CORE SERIES XI

Tame and Wild Packet Geometry

For local Langlands

Lean and evidence guide

Komorebi P.

Revised edition 1.1.0 / 18 October 2026

Reading scope

A source-pinned guide to the proved formal mechanisms and their arithmetic interfaces. The general realization endpoint is Conjecture 66. The release includes Markdown proofs, construction notes, audits and research records.

Contents

1 A reader contract for this edition 1

Navigation through the five readers 6

Evidence boundaries for this revision 7

2 Fourier projectors and Frobenius transport 8

3 Induction, return scalars and tensor interactions 10

4 Finite conductors and the tame comparison package 12

5 Heisenberg actions, central products and phase comparison 13

6 Odd phases and dyadic quadratic refinements 15

7 Matrix holonomy and global section equivalence 17

8 Monodromy and conditional atomic reconstruction 19

9 Local rings, ramification projectors and normalized Frobenius 21

10 Arithmetic type and induction adapters 23

11 Morita lattices and exact one-plus comparisons 26

12 Complex scalar cochains and honest cyclic actions 28

13 Conjecture 66 and the surviving conditional comparison 30

14 Verification lineage and reproducibility 33

A Complete selected named-source catalog 35

B Navigation across the original 66 obligations 233

C Excerpt and claim crosswalk 235

CHAPTER 1

A reader contract for this edition

Paper XI develops packet decomposition, finite representation comparisons, actual arithmetic/type adapters, and a conditional comparison interface. This guide makes the logical boundaries visible in the source language. The traditional paper gives the mathematical exposition. The visual guide illustrates the constructions. The retained Lean sources provide the formal statements and proofs at their printed hypotheses.

The general arithmetic realization endpoint is presented as **Conjecture 66**. Conjecture 66 isolates the missing realization claim from historical Claim 16.2, indexed as original obligation 66. This separation preserves the proved finite comparison and the proved conditional implication. It records that finite assembly and coherent transport have not yet supplied an actual supercuspidal realization in the original generality.

The accompanying release includes Markdown (.md) files containing detailed proofs, construction notes, proof audits, and research records. These documents distinguish established results, conditional results, and the remaining obligations associated with Conjecture 66.

1.1 What Lean certifies

Acceptance by Lean checks that the declared conclusion follows from its binders, imported declarations, and foundational axioms. It does not silently instantiate a local field, an arithmetic type, a reference correspondence, or a normalized local-factor convention. Those objects must appear in the source or in a separate mathematical attachment.

A structure may contain an existence field. A theorem that consumes that structure proves an implication using the field. It is still a substantive theorem when the comparison map, descent or uniqueness is constructed. It is not a proof of the structure's arithmetic existence. In particular, `Realization.exists_atom` is an existence input in the formal comparison interface.

1.2 Evidence available for this guide

The source catalog was generated for the inherited edition from retained production source bytes. This edition preserves its selected modules, source hashes, import edges and named lexical records. A separately dated fresh execution replays the selected source in the pinned environment and audits its actual declaration dependencies. The current result is summarized below. Historical compiler and same-kernel records later in this guide retain their original dates and scopes. The preserved catalog's historical evidence flags describe the edition in which that catalog was generated; the dated current receipts determine this release's execution status. The PDF build and the formal execution are separate operations.

> **Fresh verification, 18 October 2026: all required gates passed.** The 423 selected publication modules were compiled with their retained source hashes unchanged. All 4,342 catalogued names resolved to their recorded owners. Four source names—two private theorems and two named local instances—are matched uniquely to their kernel names using Lean's own private-name mapping; the source and kernel names are both recorded. The recovered terminal and the new comparison supplement were checked as separately counted targets.

Population	Modules	Catalogued names	Logical declarations
Selected sources	423	4,342	7,672
Recovered terminal	1	0	51
One-candidate supplement	1	1	1

The last column is the complete population of logical declarations owned by the named modules in the imported private kernel environment. It includes generated and private logical declarations. It is distinct from the retained lexical catalog, and is not a count of independent mathematical discoveries. The audit separately lists 1,636 compiler-only auxiliary names for the selected sources; these are absent from the kernel constant map.

Every audited declaration's transitive axiom dependencies belong to `Classical.choice`, `Quot.sound`, and `propext`. No forbidden dependency was found. All three aggregate imports passed. Two isolated negative controls were rejected as intended: uncatalogued public and private custom axioms in a module with a safely catalogued theorem, and a deliberately unresolved catalog name. These fixtures are not Paper XI modules. The supplement's observed axioms were `Quot.sound` and `propext`.

Fresh-source lineage and unsuccessful attempts. The first replay started from an empty Paper XI object directory and passed 421 selected modules plus the supplement. Its unfinished tower compilation was manually stopped early after a longer historical duration was found. The first run remains recorded as `PARTIAL_OR_FAIL`; it was not a source rejection, timeout, or out-of-memory event. The first continuation authenticated every completed source/object hash and dependency closure, reused only those 422 successful objects from this session, and compiled the remaining modules with a 3,600-second per-module limit. The tower and its base-change dependent completed in 1,066.8 and 827.6 seconds, respectively, completing all 423 selected modules. The terminal then emitted substantive elaboration errors, including a heartbeat-limit diagnostic. After inspection of its log, that known-failed process was deliberately terminated. This second run also remains `PARTIAL_OR_FAIL`; its stop was diagnostic-driven.

The source-revision continuation authenticated the successful source/object closure and invalidated only the revised failed terminal. Its terminal process was then killed at the environment's 8 GiB memory limit; the recorded out-of-memory counters increased. This third replay also remains failed. An unchanged-source continuation with synchronous elaboration, `Elab.async=false`, also reached the memory limit. That fourth full replay remains failed. Bounded diagnostics isolated an expensive conversion in the lower-inclusion witness. The final continuation authenticated the successful closure and compiled separately reviewed local proof repairs. No historical Paper XI object was used. Pinned official external Mathlib cache objects were imported; all Mathlib sources were not rebuilt.

Separate additions and audit history. The terminal required more than the missing explicit `PaperXI.Ramification Polyphase` import. Its repair also corrects implicit arguments, required standing-hypothesis bindings, declaration syntax and local proofs. Five private component lemmas factor the comparison proof; the lower-inclusion witness uses the retained equality in its forward direction. A separate review checks the intended conclusions and already-declared hypotheses. The original source, failed intermediate and exact diffs are retained; the absent historical receipt is not recreated. The supplement proves the supplied-candidate implication printed below. The exploratory axiom harness needed explicit counter types, a distinction between compiler auxiliaries and kernel declarations, a larger recursion bound for the catalogue literal, and private-name resolution. The failed pilot records remain alongside the successful final audits. No selected Lean theorem statement or proof body was changed.

The complete results are in `verification/lean/RELEASE_VERIFICATION.json` and its prose companion. The portable project supplies source, exact pins and replay scripts, with downloaded toolchains and generated objects omitted. This release ran fresh source compilation and complete ownership/axiom audits. It did not run a new additional full `leanchecker` pass or use an independently implemented kernel. These results establish the formal declarations at their stated hypotheses; the four open arithmetic tasks retain their separate statuses.

The recorded environment is Lean 4.34.0-rc2, commit `6a10ac8c22beadecabdbb0919c2b50214762f91d`, and Mathlib revision `aeb72f1e9981b6d35b426b16dca7759be3e7ba06`. The reported allowed transitive axiom set is `propext`, `Classical.choice`, and `Quot.sound`. Same-kernel replay is an additional invocation of Lean's kernel, not an independently implemented checker.

1.3 How the catalog is selected

The selected union contains 423 distinct module paths and 4,342 explicitly named lexical declaration commands. It starts with the 302-module v0.44 source tree, adds three checked normalized-Frobenius modules, then retains the arithmetic/type branch through authenticated base108 and checked continuations 110–113. Identical repeated module bytes are collapsed. Layer-specific provenance remains in the JSON catalog.

This is a source census, not a count of independent mathematical discoveries or kernel-generated declarations. Constructors, recursors and unnamed instances are outside this lexical count. The historical "65/66" number concerns original paper obligations, whereas the appendix includes many supporting definitions and lemmas. The conjecture receives no Lean theorem-count promotion.

1.4 Supplement: the comparison of one supplied candidate

The publication comparison fixes one admissible candidate and one admissible reference. A separating factor system identifies their isomorphism classes if their declared factor profiles agree. The additional module `PaperXI.SingleCandidateComparison` applies `LocalFactorIdentification.class_eq_iff` to exactly these supplied objects.

The expected factor profile is a parameter. H5 is assumed only for this candidate. The proof does not consume `Realization.exists_atom`, does not quantify H5 over every realizer, and does not instantiate an arithmetic correspondence. The separating factor system and admissibility conditions remain explicit inputs. This is a formal presentation of the paper's conditional implication, not an additional arithmetic existence result.

The complete supplementary module is printed below. It is counted separately from the unchanged 423-module catalog and the forty inherited teaching excerpts. Its own compiler and axiom receipt is included with the release.

```
import PaperXI.LocalFactorIdentification

/-! The one-candidate local-factor comparison used in the manuscript.
The candidate and its admissibility are supplied. Matching factors are required
only for this candidate. No atom-realization existence field is consumed.
The separating factor system remains an explicit classical input. -/
set_option autoImplicit false
noncomputable section

namespace PaperXI.SingleCandidateComparison
open LocalFactorIdentification

variable {Atom Test Value : Type*} {s : Setoid Atom}

/-- Equality of the declared factor profile identifies this supplied candidate
with the reference in the declared isomorphism quotient. -/
theorem source (S : FactorSystem s Test Value)
    (candidate reference : Atom)
    (hcandidate : S.admissible candidate) (hreference : S.admissible reference)
    (expected : Test → Value)
    (h5 : S.profile candidate = expected)
    (href : S.profile reference = expected) :
    Quotient.mk s candidate = Quotient.mk s reference := by
  apply (class_eq_iff S hcandidate hreference).mpr
  exact h5.trans href.symm

end PaperXI.SingleCandidateComparison
```

1.5 The corrected moving-space phase domain

In odd residue characteristic, the monograph's breakwise formula now applies to $M = \mathrm{im}(g - I)$ and its break summands, with g explicitly symplectic and fixed point free on M. The selected theorem `PaperXI.BreakOddPhase.source` already proves this product of moving phases. `BreakMoving` supplies the complement and nondegeneracy facts. A zero moving space contributes one. This source alignment does not add an order-prime-to-p hypothesis and does not modify the retained Lean module.

1.6 Reading literal excerpts

A displayed declaration excerpt retains the exact source text through a top-level `:=`, omitting its subsequent proof or value. Named-argument assignments inside binders do not terminate the excerpt. Structures retain all their fields. The complete declaration is available in the source file. A proof excerpt is a contiguous source slice. Ambient

assumptions are explained beside it and retained as raw setup lines in `EXCERPTS.json`; imported definitions remain necessary.

The display is not a separately compiled replacement module. Unicode supported by the monospace font appears directly. Unsupported glyphs use reversible `[U+XXXX]` tokens. The raw excerpt, display file, conversion map and SHA-256 hashes are included.

Navigation through the five readers

The formal core, the written arithmetic constructions and the research-progress record have different evidence types. This edition preserves the selected formal source and provides direct reading routes into the other four readers. The selected sources remain unchanged. The terminal replay and a one-candidate supplementary lemma are counted separately; neither supplies an arithmetic-existence axiom or a new discovery count.

Question	Reading route and formal boundary
The finite comparison	Monograph finite tame package. Its finite statement and its arithmetic interpretation retain distinct hypotheses.
The existence field	Monograph Conjecture 66, progress Section 1 and atlas V01. The source field `Realization.exists_atom` remains an input.
The pointwise comparison	Comparison of one realized candidate. H5 is about that candidate. The more general formal interface's all-realizers binder is not a stronger publication conjecture.
Ordinary arithmetic progress	Monograph Section 19 and the separate progress paper. The explicit parameter, character, conductor and marking arguments are not newly Lean-certified here.
Worked interpretation	Core visual guide and research atlas. Abstract finite examples are not certificates for an unspecified arithmetic type.

The original-obligation crosswalk

The existing 66-row historical table now links its printed-number column to the actual current monograph statements. The final row points to Conjecture 66, while the separate comparison proposition has its own link. These links are navigational. They do not assert that every printed theorem is definitionally identical to its selected entry module.

All 423 selected module paths, the 4,342 named lexical declaration records, and the 40 teaching excerpts retain their inherited source identities. The current source audit checks their correspondence and preservation. Fresh compiler and axiom receipts identify the executed targets separately from that inherited census.

Evidence boundaries for this revision

Compilation and publication are separate. This revision builds and reviews the PDF readers and records a separate fresh Lean execution. The preceding execution summary gives the current result. Historical compiler and same-kernel receipts remain attributable to their exact source revisions. No independently implemented proof checker is claimed.

The terminal has a separately dated repair and replay. The original successful v0.45 receipt remains unavailable. Adding the missing explicit `PaperXI.RamificationPolyphase` import exposed further elaboration errors. The repaired source also corrects implicit-argument calls, required standing-hypothesis bindings, declaration syntax and local proofs. The original, failed intermediate, exact diffs and hypothesis review are retained. Its compiler status is reported separately from the unchanged 423-module selection. This target is a restricted tame Weil-side attachment; it does not construct a general wild automorphic realizer.

Block 2 is a preserved-reader baseline. The 33-page first part of the research companion survives in its PDF and records. Its complete editable working archive has not been recovered. Later continuations are independently editable. This edition does not claim a unified rebuild of the missing source.

The selected classical-source gap has been closed. The Henniart–Lemaire induction interface has now been checked against the full official text, including direct visual inspection of the relevant normalization formulas. The bounded review preserves its characteristic, rank, genericity and orbit hypotheses. Other Bushnell–Kutzko, conductor, tower and descent inputs retain their own evidence. This source check is not H5 or corrected-character occurrence.

Four open obligations, not four missing formal lemmas

General arithmetic realization requires an actual common host with the declared actions and transports. H5 requires the characterizing factors for that same realizer. The corrected-character branch still needs occurrence in its fixed candidate. Where the stronger native contract is prescribed, the actual square, section and honest-phase certificate must be computed on the chosen type. The research certificate plate deliberately leaves those arithmetic fields uncomputed.

The positive transported-marking contract remains separate. Its hypotheses are not silently strengthened by the native-marking task. Likewise, a failed original carrier does not exclude every permitted enlarged inducing carrier.

Reproducible reading routes

The delivered builder resolves the local companion destinations after compiling all three original readers. Keep the two preserved research companions and the revised originals together. The source catalogs remain unchanged. The new navigation JSON records the current targets without overwriting historical page numbers or evidence dates.

CHAPTER 2

Fourier projectors and Frobenius transport

The cyclic packet is constructed by averaging against an additive character. Idempotence, orthogonality and completeness are consequences of the representation law and character orthogonality. The packet space is the actual range of the computed projector, and Frobenius transports that range by its conjugation law.

2.0.1 The averaging operator

The ambient cyclic representation and additive-character hypotheses are declared in the source module. The index is a residue label in $\mathbb{Z}/M\mathbb{Z}$. The scalar field is $\mathbb{C}$, so the finite order is invertible. The definition sums actual operators. Later results compute their products rather than postulating a family of orthogonal idempotents.

E01: `PaperXI.FourierProjectors`, source lines 13–15.

Contiguous source proof excerpt.

Source SHA-256: `02af62a8c8e5753ef7a695629d44c71db407211d18a6bd6b6a48865ba731de52`.

```
def projector (ρ : Action (K := K) (V := V) (M := M))
    (ψ : AddChar (ZMod M) K) (j : ZMod M) : Module.End K V :=
  PaperXI.FourierCovariance.fourierOperator (operator ρ) ψ j

```

2.0.2 Completeness of the packet resolution

The finite cyclic group and primitive character remain fixed. The sum ranges over all residue labels. Completeness says that every vector is the sum of its packet components. A vanishing packet is permitted, and the theorem does not assert that every residue label occurs.

E02: `PaperXI.FourierProjectors`, source lines 68–77.

Contiguous source proof excerpt.

Source SHA-256: `02af62a8c8e5753ef7a695629d44c71db407211d18a6bd6b6a48865ba731de52`.

```
theorem projector_complete (ρ : Action (K := K) (V := V) (M := M))
    (ψ : AddChar (ZMod M) K) (hψ : ψ.IsPrimitive) (hM : (M : K) ≠ 0) :
    (∑ j : ZMod M, projector ρ ψ j) = 1 := by
  have hs (a : ZMod M) : (∑ j : ZMod M, ψ (-j * a)) =
      if a = 0 then (M : K) else 0 := by
    simpa [mul_comm, neg_mul, mul_neg] using character_sum ψ hψ (-a)
  simp only [projector_formula, ← Finset.smul_sum]
  rw [Finset.sum_comm]
  simp_rw [← Finset.sum_smul, hs]
  simp [hM, smul_smul]

```

2.0.3 An actual packet equivalence

The supplied invertible Frobenius operator satisfies the declared conjugation relation with the cyclic inertia action. The multiplication direction in the residue label is fixed by that relation. The operator and its inverse restrict to the two computed packet ranges. Their inverse identities give a linear equivalence, so equal packet dimensions follow from a constructed transport.

E03: `PaperXI.PacketTransport`, source lines 62–66.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `dc7282e77c935ca61b8807fe87f4fe2fa372d720a28d0a1c8982c6c749e1a910`.

```
def packetEquiv (ρ : Action (K := ℂ) (V := V) (M := M))
    (F : V ≃ₗ[ℂ] V) (q : (ZMod M)ˣ)
    (h : F.symm.conj (operator ρ 1) = operator ρ (q : ZMod M)) (j : ZMod M) :
    packet ρ j ≃ₗ[ℂ] packet ρ ((q : ZMod M) * j) where
  toFun v :=
```

2.1 Geometric Frobenius convention and the publication repair

The geometric convention in this edition is

$$\Phi^{-1} t \Phi = t^q, \qquad \Phi P_j \Phi^{-1} = P_{qj}.$$

Under the other relation $\Phi t \Phi^{-1} = t^q$, the correct label is $q^{-1}j$. The original Proposition 3.1 proof paired that other relation with the geometric label qj. The final traditional paper repairs the displayed relation to the geometric convention, preserving its intended packet transport. The checked Lean module already states both conventions separately. No new Lean proof is claimed for this prose convention repair.

2.1.1 The existing geometric covariance theorem

The cyclic order is nonzero. The element q is a unit modulo that order, and the source hypothesis uses the conjugation by F^{-1}. This exact signature and proof support the publication correction. They distinguish the Frobenius direction from its action on projector labels, rather than changing both directions simultaneously.

E04: `PaperXI.TameProjectors`, source lines 45–52.

Contiguous source proof excerpt.

Source SHA-256: `d5881299706cbe60a5cbefc9dc60bb9bdffbc1365228529c181b9a5120b71d8e`.

```
theorem geometric_covariance (ρ : Action (K := ℂ) (V := V) (M := M))
    (F : V ≃ₗ[ℂ] V) (q : (ZMod M)ˣ)
    (h : F.symm.conj (operator ρ 1) = operator ρ (q : ZMod M)) (j : ZMod M) :
    F.conj (projector ρ j) = projector ρ ((q : ZMod M) * j) := by
  have hs := forward_generator_covariance ρ F.symm q h ((q : ZMod M) * j)
  simp only [← mul_assoc, Units.inv_mul, one_mul] at hs
  have hh := congrArg F.conj hs
  simpa only [LinearEquiv.conj_conj_symm] using hh.symm

```

CHAPTER 3

Induction, return scalars and tensor interactions

The induced representation branch uses a degree homomorphism and its degree subgroups. A normalized degree-one element gives coset representatives. Its cyclic action has a closing scalar determined by the inducing character. Tensor interactions are constructed as actual representation equivalences indexed by greatest-common-divisor classes.

3.0.1 The closing scalar is derived

An actual character of the degree subgroup and a degree-one Frobenius element are supplied. The degree n is positive. The scalar evaluates the character at the full return element. It is not an independent certificate whose agreement with the action is assumed.

E05: `PaperXI.InducedCycleInvariants`, source lines 33–33.

Contiguous source proof excerpt.

Source SHA-256: `2faaeb89211dca9d88df443a27ad69ab7e310783d4e07a3ad54d85c3c5bcae1a`.

```
33 def returnScalar : ℂˣ := ξ (degreePeriod φ δ hφ)
34
```

3.0.2 Dimension and characteristic polynomial

The basis of the genuine degree-induced space has already been constructed. Finite dimensionality and the nonzero degree hypotheses occur in the module's setup. The resulting dimension and polynomial record the entire cyclic block, including its closing scalar. Their interpretation as arithmetic local data needs the corresponding group and normalization attachments.

E06: `PaperXI.InducedCycleInvariants`, source lines 75–77.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `2faaeb89211dca9d88df443a27ad69ab7e310783d4e07a3ad54d85c3c5bcae1a`.

```
75 theorem cycle_invariants :
76     (degreeInduced δ n ξ φ).charpoly = X ^ n - C (returnScalar δ φ hφ n ξ : ℂ) ∧
77     (degreeInduced δ n ξ φ).det = (-1) ^ (n - 1) * (returnScalar δ φ hφ n ξ : ℂ) :=
```

3.0.3 The tensor decomposition is an equivalence

Two positive degrees and two characters of their degree subgroups are supplied. Representatives use φ^{-i}, fixing the direction of the shifted interaction character. There are $\gcd(n,m)$ interaction blocks of degree $\operatorname{lcm}(n,m)$. The source builds an intertwining equivalence of representations, not merely the dimension identity $\gcd(n,m)\operatorname{lcm}(n,m) = nm$.

E07: `PaperXI.TameTensorSource`, source lines 66–69.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `b52fef0aac236edae7a22103dcf267e147fa13606783529fa64e3c4c4e954d3e`.

```
def decomposition : Representation.Equiv
    ((CyclicInduced.degreeInduced ν n ξ).tprod (CyclicInduced.degreeInduced ν m η))
    (Representation.directSum (fun i : Fin (Nat.gcd n m) =>
      CyclicInduced.degreeInduced ν (Nat.lcm n m) (interactionCharacter ν φ n m ξ η i))) :=
```

CHAPTER 4

Finite conductors and the tame comparison package

The finite conductors are sums of codimensions of fixed spaces for a declared subgroup family. Arithmetic Swan and Artin conductors require that this family and its weights match the arithmetic ramification conventions. The finite residue Gauss phase is also a declared character sum before its arithmetic interpretation.

4.0.1 A declared finite ramification sum

The representation, a finite set of depths, a subgroup at each depth and rational weights are supplied. The source does not construct an absolute Weil ramification filtration here. These inputs determine the exact finite sum. The tame vanishing theorem applies when the selected positive-depth groups act trivially.

E08: `PaperXI.TameConductor`, source lines 56–58.

Contiguous source proof excerpt.

Source SHA-256: `808eb443d2a292fc79cd70ce323c369acd62c38e9393ec4c19269dc06d575dcf`.

```
def finiteSwan (π : Representation ℂ G V) (S : Finset ℕ)
    (J : ℕ → Subgroup G) (w : ℕ → ℚ) : ℚ :=
  ∑ r ∈ S, w r * (fixedCodimension π (J r) : ℚ)

```

4.0.2 Proposition 16.1 at finite scope

The exact induced spaces, degree character, cyclic relation, inertia label, finite-depth subgroup data and residue characters are supplied. The additive character is nontrivial. The package derives observations, a tensor equivalence, the total tensor dimension and the Gauss-phase magnitude. Its magnitude formula retains the trivial multiplicative-character exception. This finite statement does not certify standard local-Weil factors or an arithmetic epsilon-factor normalization.

E09: `PaperXI.FiniteTameComparison`, source lines 93–107.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `1c4357da32eaade5be0c88670eaa605f47fbc4269253888bbf9f72509d0e428d`.

```
theorem finite_comparison_package (P : Subgroup G)
    (hP : ∀ g ∈ P, ∀ v, degreeInduced δ n ξ g v = v)
    (S : Finset ℕ) (J : ℕ → Subgroup G) (w : ℕ → ℚ)
    (hJP : ∀ r ∈ S, J r ≤ P)
    (m : ℕ) [NeZero m] (η : TameDegree.degreeSubgroup δ m →* ℂˣ)
    (χ : kˣ →* ℂˣ) (ψ : AddChar k ℂ) (hψ : ψ ≠ 1) :
    observed δ ι φ n ξ q hrel S J w χ ψ =
      fromCoordinates n q j (InducedCycleInvariants.returnScalar δ φ hφ n ξ) χ ψ ∧
    Nonempty (Representation.Equiv ((degreeInduced δ n ξ).tprod (degreeInduced δ m η))
      (Representation.directSum (fun i : Fin (Nat.gcd n m) =>
        degreeInduced δ (Nat.lcm n m) (TameTensorSource.interactionCharacter δ φ n m ξ η i)))) ∧
    (∑ i : Fin (Nat.gcd n m), Module.finrank ℂ (DegreeIndSpace δ (Nat.lcm n m)
      (TameTensorSource.interactionCharacter δ φ n m ξ η i))) = n*m ∧
    ‖GaussCharacters.phase χ ψ‖ =
      if χ = 1 then (Real.sqrt (Fintype.card k))⁻¹ else 1 :=
```

CHAPTER 5

Heisenberg actions, central products and phase comparison

A finite Heisenberg action combines translations and modulation operators. Irreducibility is proved from those actual operators. Break assembly uses a genuine quotient central product and a tensor carrier. To extend the bridge through a stabilizer, the retained action on the whole central extension and the scalar discrepancy must both be controlled.

5.0.1 The finite Schrodinger action

The finite Heisenberg group and its central character are fixed by the source's preceding constructions. The function-space carrier is literal. The multiplication law is checked for the displayed operators. Invertibility, central action and irreducibility are subsequent consequences.

E10: `PaperXI.HeisenbergRepresentation`, source lines 14–16.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `386317cf254a374627df755dc02465986fc602151303b4e2b3413f2b82b46492`.

```
def representation (ψ : AddChar K ℂ) (h2 : (2 : K) ≠ 0) :
    Representation ℂ (Element K X) (X → ℂ) where
  toFun a :=
```

5.0.2 Irreducibility from actual translations and modulations

The source supplies the required nondegenerate character pairing. Its invariant-subspace proof uses coordinate projections and spans the full function space. This finite representation theorem supplies a building block. It does not by itself embed the block in a canonical arithmetic simple type.

E11: `PaperXI.HeisenbergRepresentation`, source lines 58–59.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `386317cf254a374627df755dc02465986fc602151303b4e2b3413f2b82b46492`.

```
theorem irreducible [Fintype X] (ψ : AddChar K ℂ) (hψ : ψ ≠ 1)
    (h2 : (2 : K) ≠ 0) : Representation.IsIrreducible (representation (X := X) ψ h2) :=
```

5.0.3 The assembled group is a quotient

A finite indexed family of central-extension data shares a declared center. The balanced central subgroup is defined earlier in the module. The quotient construction retains an explicit center even for an empty family. The quotient's representation must descend through the balancing relation, and that descent is proved in the supporting modules.

E12: `PaperXI.CentralProduct`, source lines 39–41.

Contiguous source proof excerpt.

Source SHA-256: `9d1e6a4ece84a862ef75383b66b97af694c9ac2298fc2047a6c4844bc6aa34af`.

```
def Group := Ambient (Z := Z) (H := H) [U+29F8] relation D

instance : _root_.Group (Group D) := inferInstanceAs (_root_.Group (_ [U+29F8] relation D))

```

5.0.4 Precisely what break assembly consumes

The finite family includes the actual groups, quotient groups and complex carriers. Its record keeps central extensions, retained lifts, central characters and relevant pairings. An initial tensor intertwiner and a final representation equivalence are not input fields. They are constructed. Dyadic square and retained-group data remain part of the input and cannot be recovered from an alternating commutator form alone.

E13: `PaperXI.FiniteBreakAssembly`, source lines 22–37.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `ecca3693f7733c7ceef597b6e821b111610c82d612fd66abfe754af0e5be434c`.

```
structure Data where
  source : ∀ i, CentralExtension.Datum Z (H i) (A i)
  target : ∀ i, CentralExtension.Datum Z (H' i) (A' i)
  lift : ∀ i, H i =* H' i
  lift_center : ∀ i z, lift i ((source i).i z) = (target i).i z
  centralCharacter : Z →* ℂˣ
  nondegenerate : ∀ i, CocycleCommutator.radical
    (CentralExtension.scalarCocycle (source i) centralCharacter) = ⊥
  leftAction : ∀ i, Representation ℂ (H i) (V i)
  rightAction : ∀ i, Representation ℂ (H' i) (W i)
  leftIrreducible : ∀ i, Representation.IsIrreducible (leftAction i)
  rightIrreducible : ∀ i, Representation.IsIrreducible (rightAction i)
  leftCentral : ∀ i z, leftAction i ((source i).i z) =
    (centralCharacter z : ℂ) • LinearMap.id
  rightCentral : ∀ i z, rightAction i ((target i).i z) =
    (centralCharacter z : ℂ) • LinearMap.id

```

5.0.5 Closing the honest stabilizer action

Honest source and target stabilizer actions are supplied together with covariance. The discrepancy is tested on a generating set. The construction kills the scalar discrepancy through the generator phase hypothesis. Matching quotient actions alone would leave this scalar issue unresolved.

E14: `PaperXI.FiniteBreakAssembly`, source lines 111–115.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `ecca3693f7733c7ceef597b6e821b111610c82d612fd66abfe754af0e5be434c`.

```
def equivOfPhase (S : Set C) (hgen : Subgroup.closure S = ⊤)
    (hp : ∀ g ∈ S, discrepancy d α R T hR hT g = 1) :
    Representation.Equiv
      (CyclicBridge.semidirectRepresentation α (sourceAction d) R hR)
      (CyclicBridge.semidirectRepresentation α (targetAction d) T hT) :=
```

CHAPTER 6

Odd phases and dyadic quadratic refinements

The paper separates odd phases defined by a determinant from dyadic phases defined by a quadratic refinement. The formal certificate computes the declared determinant phase through its characteristic polynomial. Identifying that value with an arithmetic Weil trace or a canonical beta extension requires an additional argument. In residue characteristic two, an alternating form alone does not retain the quadratic data needed for the phase.

6.0.1 The actual polynomial phase certificate

The residue characteristic is an odd prime. A finite-dimensional vector space carries an actual nondegenerate alternating form. The supplied invertible linear operator need not be fixed-point-free for this algebraic identity. The form derives even dimension. Evaluation of the actual characteristic polynomial at one gives the determinant of the difference operator with the declared sign convention. This certifies the defined phase, without assuming that it equals a separately normalized arithmetic phase.

E15: `PaperXI.OddPhaseCertificate`, source lines 46–50.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `783d84147c4f09cbca432ad799b1e57d9bd47da13f3841e92d4f071d49709f2e`.

```
theorem characteristic_certificate (hp : p ≠ 2)
    (b : LinearMap.BilinForm (ZMod p) V) (hb : b.IsAlt) (hbn : b.Nondegenerate)
    (g : V ≃ₗ[ZMod p] V) :
    phase g = quadraticChar (ZMod p) (-1) ^ (Module.finrank (ZMod p) V / 2) *
      quadraticChar (ZMod p) (g.toLinearMap.charpoly.eval 1) :=
```

6.0.2 Refinements form an affine space

Two quadratic refinements have the same bilinear form over the field with two elements. Nondegeneracy is not required for the affine-difference statement. Their difference is uniquely linear. Hence the bilinear form does not choose one refinement. When the arithmetic contract prescribes square data or a whole marked finite extension, that datum must be retained.

E16: `PaperXI.DyadicAffine`, source lines 74–75.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `06fd5b5a57b17060abd22a11fb29592fcd53b06aa09388b92314fe2fd23692e3`.

```
theorem exists_unique_linear_difference (q q₀ : Refinement b) :
    ∃! l : V →ₗ[ZMod 2] ZMod 2, addLinear q₀ l = q :=
```

6.0.3 The exact Arf Gauss sign

The finite index set parametrizes symplectic planes. Each plane has its explicit quadratic-refinement coefficients in the field with two elements. The integer character sum is $(-1)^{\mathrm{Arf}(q)}2^{d}$. The proof factors the complete finite sum into plane sums. It is an exact theorem, not an asymptotic or fitted numerical rule.

E17: `PaperXI.DyadicGauss`, source lines 51–58.

Contiguous source proof excerpt.

Source SHA-256: `634fe67a476f515f627455fa43a1ec870a4ccc66b87867cc5d40ee1227c91dea`.

```
51 theorem gauss_sum (a : ι → F2 × F2) :
52     (∑ x, sign (quadratic a x)) = sign (arf a) * 2 ^ Fintype.card ι := by
53   classical
54   simp only [quadratic, sign_sum]
55   rw [← Fintype.prod_sum (fun (i : ι) (x : F2 × F2) => sign (plane (a i) x))]
56   simp_rw [plane_gauss]
57   rw [Finset.prod_mul_distrib, ← sign_sum]
58   simp [arf]
59
```

6.0.4 Counting refinements with a prescribed Arf invariant

The supplied symplectic basis identifies the finite-dimensional nondegenerate alternating space. The source separately retains dimension zero and the odd-Arf count. The formulas count the complete refinement space, not a sampled collection. An isometry preserving only the bilinear form need not preserve the prescribed Arf invariant or its phase.

E18: `PaperXI.DyadicRefinementCounts`, source lines 42–44.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `5d9cdc46b7bde9d4f7ceed89b2a7a68ec38f36fee3142300f40b6a8d5301838b`.

```
42 theorem even_refinement_count :
43     (Fintype.card {q : Refinement (symplectic (ι := ι)) // basisArf q = 0} : ℚ) =
44       (2 : ℚ) ^ ((Fintype.card ι : ℤ) - 1) * ((2 : ℚ) ^ Fintype.card ι + 1) :=
```

CHAPTER 7

Matrix holonomy and global section equivalence

Local packet comparisons can be transported around a graph. A rooted spanning tree determines root paths. The remaining original edges become chords. Full matrix compatibility on the resulting fundamental cycles is the necessary and sufficient obstruction to a based global gauge.

7.0.1 The rooted chord criterion

The graph is a connected quiver with invertible edge transports. A root comparison is fixed. The cycle condition is an equality of the full transported linear maps. Trace, determinant and projective cycle tests are weaker than this matrix identity. The source constructs a gauge precisely when the chord tests hold and derives compatibility on every edge.

E19: `PaperXI.GraphGauge`, source lines 155–159.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `1750d323d264d4760717c62ab3aa154cbebefcdd24e80cc76ef22d7d8f5a572b`.

```
theorem connected_based_iff_chords (U : V r ≃ₗ[K] W r)
    (hc : GraphTree.Connected (α := α)) :
    (∃ F : ∀ v, V v ≃ₗ[K] W v, F r = U ∧ EdgeCompatible A B F) ↔
      ∀ e : GraphTree.Chord (GraphTree.ofConnected r hc),
        CycleCompatible A B U (GraphTree.rootPath (GraphTree.ofConnected r hc)) e.val :=
```

7.0.2 The space of compatible sections

Each vertex has a vector-space fiber and each directed edge carries a linear equivalence. A section satisfies every edge equation. The space is an actual linear subspace of the product of fibers. It can be zero. A nonzero section space, an irreducible section representation, and an arithmetic supercuspidal atom are separate assertions.

E20: `PaperXI.GluedPacketSections`, source lines 22–26.

Contiguous source proof excerpt.

Source SHA-256: `5bf8047d59226ed80f93c4668b031aedab1df38bd5ee39456a9335f2685372bf`.

```
def sections : Submodule K (∀ v, V v) where
  carrier s := ∀ {u v : α} (e : u → v), A e (s u) = s v
  zero_mem' := by intro u v e; simp
  add_mem' := by intro a b ha hb u v e; simpa using (map_add (A e) (a u) (b u)).trans (congrArg₂ (· + ·) (ha e) (hb e))
  smul_mem' := by intro c a ha u v e; simpa using (map_smul (A e) c (a u)).trans (congrArg (c • ·) (ha e))

```

7.0.3 Gluing produces a representation equivalence

Covariance of the source and target edge systems is supplied. A root intertwiner and the full fundamental-cycle compatibility equations are supplied. The resulting equivalence acts on the global compatible-section spaces. Its existence is proved by transporting the root intertwiner, rather than stored as a field in a comparison record.

E21: `PaperXI.GluedPacketSections`, source lines 138–144.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `5bf8047d59226ed80f93c4668b031aedab1df38bd5ee39456a9335f2685372bf`.

```
def fromChords {r : α} (hc : GraphTree.Connected (α := α))
    (hA : RepresentationGraph.EdgeEquivariant A ρ)
    (hB : RepresentationGraph.EdgeEquivariant B τ)
    (U : Representation.Equiv (ρ r) (τ r))
    (hch : ∀ e : GraphTree.Chord (GraphTree.ofConnected r hc),
      CycleCompatible A B U.toLinearEquiv (GraphTree.rootPath (GraphTree.ofConnected r hc)) e.val) :
    Representation.Equiv (sectionRepresentation A ρ hA) (sectionRepresentation B τ hB) :=
```

CHAPTER 8

Monodromy and conditional atomic reconstruction

Finite atomic results can propagate through a supplied decomposition and a supplied ordered-quotient interface. The reconstruction theorem constructs sorting, atomic replacement and descent. The external decomposition and ordered Langlands-quotient existence statements remain printed hypotheses.

8.0.1 The monodromy normalization

A twisting character ν and a representation are fixed. Copy i has the twisted action $\nu(g)^i\rho(g)$, and the nilpotent operator shifts copy index. The declared relation is $R(g)NR(g)^{-1} = \nu(g)^{-1}N$. The inverse on the twisting character fixes the convention; changing it would change the normalization.

E22: `PaperXI.MonodromyAtom`, source lines 64–66.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `f53d79ddc74c43995955bc45c8bce600d821bb6d118e9426d85e8478c0829bf2`.

```
theorem covariance (ν : G →* ℂˣ) (ρ : Representation ℂ G V) (m : ℕ) (g : G) :
    action ν ρ m g * monodromy m =
      (ν g : ℂ)⁻¹ • (monodromy m * action ν ρ m g) :=
```

8.0.2 Source decomposition is an explicit interface

An isomorphism relation on source blocks is fixed. The record specifies the allowed displays and their intrinsic classwise uniqueness. The formal reconstruction does not derive classification of all arithmetic source objects from arbitrary finite data. It uses this interface to make replacement independent of display choices.

E23: `PaperXI.AtomicReconstruction`, source lines 17–21.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `e848f4ab5b8717eda8465737b7a726fbb877d3e6effc3728eaeab08698da52cb`.

```
structure Decomposition {P : Type*} (parameterIso : Setoid P) (A : Type*) where
  displays : P → List (Block A) → Prop
  exists_display : ∀ p, ∃ xs, displays p xs
  intrinsic : ∀ p q xs ys, parameterIso.r p q →
    displays p xs → displays q ys → xs.Perm ys

```

8.0.3 The reconstructed map and its uniqueness

The source decomposition interface, ordered target-quotient interface and atomic replacement map are supplied. Ordering uses the specified target data. The candidate and choice independence are constructed. This conditional theorem remains useful when each interface is instantiated by separate mathematics, but it is not an unrestricted correspondence existence theorem.

E24: `PaperXI.AtomicReconstruction`, source lines 143–144.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `e848f4ab5b8717eda8465737b7a726fbb877d3e6effc3728eaeab08698da52cb`.

```
theorem formal_atomic_reconstruction (f : A → B) (D : Decomposition sP A)
    (L : OrderedQuotient B T) : ∃! F : Quotient sP → T, Reconstructs f D L F :=
```

CHAPTER 9

Local rings, ramification projectors and normalized Frobenius

The later local-field branch attaches finite residue extensions and normalized geometric Frobenius to actual algebraic objects. Lower ramification projectors average genuine finite groups. Their successive differences form depth-detail channels. A degree-only quotient forgets inertia, and the source proves exactly which detail information then vanishes.

9.0.1 Lower congruence groups

The action is on a supplied local ring, with its maximal-ideal powers. In a finite Galois DVR application these agree with the usual nonnegative lower groups under the stated dictionary. The generic definition is based on congruences of the actual action. It is not asserted to be an absolute Weil ramification filtration without the arithmetic attachment.

E25: `PaperXI.ValuationRamification`, source lines 15–28.

Contiguous source proof excerpt.

Source SHA-256: `92ddb509dc327389c2168f44ddfd5f7e16f4315af16835a54398077fe81146ca`.

```
def lower (i : ℕ) : Subgroup D where
  carrier := {g | ∀ x : R, g • x - x ∈ (IsLocalRing.maximalIdeal R) ^ (i + 1)}
  one_mem' := by intro x; simp
  mul_mem' := by
    intro g h hg hh x
    have ha := hg (h • x)
    have hb := hh x
    have hab := Ideal.add_mem ((IsLocalRing.maximalIdeal R) ^ (i+1)) ha hb
    simpa only [sub_add_sub_cancel, mul_smul] using hab
  inv_mem' := by
    intro g hg x
    have hh := hg (g⁻¹ • x)
    have hh' := ((IsLocalRing.maximalIdeal R) ^ (i+1)).neg_mem hh
    simpa using hh'

```

9.0.2 An exact depth-detail resolution

The finite-group ramification filtration, representation and stabilization hypothesis are supplied. Averaging is over actual finite subgroups. Nested fixed spaces give nested projectors. Their successive differences are orthogonal detail projectors, and the telescope reconstructs the identity at the stated stabilized depth.

E26: `PaperXI.RamificationPolyphase`, source lines 311–313.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `2f34bd694def457221d6f00b874c8b54547dcc95ee249e3afcf34d71b5b64549`.

```
theorem identity_eq_inertia_add_details {N : ℕ} (hN : lower R D N = ⊥) :
    (1 : Module.End ℂ V) = depthProjector (R := R) (D := D) (F := F) (ρ := ρ) 0 +
      ∑ i ∈ Finset.range N, detailProjector (R := R) (D := D) (F := F) (ρ := ρ) i :=
```

9.0.3 The unconditional quotient target is the degree range

The residue–Frobenius pullback has a degree homomorphism whose kernel is inertia. No degree-one lift is needed for this first-isomorphism statement. The quotient is equivalent to the image of degree. To replace that image by all integer degrees, the source adds a degree-one Frobenius-lift hypothesis or a subsequent construction that supplies it.

E27: `PaperXI.RamifiedUnramifiedQuotient`, source lines 42–43.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `755d457b0f11dfbac94eaee10326fa290bd5b6a331d8883bb8ce26811f00a747`.

```
42 noncomputable def quotientEquivDegreeRange :
43     Pullback r F [U+29F8] Inertia r F ≃* (degree r F).range :=
```

9.0.4 The constructed normalized quotient

The normalized-Frobenius continuation uses the geometric lift constructed in its preceding modules. Its exact local-field and residue hypotheses are in the source setup. This checked continuation fills the bounded normalized quotient step. Its retained report explicitly leaves full arithmetic/automorphic integration unestablished.

E28: `PaperXI.NormalizedFrobeniusQuotient`, source lines 79–79.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `917cfa62d1b914d7d47a3c31f682a52d3338dc040a1606450f1cff71ca075b1f`.

```
79 def quotientEquivIntegerDegree : Weil E L [U+29F8] Inertia E L ≃* Multiplicative ℤ :=
```

CHAPTER 10

Arithmetic type and induction adapters

The arithmetic branch uses actual local fields, continuous coordinate equivalences, literal subgroups, original characters, and induction carriers. Generic support and intertwining arguments derive irreducibility once the type dictionary is supplied. Later continuations reduce some dictionary leaves to quotient forms and actual lattice algebra.

10.0.1 A candidate from three arithmetic profile leaves

The ambient setup fixes local fields F, E, coordinates, the literal subgroup D, its character θ, the prescribed κ, residue representation σ, and the original tensor carrier. The displayed hypotheses keep trace support, character covariance and the oriented ambient index identity explicit. The proof derives the required Heisenberg intersection dimension and feeds it into the inherited candidate constructor. It preserves the original carrier and representations. The existence of canonical type data and the actual truth of its retained source hypotheses are separate mathematical attachments.

E29: `PaperXI.ActualProfileCandidate110`, source lines 34–88.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `dd434e182eec32397e64a3e6a43193cc56a3c965eff65c4e36e14f4bd0b2c27b`.

```
theorem candidate_exists_from_original_profiles
    (hF : Continuous (algebraMap F E))
    (hDo : IsOpen (D : Set (GL n F)))
    (hB : ∀ u : GL m [U+1D4AA][E],
      (coordinateAut e).symm (integralMatrix F E m u) ∈ CharacterNormalizer61.normalizer D θ)
    (hfull : ∀ g : GL n F,
      g ∈ CharacterNormalizer61.normalizer D θ ↔
        ∃ a : Eˣ, ∃ j : compact F E m n e D θ,
          unitsEmbedding F E n (scalarMatrixEmbedding F E m n e) a * (j : GL n F) = g)
    [IsIrreducible (κ.comp (firstSubgroup F E m n e D θ).subtype)] [IsIrreducible σ]
    (hκ : SmoothDegreeQuotient.IsSmooth κ)
    (hβ : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      NormalizerIntertwining60.IntertwinedAt (compact F E m n e D θ) κ g)
    (hσc : CuspidalCartan105.Cuspidal σ)
    (hc : ActualCharacterData108.Contains F E m n e D θ hDc κ)
    (hθsupport : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      ∃ j₁ j₂ : compact F E m n e D θ, ∃ y : GL m E,
        g = (j₁ : GL n F) * centralizerMap F E m n e y * (j₂ : GL n F))
    (hz : ∀ x : firstSubgroup F E m n e D θ,
      x ∉ actualScalarSubgroup F E m n e D θ →
      Representation.character (ActualIntersection108.restriction F E m n e D θ κ) x = 0)
    (hcov : ∀ y : GL m E,
      ∀ p : pairs (ActualIntersection108.embedding F E m n e D θ)
        (centralizerMap F E m n e y),
      ∀ h₁ : left (ActualIntersection108.embedding F E m n e D θ)
        (centralizerMap F E m n e y) p ∈ actualScalarSubgroup F E m n e D θ,
      ∀ h₂ : right (ActualIntersection108.embedding F E m n e D θ)
        (centralizerMap F E m n e y) p ∈ actualScalarSubgroup F E m n e D θ,
      actualScalarCharacter F E m n e D θ
        (left (ActualIntersection108.embedding F E m n e D θ)
          (centralizerMap F E m n e y) p, h₁) =
      actualScalarCharacter F E m n e D θ
        (right (ActualIntersection108.embedding F E m n e D θ)
          (centralizerMap F E m n e y) p, h₂))
    (hindex : ∀ y : GL m E,
      ((ActualIntersection108.embedding F E m n e D θ).range ⊓
        (ActualIntersection108.embedding F E m n e D θ).range.comap
```

```
          (MulAut.conj (centralizerMap F E m n e y)).toMonoidHom).relIndex
            (ActualIntersection108.embedding F E m n e D θ).range =
      (D ⊓ D.comap (MulAut.conj (centralizerMap F E m n e y)).toMonoidHom).relIndex D) :
    FiniteDimensional ℂ (W ⊗[ℂ] U) ∧
    ∃ Λ : Representation ℂ (extended F E m n e D θ) (W ⊗[ℂ] U),
      (∀ j : compact F E m n e D θ,
        Λ (PrescribedFieldType70.compactInclusion F E n (scalarMatrixEmbedding F E m n e)
          (compact F E m n e D θ) j) =
          CompactTypeTensor58.action (reduction F E m n e D θ hmodel) κ σ j) ∧
      IsIrreducible Λ ∧ SmoothDegreeQuotient.IsSmooth Λ ∧
      FullIntertwiningLocus69.locus (extended F E m n e D θ) Λ =
        (extended F E m n e D θ : Set (GL n F)) ∧
      Function.Injective (TensorTypeOperators70.realize (extended F E m n e D θ) Λ) ∧
      Nontrivial (IndV (extended F E m n e D θ).subtype Λ) ∧
      SmoothDegreeQuotient.IsSmooth (ind (extended F E m n e D θ).subtype Λ) ∧
      IsIrreducible (ind (extended F E m n e D θ).subtype Λ) :=
```

10.0.2 The residue-rank-one specialization

The residue rank is Fin 1. The original character, carrier, support dictionary, beta-extension intertwining and occurrence input remain fixed. The finite cuspidal residue factor and several normalization/support leaves are derived in this branch. It still does not manufacture every canonical simple character or establish the entire original claim.

E30: `PaperXI.RankOneSourceCandidate110`, source lines 29–60.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `7f57ede141636d625efea3d9b67e905b1f34bcd5dd6c475f31a7e2ec2a7b2928`.

```
theorem candidate_exists
    (hF : Continuous (algebraMap F E)) (hDo : IsOpen (D : Set (GL n F)))
    (hJ : J₁ ≤ matrixCongruence F E (Fin 1) n e)
    (hJN : J₁ ≤ CharacterNormalizer61.normalizer D θ)
    (hθcentralizer : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      ∃ a : J₁, ∃ b : (centralizerMap F E (Fin 1) n e).range,
      ∃ c : J₁, (a : GL n F) * (b : GL n F) * (c : GL n F) = g)
    (hfull : ∀ g : GL n F,
      g ∈ CharacterNormalizer61.normalizer D θ ↔
        ∃ a : Eˣ, ∃ j : compact F E (Fin 1) n e D θ,
          unitsEmbedding F E n (scalarMatrixEmbedding F E (Fin 1) n e) a *
            (j : GL n F) = g)
    [IsIrreducible (κ.comp (firstSubgroup F E (Fin 1) n e D θ).subtype)]
    (hκ : SmoothDegreeQuotient.IsSmooth κ)
    (hβ : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      NormalizerIntertwining60.IntertwinedAt (compact F E (Fin 1) n e D θ) κ g)
    (hc : ActualCharacterData108.Contains F E (Fin 1) n e D θ hDc κ) :
    FiniteDimensional ℂ (W ⊗[ℂ] ℂ) ∧
    ∃ Λ : Representation ℂ (extended F E (Fin 1) n e D θ) (W ⊗[ℂ] ℂ),
      (∀ j : compact F E (Fin 1) n e D θ,
        Λ (PrescribedFieldType70.compactInclusion F E n
          (scalarMatrixEmbedding F E (Fin 1) n e)
          (compact F E (Fin 1) n e D θ) j) =
          (κ.tprod (Representation.trivial ℂ (compact F E (Fin 1) n e D θ) ℂ)) j) ∧
      IsIrreducible Λ ∧ SmoothDegreeQuotient.IsSmooth Λ ∧
      FullIntertwiningLocus69.locus (extended F E (Fin 1) n e D θ) Λ =
        (extended F E (Fin 1) n e D θ : Set (GL n F)) ∧
      Nontrivial (IndV (extended F E (Fin 1) n e D θ).subtype Λ) ∧
      SmoothDegreeQuotient.IsSmooth (ind (extended F E (Fin 1) n e D θ).subtype Λ) ∧
      IsIrreducible (ind (extended F E (Fin 1) n e D θ).subtype Λ) :=
```

10.1 Why orientation is load-bearing

For the literal pair group, $\iota(b) = g\iota(a)g^{-1}$. The first coordinate therefore belongs to

$$I_g = \{a : g\iota(a)g^{-1} \in \iota(K)\}, \qquad \iota(I_g) = \iota(K) \cap g^{-1}\iota(K)g.$$

The scalar intersection uses the same orientation. Under whole-action irreducibility and the original scalar/trace profile,

$$d^2 = [K : H].$$

The positive oriented index identity $[K : I_g] = [H : C_g]$ then yields $[I_g : C_g] = d^2$, giving the one-dimensional intertwining space. Schur is applied to the whole irreducible action, not to arbitrary intersection restrictions. Infinite subgroup index is encoded by zero in the relevant natural-valued index API, so positivity is essential to cancellation.

CHAPTER 11

Morita lattices and exact one-plus comparisons

The centralizer coefficient ring acts inside a potentially noncommutative endomorphism ring. It must not be treated as central in the full ambient algebra. Morita corner arguments produce bases, coordinate quotient equivalences and actual additive indices. Multiplicative one-plus indices require genuine group closure and a common deep ideal.

11.0.1 The actual Cartan index

Actual ring homomorphisms embed centralizer matrices in the endomorphism ring. The finite/full lattice and radical-commutation conditions are explicit. The index follows from actual coordinate conjugation and the preceding quotient computation. A numerical index formula is not inserted as a premise, but the arithmetic lattice instance must satisfy the printed hypotheses.

E31: `PaperXI.MoritaActualCartan110`, source lines 137–152.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `dd18d3a60333f0e737ef4ff22c34db295f49460de701a9fddf9fcd3dc22d293a`.

```
theorem relIndex_of_actual_cartan [Nonempty m]
    (L : Submodule [U+1D4AA][E] (Matrix m m (Module.End F E)))
    (hL : IsMatrixBimodule (coefficientMap F E) L.toAddSubgroup)
    (hfg : L.FG)
    (hfull : Submodule.span F (L : Set (Matrix m m (Module.End F E))) = ⊤)
    (hcomm : RadicalCommutes F E m L.toAddSubgroup)
    (y : GL m E) (c : Eˣ) (l r : GL m [U+1D4AA][E]) (a : m → ℕ)
    (hy : y = GeneralLinearGroup.scalar m c *
      (ActualIntegralGeneralLinear.embedding E m l *
        PaperXI.SplitRadical105.cartan (ActualLocalUnitDecomposition.uniformizer E : E)
          (ActualLocalUnitDecomposition.uniformizer_ne_zero E) a *
            ActualIntegralGeneralLinear.embedding E m r)) :
    (L.toAddSubgroup ⊓ L.toAddSubgroup.map
      (unitConjugation (centralizerEmbedding F E m y)).toAddMonoidHom).relIndex
        L.toAddSubgroup =
      Nat.card [U+1D4C0][E] ^ (Module.finrank F E * ∑ i, ∑ j, ((a i : ℤ) - (a j : ℤ)).toNat) :=
```

11.0.2 From additive lattices to unit groups

The source keeps the original integral-order ring and its actual one-plus subgroups. A common ideal and a finite quotient allow the exact coset comparison. An arbitrary additive lattice is not automatically a subgroup under $1 + x$, and a conjugate lattice need not be an ideal of the full field algebra. The theorem's carrier and group hypotheses prevent either substitution.

E32: `PaperXI.OnePlusIntersection110`, source lines 179–189.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `512e3b7a894d7db7474f8586c4c652dddd45bfd98197f637f7be8e5a8fec3e7e`.

```
theorem conjugate_intersection_relIndex_eq (f : R →+* S) (y : Sˣ) (I : Ideal R)
    (L : AddSubgroup R) (K : Subgroup Rˣ)
    (hK : ∀ g : K, ((g : Rˣ) : R) - 1 ∈ L)
    (hL : ∀ x : L, ∃ g : K, ((g : Rˣ) : R) = 1 + (x : R))
```

```
    (hI : I.toAddSubgroup ≤ pullbackLattice f L
      (L.map ((innerRing y).toRingHom.comp f).toAddMonoidHom))
    [Finite (L [U+29F8] I.toAddSubgroup.addSubgroupOf L)] :
    (K.map (Units.map ((innerRing y).toRingHom.comp f).toMonoidHom)).relIndex
        (K.map (Units.map f.toMonoidHom)) =
      (L.map ((innerRing y).toRingHom.comp f).toAddMonoidHom).relIndex
        (L.map f.toAddMonoidHom) :=
```

11.0.3 The original carrier after dictionary reductions

The literal type quotient/form, actual one-plus groups, Morita lattices and their coordinate matches are supplied. The source fixes the original character and beta extension. Trace support, covariance, finite quotient, degree-square and index equality are derived from those lower-level instances. The canonical type dictionary and arithmetic category/factor interpretation still require an actual instance.

E33: `PaperXI.ActualSourceCandidate111`, source lines 40–84.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `c802c3b3d1be72ca3d961a4edaa42cca066eb81cbc902a0a0abe9eee58d60005`.

```
theorem candidate_exists_from_source_dictionary
    (hF : Continuous (algebraMap F E))
    (hDo : IsOpen (D : Set (GL n F)))
    (hB : ∀ u : GL m [U+1D4AA][E],
      (coordinateAut e).symm (integralMatrix F E m u) ∈ CharacterNormalizer61.normalizer D θ)
    (hfull : ∀ g : GL n F,
      g ∈ CharacterNormalizer61.normalizer D θ ↔
        ∃ a : Eˣ, ∃ j : compact F E m n e D θ,
          unitsEmbedding F E n (scalarMatrixEmbedding F E m n e) a * (j : GL n F) = g)
    [IsIrreducible (κ.comp (firstSubgroup F E m n e D θ).subtype)] [IsIrreducible σ]
    (hκ : SmoothDegreeQuotient.IsSmooth κ)
    (hβ : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      NormalizerIntertwining60.IntertwinedAt (compact F E m n e D θ) κ g)
    (hσc : CuspidalCartan105.Cuspidal σ)
    (hc : ActualCharacterData108.Contains F E m n e D θ hDc κ)
    (hθsupport : ∀ g : GL n F,
      NormalizerIntertwining60.IntertwinedAt D (MonomialInduction.scalarRepresentation θ) g →
      ∃ j₁ j₂ : compact F E m n e D θ, ∃ y : GL m E,
        g = (j₁ : GL n F) * centralizerMap F E m n e y * (j₂ : GL n F))
    (τ : ((firstSubgroup F E m n e D θ) [U+29F8]
      actualScalarSubgroup F E m n e D θ) ≃* Multiplicative X)
    (B : X →ₗ[k] X →ₗ[k] k) (ψ : AddChar k ℂˣ) (hψ : ψ ≠ 1)
    (hnd : ∀ v : X, (∀ u : X, B u v = 0) → v = 0)
    (hform : ∀ x y : firstSubgroup F E m n e D θ,
      ψ (B (coordinate (actualScalarSubgroup F E m n e D θ) τ x)
        (coordinate (actualScalarSubgroup F E m n e D θ) τ y)) =
      characterCommutator (actualScalarSubgroup F E m n e D θ) τ
        (actualScalarCharacter F E m n e D θ) x y)
    (hθcentralizer : ∀ y : GL m E, NormalizerIntertwining60.IntertwinedAt D
      (MonomialInduction.scalarRepresentation θ) (centralizerMap F E m n e y))
    (data : ActualLatticeData F E m n e D θ) :
    FiniteDimensional ℂ (W ⊗[ℂ] U) ∧
    ∃ Λ : Representation ℂ (extended F E m n e D θ) (W ⊗[ℂ] U),
      (∀ j : compact F E m n e D θ,
        Λ (PrescribedFieldType70.compactInclusion F E n (scalarMatrixEmbedding F E m n e)
          (compact F E m n e D θ) j) =
          CompactTypeTensor58.action (reduction F E m n e D θ hmodel) κ σ j) ∧
      IsIrreducible Λ ∧ SmoothDegreeQuotient.IsSmooth Λ ∧
      FullIntertwiningLocus69.locus (extended F E m n e D θ) Λ =
        (extended F E m n e D θ : Set (GL n F)) ∧
      Function.Injective (TensorTypeOperators70.realize (extended F E m n e D θ) Λ) ∧
      Nontrivial (IndV (extended F E m n e D θ).subtype Λ) ∧
      SmoothDegreeQuotient.IsSmooth (ind (extended F E m n e D θ).subtype Λ) ∧
      IsIrreducible (ind (extended F E m n e D θ).subtype Λ) :=
```

CHAPTER 12

Complex scalar cochains and honest cyclic actions

Equal scalar commutators determine a complex scalar cocycle class under the stated divisible-coefficient hypotheses. This can construct an actual representation of a supplied group. The original retained finite-group lift and the arithmetic source attachment remain separate when more structure is prescribed.

12.0.1 What equal commutators supply over complex units

The two normalized cocycles have coefficients in $\mathbb{C}^\times$. Their commutators agree. A cochain is constructed whose coboundary gives the cocycle ratio. This scalar-pushout result does not identify distinct marked finite central extensions such as the dyadic retained-group examples.

E34: `PaperXI.ScalarCocycle112`, source lines 171–174.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `351ac3d1e21afeb0a8897f6eea1ba99f83bb63eaeee390eebefa44c1dfbb0f4e`.

```
theorem same_commutator_exists_cochain
    (c d : Cocycle A ℂˣ) (h : ∀ a b, commutator c a b = commutator d a b) :
    ∃ s : A → ℂˣ, s 1 = 1 ∧
      ∀ a b, c.coeff a b = (s a * s b / s (a * b)) * d.coeff a b :=
```

12.0.2 The equivariant actual-group action

The actual groups and their full automorphisms are supplied. The scalar bridge has been corrected using the explicit cyclic coprimality conditions. The result retains the original theta restriction and intertwines the prescribed packet generator. It requires full actions, not only an action on the symplectic quotient.

E35: `PaperXI.EquivariantRepresentation113`, source lines 67–91.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `f1cf8ff3343f2647c0994270f6de5bc3478925a25406ea3a221861807b08b6c9`.

```
theorem exists_equivariant_actual_action
    (D : Subgroup G) [D.Normal] (θ : D →* ℂˣ)
    (hinv : TypeSideDescent.Invariant D θ)
    (q : G →* A) (hker : q.ker = D)
    (t : A → G) (ht : ∀ a, q (t a) = a) (ht1 : t 1 = 1)
    (H : Subgroup P) [H.Normal] (χ : H →* ℂˣ)
    (hχinv : TypeSideDescent.Invariant H χ)
    (r : P →* A) (hkerP : r.ker = H)
    (u : A → P) (hu : ∀ a, r (u a) = a) (hu1 : u 1 = 1)
    (ρ : Representation ℂ P V) [IsIrreducible ρ]
    (hρ : ∀ h : H, ρ h = (χ h : ℂ) • LinearMap.id)
    (α : G ≃* G) (γ : P ≃* P) (β : A ≃* A)
    (hq : ∀ g, q (α g) = β (q g)) (hr : ∀ p, r (γ p) = β (r p))
    (hθ : ∀ d : D, θ (subgroupMap D q hker α β hq d) = θ d)
    (hχ : ∀ h : H, χ (subgroupMap H r hkerP γ β hr h) = χ h)
    (U : V ≃ₗ[ℂ] V) (hU : ∀ p v, U (ρ p v) = ρ (γ p) (U v))
    (n m : ℕ) (hα : α ^ n = 1) (hγ : γ ^ n = 1)
    (hm : ∀ a : A, a ^ m = 1) (hcop : n.Coprime m)
    (hcomm : ∀ a b,
      sectionCocycle D θ q hker t ht a b / sectionCocycle D θ q hker t ht b a =
```

```
        sectionCocycle H χ r hkerP u hu a b / sectionCocycle H χ r hkerP u hu b a) :
    ∃ Ξ : Representation ℂ G V,
      IsIrreducible Ξ ∧ TypeSideDescent.Contains D θ Ξ ∧
        (∀ d : D, Ξ d = (θ d : ℂ) • LinearMap.id) ∧
        (∀ g v, U (Ξ g v) = Ξ (α g) (U v)) :=
```

12.0.3 Determinant-one normalization

The generator implements the supplied full automorphism on the same irreducible carrier. Its cyclic order and carrier dimension satisfy the required coprimality. Scalar rescaling can impose order and determinant normalization with uniqueness at the stated scope. The theorem does not identify the implementer with the independently prescribed arithmetic beta extension.

E36: `PaperXI.CyclicNormalization113`, source lines 167–173.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `607587d94a544d2c4e014d81b21e4c9241d852feddfaa826df57f9715d396ce7`.

```
theorem normalized_generator_existsUnique (ρ : Representation ℂ P V) [IsIrreducible ρ]
    (α : MulAut P) (A : V ≃ₗ[ℂ] V)
    (hA : ∀ p v, A (ρ p v) = ρ (α p) (A v))
    (n : ℕ) (hn : 0 < n) (hα : α ^ n = 1)
    (hcop : Nat.Coprime n (Module.finrank ℂ V)) :
    ∃! B : V ≃ₗ[ℂ] V, B ^ n = 1 ∧ LinearEquiv.det B = 1 ∧
      ∀ p v, B (ρ p v) = ρ (α p) (B v) :=
```

CHAPTER 13

Conjecture 66 and the surviving conditional comparison

The finite bridge, the coherent graph equivalence, arithmetic realization and local-factor characterization are four different assertions. The final edition retains their order. Conjecture 66 concerns the general arithmetic realization step. The conditional comparison remains a theorem once the actual realization and the original factor characterization are supplied.

13.0.1 The factor profile must characterize the declared range

An atom type, its isomorphism setoid, a test type and value type are fixed. The record contains admissibility, invariant profiles and separation on that admissible range. The separation field is the external characterization input. In an arithmetic application, all normalizations and the original characterizing family must be instantiated. A finite-group character instance is an example of the abstract interface, not a local-Weil converse theorem.

E37: `PaperXI.LocalFactorIdentification`, source lines 14–18.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `3e81e1ce1c7b50d2771432e73d46f02de2a2ce4910e0a3420aa67e6b02a514f3`.

```
structure FactorSystem {Atom : Type*} (s : Setoid Atom) (Test Value : Type*) where
  admissible : Atom → Prop
  profile : Atom → Test → Value
  invariant : ∀ a b, s.r a b → profile a = profile b
  separates : ∀ a b, admissible a → admissible b → profile a = profile b → s.r a b

```

13.0.2 Existence is an explicit arithmetic field

The factor system and allowed datum type are fixed. A realizes relation connects each datum to possible actual atoms. The record's `exists_atom` supplies existence. Neither atom uniqueness nor reference agreement is stored here. General arithmetic existence of this record is the unresolved endpoint.

E38: `PaperXI.LocalFactorIdentification`, source lines 25–28.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `3e81e1ce1c7b50d2771432e73d46f02de2a2ce4910e0a3420aa67e6b02a514f3`.

```
structure Realization (D : Type*) where
  realizes : D → Atom → Prop
  exists_atom : ∀ d, ∃ a, realizes d a
  admissible : ∀ d a, realizes d a → S.admissible a

```

13.0.3 H5 acts on the same realized candidate

An actual realization is supplied. The realized atom's profile and the reference atom's profile agree with the same expected profile on the original test family. Separation then proves equality of atom classes. H5 must refer to this same chosen completion. Replacing it with a factor identity for another candidate would not discharge the displayed hypothesis.

E39: `PaperXI.LocalFactorIdentification`, source lines 49–55.

Contiguous source proof excerpt.

Source SHA-256: `3e81e1ce1c7b50d2771432e73d46f02de2a2ce4910e0a3420aa67e6b02a514f3`.

```
theorem candidate_eq_reference (d : D) (reference : Atom)
    (hr : S.admissible reference) (expected : Test → Value)
    (hreal : ∀ a, R.realizes d a → S.profile a = expected)
    (href : S.profile reference = expected) :
    candidate S R d = Quotient.mk s reference := by
  apply (class_eq_iff S (R.admissible d _ (chosenAtom_realizes S R d)) hr).mpr
  exact (hreal _ (chosenAtom_realizes S R d)).trans href.symm
```

13.0.4 The full formal conditional signature

The finite source data construct a tensor bridge. Honest actions, phase compatibility and matrix holonomy supply the coherent finite comparison. The allowed target diagram, an arithmetic realization, admissibility and factor profiles are then supplied explicitly. The conclusion contains a concrete compatible gauge, an equivalence of global section representations, and equality of the realized atom class with the reference. The source does not assert realization of every arbitrary finite datum. Its theorem name and successful Lean status must be read together with these hypotheses.

E40: `PaperXI.ConditionalAtomicComparison`, source lines 127–143.

Literal declaration excerpt. A proof/value after a top-level assignment is omitted; structure fields are retained.

Source SHA-256: `ce8d1ea0f8658d4d968aeb6665d5f1d3e4cc6bb1d472ebfbfbce6215f1103a58`.

```
theorem source {Atom Test Value : Type*} {atomIso : Setoid Atom}
    (sys : FactorSystem atomIso Test Value)
    (allowed : Set (Diagram X ℂ (CentralProduct.Group d.source [U+22CA][α] C)
      (PiTensorProduct ℂ Wb)))
    (ha : targetDiagram a0 ∈ allowed)
    (realization : Realization sys allowed)
    (reference : Atom) (href : sys.admissible reference)
    (expected : Test → Value)
    (hfactors : ∀ b, realization.realizes (targetDiagram a0, ha) b →
      sys.profile b = expected)
    (hreference : sys.profile reference = expected) :
    (∃ F : X → PiTensorProduct ℂ Vb ≃ₗ[ℂ] PiTensorProduct ℂ Wb,
      F a0.root = FiniteBreakAssembly.tensorBridge d ∧
      EdgeCompatible a0.sourceEdge a0.targetEdge F ∧
      RepresentationGraph.Intertwines a0.left a0.right F) ∧
    Nonempty (Representation.Equiv (leftSections a0) (rightSections a0)) ∧
      candidate sys realization (targetDiagram a0, ha) = Quotient.mk atomIso reference :=
```

13.1 The publication statement

Fix the declared class $\mathcal{A}$ of wild atoms and the paper's normalizations. For every $\rho \in \mathcal{A}$ with supplied H1–H4 assembled finite packet datum d_ρ, Conjecture 66 asserts that there exists one actual admissible supercuspidal representation π of $\mathrm{GL}_{\dim \rho}(F)$ together with an arithmetic realization witness $R(d_\rho, \pi)$, recovering the prescribed packet and preserving the declared stabilizer action, central-extension/phase data, and compatible transports. This witness uses actual arithmetic inducing data and explicit comparison/recovery. It does not require the whole assembled packet to be the original maximal-simple-type quotient.

Given that same π and realization witness, the conditional comparison proposition assumes original H5 for π on the stipulated characterizing family with the fixed normalization. The supplied characterization then gives $\pi \simeq \mathrm{LLC}(\rho)$. This proposition has no claim that H5 produces a candidate, and no requirement that H5 hold for all possible realizers.

The retained general Lean interface has an *all-realizers* profile binder because its `Realization` relation may have several realizers. The publication proposition needs the factor equality only for its one supplied candidate. The pointwise separation lemma `class_eq_iff` gives that implication directly. The broader interface should not be misread as strengthening the publication conjecture.

The traditional paper gives the realization definition, the declared source class with its supplied finite hypothesis data, and the retained hypothesis numbering. The release records the broad arithmetic realization ambition as a conjecture, the factor comparison as a conditional theorem, and the restricted constructions of Blocks 166–167 as ordinary mathematical realization results within their printed hypotheses. The guide does not add a Lean axiom named after the conjecture and does not substitute a `sorry` declaration for a proof.

The restricted constructions are not Lean-certified local-field/simple-character developments. Their proof notes and audits accompany the release. They retain the same-candidate H5 boundary. The dyadic low-rank enlargement changes the arithmetic assembly carrier while preserving the original candidate through induction in stages; it must not be described as enlarging a fixed maximal-type quotient that the dimension calculation obstructs.

CHAPTER 14

Verification lineage and reproducibility

The retained verification reports certify specific source revisions and specific dependency closures. A full project-source scan, a focused compiler run, a transitive axiom audit, and a kernel replay are distinct checks. The evidence remains attributable to its original layer rather than collapsed into one global PASS flag.

14.1 Recorded scopes

The v0.44 record checks the new ramified/unramified quotient module, its 28 public declarations, focused imports and same-kernel replay. Its source-only reconstruction freshly builds the 20-module required project closure, reusing pinned Lean and Mathlib objects. The cumulative source scan covers 302 production modules.

The normalized-Frobenius record checks three new modules and 50 public declarations in two source states. Its clean project closure has 29 modules and agrees byte-for-byte with the primary object closure. It explicitly records that the full Block45 arithmetic/automorphic contract is not established.

Continuation110 records 20 source-set modules and 223 audited public exports. Continuation111 records nine new modules and 78 audited exports. Continuation112 records three new modules and 60 exports. Continuation113 records four modules and 76 exports, with a separately recorded internal compile/kernel audit. In each continuation, inherited modules and Mathlib are authenticated or reused at the documented scope rather than rebuilt in full.

14.2 Failures remain part of the evidence

Continuation110 retains an out-of-memory full-baseline attempt under an 8 GiB virtual-address-space cap and an interrupted larger attempt. The passing bounded new-module replay is not a claim that those failed full-baseline attempts succeeded.

Continuation113 retains the rejected theta0 arithmetic attachment, a corrected sign convention, failed harness/parser attempts and an overwritten early arithmetic receipt disclosed as unavailable. Its final corrected replay is a new accepted run, not a reconstruction of that missing receipt. The finite formal results and the failed arithmetic candidate are different claims.

14.3 Portable runtime

The preserved portable-runtime README describes an offline Linux x86_64 kit for Lean 4.34.0-rc2 and the pinned Mathlib revision. Its full executable distribution is an external dependency where the source handoff does not include it. Native Windows launchers and the Linux/WSL path have different prerequisites. The release documentation gives the exact retained resources and availability.

A reader should first authenticate the runtime, Mathlib revision and branch archive hashes, then replay in a fresh output directory. Do not overwrite original receipts. The documented Continuation113 verifier checks the baseline110–112 archives and inherited objects, compiles only its four new modules, audits named exports and runs `leanchecker`. It does not claim to rebuild every historical Paper XI module or all of Mathlib.

```
python scripts/verify_new113.py \
  --kit /absolute/path/to/Komorebi_Portable_Lean_4.34.0-rc2 \
```

```
  --archives /absolute/path/to/baseline_archives \
  --out evidence/replay_on_new_machine
```

14.4 Rebuilding this guide

The editable folder contains the LaTeX source, displayed and raw excerpts, complete selected Lean source copies, source catalog and this PDF builder. From the checkpoint root, run the following command with a new output directory outside the checkpoint. XeLaTeX and the declared fonts/packages must already be installed. The builder installs nothing, uses no network and fixes the source date epoch for reproducibility.

```
python3 tools/build_readers.py --output /absolute/path/new-output
```

APPENDIX A

Complete selected named-source catalog

Every selected module below is copied unchanged into the editable guide's `sources/PaperXI` folder. The catalog gives module identity, selected layer, SHA-256, direct project imports, module documentation and every named lexical declaration with its raw source line. These line pins locate declarations in the complete source, not independently compiled snippets. Earlier repeated source-layer identities remain in `SOURCE_VERSIONS.json`.

A.1 `PaperXI.AbstractHeisenberg`

Selected layer: v0.44 parent. **Source lines:** 184.

SHA-256: `d83045df228867234c49fb9dd9e838bbf810394614dd22dce2401c6744e3dacd`.

```
Actual Heisenberg groups on abstract bilinear spaces, their central extensions,
and center-preserving lifts of arbitrary form-preserving linear equivalences.
No choice of symplectic coordinates is used.
```

Direct project imports: `PaperXI.CentralExtension`.

Line	Kind	Name
18	structure	`PaperXI.AbstractHeisenberg.Element`
24	instance	`PaperXI.AbstractHeisenberg.instMul`
28	instance	`PaperXI.AbstractHeisenberg.instOne`
33	instance	`PaperXI.AbstractHeisenberg.instInv`
36	theorem	`PaperXI.AbstractHeisenberg.mul_vector`
37	theorem	`PaperXI.AbstractHeisenberg.mul_central`
39	theorem	`PaperXI.AbstractHeisenberg.one_vector`
40	theorem	`PaperXI.AbstractHeisenberg.one_central`
41	theorem	`PaperXI.AbstractHeisenberg.inv_vector`
42	theorem	`PaperXI.AbstractHeisenberg.inv_central`
45	instance	`PaperXI.AbstractHeisenberg.instGroup`
55	def	`PaperXI.AbstractHeisenberg.coordinates`
61	instance	`PaperXI.AbstractHeisenberg.instFinite`
65	theorem	`PaperXI.AbstractHeisenberg.finite_of_finite_field`
70	def	`PaperXI.AbstractHeisenberg.centralElement`
72	theorem	`PaperXI.AbstractHeisenberg.centralElement_vector`
73	theorem	`PaperXI.AbstractHeisenberg.centralElement_central`
74	theorem	`PaperXI.AbstractHeisenberg.centralElement_zero`
75	theorem	`PaperXI.AbstractHeisenberg.centralElement_mul`
79	theorem	`PaperXI.AbstractHeisenberg.centralElement_commute`
84	def	`PaperXI.AbstractHeisenberg.inclusion`
90	def	`PaperXI.AbstractHeisenberg.quotient`
95	theorem	`PaperXI.AbstractHeisenberg.inclusion_apply`
97	theorem	`PaperXI.AbstractHeisenberg.quotient_apply`
101	def	`PaperXI.AbstractHeisenberg.datum`
121	theorem	`PaperXI.AbstractHeisenberg.datum_i`

Line	Kind	Name
122	theorem	`PaperXI.AbstractHeisenberg.datum_q`
125	theorem	`PaperXI.AbstractHeisenberg.inv_central_of_isAlt`
129	theorem	`PaperXI.AbstractHeisenberg.mul_swap`
140	theorem	`PaperXI.AbstractHeisenberg.commutator`
151	def	`PaperXI.AbstractHeisenberg.lift`
163	theorem	`PaperXI.AbstractHeisenberg.lift_vector`
167	theorem	`PaperXI.AbstractHeisenberg.lift_central`
171	theorem	`PaperXI.AbstractHeisenberg.lift_inclusion`
177	theorem	`PaperXI.AbstractHeisenberg.lift_quotient`

A.2 `PaperXI.AbstractHeisenbergExtension`

Selected layer: v0.44 parent. **Source lines:** 58.

SHA-256: `64c993ffe68d2705763bdb7173dcdb8096ddb3540a82363620d5ac0e820f8d06`.

```
The actual commutator of the abstract odd-characteristic Heisenberg extension
is the supplied alternating form, independently of the chosen quotient section.
```

Direct project imports: `PaperXI.AbstractHeisenberg`, `PaperXI.CentralExtensionCocycle`, `PaperXI.SymplecticCentralExtension`.

Line	Kind	Name
15	theorem	`PaperXI.AbstractHeisenberg.centralCommutator_eq`
23	theorem	`PaperXI.AbstractHeisenberg.scalar_commutator`
33	theorem	`PaperXI.AbstractHeisenberg.scalar_radical_eq_bot`
43	theorem	`PaperXI.AbstractHeisenberg.scalar_radical_eq_bot_scalarCharacter`
52	theorem	`PaperXI.AbstractHeisenberg.center_eq_inclusion_range`

A.3 `PaperXI.ActualCartanCandidate106`

Selected layer: base108 arithmetic branch. **Source lines:** 79.

SHA-256: `e87ad679e0338ba7b5ffda904079152531d758cd29a472151aa21431bc543d06`.

```
The same candidate constructor with Cartan decomposition discharged from
actual local-field structure. The BK tensor reduction remains explicit.
```

Direct project imports: `PaperXI.ActualLocalCartan106`, `PaperXI.ActualCuspidalCandidate105`.

Line	Kind	Name
31	theorem	`PaperXI.ActualCartanCandidate106.candidate_exists`

A.4 `PaperXI.ActualCharacterCandidate108`

Selected layer: base108 arithmetic branch. **Source lines:** 89.

SHA-256: `a311e3796f48d7d7a6c2cafd95e7ff962b8fb7543a0cdb7a81d04d9d12dbe4f2`.

```
Character attachment removes a redundant beta-support input. Character
support and the Heisenberg Hom dimension are still arithmetic hypotheses.
```

Direct project imports: `PaperXI.ActualCharacterData108`.

Line	Kind	Name
31	theorem	`PaperXI.ActualCharacterCandidate108.candidate_exists`

A.5 `PaperXI.ActualCharacterData108`

Selected layer: base108 arithmetic branch. **Source lines:** 49.

SHA-256: `adcb8f31671a6ce4587678730fd689cdda4c87ebd3f1c8c67391cf42ee921bc4`.

```
Character occurrence on the actual prescribed kappa carrier, with its
normality and invariance derived from the literal character normalizer.
```

Direct project imports: `PaperXI.ActualHeisenbergCandidate107`, `PaperXI.CharacterSupport108`.

Line	Kind	Name
22	theorem	`PaperXI.ActualCharacterData108.domain_le_compact`
26	def	`PaperXI.ActualCharacterData108.intoCompact`
30	theorem	`PaperXI.ActualCharacterData108.intoCompact_mem_first`
37	def	`PaperXI.ActualCharacterData108.Contains`
41	theorem	`PaperXI.ActualCharacterData108.scalar_restriction`

A.6 `PaperXI.ActualCharacterIdentification61`

Selected layer: base108 arithmetic branch. **Source lines:** 100.

SHA-256: `8507c7ccc941069745d0352c942ee3b600e8fb68441dc2ecd4d458f31ce367fc`.

```
# Identification with the supplied genuine compact type group

The exact standard-theory input is the actual full character-normalizer
formula N(theta)=E^× J0, together with integral E-units in J0 and preservation
of the lattice chain. These imply that the constructed compact intersection
is literally J0. No equality J1=U_a1 is used or concluded.
```

Direct project imports: `PaperXI.ActualCharacterType61`.

Line	Kind	Name
27	theorem	`PaperXI.ActualCharacterIdentification61.compactGroup_eq_of_full_normalizer`
59	theorem	`PaperXI.ActualCharacterIdentification61.uniformizer_mem_full_normalizer`
69	theorem	`PaperXI.ActualCharacterIdentification61.extendedSubgroup_eq_full_normalizer`

A.7 `PaperXI.ActualCharacterResidue61`

Selected layer: base108 arithmetic branch. **Source lines:** 133.

SHA-256: `401e10d5858bab78cc6a399bac1bd8c3d594a891773157e9ff975c2a7f0b805f`.

```
# Actual residue reduction on the theta-dependent compact group

The map is the restriction of the already constructed literal lattice residue
map. Its kernel is the actual intersection C(theta)nU_a1. Containment in the
lattice-product group and normalization of theta by integral centralizer
matrices are explicit standard-type inputs. They do not assert C(theta)=B U_a1.
```

Direct project imports: `PaperXI.ActualCharacterType61`, `PaperXI.ActualELatticeFieldJoin59`.

Line	Kind	Name
29	def	`PaperXI.ActualCharacterResidue61.intoModel`
35	def	`PaperXI.ActualCharacterResidue61.reduction`
40	theorem	`PaperXI.ActualCharacterResidue61.reduction_kernel`
48	theorem	`PaperXI.ActualCharacterResidue61.integralMatrix_mem_compactGroup`
60	theorem	`PaperXI.ActualCharacterResidue61.integralUnit_mem_compactGroup`
76	theorem	`PaperXI.ActualCharacterResidue61.reduction_surjective`
110	theorem	`PaperXI.ActualCharacterResidue61.reduction_conjugation`

A.8 `PaperXI.ActualCharacterType61`

Selected layer: base108 arithmetic branch. **Source lines:** 163.

SHA-256: `31cbff49441e1406a497ce67cbba0965509582286af1f2fc8337525909e25c1f`.

```
# The compact lattice part of an actual character normalizer

This group depends on the actual subgroup-character pair. It is not the larger
literal lattice-product group from Block59. Its first subgroup is its actual
intersection with adjacent-lattice congruence, never an equality with all U1.
```

Direct project imports: `PaperXI.CharacterNormalizer61`, `PaperXI.ActualELatticeTopology60`.

Line	Kind	Name
23	theorem	`PaperXI.ActualCharacterType61.power_normalizes_chainStabilizer`
53	def	`PaperXI.ActualCharacterType61.chainGroup`
58	theorem	`PaperXI.ActualCharacterType61.matrixCongruence_le_chainGroup`
62	theorem	`PaperXI.ActualCharacterType61.chainGroup_open`
68	theorem	`PaperXI.ActualCharacterType61.chainGroup_compact`
83	theorem	`PaperXI.ActualCharacterType61.coordinateAut_unitsEmbedding`
96	theorem	`PaperXI.ActualCharacterType61.coordinateAut_embeddedUniformizer`
110	theorem	`PaperXI.ActualCharacterType61.uniformizer_normalizes_chainGroup`
126	def	`PaperXI.ActualCharacterType61.compactGroup`
130	def	`PaperXI.ActualCharacterType61.firstSubgroup`
135	theorem	`PaperXI.ActualCharacterType61.domain_le_compactGroup`
140	theorem	`PaperXI.ActualCharacterType61.compactGroup_compact_open`
148	theorem	`PaperXI.ActualCharacterType61.firstSubgroup_open`
154	theorem	`PaperXI.ActualCharacterType61.uniformizer_normalizes_compactGroup`

A.9 `PaperXI.ActualCompactTypeCompletion58`

Selected layer: base108 arithmetic branch. **Source lines:** 97.

SHA-256: `9c0e0c63cd7aa7f6c5ce5841cf9dbd3b97709ee0300314cb808dae8caef66b2a`.

```
# Same-factor actual compact type extension

The literal tensor action retains the prescribed Heisenberg factor. Its
irreducibility and smoothness, and the whole-type normalizer intertwiner, are
constructed. Actual embedded-field arithmetic supplies cyclic freeness.
Normalization of the actual compact subgroup, quotient-triviality, and one
nonzero first-factor intertwiner remain explicit type-theoretic inputs.
```

Direct project imports: `PaperXI.ActualNormalizerTypeExtension58`, `PaperXI.CompactTypeTensorNormalizer58`.

Line	Kind	Name
25	theorem	`PaperXI.ActualCompactTypeCompletion58.firstFactor_irreducible`
53	theorem	`PaperXI.ActualCompactTypeCompletion58.completed`

A.10 `PaperXI.ActualCoordinateDeepIdeal111`

Selected layer: Continuation111. **Source lines:** 80.

SHA-256: `1478145467bf2f189641de88b717440f1439dc649d84164ad972b32966afeb91`.

```
The actual integral coordinate-order instance and its one-plus index
application. Ideal containment and quotient finiteness are conclusions.
```

Direct project imports: `PaperXI.CoordinateDeepIdeal111`.

Line	Kind	Name
14	def	`PaperXI.ActualCoordinateDeepIdeal111.innerHomeomorph`
26	def	`PaperXI.ActualCoordinateDeepIdeal111.integralEmbedding`
29	theorem	`PaperXI.ActualCoordinateDeepIdeal111.integralEmbedding_open`
37	theorem	`PaperXI.ActualCoordinateDeepIdeal111.conjugate_lattice_open`
49	theorem	`PaperXI.ActualCoordinateDeepIdeal111.exists_conjugate_common_ideal`
66	theorem	`PaperXI.ActualCoordinateDeepIdeal111.onePlus_conjugate_relIndex`

A.11 `PaperXI.ActualCuspidalCandidate105`

Selected layer: base108 arithmetic branch. **Source lines:** 118.

SHA-256: `b37fd23618fec1d17f60baca5392c3d0d4e8c255da68dfff131b5ca746840100`.

```
The actual character-dependent constructor of Block 70 now returns an
irreducible induced candidate, not a conditional promise requiring its full
intertwining locus. The residual cuspidality, arithmetic Cartan decomposition,
and BK tensor-to-residue reduction remain explicit.
```

Direct project imports: `PaperXI.FullTypeIntertwining105`, `PaperXI.ActualPacketTypeRealization70`.

Line	Kind	Name
26	def	`PaperXI.ActualCuspidalCandidate105.centralizerMap`
29	abbrev	`PaperXI.ActualCuspidalCandidate105.compact`
30	abbrev	`PaperXI.ActualCuspidalCandidate105.extended`
36	theorem	`PaperXI.ActualCuspidalCandidate105.centralizer_scalar`
54	theorem	`PaperXI.ActualCuspidalCandidate105.candidate_exists`

A.12 `PaperXI.ActualDepthZeroExtension`

Selected layer: base108 arithmetic branch. **Source lines:** 155.

SHA-256: `ff5750ec70106f6aa0ce85b66fc16095d1d17973b914f7b37b2e2c1efdcda17c`.

```
# Actual depth-zero inducing data extended by the scalar uniformizer

The subgroup, degree, free central direction and smooth inducing representation
are constructed from the given local field. The supplied finite representation
is inflated through literal residue reduction. Induced irreducibility is a
separate theorem: no Mackey vanishing is assumed in this construction.
```

Direct project imports: `PaperXI.ActualIntegralGeneralLinear`, `PaperXI.ActualLocalUnitDecomposition`, `PaperXI.CentralExtensionInductionConsequences`.

Line	Kind	Name
25	def	`PaperXI.ActualDepthZeroExtension.determinantDegree`
29	def	`PaperXI.ActualDepthZeroExtension.scalarUniformizer`
32	theorem	`PaperXI.ActualDepthZeroExtension.scalarUniformizer_central`
35	theorem	`PaperXI.ActualDepthZeroExtension.determinantDegree_embedding`
41	theorem	`PaperXI.ActualDepthZeroExtension.integralSubgroup_le_degree_kernel`
46	theorem	`PaperXI.ActualDepthZeroExtension.determinantDegree_scalarUniformizer`
56	theorem	`PaperXI.ActualDepthZeroExtension.scalarUniformizer_free`
66	abbrev	`PaperXI.ActualDepthZeroExtension.extendedSubgroup`
71	theorem	`PaperXI.ActualDepthZeroExtension.integral_le_extended`
76	theorem	`PaperXI.ActualDepthZeroExtension.scalar_unit_mem_integral`
83	theorem	`PaperXI.ActualDepthZeroExtension.extendedSubgroup_eq_scalar_sup`
106	theorem	`PaperXI.ActualDepthZeroExtension.extendedSubgroup_open`
111	abbrev	`PaperXI.ActualDepthZeroExtension.inclusion`
118	abbrev	`PaperXI.ActualDepthZeroExtension.extendedRepresentation`
122	theorem	`PaperXI.ActualDepthZeroExtension.extendedRepresentation_restrict`
126	theorem	`PaperXI.ActualDepthZeroExtension.extendedRepresentation_irreducible`
131	theorem	`PaperXI.ActualDepthZeroExtension.extendedRepresentation_smooth`
137	abbrev	`PaperXI.ActualDepthZeroExtension.inducedRepresentation`
142	theorem	`PaperXI.ActualDepthZeroExtension.induced_smooth`
146	theorem	`PaperXI.ActualDepthZeroExtension.induced_nontrivial`
151	theorem	`PaperXI.ActualDepthZeroExtension.induced_scalar`

A.13 `PaperXI.ActualELatticeFieldJoin59`

Selected layer: base108 arithmetic branch. **Source lines:** 80.

SHA-256: `cd8801f07f19c6130f8e0cb4b83972ac28c399481be71c0f8cee79f7e9bbd660`.

```
# The literal field-times-lattice inducing subgroup

Integral extension-field units already lie in the actual integral matrix
factor. The unit/uniformizer decomposition of E× therefore identifies the
subgroup generated by J and pi_E with the subgroup generated by J and E×.
```

Direct project imports: `PaperXI.ActualELatticeQuotient59`, `PaperXI.ActualELatticeFreeness59`.

Line	Kind	Name
24	theorem	`PaperXI.ActualELatticeFieldJoin59.integralMatrix_scalar`
34	theorem	`PaperXI.ActualELatticeFieldJoin59.integralUnit_mem_typeGroup`
40	theorem	`PaperXI.ActualELatticeFieldJoin59.extendedSubgroup_eq_field_join`
60	theorem	`PaperXI.ActualELatticeFieldJoin59.field_normalizes_typeGroup`
71	theorem	`PaperXI.ActualELatticeFieldJoin59.extendedSubgroup_coe_eq_field_mul`

A.14 PaperXI.ActualELatticeFreeness59

Selected layer: base108 arithmetic branch. **Source lines:** 73.

SHA-256: `b9c837f72cdde593dc7a1bf64f366e0e5453e390448eedf225b0024251c84a47`.

```
# Uniformizer freeness for the actual lattice-product group

Freeness is proved directly from the integral lattice and the strict valuation
of the actual extension-field uniformizer. It does not assume compactness of
the group or a matrix determinant identity.
```

Direct project imports: `PaperXI.ActualELatticeTypeGroup59`.

Line	Kind	Name
21	def	`PaperXI.ActualELatticeFreeness59.scalarAction`
28	theorem	`PaperXI.ActualELatticeFreeness59.scalarAction_apply`
33	theorem	`PaperXI.ActualELatticeFreeness59.uniformizerAction_zpow_apply`
41	theorem	`PaperXI.ActualELatticeFreeness59.typeGroup_le_stabilizer`
49	theorem	`PaperXI.ActualELatticeFreeness59.uniformizerAction_exponent_nonneg`
66	theorem	`PaperXI.ActualELatticeFreeness59.uniformizerAction_free`

A.15 PaperXI.ActualELatticeNormalizer59

Selected layer: base108 arithmetic branch. **Source lines:** 129.

SHA-256: `7328521c5eb95fe067aaae8fbb771a976d75965543e6cb1619720f6df9d68c17`.

```
# Periodic lattices and their literal first congruence group

The chain used below is the actual sequence of scalar-uniformizer translates
of the integral lattice in `E^m`, viewed as a vector space over `K`. Its first
congruence group consists of automorphisms preserving every lattice and acting
as the identity modulo the next lattice. Uniformizer normalization is proved
from the coordinate definitions. This is not an identification with the
simple-character group of an arbitrary stratum.
```

Direct project imports: `PaperXI.ActualIntegralGeneralLinear`, `PaperXI.ActualLocalUnitDecomposition`, `PaperXI.NormalizerExtensionInducing58`.

Line	Kind	Name
24	def	`PaperXI.ActualELatticeNormalizer59.chainStabilizer`
35	def	`PaperXI.ActualELatticeNormalizer59.firstCongruence`
59	theorem	`PaperXI.ActualELatticeNormalizer59.firstCongruence_le`
63	theorem	`PaperXI.ActualELatticeNormalizer59.stabilizer_normalizes_firstCongruence`
84	def	`PaperXI.ActualELatticeNormalizer59.periodicChain`
88	theorem	`PaperXI.ActualELatticeNormalizer59.periodicChain_shift`
96	theorem	`PaperXI.ActualELatticeNormalizer59.power_normalizes_firstCongruence`

A.16 PaperXI.ActualELatticeQuotient59

Selected layer: base108 arithmetic branch. **Source lines:** 89.

SHA-256: `a49f9630deecf93df96b4f831fb702cd48b0154c5563d3fe4f919ad2c9f47d24`.

```
# Finite residue quotient of the literal extension-field lattice product

For the actual subgroup `GL_n(O_E) U^1` in `Aut_K(E^n)`, the quotient by
literal adjacent-lattice congruence is `GL_n(k_E)`. Conjugation by the
extension-field uniformizer is the identity on this quotient.
```

Direct project imports: `PaperXI.ActualELatticeTypeGroup59`, `PaperXI.ProductNormalizerQuotient59`.

Line	Kind	Name
22	def	`PaperXI.ActualELatticeQuotient59.typeCongruence`
25	instance	`PaperXI.ActualELatticeQuotient59.typeCongruence_normal`
29	def	`PaperXI.ActualELatticeQuotient59.residueEquiv`
37	def	`PaperXI.ActualELatticeQuotient59.reduction`
40	theorem	`PaperXI.ActualELatticeQuotient59.reduction_surjective`
43	theorem	`PaperXI.ActualELatticeQuotient59.reduction_kernel`
50	theorem	`PaperXI.ActualELatticeQuotient59.finite_quotient`
54	theorem	`PaperXI.ActualELatticeQuotient59.uniformizer_centralizes`
62	theorem	`PaperXI.ActualELatticeQuotient59.uniformizer_normalizes_congruence`
68	theorem	`PaperXI.ActualELatticeQuotient59.uniformizer_normalizes`
75	def	`PaperXI.ActualELatticeQuotient59.uniformizerConjugation`
81	theorem	`PaperXI.ActualELatticeQuotient59.reduction_conjugation`

A.17 `PaperXI.ActualELatticeResidue59`

Selected layer: base108 arithmetic branch. **Source lines:** 157.

SHA-256: `7adc864addc36f19b3d94431dcbaa62f118e45633f9bc356692c1c8c0698d5a3`.

```
# Literal residue action on an extension-field lattice

The group and the congruence kernel below are the actual integral matrices on
`E^n`, viewed as `K`-linear operators. The next lattice is multiplication of the
integral lattice by the chosen actual uniformizer.
```

Direct project imports: `PaperXI.ActualIntegralGeneralLinear`, `PaperXI.ActualLocalUnitDecomposition`.

Line	Kind	Name
20	def	`PaperXI.ActualELatticeResidue59.restrictScalarMatrix`
25	theorem	`PaperXI.ActualELatticeResidue59.restrictScalarMatrix_apply`
28	theorem	`PaperXI.ActualELatticeResidue59.restrictScalarMatrix_injective`
37	def	`PaperXI.ActualELatticeResidue59.integralMatrix`
40	theorem	`PaperXI.ActualELatticeResidue59.integralMatrix_injective`
44	def	`PaperXI.ActualELatticeResidue59.integralLattice`
51	theorem	`PaperXI.ActualELatticeResidue59.mem_integralLattice`
55	theorem	`PaperXI.ActualELatticeResidue59.integralLattice_coe`
58	theorem	`PaperXI.ActualELatticeResidue59.integralMatrix_coe`
64	theorem	`PaperXI.ActualELatticeResidue59.integralMatrix_preserves`
74	def	`PaperXI.ActualELatticeResidue59.nextLattice`
79	theorem	`PaperXI.ActualELatticeResidue59.mem_nextLattice`
84	theorem	`PaperXI.ActualELatticeResidue59.coe_mem_nextLattice_iff`
111	theorem	`PaperXI.ActualELatticeResidue59.reduction_kernel_iff`
139	def	`PaperXI.ActualELatticeResidue59.integralImage`
141	def	`PaperXI.ActualELatticeResidue59.integralImageEquiv`
144	def	`PaperXI.ActualELatticeResidue59.residueAction`

Line	Kind	Name
147	theorem	`PaperXI.ActualELatticeResidue59.residueAction_surjective`
152	def	`PaperXI.ActualELatticeResidue59.residueQuotientEquiv`

A.18 `PaperXI.ActualELatticeScalarPeriod60`

Selected layer: base108 arithmetic branch. **Source lines:** 87.

SHA-256: `10bfffe3207aaacbc9291830df80f34c502d7383c79a8a9e1b7b723e7e3021f0`.

```
# A base-field scalar period for the literal extension-field lattice chain

The actual base-field uniformizer has nonzero extension-field degree. Otherwise
its image would be an integral extension-field unit, whose matrix determinant
degree vanishes, contradicting its known scalar-matrix degree. Choosing it or
its inverse supplies a positive period without assuming a ramification formula.
```

Direct project imports: `PaperXI.ActualELatticeFieldJoin59`, `PaperXI.ActualEmbeddedUniformizer58`.

Line	Kind	Name
26	theorem	`PaperXI.ActualELatticeScalarPeriod60.base_uniformizer_degree_ne_one`
42	theorem	`PaperXI.ActualELatticeScalarPeriod60.exists_positive_base_normalForm`
67	theorem	`PaperXI.ActualELatticeScalarPeriod60.chain_period_of_normalForm`
81	theorem	`PaperXI.ActualELatticeScalarPeriod60.exists_chain_period`

A.19 `PaperXI.ActualELatticeTopology60`

Selected layer: base108 arithmetic branch. **Source lines:** 170.

SHA-256: `4936918f1210fcc205c45cc2b2e787c6dfa51e1d1399c1c9e3988e6d9e791981`.

```
# Compact open topology for the literal extension-field lattice type group

The exact Block59 lattice groups are realized in actual `GL_d(K)` through
finite continuous coordinates on `E^n`. Compactness, openness, and the open
kernel of actual residue reduction are proved. No identification with the
simple-character group of an arbitrary stratum is asserted.
```

Direct project imports: `PaperXI.LatticeCongruenceTopology60`, `PaperXI.ActualELatticeScalarPeriod60`.

Line	Kind	Name
29	theorem	`PaperXI.ActualELatticeTopology60.field_nhdsNE_neBot`
41	def	`PaperXI.ActualELatticeTopology60.coordinatesOfBasis`
56	def	`PaperXI.ActualELatticeTopology60.scalarMatrixEmbedding`
62	theorem	`PaperXI.ActualELatticeTopology60.integralLattice_compact`
67	theorem	`PaperXI.ActualELatticeTopology60.integralLattice_open`
74	theorem	`PaperXI.ActualELatticeTopology60.latticeChain_compact`
86	theorem	`PaperXI.ActualELatticeTopology60.latticeChain_open`
95	def	`PaperXI.ActualELatticeTopology60.matrixType`
99	def	`PaperXI.ActualELatticeTopology60.matrixCongruence`
105	theorem	`PaperXI.ActualELatticeTopology60.matrixCongruence_open`
115	theorem	`PaperXI.ActualELatticeTopology60.matrixType_open`
124	theorem	`PaperXI.ActualELatticeTopology60.matrixType_compact`

Line	Kind	Name
139	theorem	`PaperXI.ActualELatticeTopology60.matrixCongruence_compact`
148	theorem	`PaperXI.ActualELatticeTopology60.compact_open_of_basis`
155	def	`PaperXI.ActualELatticeTopology60.finiteExtensionCoordinates`
161	theorem	`PaperXI.ActualELatticeTopology60.finiteExtension_compact_open`

A.20 `PaperXI.ActualELatticeTypeGroup59`

Selected layer: base108 arithmetic branch. **Source lines:** 136.

SHA-256: `985e4b5cffb7f0083ef712004a55d0dd95409019eac2f83d57c77cd971ef0455`.

```
# The actual lattice-product type group

The lattice and its first congruence subgroup are literal subgroups of
`Aut_K(E^n)`. The centralizer integral matrices normalize congruence, and their
intersection with congruence is exactly their residue kernel.
```

Direct project imports: `PaperXI.ActualELatticeNormalizer59`, `PaperXI.ActualELatticeResidue59`.

Line	Kind	Name
22	def	`PaperXI.ActualELatticeTypeGroup59.uniformizerAction`
27	theorem	`PaperXI.ActualELatticeTypeGroup59.uniformizerAction_apply`
31	def	`PaperXI.ActualELatticeTypeGroup59.latticeChain`
35	theorem	`PaperXI.ActualELatticeTypeGroup59.latticeChain_zero`
40	theorem	`PaperXI.ActualELatticeTypeGroup59.latticeChain_one`
56	theorem	`PaperXI.ActualELatticeTypeGroup59.integralMatrix_preserves_iff`
65	theorem	`PaperXI.ActualELatticeTypeGroup59.integralMatrix_commute`
74	theorem	`PaperXI.ActualELatticeTypeGroup59.integralMatrix_mem_stabilizer`
83	def	`PaperXI.ActualELatticeTypeGroup59.congruence`
87	theorem	`PaperXI.ActualELatticeTypeGroup59.integralImage_normalizes`
94	theorem	`PaperXI.ActualELatticeTypeGroup59.integralMatrix_mem_congruence_iff`
124	theorem	`PaperXI.ActualELatticeTypeGroup59.congruence_subgroupOf_integralImage`
133	def	`PaperXI.ActualELatticeTypeGroup59.typeGroup`

A.21 `PaperXI.ActualEmbeddedUniformizer58`

Selected layer: base108 arithmetic branch. **Source lines:** 177.

SHA-256: `3f8d9f340d7a6f79785e6b82d82a800344788509d9b97b51c77aa96429018a33`.

```
# Uniformizers of actual matrix-embedded local fields

For a continuous algebra embedding of a local field `E` into matrices over
`F`, its actual uniformizer has nonzero determinant degree over `F`.
The proof uses actual compact integral units and actual uniformizer coordinates;
it does not assume a determinant-norm formula or a ramification index.
Consequently no nonzero power belongs to any compact matrix subgroup.
The embedding may be noncentral. No simple-type realization is asserted here.
```

Direct project imports: `PaperXI.ActualDepthZeroExtension`, `PaperXI.CompactDegreeSeparation`.

Line	Kind	Name
30	def	`PaperXI.ActualEmbeddedUniformizer58.unitsEmbedding`

Line	Kind	Name
34	def	PaperXI.ActualEmbeddedUniformizer58.embeddedUniformizer
39	theorem	PaperXI.ActualEmbeddedUniformizer58.unitsEmbedding_continuous
42	theorem	PaperXI.ActualEmbeddedUniformizer58.determinantDegree_continuous
48	theorem	PaperXI.ActualEmbeddedUniformizer58.unitsEmbedding_baseScalar
57	theorem	PaperXI.ActualEmbeddedUniformizer58.determinantDegree_integral_unit
71	theorem	PaperXI.ActualEmbeddedUniformizer58.algebraEmbedding_continuous
89	theorem	PaperXI.ActualEmbeddedUniformizer58.embeddedUniformizer_degree_ne_one
107	theorem	PaperXI.ActualEmbeddedUniformizer58.embeddedUniformizer_free
117	theorem	PaperXI.ActualEmbeddedUniformizer58.embeddedUniformizer_free_of_algebraMap
126	theorem	PaperXI.ActualEmbeddedUniformizer58.normalForm_injective
152	theorem	PaperXI.ActualEmbeddedUniformizer58.field_product_normalForm
166	theorem	PaperXI.ActualEmbeddedUniformizer58.field_product_unique_normalForm

A.22 PaperXI.ActualFieldTypeQuotient69

Selected layer: base108 arithmetic branch. **Source lines:** 157.

SHA-256: fd2a01056fe19cf73010c917683c3bcdac14d241c1d9f3bc3accb3584b709bd6.

```
Compactness modulo the actual base-field scalar matrices for E-times-J.
```

Direct project imports: PaperXI.ActualNormalizerTypeExtension58, PaperXI.CompactQuotientPeriod69.

Line	Kind	Name
21	abbrev	PaperXI.ActualFieldTypeQuotient69.fieldGroup
25	def	PaperXI.ActualFieldTypeQuotient69.scalarHom
31	def	PaperXI.ActualFieldTypeQuotient69.scalarSubgroup
36	theorem	PaperXI.ActualFieldTypeQuotient69.scalar_central
43	instance	PaperXI.ActualFieldTypeQuotient69.scalar_normal
51	def	PaperXI.ActualFieldTypeQuotient69.compactPart
55	theorem	PaperXI.ActualFieldTypeQuotient69.degree_kernel
98	theorem	PaperXI.ActualFieldTypeQuotient69.quotient_compact

A.23 PaperXI.ActualGroupIndex111

Selected layer: Continuation111. **Source lines:** 273.

SHA-256: 569cdbec7bc0fcd987fb8fc614db2c4369221abc0007eb98bfdff0896e0800c2.

```
The exact ambient group index from two supplied open coordinate lattices
and two finite full Morita lattices. Canonical H1/J1 identifications remain
source matches; no index identity or common ideal is assumed.
```

Direct project imports: PaperXI.ActualCoordinateDeepIdeal111, PaperXI.ActualOnePlusTopology111, PaperXI.MoritaEndomorphismCoordinates111, PaperXI.MoritaPairIndex110, PaperXI.MoritaRadicalDictionary110, PaperXI.DegreeIndex110.

Line	Kind	Name
24	theorem	PaperXI.ActualGroupIndex111.units_innerRing_inv
33	theorem	PaperXI.ActualGroupIndex111.map_inner_inv_eq_comap

Line	Kind	Name
47	theorem	PaperXI.ActualGroupIndex111.integralUnitMap_eq_actualEmbedding
53	theorem	PaperXI.ActualGroupIndex111.onePlus_backward_relIndex
93	theorem	PaperXI.ActualGroupIndex111.onePlus_index_of_Morita_image
125	theorem	PaperXI.ActualGroupIndex111.ambient_pair_index
163	theorem	PaperXI.ActualGroupIndex111.ambient_pair_index_of_radical_products
202	structure	PaperXI.ActualGroupIndex111.ActualLatticeData
244	theorem	PaperXI.ActualGroupIndex111.ActualLatticeData.actual_index

A.24 PaperXI.ActualHeisenbergCandidate107

Selected layer: base108 arithmetic branch. **Source lines:** 89.

SHA-256: 7dd358423f2d83b2201f0e8b9eed0484c2e577f041d171fe447c59806e11d971.

```
The same arithmetic candidate, with the BK tensor-to-residue implication
proved internally from explicit Heisenberg support and intertwining-line inputs.
Those arithmetic Heisenberg inputs remain unproved here.
```

Direct project imports: PaperXI.ActualCartanCandidate106, PaperXI.ResidueRestriction107, PaperXI.HeisenbergReduction107, PaperXI.IntertwiningSpace107.

Line	Kind	Name
34	theorem	PaperXI.ActualHeisenbergCandidate107.candidate_exists

A.25 PaperXI.ActualIntegerTower111

Selected layer: Continuation111. **Source lines:** 178.

SHA-256: 6effc42145758945edec994d27d6de01bfae48a6752ba7b8f53ac9f8c9ee0300.

```
Actual integer-ring compatibility from the original continuous local-field
embedding. Scalar towers are constructed; none is a supplied arithmetic input.
```

Direct project imports: PaperXI.MoritaActualCartan110.

Line	Kind	Name
20	theorem	PaperXI.ActualIntegerTower111.map_integer_of_continuous
43	def	PaperXI.ActualIntegerTower111.integerMap
47	theorem	PaperXI.ActualIntegerTower111.integerMap_coe
51	theorem	PaperXI.ActualIntegerTower111.integer_of_map_integer
79	theorem	PaperXI.ActualIntegerTower111.integer_comap_eq
88	theorem	PaperXI.ActualIntegerTower111.valuationExtension
93	def	PaperXI.ActualIntegerTower111.integerAlgebra
97	theorem	PaperXI.ActualIntegerTower111.integerAlgebra_map_coe
102	theorem	PaperXI.ActualIntegerTower111.integer_tower
109	theorem	PaperXI.ActualIntegerTower111.integerAlgebra_map_eq
117	theorem	PaperXI.ActualIntegerTower111.map_valuation_le_iff
126	theorem	PaperXI.ActualIntegerTower111.integer_end_smul_apply
131	theorem	PaperXI.ActualIntegerTower111.integer_end_tower
143	theorem	PaperXI.ActualIntegerTower111.integer_matrix_smul_apply

Line	Kind	Name
149	theorem	PaperXI.ActualIntegerTower111.integer_matrix_tower
162	theorem	PaperXI.ActualIntegerTower111.exists_actual_cartan_index_of_integer_lattice

A.26 PaperXI.ActualIntegralGeneralLinear

Selected layer: base108 arithmetic branch. **Source lines:** 219.

SHA-256: 37fadf54dfd99b1032584b0bd3b561824077296be369ecc43f74deffb3847996.

```
# The actual compact open integral subgroup of `GL_n(K)`

The subgroup is the image of `GL_n([U+1D4AA][K])` in the given local field's matrix
group. Its compactness, openness and actual residue quotient are derived from
the entrywise maps. Inflation therefore gives smooth irreducible actions on
this actual subgroup from actual irreducible finite residue representations.
```

Direct project imports: PaperXI.ActualResidueGeneralLinear, PaperXI.QuotientRepresentation, PaperXI.SmoothDegreeQuotient, PaperXI.FiniteRepresentation.

Line	Kind	Name
27	def	PaperXI.ActualIntegralGeneralLinear.embedding
30	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_apply
33	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_injective
41	def	PaperXI.ActualIntegralGeneralLinear.integralSubgroup
44	def	PaperXI.ActualIntegralGeneralLinear.integralEquiv
47	theorem	PaperXI.ActualIntegralGeneralLinear.integralEquiv_coe
50	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_integralEquiv_symm
55	theorem	PaperXI.ActualIntegralGeneralLinear.mem_integralSubgroup
84	theorem	PaperXI.ActualIntegralGeneralLinear.field_t2Space
92	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_isEmbedding
96	theorem	PaperXI.ActualIntegralGeneralLinear.embedding_continuous
100	theorem	PaperXI.ActualIntegralGeneralLinear.integralEquiv_symm_continuous
111	theorem	PaperXI.ActualIntegralGeneralLinear.integralSubgroup_compact
120	theorem	PaperXI.ActualIntegralGeneralLinear.integralSubgroup_open
131	def	PaperXI.ActualIntegralGeneralLinear.reductionOnSubgroup
134	theorem	PaperXI.ActualIntegralGeneralLinear.reductionOnSubgroup_integralEquiv
138	theorem	PaperXI.ActualIntegralGeneralLinear.reductionOnSubgroup_surjective
142	theorem	PaperXI.ActualIntegralGeneralLinear.mem_reduction_kernel
156	theorem	PaperXI.ActualIntegralGeneralLinear.reduction_kernel_open
171	theorem	PaperXI.ActualIntegralGeneralLinear.reductionOnSubgroup_kernel_open
177	def	PaperXI.ActualIntegralGeneralLinear.residueQuotientEquiv
182	theorem	PaperXI.ActualIntegralGeneralLinear.finite_residue_quotient
191	def	PaperXI.ActualIntegralGeneralLinear.inflated
194	theorem	PaperXI.ActualIntegralGeneralLinear.inflated_integralEquiv
199	theorem	PaperXI.ActualIntegralGeneralLinear.inflated_irreducible
205	theorem	PaperXI.ActualIntegralGeneralLinear.inflated_isSmooth
216	theorem	PaperXI.ActualIntegralGeneralLinear.inflated_finiteDimensional

A.27 PaperXI.ActualIntersection108

Selected layer: base108 arithmetic branch. **Source lines:** 59.

SHA-256: fede966a6ccc302109f79282ebd6c8a1b66b3c9dcbafc6b1cccd28b1160bd34b.

```
The canonical finite quotient for the actual first-subgroup intertwining
space. This proves existence of the finite model, not its arithmetic profile.
```

Direct project imports: `PaperXI.ActualCharacterCandidate108`, `PaperXI.FiniteIntersectionCriterion108`.

Line	Kind	Name
20	abbrev	`PaperXI.ActualIntersection108.embedding`
23	abbrev	`PaperXI.ActualIntersection108.restriction`
25	abbrev	`PaperXI.ActualIntersection108.quotient`
30	theorem	`PaperXI.ActualIntersection108.finite`
47	theorem	`PaperXI.ActualIntersection108.dimension`

A.28 `PaperXI.ActualLocalCartan106`

Selected layer: base108 arithmetic branch. **Source lines:** 44.

SHA-256: `ad8205ed61e0d305b7023ba61e8f17b266407a62ec64e846e7993a90d62882b2`.

```
The arithmetic Cartan and residue support theorems on actual local-field
matrices, with a uniformizer chosen from the proved valuation-ring structure.
```

Direct project imports: `PaperXI.DVRCartan106`, `PaperXI.ActualLocalUnitDecomposition`, `PaperXI.ActualIntegralGeneralLinear`, `PaperXI.ActualResidueGeneralLinear`.

Line	Kind	Name
16	theorem	`PaperXI.ActualLocalCartan106.uniformizer_residue_zero`
22	theorem	`PaperXI.ActualLocalCartan106.hasCartan`
31	theorem	`PaperXI.ActualLocalCartan106.intertwining_support`

A.29 `PaperXI.ActualLocalUnitDecomposition`

Selected layer: base108 arithmetic branch. **Source lines:** 128.

SHA-256: `a8395974d47aefc05e157314ff5c0f41b5ec8a33c9bfb98579de8b62ab5e273d`.

```
# Actual local-field unit and uniformizer coordinates

The uniformizer is chosen from the proved DVR structure of the actual ring of
integers. Existence of coordinates comes from fraction-field factorization;
uniqueness comes from the actual valuation. No split group is supplied.
```

Direct project imports: `PaperXI.ActualLocalUnitTopology`.

Line	Kind	Name
22	def	`PaperXI.ActualLocalUnitDecomposition.uniformizer`
24	theorem	`PaperXI.ActualLocalUnitDecomposition.uniformizer_irreducible`
27	theorem	`PaperXI.ActualLocalUnitDecomposition.uniformizer_ne_zero`
31	def	`PaperXI.ActualLocalUnitDecomposition.uniformizerUnit`
33	theorem	`PaperXI.ActualLocalUnitDecomposition.uniformizerUnit_coe`
35	theorem	`PaperXI.ActualLocalUnitDecomposition.uniformizer_value_pos`
38	theorem	`PaperXI.ActualLocalUnitDecomposition.uniformizer_value_lt_one`
42	def	`PaperXI.ActualLocalUnitDecomposition.normalFormHom`

Line	Kind	Name
50	theorem	PaperXI.ActualLocalUnitDecomposition.normalFormHom_apply
54	theorem	PaperXI.ActualLocalUnitDecomposition.normalForm_surjective
65	theorem	PaperXI.ActualLocalUnitDecomposition.normalForm_injective
82	def	PaperXI.ActualLocalUnitDecomposition.normalFormEquiv
85	theorem	PaperXI.ActualLocalUnitDecomposition.normalFormEquiv_apply
89	def	PaperXI.ActualLocalUnitDecomposition.degree
92	theorem	PaperXI.ActualLocalUnitDecomposition.degree_normalForm
97	theorem	PaperXI.ActualLocalUnitDecomposition.degree_unit
100	theorem	PaperXI.ActualLocalUnitDecomposition.degree_unitEmbedding
103	theorem	PaperXI.ActualLocalUnitDecomposition.degree_uniformizer
107	theorem	PaperXI.ActualLocalUnitDecomposition.ker_degree
119	theorem	PaperXI.ActualLocalUnitDecomposition.degree_continuous

A.30 PaperXI.ActualLocalUnitTopology

Selected layer: base108 arithmetic branch. **Source lines:** 102.

SHA-256: 2d87df54c86bef03b5bf3277d309f08733cde2c39d287c2226b9d04bcbe388ea.

```
# The actual compact open unit subgroup of a nonarchimedean local field

The subgroup is the image of the unit group of the valuation ring under its
canonical embedding in the field. Its topological properties are derived from
the local-field hypotheses; they are not assumptions on an abstract subgroup.
No characteristic-zero hypothesis is used.
```

Direct project imports: none.

Line	Kind	Name
23	def	PaperXI.ActualLocalUnitTopology.unitEmbedding
27	def	PaperXI.ActualLocalUnitTopology.compactUnitSubgroup
29	theorem	PaperXI.ActualLocalUnitTopology.coe_unitEmbedding
32	theorem	PaperXI.ActualLocalUnitTopology.unitEmbedding_injective
35	theorem	PaperXI.ActualLocalUnitTopology.mem_compactUnitSubgroup_iff
50	theorem	PaperXI.ActualLocalUnitTopology.compactUnitSubgroup_eq_units
62	theorem	PaperXI.ActualLocalUnitTopology.continuous_unitEmbedding
67	instance	PaperXI.ActualLocalUnitTopology.localFieldT2Space
73	theorem	PaperXI.ActualLocalUnitTopology.isCompact_compactUnitSubgroup
78	theorem	PaperXI.ActualLocalUnitTopology.isOpen_compactUnitSubgroup
84	def	PaperXI.ActualLocalUnitTopology.unitEmbeddingEquiv
92	theorem	PaperXI.ActualLocalUnitTopology.coe_unitEmbeddingEquiv
96	theorem	PaperXI.ActualLocalUnitTopology.unitEmbedding_isOpenMap

A.31 PaperXI.ActualNormalizerTypeExtension58

Selected layer: base108 arithmetic branch. **Source lines:** 82.

SHA-256: eed55102f4bdce2532e92a4655f4da687a9fa494d89917d61f2a8cef613b46f1.

```
# Actual embedded-field normalizer extension

The noncentral matrix uniformizer's freeness is proved from compactness and
actual local-field arithmetic. Given its actual normalization of the compact
```

```
subgroup and one nonzero intertwiner, the literal representation is extended.
Simple-character normalization and beta-extension intertwining are explicit
remaining arithmetic inputs; no claim that they have been constructed is made.
```

Direct project imports: `PaperXI.ActualEmbeddedUniformizer58`, `PaperXI.NormalizerExtensionInducing58`.

Line	Kind	Name
32	theorem	`PaperXI.ActualNormalizerTypeExtension58.extendedSubgroup_eq_field_join`
60	theorem	`PaperXI.ActualNormalizerTypeExtension58.exists_smooth_irreducible_extension`

A.32 `PaperXI.ActualOnePlusTopology111`

Selected layer: Continuation111. **Source lines:** 114.

SHA-256: `cdef6d0ce5c93ad213cc8af1fef105ca1375f9566fdf088cdaad7677198cb9ac`.

```
Openness of an actual one-plus group supplies additive lattice openness.
```

Direct project imports: `PaperXI.ActualCoordinateDeepIdeal111`, `PaperXI.ActualIntegralGeneralLinear`.

Line	Kind	Name
19	def	`PaperXI.ActualOnePlusTopology111.matrixUnitsHomeomorph`
38	theorem	`PaperXI.ActualOnePlusTopology111.matrix_isOpenUnits`
53	theorem	`PaperXI.ActualOnePlusTopology111.actual_matrix_isOpenUnits`
59	theorem	`PaperXI.ActualOnePlusTopology111.additive_open_of_image_group_open`
100	theorem	`PaperXI.ActualOnePlusTopology111.onePlus_conjugate_relIndex_of_group_open`

A.33 `PaperXI.ActualPacketTypeRealization70`

Selected layer: base108 arithmetic branch. **Source lines:** 109.

SHA-256: `079fe5114db1fe95bd51e370b7932a6394bce0b44bad32671fa820774e806037`.

```
Same-theta, same-kappa construction at the actual residue map and field join.
This replaces the old compact-subgroup Mackey premise by a full-locus condition
on the newly constructed extension. Standard character/type leaves remain
explicit; this is not a formalization of their classification theorem.
```

Direct project imports: `PaperXI.PacketTypeAttachment70`, `PaperXI.PrescribedFieldType70`, `PaperXI.ArithmeticTypeCertificate69`, `PaperXI.ActualCharacterIdentification61`, `PaperXI.ActualCharacterResidue61`.

Line	Kind	Name
36	theorem	`PaperXI.ActualPacketTypeRealization70.candidate_exists`

A.34 `PaperXI.ActualProfileCandidate110`

Selected layer: Continuation110. **Source lines:** 94.

SHA-256: `dd434e182eec32397e64a3e6a43193cc56a3c965eff65c4e36e14f4bd0b2c27b`.

```
The existing actual candidate from original eta trace support, centralizer
character covariance and the oriented ambient index identity. The degree,
finite quotient and Heisenberg Hom dimension are derived, not supplied.
Actual simple-character and category instantiation remains separate.
```

Direct project imports: `PaperXI.DegreeIndex110`.

Line	Kind	Name
34	theorem	`PaperXI.ActualProfileCandidate110.candidate_exists_from_original_profiles`

A.35 `PaperXI.ActualResidueGeneralLinear`

Selected layer: base108 arithmetic branch. **Source lines:** 69.

SHA-256: `33f125bdc926fb2a521cd18f1d46ba7c0d70d277e4d96bf0827d0fc610cd3459`.

```
Entrywise reduction of actual integral general linear groups.

The lifting argument works over any commutative local ring: lift each entry,
then use the nonzero reduced determinant to prove that the lifted determinant
is a unit. The final declarations specialize this to the given nonarchimedean
local field's own ring of integers and finite residue field.
```

Direct project imports: none.

Line	Kind	Name
21	def	`PaperXI.ActualResidueGeneralLinear.residueMap`
25	theorem	`PaperXI.ActualResidueGeneralLinear.residueMap_apply`
30	theorem	`PaperXI.ActualResidueGeneralLinear.residueMap_surjective`
52	def	`PaperXI.ActualResidueGeneralLinear.reduction`
55	theorem	`PaperXI.ActualResidueGeneralLinear.reduction_apply`
59	theorem	`PaperXI.ActualResidueGeneralLinear.reduction_surjective`
65	theorem	`PaperXI.ActualResidueGeneralLinear.finite_reduction_target`

A.36 `PaperXI.ActualScalarLift113`

Selected layer: Continuation113. **Source lines:** 221.

SHA-256: `9dca2abac60d683f19c4b2bd2bd257b8f18e5b5f459cf39df9948de4ccf41ce0`.

```
Honest actions on an actual group induce honest scalar-pushout actions.
The section phase and its multiplication law are conclusions. The original
theta, its literal kernel subgroup, and the quotient labels are retained.
```

Direct project imports: `PaperXI.ScalarRepresentation112`, `PaperXI.CyclicDefect113`.

Line	Kind	Name
19	def	`PaperXI.ActualScalarLift113.actualPushout`
30	theorem	`PaperXI.ActualScalarLift113.actualPushout_section`
37	theorem	`PaperXI.ActualScalarLift113.actualPushout_subgroup`
50	theorem	`PaperXI.ActualScalarLift113.quotient_action_power`
62	theorem	`PaperXI.ActualScalarLift113.quotient_power_eq_one`
72	def	`PaperXI.ActualScalarLift113.subgroupMap`
88	def	`PaperXI.ActualScalarLift113.actualPhase`
92	theorem	`PaperXI.ActualScalarLift113.actualPhase_one`
98	theorem	`PaperXI.ActualScalarLift113.sectionScalar_action`
127	theorem	`PaperXI.ActualScalarLift113.actualPhase_mul`
137	def	`PaperXI.ActualScalarLift113.actualLift`

Line	Kind	Name
161	theorem	PaperXI.ActualScalarLift113.actualLift_phase
167	theorem	PaperXI.ActualScalarLift113.actualLift_intertwines
175	theorem	PaperXI.ActualScalarLift113.actualLift_power_intertwines
191	theorem	PaperXI.ActualScalarLift113.actualLift_power_eq_one

A.37 PaperXI.ActualSourceCandidate111

Selected layer: Continuation111. **Source lines:** 102.

SHA-256: c802c3b3d1be72ca3d961a4edaa42cca066eb81cbc902a0a0abe9eee58d60005.

```
The original candidate from exact type quotient/form data and literal
one-plus/Morita lattice matches. Trace support, covariance, finite quotient,
degree-square and the ambient index equality are derived. The input record
still requires its canonical type-theoretic instance; this is not Claim16.2.
```

Direct project imports: PaperXI.ActualProfileCandidate110, PaperXI.HeisenbergCharacterProfile111, PaperXI.ActualGroupIndex111.

Line	Kind	Name
40	theorem	PaperXI.ActualSourceCandidate111.candidate_exists_from_source_dictionary

A.38 PaperXI.ActualUnramifiedBaseChange

Selected layer: v0.44 parent. **Source lines:** 115.

SHA-256: 8f4e3c5117f404df075772a6e9b3782e2463b7e84973d380f248d9c9c5c4f87f.

```
# Actual unramified base change through the recursive local-field tower

Block 40 proved Frobenius-power and Euler-factor formulas for an internal degree
dilation of the integer-degree unramified Weil model. Block 41 constructs an
actual recursive local-field tower and its restriction map. This module proves
that the two constructions give the same Frobenius polynomials and local
factors, and records conductor zero on both sides of the actual tower.

This remains scoped to the bounded unramified Weil models, not absolute Weil
groups or automorphic base change.
```

Direct project imports: PaperXI.UnramifiedFieldTower.

Line	Kind	Name
43	theorem	PaperXI.ActualUnramifiedBaseChange.frobeniusPolynomial_eq_modelBaseChange
52	theorem	PaperXI.ActualUnramifiedBaseChange.localFactor_eq_modelBaseChange
62	theorem	PaperXI.ActualUnramifiedBaseChange.restricted_inertiaCodimension_eq_zero
71	theorem	PaperXI.ActualUnramifiedBaseChange.restricted_swanConductorExponent_eq_zero
79	theorem	PaperXI.ActualUnramifiedBaseChange.restricted_artinConductorExponent_eq_zero
87	theorem	PaperXI.ActualUnramifiedBaseChange.artinConductorExponent_restriction
98	theorem	PaperXI.ActualUnramifiedBaseChange.irreducible_restriction_frobeniusPolynomial
107	theorem	PaperXI.ActualUnramifiedBaseChange.irreducible_restriction_localFactor

A.39 PaperXI.AlgebraActionModule

Selected layer: v0.44 parent. **Source lines:** 250.

SHA-256: 5ceb841756c4ca3a00e4fe21b5969470175a83df02d2c417316405eced836820.

```
An algebra action gives an actual module on a separate copy of its vector space.
Submodules and intertwiners are identified with the corresponding complex-linear data.
```

Direct project imports: none.

Line	Kind	Name
19	def	PaperXI.AlgebraActionModule.Space
25	def	PaperXI.AlgebraActionModule.equiv
27	instance	PaperXI.AlgebraActionModule.instModule
29	theorem	PaperXI.AlgebraActionModule.equiv_smul
32	theorem	PaperXI.AlgebraActionModule.equiv_symm_apply
34	instance	PaperXI.AlgebraActionModule.instScalarTower
41	def	PaperXI.AlgebraActionModule.Stable
45	def	PaperXI.AlgebraActionModule.toSubspace
47	theorem	PaperXI.AlgebraActionModule.mem_toSubspace
50	theorem	PaperXI.AlgebraActionModule.stable_toSubspace
55	def	PaperXI.AlgebraActionModule.ofSubspace
61	theorem	PaperXI.AlgebraActionModule.mem_ofSubspace
64	theorem	PaperXI.AlgebraActionModule.toSubspace_ofSubspace
67	theorem	PaperXI.AlgebraActionModule.ofSubspace_toSubspace
71	def	PaperXI.AlgebraActionModule.submoduleOrderIso
78	theorem	PaperXI.AlgebraActionModule.toSubspace_bot
79	theorem	PaperXI.AlgebraActionModule.toSubspace_top
82	theorem	PaperXI.AlgebraActionModule.simple_iff
119	def	PaperXI.AlgebraActionModule.Intertwines
123	def	PaperXI.AlgebraActionModule.toModuleMap
130	def	PaperXI.AlgebraActionModule.toComplexMap
133	theorem	PaperXI.AlgebraActionModule.toModuleMap_apply
136	theorem	PaperXI.AlgebraActionModule.toComplexMap_apply
139	theorem	PaperXI.AlgebraActionModule.intertwines_toComplexMap
143	def	PaperXI.AlgebraActionModule.intertwinerEquiv
150	theorem	PaperXI.AlgebraActionModule.toComplexMap_toModuleMap
153	theorem	PaperXI.AlgebraActionModule.toModuleMap_toComplexMap
159	theorem	PaperXI.AlgebraActionModule.toComplexMap_id
165	theorem	PaperXI.AlgebraActionModule.toComplexMap_comp
173	def	PaperXI.AlgebraActionModule.toModuleEquiv
183	def	PaperXI.AlgebraActionModule.toComplexEquiv
186	theorem	PaperXI.AlgebraActionModule.toModuleEquiv_apply
190	theorem	PaperXI.AlgebraActionModule.toComplexEquiv_apply
193	theorem	PaperXI.AlgebraActionModule.intertwines_toComplexEquiv
197	def	PaperXI.AlgebraActionModule.equivalenceEquiv
204	theorem	PaperXI.AlgebraActionModule.toComplexEquiv_toModuleEquiv
208	theorem	PaperXI.AlgebraActionModule.toModuleEquiv_toComplexEquiv
213	theorem	PaperXI.AlgebraActionModule.toComplexEquiv_refl
218	theorem	PaperXI.AlgebraActionModule.toComplexEquiv_trans
232	def	PaperXI.AlgebraActionModule.ofModuleEquiv
240	theorem	PaperXI.AlgebraActionModule.ofModuleEquiv_apply

Line	Kind	Name
244	theorem	PaperXI.AlgebraActionModule.ofModule_simple_iff

A.40 PaperXI.ArithmeticAttachment110

Selected layer: Continuation110. **Source lines:** 161.

SHA-256: e67a6b640cb79d7ea347d00621361e950ea439519e53e1f472ef6758460a8693.

A genuine residue-rank-one specialization. The finite residue representation
is constructed, and the full extended support follows directly from character
support because GL_1(E) consists of field scalars. No Heisenberg intersection
dimension, BK lattice index, or supplied finite cuspidal representation occurs
as an input. The actual simple-character/model/category leaves remain explicit.

Direct project imports: PaperXI.ActualCharacterCandidate108.

Line	Kind	Name
18	def	PaperXI.ArithmeticAttachment110.residueOne
21	theorem	PaperXI.ArithmeticAttachment110.residueOne_irreducible
27	theorem	PaperXI.ArithmeticAttachment110.residueOne_cuspidal
37	theorem	PaperXI.ArithmeticAttachment110.gl_one_scalar
45	def	PaperXI.ArithmeticAttachment110.tensorUnitEquiv
66	theorem	PaperXI.ArithmeticAttachment110.integral_normalization_of_full_rank_one
88	theorem	PaperXI.ArithmeticAttachment110.rank_one_candidate_exists

A.41 PaperXI.ArithmeticFieldInduction69

Selected layer: base108 arithmetic branch. **Source lines:** 83.

SHA-256: 84d9fdbbdab35cebcb07e0b60f4363cd4e176ac7b0483a850db3af4ebe56588e.

The compact-mod-center induction theorem specialized to literal local-field matrices.
The actual field scalar subgroup and compact quotient are constructed. The common
scalar character is derived from finite-dimensional irreducibility, not supplied.

Direct project imports: PaperXI.ActualFieldTypeQuotient69, PaperXI.DerivedCentralCharacter69, PaperXI.CompactModCenterInduction68.

Line	Kind	Name
27	theorem	PaperXI.ArithmeticFieldInduction69.fieldGroup_open
32	def	PaperXI.ArithmeticFieldInduction69.scalarCharacter
38	theorem	PaperXI.ArithmeticFieldInduction69.scalarCharacter_action
46	theorem	PaperXI.ArithmeticFieldInduction69.induction_package
72	theorem	PaperXI.ArithmeticFieldInduction69.induced_scalar

A.42 PaperXI.ArithmeticTypeCertificate69

Selected layer: base108 arithmetic branch. **Source lines:** 106.

SHA-256: a360d478cc7c93440aa4806bf37892a9cbc4c847d393a8be3079c7fc9da439ac.

The actual base-field central character and the full intertwining-locus
application. No auxiliary scalar action or quotient compactness is assumed.

Direct project imports: `PaperXI.ArithmeticFieldInduction69`, `PaperXI.FullIntertwiningLocus69`.

Line	Kind	Name
24	def	`PaperXI.ArithmeticTypeCertificate69.centralCharacter`
29	theorem	`PaperXI.ArithmeticTypeCertificate69.centralCharacter_action`
35	theorem	`PaperXI.ArithmeticTypeCertificate69.induced_centralCharacter`
41	theorem	`PaperXI.ArithmeticTypeCertificate69.scalarHom_continuous`
56	theorem	`PaperXI.ArithmeticTypeCertificate69.centralCharacter_locallyConstant`
87	theorem	`PaperXI.ArithmeticTypeCertificate69.from_full_intertwining_locus`

A.43 `PaperXI.AtomicMultiset`

Selected layer: v0.44 parent. **Source lines:** 122.

SHA-256: `e9b0e86f94713a76638c5bde3c53ba97140aec5c543aad941c9bba61ede4d3f6`.

```
The block-multiset part of Paper XI, Theorem 2.5.
An atom is an isomorphism class. A block retains its strictly positive monodromy
length. Mapping atoms does not discard lengths, multiplicities, or repetitions.
This module contains no local-Weil or Langlands-classification existence axiom.
```

Direct project imports: none.

Line	Kind	Name
14	structure	`PaperXI.AtomicMultiset.Block`
21	def	`PaperXI.AtomicMultiset.blockMap`
24	theorem	`PaperXI.AtomicMultiset.blockMap_atom`
27	theorem	`PaperXI.AtomicMultiset.blockMap_length`
30	theorem	`PaperXI.AtomicMultiset.blockMap_id`
33	theorem	`PaperXI.AtomicMultiset.blockMap_comp`
36	def	`PaperXI.AtomicMultiset.extend`
41	theorem	`PaperXI.AtomicMultiset.extend_singleton`
44	theorem	`PaperXI.AtomicMultiset.extend_cons`
47	theorem	`PaperXI.AtomicMultiset.extend_card`
50	theorem	`PaperXI.AtomicMultiset.extend_id`
54	theorem	`PaperXI.AtomicMultiset.extend_comp`
59	theorem	`PaperXI.AtomicMultiset.extend_unique`
70	theorem	`PaperXI.AtomicMultiset.extend_injective`
80	def	`PaperXI.AtomicMultiset.extendEquiv`
94	def	`PaperXI.AtomicMultiset.dimension`
97	theorem	`PaperXI.AtomicMultiset.dimension_extend`
107	def	`PaperXI.AtomicMultiset.descend`
111	theorem	`PaperXI.AtomicMultiset.descend_mk`
116	theorem	`PaperXI.AtomicMultiset.extend_natural`

A.44 `PaperXI.AtomicNaturality`

Selected layer: v0.44 parent. **Source lines:** 106.

SHA-256: `ba747e172fb8b6398cc4ce4076c4f8dc2c3f5fcff98a6a24f1e3f1b4a4584873`.

```
Naturality and the exact information-preservation boundary for Theorem 2.5.
Twist laws are attached to the primitive decomposition and ordered quotient
relations. Equivariance of the final candidate is then a theorem, not an input.
```

Direct project imports: `PaperXI.AtomicReconstruction`.

Line	Kind	Name
15	theorem	`PaperXI.AtomicNaturality.reconstruction_from_representatives`
22	theorem	`PaperXI.AtomicNaturality.blocks_twist`
36	theorem	`PaperXI.AtomicNaturality.realize_twist`
53	theorem	`PaperXI.AtomicNaturality.candidate_twist`
67	theorem	`PaperXI.AtomicNaturality.extend_injective_iff`
84	theorem	`PaperXI.AtomicNaturality.candidate_injective`
92	theorem	`PaperXI.AtomicNaturality.decomposition_independence`

A.45 `PaperXI.AtomicReconstruction`

Selected layer: v0.44 parent. **Source lines:** 153.

SHA-256: `e848f4ab5b8717eda8465737b7a726fbb877d3e6effc3728eaeab08698da52cb`.

```
Paper XI, Theorem 2.5: choice-independent conditional reconstruction.
External interfaces here are exactly (1) existence and intrinsic uniqueness of
indecomposable source displays and (2) existence and uniqueness of irreducible
Langlands quotients of ordered target factors. They do not contain the final
candidate map. Sorting, atomic replacement, descent, and that map are constructed.
The theorem is conditional on these classical interfaces; it does not formalize
the construction/classification of local Weil groups or p-adic representations.
```

Direct project imports: `PaperXI.AtomicMultiset`.

Line	Kind	Name
17	structure	`PaperXI.AtomicReconstruction.Decomposition`
25	def	`PaperXI.AtomicReconstruction.rawBlocks`
28	theorem	`PaperXI.AtomicReconstruction.rawBlocks_eq`
33	theorem	`PaperXI.AtomicReconstruction.rawBlocks_iso`
41	def	`PaperXI.AtomicReconstruction.blocks`
44	theorem	`PaperXI.AtomicReconstruction.blocks_of_display`
52	structure	`PaperXI.AtomicReconstruction.OrderedQuotient`
63	def	`PaperXI.AtomicReconstruction.ordered`
67	theorem	`PaperXI.AtomicReconstruction.ordered_pairwise`
74	theorem	`PaperXI.AtomicReconstruction.ordered_multiset`
83	def	`PaperXI.AtomicReconstruction.realize`
86	theorem	`PaperXI.AtomicReconstruction.realize_eq`
95	theorem	`PaperXI.AtomicReconstruction.realize_spec`
103	def	`PaperXI.AtomicReconstruction.candidate`
108	def	`PaperXI.AtomicReconstruction.Reconstructs`
114	theorem	`PaperXI.AtomicReconstruction.candidate_spec`
121	theorem	`PaperXI.AtomicReconstruction.candidate_eq`
130	theorem	`PaperXI.AtomicReconstruction.candidate_unique`
143	theorem	`PaperXI.AtomicReconstruction.formal_atomic_reconstruction`
148	theorem	`PaperXI.AtomicReconstruction.candidate_isomorphism`

A.46 `PaperXI.BilinearHeisenberg`

Selected layer: v0.44 parent. **Source lines:** 152.

SHA-256: `d4615e543af67b750697c7fdbf5f7d3b790aa0b4d6992aa2246a3aa00399e7d9`.

```
Actual finite central extensions from bilinear factor sets.  The extension,
commutator, central character and Stone--von Neumann endpoint are distinct data.
No multiplication law, irrep, dimension or initial intertwiner is assumed.
```

Direct project imports: `PaperXI.SymplecticHeisenbergUniqueness`.

Line	Kind	Name
13	structure	`PaperXI.BilinearHeisenberg.Element`
19	instance	`PaperXI.BilinearHeisenberg.instMul`
21	instance	`PaperXI.BilinearHeisenberg.instOne`
22	instance	`PaperXI.BilinearHeisenberg.instInv`
25	theorem	`PaperXI.BilinearHeisenberg.mul_vector`
27	theorem	`PaperXI.BilinearHeisenberg.mul_central`
29	theorem	`PaperXI.BilinearHeisenberg.one_vector`
30	theorem	`PaperXI.BilinearHeisenberg.one_central`
31	theorem	`PaperXI.BilinearHeisenberg.inv_vector`
32	theorem	`PaperXI.BilinearHeisenberg.inv_central`
35	instance	`PaperXI.BilinearHeisenberg.instGroup`
44	def	`PaperXI.BilinearHeisenberg.i`
49	def	`PaperXI.BilinearHeisenberg.q`
54	theorem	`PaperXI.BilinearHeisenberg.i_vector`
55	theorem	`PaperXI.BilinearHeisenberg.i_central`
56	theorem	`PaperXI.BilinearHeisenberg.q_apply`
59	def	`PaperXI.BilinearHeisenberg.datum`
78	theorem	`PaperXI.BilinearHeisenberg.commutator`
88	theorem	`PaperXI.BilinearHeisenberg.centralCommutator`
96	instance	`PaperXI.BilinearHeisenberg.instFinite`
100	theorem	`PaperXI.BilinearHeisenberg.order`
109	theorem	`PaperXI.BilinearHeisenberg.source_commutator`
121	theorem	`PaperXI.BilinearHeisenberg.exists_irreducible`
136	theorem	`PaperXI.BilinearHeisenberg.exists_equiv`
145	theorem	`PaperXI.BilinearHeisenberg.finrank_sqrt`

A.47 `PaperXI.BoundaryExamples`

Selected layer: v0.44 parent. **Source lines:** 30.

SHA-256: `d847590d5166021de5e552840a8b6bac45da991bbac45860b3cd96c3aaeabf89`.

Direct project imports: `PaperXI.GeneralChain`, `PaperXI.ChainSharpness`, `PaperXI.ScalarCovariance`, `PaperXI.FourierCovariance`, `PaperXI.HolonomyResidual`.

Line	Kind	Name
11	theorem	`PaperXI.BoundaryExamples.empty_chain`
14	theorem	`PaperXI.BoundaryExamples.singleton_chain`
17	theorem	`PaperXI.BoundaryExamples.length_three_square_nonzero`
21	theorem	`PaperXI.BoundaryExamples.length_three_cube_zero`
24	theorem	`PaperXI.BoundaryExamples.inverse_two_mod_five`
26	theorem	`PaperXI.BoundaryExamples.inverse_label_differs`

A.48 `PaperXI.BreakBridge`

Selected layer: v0.44 parent. **Source lines:** 104.

SHA-256: `8867bca4f28ab57140255d385f80f0d940f50f9d7bfd3de435c9b4b6385937d9`.

```
Finite break tensor bridges on the actual central quotient. The total
stabilizer phase is the product of the actual factor discrepancies.
```

Direct project imports: `PaperXI.BreakTensorMap`, `PaperXI.HeisenbergTensor`, `PaperXI.CharacterGenerators`.

Line	Kind	Name
18	theorem	`PaperXI.BreakBridge.ambient_intertwines`
30	theorem	`PaperXI.BreakBridge.quotient_intertwines`
47	theorem	`PaperXI.BreakBridge.stabilizer_intertwines`
61	theorem	`PaperXI.BreakBridge.stabilizer_intertwines_of_factor_phases`
75	theorem	`PaperXI.BreakBridge.semidirect_intertwines`

A.49 `PaperXI.BreakBridgeAlong`

Selected layer: v0.44 parent. **Source lines:** 80.

SHA-256: `cc313014e54d45de8e7391509928e831d857cc92c21b22d029f4c5541b823658`.

```
Tensor bridges between different actual break extensions, with discrepancy
characters derived from the chosen honest stabilizer actions.
```

Direct project imports: `PaperXI.BreakBridge`, `PaperXI.CentralProductMap`, `PaperXI.HeisenbergTensorFactorization`.

Line	Kind	Name
19	theorem	`PaperXI.BreakBridgeAlong.quotient_intertwines`
45	theorem	`PaperXI.BreakBridgeAlong.source`

A.50 `PaperXI.BreakDyadicPhase`

Selected layer: v0.44 parent. **Source lines:** 115.

SHA-256: `6b938157b4b523f51398fb5d57351690ba2200647a98c90f9d715a12030349df`.

```
Paper XI 15.3, characteristic two: finite-family products of actual
normalized Gauss sums, and transfer through the supplied quadratic decomposition.
```

Direct project imports: `PaperXI.BreakQuadratic`.

Line	Kind	Name
20	theorem	`PaperXI.BreakDyadicPhase.gaussSum_product`
31	theorem	`PaperXI.BreakDyadicPhase.normalizedGaussSign_product`
42	theorem	`PaperXI.BreakDyadicPhase.arf_sum`
53	theorem	`PaperXI.BreakDyadicPhase.normalizedGaussSign_empty`
66	theorem	`PaperXI.BreakDyadicPhase.transport_refinement`
77	theorem	`PaperXI.BreakDyadicPhase.arf_of_decomposition`
95	theorem	`PaperXI.BreakDyadicPhase.normalizedGaussSign_of_decomposition`
110	theorem	`PaperXI.BreakDyadicPhase.half_finrank_of_decomposition`

A.51 PaperXI.BreakHeisenbergSum

Selected layer: v0.44 parent. **Source lines:** 143.

SHA-256: `8f7d2102714c520e91a760b6fe970dd48fb99ed979d8d4cd14280b6ffc18e50c`.

```
Actual half-form Heisenberg groups of a finite orthogonal sum are the
balanced central product of the summand groups. The retained center handles
an empty family. The group isomorphism is constructed, not assumed.
```

Direct project imports: `PaperXI.AbstractHeisenberg`, `PaperXI.CentralProduct`.

Line	Kind	Name
19	def	`PaperXI.BreakHeisenbergSum.form`
27	theorem	`PaperXI.BreakHeisenbergSum.form_apply`
30	theorem	`PaperXI.BreakHeisenbergSum.form_isAlt`
36	def	`PaperXI.BreakHeisenbergSum.fromProduct`
49	theorem	`PaperXI.BreakHeisenbergSum.fromProduct_vector`
53	theorem	`PaperXI.BreakHeisenbergSum.fromProduct_central`
58	theorem	`PaperXI.BreakHeisenbergSum.fromProduct_surjective`
65	theorem	`PaperXI.BreakHeisenbergSum.fromProduct_balanced`
72	def	`PaperXI.BreakHeisenbergSum.sumMap`
76	theorem	`PaperXI.BreakHeisenbergSum.sumMap_mk`
80	theorem	`PaperXI.BreakHeisenbergSum.sumMap_center`
84	theorem	`PaperXI.BreakHeisenbergSum.sumMap_surjective`
91	theorem	`PaperXI.BreakHeisenbergSum.sumMap_injective`
116	def	`PaperXI.BreakHeisenbergSum.sumEquiv`
120	theorem	`PaperXI.BreakHeisenbergSum.sumEquiv_mk`
124	theorem	`PaperXI.BreakHeisenbergSum.sumEquiv_center`
130	def	`PaperXI.BreakHeisenbergSum.decompositionEquiv`
136	theorem	`PaperXI.BreakHeisenbergSum.decompositionEquiv_center`

A.52 PaperXI.BreakMoving

Selected layer: v0.44 parent. **Source lines:** 116.

SHA-256: `e72bb01b58c7e900a66b0919ad0b78a71e7ae979c3ed55025b524b2d80519da9`.

```
Actual finite-family moving spaces. The maps below use the image of the
actual difference, not an independently supplied vector space of the same size.
```

Direct project imports: `PaperXI.MovingPhase`, `PaperXI.OrthogonalPhase`.

Line	Kind	Name
17	def	`PaperXI.BreakMoving.productEquiv`
30	theorem	`PaperXI.BreakMoving.productEquiv_coe`
34	theorem	`PaperXI.BreakMoving.productEquiv_symm_coe`
39	theorem	`PaperXI.BreakMoving.productEquiv_intertwines`
51	def	`PaperXI.BreakMoving.equiv`
57	theorem	`PaperXI.BreakMoving.equiv_coe`
62	theorem	`PaperXI.BreakMoving.equiv_intertwines`
72	theorem	`PaperXI.BreakMoving.equiv_preserves`
89	theorem	`PaperXI.BreakMoving.isCompl_of_moving_fixedPointFree`

Line	Kind	Name
109	theorem	`PaperXI.BreakMoving.movingForm_nondegenerate`

A.53 `PaperXI.BreakOddPhase`

Selected layer: v0.44 parent. **Source lines:** 164.

SHA-256: `32575366712c4c41dec1087b72f1a84ef94ced9b0af536673412a053eec700cf`.

```
Source 15.3 in odd characteristic, using the moving phase of Definition
12.7. Fixed vectors do not enter a determinant on the full singular space.
```

Direct project imports: `PaperXI.BreakMoving`.

Line	Kind	Name
20	theorem	`PaperXI.BreakOddPhase.component_fixedPointFree`
42	theorem	`PaperXI.BreakOddPhase.product_of_moving_forms`
58	theorem	`PaperXI.BreakOddPhase.product`
74	theorem	`PaperXI.BreakOddPhase.determinants_ne_zero`
85	theorem	`PaperXI.BreakOddPhase.empty`
92	theorem	`PaperXI.BreakOddPhase.zero_moving_factor`
100	theorem	`PaperXI.BreakOddPhase.internal_intertwines`
112	theorem	`PaperXI.BreakOddPhase.internal_form`
126	theorem	`PaperXI.BreakOddPhase.internal_moving_form`
147	theorem	`PaperXI.BreakOddPhase.source`

A.54 `PaperXI.BreakQuadratic`

Selected layer: v0.44 parent. **Source lines:** 104.

SHA-256: `37bf741c07a7c2c90e5139d5bd490d61a60eccc0d2573bcf0d51e59ce81f4f6a`.

```
Actual finite orthogonal products of dyadic quadratic refinements.
The index type may be empty and the factors may have dimension zero.
```

Direct project imports: `PaperXI.DyadicAbstractOrthogonal`.

Line	Kind	Name
16	def	`PaperXI.BreakQuadratic.form`
25	theorem	`PaperXI.BreakQuadratic.form_apply`
28	theorem	`PaperXI.BreakQuadratic.form_alternating`
32	theorem	`PaperXI.BreakQuadratic.form_single_right`
42	theorem	`PaperXI.BreakQuadratic.form_nondegenerate`
51	def	`PaperXI.BreakQuadratic.refinement`
57	theorem	`PaperXI.BreakQuadratic.refinement_apply`
63	theorem	`PaperXI.BreakQuadratic.refinement_polar`
71	theorem	`PaperXI.BreakQuadratic.gauss_sum_product`
82	theorem	`PaperXI.BreakQuadratic.finrank_eq_twice_half`
91	theorem	`PaperXI.BreakQuadratic.half_finrank_product`

A.55 PaperXI.BreakTensorFactorization

Selected layer: v0.44 parent. **Source lines:** 77.

SHA-256: `9201c286d4ce3d4c53eda5ba1a756940f87a36ce7ddfb768a007ec158398ec24`.

```
Source 15.2: actual tensor factorization on the declared full group.
Compatible factor embeddings construct the group comparison. In the odd
symplectic branch both this comparison and scalar nondegeneracy are derived.
```

Direct project imports: `PaperXI.CentralProductComparison`, `PaperXI.HeisenbergTensorComparison`, `PaperXI.BreakHeisenbergSum`, `PaperXI.AbstractHeisenbergExtension`.

Line	Kind	Name
23	def	`PaperXI.BreakTensorFactorization.source`
53	theorem	`PaperXI.BreakTensorFactorization.odd_source`

A.56 PaperXI.BreakTensorMap

Selected layer: v0.44 parent. **Source lines:** 100.

SHA-256: `52fc48e66b2d78e5a85d528b3d8eccdee4b0668c3dc10b4ba49f3c44482caebb`.

```
Actual finite tensor maps of the supplied break intertwiners.
```

Direct project imports: `PaperXI.FiniteTensor`.

Line	Kind	Name
14	def	`PaperXI.BreakTensorMap.tensor`
17	theorem	`PaperXI.BreakTensorMap.tensor_tprod`
22	theorem	`PaperXI.BreakTensorMap.tensor_symm_tprod`
27	theorem	`PaperXI.BreakTensorMap.tensor_symm`
35	theorem	`PaperXI.BreakTensorMap.intertwines`
49	theorem	`PaperXI.BreakTensorMap.descends`
62	def	`PaperXI.BreakTensorMap.stabilizer`
66	theorem	`PaperXI.BreakTensorMap.stabilizer_tprod`
73	def	`PaperXI.BreakTensorMap.productCharacter`
78	theorem	`PaperXI.BreakTensorMap.productCharacter_apply`
82	theorem	`PaperXI.BreakTensorMap.scalar_intertwines`

A.57 PaperXI.CanonicalDVRValuation

Selected layer: v0.44 parent. **Source lines:** 225.

SHA-256: `d30233e7e4f5fffd0c9ee97fc3e54a765389da26cf8f917bd7c158e249eae6a1`.

```
# The canonical adic valuation of the constructed upper DVR

The fraction field constructed in `LocalFieldResidueExtension` is literally the
fraction field of the constructed DVR.  Consequently Mathlib's maximal-ideal
valuation identifies that DVR with the valuation unit ball.  This file records
that identification and proves the unit/uniformizer normalization required by
the local-field attachment.
```

Direct project imports: `PaperXI.LocalFieldResidueExtension`.

Line	Kind	Name
28	abbrev	`PaperXI.CanonicalDVRValuation.canonicalValuation`
32	def	`PaperXI.CanonicalDVRValuation.integerRingEquiv`
37	theorem	`PaperXI.CanonicalDVRValuation.integerRingEquiv_coe`
43	theorem	`PaperXI.CanonicalDVRValuation.integerRing_image`
50	theorem	`PaperXI.CanonicalDVRValuation.integer_iff_exists_lift`
59	theorem	`PaperXI.CanonicalDVRValuation.unit_value`
65	theorem	`PaperXI.CanonicalDVRValuation.maximalIdeal_iff_value_lt_one`
72	theorem	`PaperXI.CanonicalDVRValuation.uniformizer_isUniformizer`
85	theorem	`PaperXI.CanonicalDVRValuation.uniformizer_value`
92	theorem	`PaperXI.CanonicalDVRValuation.uniformizer_value_lt_one`
108	abbrev	`PaperXI.CanonicalDVRValuation.upperRing`
109	abbrev	`PaperXI.CanonicalDVRValuation.upperField`
110	abbrev	`PaperXI.CanonicalDVRValuation.upperValuation`
115	def	`PaperXI.CanonicalDVRValuation.upperIntegerRingEquiv`
121	theorem	`PaperXI.CanonicalDVRValuation.upperIntegerRingEquiv_coe`
129	theorem	`PaperXI.CanonicalDVRValuation.upper_integerRing_image`
136	theorem	`PaperXI.CanonicalDVRValuation.upper_integer_iff_exists_lift`
144	theorem	`PaperXI.CanonicalDVRValuation.upper_unit_value`
152	theorem	`PaperXI.CanonicalDVRValuation.base_unit_value`
161	theorem	`PaperXI.CanonicalDVRValuation.base_uniformizer_generates_upper`
171	theorem	`PaperXI.CanonicalDVRValuation.base_uniformizer_isUniformizer`
182	theorem	`PaperXI.CanonicalDVRValuation.base_uniformizer_value`
191	def	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer`
195	theorem	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer_irreducible`
200	theorem	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer_span`
207	theorem	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer_isUpperUniformizer`
216	theorem	`PaperXI.CanonicalDVRValuation.chosenBaseUniformizer_value`

A.58 `PaperXI.CentralCharacterAction`

Selected layer: v0.44 parent. **Source lines:** 111.

SHA-256: `7d5028051c4d9345e5ff3724fa65990af1cb4900a25053f54c82f6e7dafcf04f`.

```
An actual representation with a prescribed central character yields an
actual action of its section's twisted group algebra.
```

Direct project imports: `PaperXI.CentralExtensionCocycle`, `PaperXI.TwistedAction`.

Line	Kind	Name
15	theorem	`PaperXI.CentralCharacter.section_operators_one`
20	theorem	`PaperXI.CentralCharacter.section_operators_mul`
32	abbrev	`PaperXI.CentralCharacter.Algebra`
38	abbrev	`PaperXI.CentralCharacter.basis`
42	def	`PaperXI.CentralCharacter.toAlgebraAction`
50	theorem	`PaperXI.CentralCharacter.toAlgebraAction_basis`
59	theorem	`PaperXI.CentralCharacter.action_normal_form`
76	theorem	`PaperXI.CentralCharacter.action_stable_iff`

Line	Kind	Name
93	theorem	`PaperXI.CentralCharacter.action_intertwines_iff`

A.59 `PaperXI.CentralCharacterInduction`

Selected layer: base108 arithmetic branch. **Source lines:** 62.

SHA-256: `26563eced498d52375622ece6a9b1081ff7d338c5f94399babdd60a81e749349`.

```
# The central action of actual induction

The scalar on a central element of the inducing subgroup remains the scalar on
the actual induced representation. Consequently, two different such scalars
cannot be related by a nonzero ambient intertwiner. No irreducibility hypothesis
or finite-index hypothesis is used.
```

Direct project imports: `PaperXI.OpenSubgroupSmoothInduction`.

Line	Kind	Name
22	theorem	`PaperXI.CentralCharacterInduction.ind_central_scalar`
37	theorem	`PaperXI.CentralCharacterInduction.intertwiner_eq_zero_of_scalar_ne`
51	theorem	`PaperXI.CentralCharacterInduction.ind_intertwiner_eq_zero`

A.60 `PaperXI.CentralCharacterIrreducible`

Selected layer: v0.44 parent. **Source lines:** 49.

SHA-256: `bb4d95bc7100a200f2b5e915893df883a1e7768c06b5e5006432e5b6c4b3d367`.

```
Irreducibility of an actual central-character representation is equivalent
to simplicity of its constructed twisted-algebra module.
```

Direct project imports: `PaperXI.CentralCharacterAction`, `PaperXI.CentralCharacterRepresentation`, `PaperXI.AlgebraActionModule`, `PaperXI.RepresentationStability`.

Line	Kind	Name
18	theorem	`PaperXI.CentralCharacter.irreducible_iff_simple`
34	theorem	`PaperXI.CentralCharacter.simpleModule`
42	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_irreducible_iff`

A.61 `PaperXI.CentralCharacterRepresentation`

Selected layer: v0.44 parent. **Source lines:** 83.

SHA-256: `115c7bf9c958e23517b9990efc2179bb7595d5c416e839a96ee9705c1448d3fe`.

```
Reconstruction of the actual central-character representation from an
algebra action, with both action roundtrips.
```

Direct project imports: `PaperXI.CentralCharacterAction`.

Line	Kind	Name
14	def	`PaperXI.CentralCharacter.ofAlgebraAction`
27	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_apply`
33	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_central`
40	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_section`
46	theorem	`PaperXI.CentralCharacter.ofAlgebraAction_isUnit`
53	theorem	`PaperXI.CentralCharacter.of_toAlgebraAction`
62	theorem	`PaperXI.CentralCharacter.to_ofAlgebraAction`
74	def	`PaperXI.CentralCharacter.actionEquiv`

A.62 `PaperXI.CentralExtension`

Selected layer: v0.44 parent. **Source lines:** 85.

SHA-256: `738dea99a4dea853d5968a45499d8cd3804c8aae571749faa7ecd0f2beada19a`.

```
An actual central extension and normalized coordinates on its middle group.
No characteristic, finiteness, or scalar-character assumption is used.
```

Direct project imports: `PaperXI.CocycleCommutator`.

Line	Kind	Name
11	structure	`PaperXI.CentralExtension.Datum`
22	theorem	`PaperXI.CentralExtension.q_i`
28	def	`PaperXI.CentralExtension.sectionMap`
32	theorem	`PaperXI.CentralExtension.section_one`
35	theorem	`PaperXI.CentralExtension.q_section`
41	theorem	`PaperXI.CentralExtension.exists_coordinate`
49	def	`PaperXI.CentralExtension.coordinate`
51	theorem	`PaperXI.CentralExtension.coordinate_spec`
56	theorem	`PaperXI.CentralExtension.normal_form`
60	theorem	`PaperXI.CentralExtension.coordinate_eq_of_eq`
67	theorem	`PaperXI.CentralExtension.coordinate_one`
70	theorem	`PaperXI.CentralExtension.coordinate_section`
73	theorem	`PaperXI.CentralExtension.coordinate_i`
77	theorem	`PaperXI.CentralExtension.normal_form_unique`

A.63 `PaperXI.CentralExtensionCocycle`

Selected layer: v0.44 parent. **Source lines:** 127.

SHA-256: `20408e0fc0cb3e5f4edced01e3f8829ed5b80a905ed55a06d4545e27170151fa`.

```
The normalized central and scalar cocycles of an actual central extension.
The defect and its cocycle law are derived from group multiplication.
```

Direct project imports: `PaperXI.CentralExtension`.

Line	Kind	Name
15	def	`PaperXI.CentralExtension.defect`
17	theorem	`PaperXI.CentralExtension.section_mul`

Line	Kind	Name
22	theorem	`PaperXI.CentralExtension.defect_one_left`
25	theorem	`PaperXI.CentralExtension.defect_one_right`
28	theorem	`PaperXI.CentralExtension.normal_form_mul`
38	theorem	`PaperXI.CentralExtension.coordinate_mul`
49	theorem	`PaperXI.CentralExtension.defect_assoc`
56	def	`PaperXI.CentralExtension.centralCocycle`
60	theorem	`PaperXI.CentralExtension.normalized_centralCocycle`
64	def	`PaperXI.CentralExtension.scalarCocycle`
69	theorem	`PaperXI.CentralExtension.scalarCocycle_coeff`
72	theorem	`PaperXI.CentralExtension.normalized_scalarCocycle`
76	theorem	`PaperXI.CentralExtension.product_normal_form`
82	theorem	`PaperXI.CentralExtension.lift_commutation`
100	def	`PaperXI.CentralExtension.centralCommutator`
102	theorem	`PaperXI.CentralExtension.centralCommutator_eq`
110	theorem	`PaperXI.CentralExtension.i_centralCommutator`
116	theorem	`PaperXI.CentralExtension.centralCommutator_lift_independent`
123	theorem	`PaperXI.CentralExtension.scalar_commutator`

A.64 `PaperXI.CentralExtensionComparison`

Selected layer: v0.44 parent. **Source lines:** 61.

SHA-256: `e36d9e9e7da551d6db27596259bc69841e32a06e82d7b1c79f3bb69942a35082`.

```
The initial representation comparison along an actual extension lift,
without an odd-characteristic or coordinate-group restriction.
```

Direct project imports: `PaperXI.CentralExtensionUniqueness`, `PaperXI.HeisenbergTransport`.

Line	Kind	Name
15	def	`PaperXI.CentralExtensionComparison.equivAlong`
25	theorem	`PaperXI.CentralExtensionComparison.equivAlong_intertwines`
38	theorem	`PaperXI.CentralExtensionComparison.exists_unique_along`

A.65 `PaperXI.CentralExtensionExistence`

Selected layer: v0.44 parent. **Source lines:** 43.

SHA-256: `c073b326ebaed1aafad9eeaf1e4cbb0707dfa8112ae901e0b421f9d114227a2a`.

```
Existence is obtained from an actual simple ideal of the constructed
twisted algebra, then the reverse action construction.
```

Direct project imports: `PaperXI.CentralExtensionUniqueness`, `PaperXI.CentralCharacterRepresentation`.

Line	Kind	Name
16	theorem	`PaperXI.CentralExtensionExistence.exists_irreducible`

A.66 PaperXI.CentralExtensionInducing

Selected layer: base108 arithmetic branch. **Source lines:** 208.

SHA-256: a435de848de5be66c076c64e660fe71344453fd66a68dc1d2f842854b984317f.

```
# Extending inducing data by a central infinite cyclic direction

The extension is a representation of an actual subgroup of the ambient group.
Its hypotheses require that the adjoined central element has no nonzero power
in the original subgroup.  No assertion about simple types, intertwining, or
admissibility is supplied by this construction.
```

Direct project imports: PaperXI.SmoothDegreeQuotient, PaperXI.RepresentationStability.

Line	Kind	Name
21	theorem	PaperXI.CentralExtensionInducing.hfree_from_degree
32	def	PaperXI.CentralExtensionInducing.normalFormHom
45	theorem	PaperXI.CentralExtensionInducing.normalFormHom_apply
50	def	PaperXI.CentralExtensionInducing.extendedSubgroup
53	theorem	PaperXI.CentralExtensionInducing.le_extendedSubgroup
58	theorem	PaperXI.CentralExtensionInducing.central_mem_extendedSubgroup
63	theorem	PaperXI.CentralExtensionInducing.extendedSubgroup_eq_sup
75	theorem	PaperXI.CentralExtensionInducing.normalFormHom_injective
91	def	PaperXI.CentralExtensionInducing.normalFormEquiv
97	theorem	PaperXI.CentralExtensionInducing.normalFormEquiv_coe
104	def	PaperXI.CentralExtensionInducing.subgroupInclusion
109	def	PaperXI.CentralExtensionInducing.centralElement
116	def	PaperXI.CentralExtensionInducing.productRepresentation
127	def	PaperXI.CentralExtensionInducing.extendedRepresentation
135	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_normalForm
143	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_restrict
153	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_central
167	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_irreducible
183	theorem	PaperXI.CentralExtensionInducing.extendedSubgroup_isOpen
189	theorem	PaperXI.CentralExtensionInducing.extendedRepresentation_smooth

A.67 PaperXI.CentralExtensionInductionConsequences

Selected layer: base108 arithmetic branch. **Source lines:** 95.

SHA-256: 6e51c61426161f53ded9cdde06049b7358cf387ee47e88de993af2d98a6838f1.

```
# Actual induced candidates with prescribed central scalar

The central extension constructs nonzero smooth induced representations with
an arbitrary unit scalar on the adjoined central element.  Different scalars
give no nonzero ambient intertwiner.  These statements do not assert that the
induced representations are irreducible, admissible, or realizations of a
specified supercuspidal parameter.
```

Direct project imports: PaperXI.CentralExtensionInducing, PaperXI.CentralCharacterInduction.

Line	Kind	Name
22	def	PaperXI.CentralExtensionInductionConsequences.inducedRepresentation

Line	Kind	Name
31	theorem	PaperXI.CentralExtensionInductionConsequences.induced_central_scalar
41	theorem	PaperXI.CentralExtensionInductionConsequences.induced_nontrivial
50	theorem	PaperXI.CentralExtensionInductionConsequences.induced_intertwiner_eq_zero
62	theorem	PaperXI.CentralExtensionInductionConsequences.induced_not_equiv
84	theorem	PaperXI.CentralExtensionInductionConsequences.induced_isSmooth

A.68 PaperXI.CentralExtensionSectionChange

Selected layer: v0.44 parent. **Source lines:** 267.

SHA-256: 5360a7116c397b9588d620f7febc89b633f697bd8cec091dd4ecc76468e03f4f.

Independence of normalized quotient section for the actual cocycle and twisted algebra of a central extension. The alternate defect is computed in the middle group, then identified with the derived coboundary.

Direct project imports: PaperXI.CentralExtensionCocycle, PaperXI.TwistedRescaling, PaperXI.CentralCharacterAction.

Line	Kind	Name
18	theorem	PaperXI.CentralExtension.cocycle_eq_of_coeff_eq
27	def	PaperXI.CentralExtension.sectionFactor
29	theorem	PaperXI.CentralExtension.section_factor
33	theorem	PaperXI.CentralExtension.sectionFactor_one
38	def	PaperXI.CentralExtension.sectionDefect
41	theorem	PaperXI.CentralExtension.i_sectionDefect
46	theorem	PaperXI.CentralExtension.sectionDefect_mul
51	theorem	PaperXI.CentralExtension.sectionDefect_eq_rescale
70	def	PaperXI.CentralExtension.sectionCocycle
76	theorem	PaperXI.CentralExtension.sectionCocycle_eq_rescale
81	theorem	PaperXI.CentralExtension.normalized_sectionCocycle
88	def	PaperXI.CentralExtension.sectionScalarCocycle
94	theorem	PaperXI.CentralExtension.sectionScalarCocycle_eq_rescale
106	theorem	PaperXI.CentralExtension.normalized_sectionScalarCocycle
114	theorem	PaperXI.CentralExtension.sectionScalarCocycle_commutator
121	theorem	PaperXI.CentralExtension.sectionScalarCocycle_radical
130	def	PaperXI.CentralExtension.sectionAlgebraEquiv
142	theorem	PaperXI.CentralExtension.sectionAlgebraEquiv_basis
152	theorem	PaperXI.CentralExtension.sectionAlgebraEquiv_symm_basis
165	def	PaperXI.CentralExtension.betweenSectionsEquiv
174	theorem	PaperXI.CentralExtension.betweenSectionsEquiv_basis
191	theorem	PaperXI.CentralExtension.sectionOperators_factor
202	theorem	PaperXI.CentralExtension.sectionOperators_mul
214	def	PaperXI.CentralExtension.sectionAction
224	theorem	PaperXI.CentralExtension.sectionAction_basis
235	theorem	PaperXI.CentralExtension.sectionAction_eq_transport
254	theorem	PaperXI.CentralExtension.sectionAction_betweenSections

A.69 PaperXI.CentralExtensionStructure

Selected layer: v0.44 parent. **Source lines:** 117.

SHA-256: `601bab4de431a620e5b9f2c84ea154cd63b7336382468bdf4dd6cca2a0a1ee35`.

```
Coordinates, cardinality, and intrinsic scalar nondegeneracy of an actual
central extension. Faithfulness is an explicit hypothesis when a central-valued
nondegeneracy statement is transported through a central character.
```

Direct project imports: `PaperXI.CentralExtensionCocycle`.

Line	Kind	Name
17	def	`PaperXI.CentralExtension.coordinates`
26	theorem	`PaperXI.CentralExtension.coordinates_apply`
29	theorem	`PaperXI.CentralExtension.coordinates_symm_apply`
33	theorem	`PaperXI.CentralExtension.finite_middle`
38	theorem	`PaperXI.CentralExtension.natCard_eq`
42	theorem	`PaperXI.CentralExtension.card_eq`
48	theorem	`PaperXI.CentralExtension.radical_scalarCocycle_eq_bot_iff`
69	theorem	`PaperXI.CentralExtension.radical_scalarCocycle_eq_bot_of_faithful`
81	theorem	`PaperXI.CentralExtension.radical_scalarCocycle_of_injective`
95	theorem	`PaperXI.CentralExtension.radical_scalarCocycle_eq_bot_of_central`
102	theorem	`PaperXI.CentralExtension.center_eq_range`

A.70 `PaperXI.CentralExtensionUniqueness`

Selected layer: v0.44 parent. **Source lines:** 99.

SHA-256: `2e5ae550279047849c064e92382606a085429ff1dabb04de593c14006d39a9d2`.

```
Finite Stone–von Neumann for actual central extensions with nondegenerate
scalar commutator. Only the quotient is required finite. The argument works
in every characteristic and derives the dimensions of the given irreducibles.
```

Direct project imports: `PaperXI.CentralCharacterIrreducible`, `PaperXI.TwistedModules`, `PaperXI.SchurScalar`.

Line	Kind	Name
18	theorem	`PaperXI.CentralExtensionUniqueness.finiteDimensional`
33	theorem	`PaperXI.CentralExtensionUniqueness.finrank_sqrt`
47	theorem	`PaperXI.CentralExtensionUniqueness.exists_equiv`
68	def	`PaperXI.CentralExtensionUniqueness.equiv`
76	theorem	`PaperXI.CentralExtensionUniqueness.equiv_intertwines`
87	theorem	`PaperXI.CentralExtensionUniqueness.unique_scalar`

A.71 `PaperXI.CentralProduct`

Selected layer: v0.44 parent. **Source lines:** 159.

SHA-256: `9d1e6a4ece84a862ef75383b66b97af694c9ac2298fc2047a6c4844bc6aa34af`.

```
The finite central product, with an explicit central factor so that the empty
family has the correct tensor-unit center. The group is an actual quotient.
```

Direct project imports: `PaperXI.CentralExtension`.

Line	Kind	Name
15	abbrev	`PaperXI.CentralProduct.Ambient`
18	def	`PaperXI.CentralProduct.balanced`
23	def	`PaperXI.CentralProduct.relation`
25	theorem	`PaperXI.CentralProduct.balanced_central`
32	instance	`PaperXI.CentralProduct.relation_normal`
39	def	`PaperXI.CentralProduct.Group`
43	def	`PaperXI.CentralProduct.mk`
45	theorem	`PaperXI.CentralProduct.mk_surjective`
47	theorem	`PaperXI.CentralProduct.mk_balanced`
50	def	`PaperXI.CentralProduct.center`
52	theorem	`PaperXI.CentralProduct.center_apply`
54	theorem	`PaperXI.CentralProduct.center_injective`
69	def	`PaperXI.CentralProduct.ambientProjection`
74	theorem	`PaperXI.CentralProduct.relation_le_ker_projection`
79	def	`PaperXI.CentralProduct.projection`
82	theorem	`PaperXI.CentralProduct.projection_mk`
85	theorem	`PaperXI.CentralProduct.projection_center`
89	theorem	`PaperXI.CentralProduct.projection_surjective`
96	theorem	`PaperXI.CentralProduct.mk_centers`
105	theorem	`PaperXI.CentralProduct.range_center_eq_ker_projection`
122	theorem	`PaperXI.CentralProduct.center_commutes`
131	def	`PaperXI.CentralProduct.datum`
139	theorem	`PaperXI.CentralProduct.datum_i`
141	theorem	`PaperXI.CentralProduct.datum_q`
144	def	`PaperXI.CentralProduct.descend`
148	theorem	`PaperXI.CentralProduct.descend_mk`
153	theorem	`PaperXI.CentralProduct.hom_ext`

A.72 `PaperXI.CentralProductCommutator`

Selected layer: v0.44 parent. **Source lines:** 72.

SHA-256: `00857ad77d53b9578db693336a58114a88ad7c58f683a74132a7730fff7c4dbe`.

```
The actual quotient commutator is the product of the factor commutators.
Scalar nondegeneracy is inherited without assuming a faithful central character.
```

Direct project imports: `PaperXI.CentralProduct`, `PaperXI.CentralExtensionStructure`.

Line	Kind	Name
16	theorem	`PaperXI.CentralProduct.commutator_mk`
32	theorem	`PaperXI.CentralProduct.scalar_commutator_product`
48	theorem	`PaperXI.CentralProduct.radical_eq_bot`
67	theorem	`PaperXI.CentralProduct.full_center`

A.73 `PaperXI.CentralProductComparison`

Selected layer: v0.44 parent. **Source lines:** 149.

SHA-256: `0c1567c5e43684d459663ec1d8c0995f0cdd4d26a9b3b84b56e719e76dcf1a67`.

```
Actual compatible factor embeddings construct the comparison with a declared
full central extension. The desired group isomorphism is derived from exactness.
```

Direct project imports: `PaperXI.CentralProductFactors`.

Line	Kind	Name
22	def	`PaperXI.CentralProductComparison.factors`
24	theorem	`PaperXI.CentralProductComparison.factors_single`
30	theorem	`PaperXI.CentralProductComparison.factors_projection`
45	def	`PaperXI.CentralProductComparison.centerProduct`
52	theorem	`PaperXI.CentralProductComparison.factors_centers`
72	def	`PaperXI.CentralProductComparison.ambient`
76	theorem	`PaperXI.CentralProductComparison.ambient_apply`
81	theorem	`PaperXI.CentralProductComparison.ambient_balanced`
87	def	`PaperXI.CentralProductComparison.comparison`
90	theorem	`PaperXI.CentralProductComparison.comparison_mk`
93	theorem	`PaperXI.CentralProductComparison.comparison_center`
99	theorem	`PaperXI.CentralProductComparison.comparison_projection`
108	theorem	`PaperXI.CentralProductComparison.comparison_injective`
126	theorem	`PaperXI.CentralProductComparison.comparison_surjective`
138	def	`PaperXI.CentralProductComparison.equiv`
142	theorem	`PaperXI.CentralProductComparison.equiv_center`
146	theorem	`PaperXI.CentralProductComparison.equiv_mk`

A.74 `PaperXI.CentralProductFactors`

Selected layer: v0.44 parent. **Source lines:** 89.

SHA-256: `2c019b264e9f537a7886668193c3e1777ac631b16c10cf02acbd25e8f30b3e82`.

```
The factor injections and the empty-family center of the actual finite central product.
```

Direct project imports: `PaperXI.CentralProduct`.

Line	Kind	Name
13	def	`PaperXI.CentralProduct.factor`
16	theorem	`PaperXI.CentralProduct.factor_apply`
20	theorem	`PaperXI.CentralProduct.projection_factor`
28	theorem	`PaperXI.CentralProduct.factor_center`
43	theorem	`PaperXI.CentralProduct.factor_injective`
61	theorem	`PaperXI.CentralProduct.factor_commute`
72	theorem	`PaperXI.CentralProduct.center_surjective_of_isEmpty`
81	def	`PaperXI.CentralProduct.emptyEquiv`
86	theorem	`PaperXI.CentralProduct.emptyEquiv_center`

A.75 `PaperXI.CentralProductMap`

Selected layer: v0.44 parent. **Source lines:** 64.

SHA-256: `a5dad822167784fc5033f5187a85a8e99a172466c1f65742380c20fa01d6e4d8`.

```
Actual maps and isomorphisms of the central quotients induced by compatible factor maps.
```

Direct project imports: `PaperXI.CentralProduct`.

Line	Kind	Name
15	def	`PaperXI.CentralProduct.ambientMap`
21	def	`PaperXI.CentralProduct.map`
29	theorem	`PaperXI.CentralProduct.map_mk`
34	theorem	`PaperXI.CentralProduct.map_center`
40	def	`PaperXI.CentralProduct.congr`
55	theorem	`PaperXI.CentralProduct.congr_mk`
60	theorem	`PaperXI.CentralProduct.congr_center`

A.76 `PaperXI.ChainSharpness`

Selected layer: v0.44 parent. **Source lines:** 25.

SHA-256: `9dc676cc958252b1a0d2ad618ee5f85bcded1dbd8813022e05f9a32c85129ece`.

Direct project imports: `PaperXI.MonodromyChain`.

Line	Kind	Name
9	theorem	`PaperXI.MonodromyChain.identity_path`
16	theorem	`PaperXI.MonodromyChain.identity_shift_power_ne_zero`

A.77 `PaperXI.CharacterAttachment108`

Selected layer: base108 arithmetic branch. **Source lines:** 73.

SHA-256: `3ab5684d139d5f725153a9bc2d64d616384a28734b3839f0bc3c1f89856219fe`.

```
Attach the actual character to the prescribed carrier. A single character
vector and character normalization imply scalar action, by irreducibility.
```

Direct project imports: `PaperXI.TypeSideDescent`, `PaperXI.CharacterNormalizer61`, `PaperXI.RepresentationStability`, `PaperXI.IntertwiningAlong105`.

Line	Kind	Name
14	theorem	`PaperXI.CharacterAttachment108.irreducible_of_restriction`
23	def	`PaperXI.CharacterAttachment108.insideCharacter`
27	theorem	`PaperXI.CharacterAttachment108.scalar_of_contains`
50	theorem	`PaperXI.CharacterAttachment108.character_of_scalar`

A.78 `PaperXI.CharacterGenerators`

Selected layer: v0.44 parent. **Source lines:** 59.

SHA-256: `5f499da97dbae7d5948989b633f51e14f7adac1468a62e22af5ec653fbc2b1a6`.

```
Triviality of the actual stabilizer discrepancy on a generating set.
```

Direct project imports: `PaperXI.CyclicBridge`.

Line	Kind	Name
12	theorem	`PaperXI.CharacterGenerators.eq_one_on_closure`
18	theorem	`PaperXI.CharacterGenerators.trivial`
25	theorem	`PaperXI.CharacterGenerators.trivial_iff`
35	theorem	`PaperXI.CharacterGenerators.bridge_from_actions`

A.79 `PaperXI.CharacterNormalizer61`

Selected layer: base108 arithmetic branch. **Source lines:** 104.

SHA-256: `4d1fb07d5d106c49247b3900dea666ce7859054359e7328de46b86d3cf3cdf1f`.

```
# The literal normalizer of a subgroup-character pair

The group is defined from the graph of the actual character. It is therefore
sensitive to both its domain and its values, rather than a separately supplied
group label. Compact parts are formed by intersection with a compact subgroup.
```

Direct project imports: `PaperXI.NormalizerIntertwining60`.

Line	Kind	Name
18	def	`PaperXI.CharacterNormalizer61.graph`
21	def	`PaperXI.CharacterNormalizer61.normalizer`
25	theorem	`PaperXI.CharacterNormalizer61.conjugation_mem`
34	theorem	`PaperXI.CharacterNormalizer61.character_covariance`
45	theorem	`PaperXI.CharacterNormalizer61.normalizes_domain`
57	theorem	`PaperXI.CharacterNormalizer61.domain_le_normalizer`
76	theorem	`PaperXI.CharacterNormalizer61.character_intertwinedAt`
85	def	`PaperXI.CharacterNormalizer61.compactPart`
91	theorem	`PaperXI.CharacterNormalizer61.normalizer_open`
95	theorem	`PaperXI.CharacterNormalizer61.compactPart_compact_open`

A.80 `PaperXI.CharacterPairing108`

Selected layer: base108 arithmetic branch. **Source lines:** 78.

SHA-256: `e86637f400e8ad723c2b13db11b5fc48fb9110e11a1d764d1925fed4a1e06dfb`.

```
Exact character pairing on a finite intersection. The index equality used
for dimension one is an explicit, separate arithmetic obligation.
```

Direct project imports: `PaperXI.HeisenbergTrace108`.

Line	Kind	Name
22	theorem	`PaperXI.CharacterPairing108.dimension_formula`
65	theorem	`PaperXI.CharacterPairing108.dimension_one_iff`

A.81 `PaperXI.CharacterProjector`

Selected layer: v0.44 parent. **Source lines:** 100.

SHA-256: `c3d3017695299f3ef67a28bee5c519bbe37aa9c0768e52c4fdbb8c9be912d7cc`.

```
The literal character element of a finite complex group algebra.
Centrality and scalar action are proved from character orthogonality.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.CharacterProjector.element`
20	theorem	`PaperXI.CharacterProjector.coeff_element`
26	theorem	`PaperXI.CharacterProjector.central`
42	theorem	`PaperXI.CharacterProjector.action`
50	theorem	`PaperXI.CharacterProjector.central_scalar`
58	theorem	`PaperXI.CharacterProjector.nontrivial_space`
66	theorem	`PaperXI.CharacterProjector.degree_ne_zero`
72	theorem	`PaperXI.CharacterProjector.irreducible_action`

A.82 `PaperXI.CharacterSupport108`

Selected layer: base108 arithmetic branch. **Source lines:** 71.

SHA-256: `4a1a02bf6685a09772dbbe49ad1f29037975b2bbc299e736b1eba53df4aa8960`.

```
Equality of actual intertwining supports after character attachment. The
centralizer beta-intertwining input follows from positive Heisenberg dimension.
```

Direct project imports: `PaperXI.CharacterAttachment108`, `PaperXI.FullTypeIntertwining105`, `PaperXI.IntertwiningSpace107`.

Line	Kind	Name
16	def	`PaperXI.CharacterSupport108.intoFirst`
26	theorem	`PaperXI.CharacterSupport108.restricted_iff_character`
38	theorem	`PaperXI.CharacterSupport108.full_iff_character`
48	theorem	`PaperXI.CharacterSupport108.full_iff_restricted`
54	theorem	`PaperXI.CharacterSupport108.existsAt_of_dimension_pos`
64	theorem	`PaperXI.CharacterSupport108.beta_of_dimension_pos`

A.83 `PaperXI.CharacterTwist`

Selected layer: v0.44 parent. **Source lines:** 54.

SHA-256: `2f04977b3efe71419a28d3985c9ca035ba1e9f43cccb4779638ea69098b5f58e`.

```
Scalar twisting and its literal tensor comparison.
```

Direct project imports: `PaperXI.CyclicInducedCoordinates`.

Line	Kind	Name
11	def	`PaperXI.CharacterTwist.twist`
16	theorem	`PaperXI.CharacterTwist.twist_apply`
20	def	`PaperXI.CharacterTwist.tensorEquiv`
27	def	`PaperXI.CharacterTwist.mapEquiv`
37	theorem	`PaperXI.CharacterTwist.restrict_eq`

Line	Kind	Name
42	theorem	`PaperXI.CharacterTwist.power_action`
48	theorem	`PaperXI.CharacterTwist.closing_scalar`

A.84 `PaperXI.CliffordAction`

Selected layer: v0.44 parent. **Source lines:** 86.

SHA-256: `aff52bad200146645c61ebf8647615481ee1f908eb4fa8aa5ccdd39bf79393bf`.

```
The full Clifford choice comparison and irreducibility for every
section, intertwiner family, and equivariant tensor identification.
```

Direct project imports: `PaperXI.CliffordIrreducible`, `PaperXI.CliffordChoices`.

Line	Kind	Name
31	theorem	`PaperXI.CliffordAction.all_choices`
56	theorem	`PaperXI.CliffordAction.irreducible_all_choices`

A.85 `PaperXI.CliffordChoices`

Selected layer: v0.44 parent. **Source lines:** 148.

SHA-256: `1e5599eae92df49fe01d3a846e8a21e070e97ca8aa79461ebafdfaf666c9a9e0`.

```
Choice comparisons on the literal Clifford multiplicity operators.
The comparing scalar is derived from Schur's lemma.
```

Direct project imports: `PaperXI.CliffordMultiplicity`, `PaperXI.ProjectiveCocycle`.

Line	Kind	Name
16	theorem	`PaperXI.CliffordChoices.operator_rescale`
32	theorem	`PaperXI.CliffordChoices.existsUnique_intertwiner_scalar`
46	def	`PaperXI.CliffordChoices.intertwinerScalar`
52	theorem	`PaperXI.CliffordChoices.intertwinerScalar_spec`
61	theorem	`PaperXI.CliffordChoices.operator_choice`
72	def	`PaperXI.CliffordChoices.representativeFactor`
79	theorem	`PaperXI.CliffordChoices.representative_factorization`
85	def	`PaperXI.CliffordChoices.representativeIntertwiner`
103	theorem	`PaperXI.CliffordChoices.representativeIntertwiner_apply`
110	theorem	`PaperXI.CliffordChoices.operator_representative`
126	def	`PaperXI.CliffordChoices.choiceCochain`
135	theorem	`PaperXI.CliffordChoices.arbitrary_choices`

A.86 `PaperXI.CliffordCocycle`

Selected layer: v0.44 parent. **Source lines:** 98.

SHA-256: `915d42076be1fed3ce0abb96e2e74367ebbd9d51d78dd6220a3cd73f37a04354`.

```
The Clifford cocycle is extracted from the literal multiplicity action.
The scalar on the multiplicity space is the inverse of the Schur scalar
comparing the subgroup intertwiners.
```

Direct project imports: `PaperXI.CliffordChoices`.

Line	Kind	Name
16	def	`PaperXI.CliffordCocycle.scalar`
25	theorem	`PaperXI.CliffordCocycle.scalar_spec`
38	theorem	`PaperXI.CliffordCocycle.operator_defect`
52	theorem	`PaperXI.CliffordCocycle.sectionOperator_mul`
67	theorem	`PaperXI.CliffordCocycle.sectionOperator_ne_zero`
77	def	`PaperXI.CliffordCocycle.cocycle`
85	theorem	`PaperXI.CliffordCocycle.cocycle_coeff`
91	theorem	`PaperXI.CliffordCocycle.normalized`

A.87 `PaperXI.CliffordCocycleChoices`

Selected layer: v0.44 parent. **Source lines:** 87.

SHA-256: `b81243b34012e08c729b15becd2058df413240d7047d7975e8f6d119746f4bd8`.

```
Explicit cocycle and commutator comparisons for all actual section and
intertwiner choices in the Clifford construction.
```

Direct project imports: `PaperXI.CliffordCocycle`.

Line	Kind	Name
18	def	`PaperXI.CliffordCocycleChoices.choiceCocycle`
23	theorem	`PaperXI.CliffordCocycleChoices.choiceOperator_mul`
35	theorem	`PaperXI.CliffordCocycleChoices.choiceCocycle_unique`
55	theorem	`PaperXI.CliffordCocycleChoices.commutator_choice`
62	theorem	`PaperXI.CliffordCocycleChoices.radical_choice`
68	theorem	`PaperXI.CliffordCocycleChoices.choiceCochain_one`
80	theorem	`PaperXI.CliffordCocycleChoices.choiceCocycle_normalized`

A.88 `PaperXI.CliffordCocycleClass`

Selected layer: v0.44 parent. **Source lines:** 111.

SHA-256: `88207bfc7c10216386294ac5631533d2b07f6a592136a204d91b41553bd7711c`.

```
The class of the actual Clifford multiplicity cocycle in the explicit
quotient of normalized cocycles by normalized scalar coboundaries.
```

Direct project imports: `PaperXI.CocycleClass`, `PaperXI.CliffordCocycleChoices`, `PaperXI.CliffordTransport`, `PaperXI.CliffordSource`.

Line	Kind	Name
21	def	`PaperXI.CliffordCocycleClass.value`
24	theorem	`PaperXI.CliffordCocycleClass.commutator_value`
28	theorem	`PaperXI.CliffordCocycleClass.radical_value`
32	theorem	`PaperXI.CliffordCocycleClass.choice_class`
42	theorem	`PaperXI.CliffordCocycleClass.class_of_multiplication`
67	theorem	`PaperXI.CliffordCocycleClass.value_transport`

Line	Kind	Name
82	def	`PaperXI.CliffordCocycleClass.sourceValue`
101	theorem	`PaperXI.CliffordCocycleClass.source_transport`

A.89 `PaperXI.CliffordDatum`

Selected layer: v0.44 parent. **Source lines:** 112.

SHA-256: `7321b347ed121e9981fdb5e48e5cdcc56ffa0b1ed4c556fb547dcc4a7eab8b5b`.

```
Actual restriction, conjugation and normalized quotient representatives
for the Clifford multiplicity construction.
```

Direct project imports: `PaperXI.QuotientRepresentation`.

Line	Kind	Name
14	abbrev	`PaperXI.CliffordDatum.restriction`
18	def	`PaperXI.CliffordDatum.conjugation`
23	theorem	`PaperXI.CliffordDatum.conjugation_coe`
26	theorem	`PaperXI.CliffordDatum.conjugation_one`
29	theorem	`PaperXI.CliffordDatum.conjugation_mul`
34	theorem	`PaperXI.CliffordDatum.conjugation_inv_cancel`
40	abbrev	`PaperXI.CliffordDatum.conjugate`
44	def	`PaperXI.CliffordDatum.Invariant`
48	def	`PaperXI.CliffordDatum.sectionMap`
53	theorem	`PaperXI.CliffordDatum.section_one`
56	theorem	`PaperXI.CliffordDatum.quotient_section`
64	def	`PaperXI.CliffordDatum.defect`
71	theorem	`PaperXI.CliffordDatum.defect_coe`
74	theorem	`PaperXI.CliffordDatum.section_mul`
78	theorem	`PaperXI.CliffordDatum.defect_one_left`
81	theorem	`PaperXI.CliffordDatum.defect_one_right`
85	def	`PaperXI.CliffordDatum.coordinate`
92	theorem	`PaperXI.CliffordDatum.normal_form`
97	def	`PaperXI.CliffordDatum.sectionIntertwiner`
106	theorem	`PaperXI.CliffordDatum.sectionIntertwiner_one`

A.90 `PaperXI.CliffordEnvelope`

Selected layer: v0.44 parent. **Source lines:** 147.

SHA-256: `550b0e257a986100dd3ddf95022f7766f8366363760787b3b28b03c08cc04548`.

```
The actual elementary commutator envelope of the source Clifford datum.
No cocycle, projective operator family, bilinear form or transport on the
multiplicity space is supplied in place of its construction.
```

Direct project imports: `PaperXI.CliffordSource`, `PaperXI.CliffordTransport`, `PaperXI.ElementaryEnvelopeTransport`.

Line	Kind	Name
23	def	PaperXI.CliffordEnvelope.pairing
26	theorem	PaperXI.CliffordEnvelope.pairing_coe
31	theorem	PaperXI.CliffordEnvelope.pairing_self
35	abbrev	PaperXI.CliffordEnvelope.Space
40	def	PaperXI.CliffordEnvelope.form
43	theorem	PaperXI.CliffordEnvelope.form_isAlt
46	theorem	PaperXI.CliffordEnvelope.form_nondegenerate
49	theorem	PaperXI.CliffordEnvelope.finiteDimensional
54	theorem	PaperXI.CliffordEnvelope.form_on_classes
77	theorem	PaperXI.CliffordEnvelope.commutator_preserved
92	theorem	PaperXI.CliffordEnvelope.exists_normalized_cochain
110	theorem	PaperXI.CliffordEnvelope.pairing_preserved
118	def	PaperXI.CliffordEnvelope.equiv
124	theorem	PaperXI.CliffordEnvelope.equiv_mk
131	theorem	PaperXI.CliffordEnvelope.equiv_form
139	theorem	PaperXI.CliffordEnvelope.equiv_independent

A.91 PaperXI.CliffordEvaluation

Selected layer: v0.44 parent. **Source lines:** 242.

SHA-256: 405e9bf8ba1bdf8eba00ab3d6063ec48594e3b24d4d919d5a5b862e76bd4effd.

Direct project imports: PaperXI.IsotypicEvaluation, PaperXI.FiniteRepresentation, PaperXI.CliffordMultiplicity.

Line	Kind	Name
19	def	PaperXI.CliffordEvaluation.map
28	theorem	PaperXI.CliffordEvaluation.map_tmul
32	theorem	PaperXI.CliffordEvaluation.map_action
40	theorem	PaperXI.CliffordEvaluation.bijective
55	def	PaperXI.CliffordEvaluation.equiv
60	theorem	PaperXI.CliffordEvaluation.equiv_tmul
64	theorem	PaperXI.CliffordEvaluation.equiv_action
71	theorem	PaperXI.CliffordEvaluation.multiplicity_nontrivial
80	def	PaperXI.CliffordEvaluation.choiceMap
90	theorem	PaperXI.CliffordEvaluation.choiceMap_apply
96	theorem	PaperXI.CliffordEvaluation.choiceMap_injective
107	def	PaperXI.CliffordEvaluation.choiceEquiv
116	theorem	PaperXI.CliffordEvaluation.choiceEquiv_apply
123	theorem	PaperXI.CliffordEvaluation.exists_unique_choice
146	theorem	PaperXI.CliffordEvaluation.map_operator
156	theorem	PaperXI.CliffordEvaluation.operator_unique
166	theorem	PaperXI.CliffordEvaluation.choice_operator_conjugation
184	def	PaperXI.CliffordEvaluation.choiceOperator
194	theorem	PaperXI.CliffordEvaluation.choiceOperator_tensor
206	theorem	PaperXI.CliffordEvaluation.exists_unique_choice_operator
227	theorem	PaperXI.CliffordEvaluation.restriction_bijective
234	theorem	PaperXI.CliffordEvaluation.restriction_multiplicity_nontrivial

A.92 PaperXI.CliffordHeisenbergAttachment

Selected layer: v0.44 parent. **Source lines:** 144.

SHA-256: bfc127d6190ce5ad509d5e37b1b9507b92996f620a90657c855d90516c6648e7.

Source-facing Corollary 9.9. The form comes from the actual Clifford
multiplicity cocycle. Odd characteristic uses its intrinsic half-form group;
characteristic two retains a declared quadratic refinement and constructs a
bilinear lift. No irreducible Heisenberg representation is supplied as input.

Direct project imports: PaperXI.OddEnvelopeHeisenberg, PaperXI.DyadicHeisenbergLift, PaperXI.CliffordEnvelope.

Line	Kind	Name
27	def	PaperXI.CliffordHeisenbergAttachment.oddFactor
32	theorem	PaperXI.CliffordHeisenbergAttachment.odd_commutator_on_classes
50	theorem	PaperXI.CliffordHeisenbergAttachment.odd_exists
72	theorem	PaperXI.CliffordHeisenbergAttachment.odd_unique
91	def	PaperXI.CliffordHeisenbergAttachment.dyadicFactor
98	theorem	PaperXI.CliffordHeisenbergAttachment.dyadic_square
106	theorem	PaperXI.CliffordHeisenbergAttachment.dyadic_exists
128	theorem	PaperXI.CliffordHeisenbergAttachment.dyadic_unique

A.93 PaperXI.CliffordIrreducible

Selected layer: v0.44 parent. **Source lines:** 98.

SHA-256: 39476ad63852c2ef28b547eed56fe8805910a7beabbdde97f4cdbef6b5b83b63.

Canonical evaluation and irreducibility on the actual Clifford
multiplicity space. All finite dimensions and tensor-image properties are
derived from the finite-group irreducibles and genuine module isotypy.

Direct project imports: PaperXI.CliffordMultiplicity, PaperXI.CliffordEvaluation, PaperXI.TensorSubspace.

Line	Kind	Name
21	abbrev	PaperXI.CliffordIrreducible.Isotypic
25	theorem	PaperXI.CliffordIrreducible.multiplicity_finiteDimensional
31	theorem	PaperXI.CliffordIrreducible.multiplicity_nontrivial
37	def	PaperXI.CliffordIrreducible.evaluation
41	theorem	PaperXI.CliffordIrreducible.evaluation_tmul
47	theorem	PaperXI.CliffordIrreducible.section_action
64	theorem	PaperXI.CliffordIrreducible.group_action
80	theorem	PaperXI.CliffordIrreducible.irreducible

A.94 PaperXI.CliffordMultiplicity

Selected layer: v0.44 parent. **Source lines:** 241.

SHA-256: 9ad23300239f1a6cb777e10ea3a8efb66596640c0644862baf2db01c0089b1df.

Invertible operators on the actual space of subgroup intertwiners.

Direct project imports: `PaperXI.CliffordDatum`.

Line	Kind	Name
13	abbrev	`PaperXI.CliffordMultiplicity.Space`
16	theorem	`PaperXI.CliffordMultiplicity.space_intertwines`
20	theorem	`PaperXI.CliffordMultiplicity.inverse_intertwines`
27	theorem	`PaperXI.CliffordMultiplicity.action_conjugation`
35	def	`PaperXI.CliffordMultiplicity.operatorMap`
48	theorem	`PaperXI.CliffordMultiplicity.operatorMap_apply`
54	def	`PaperXI.CliffordMultiplicity.inverseMap`
74	theorem	`PaperXI.CliffordMultiplicity.inverseMap_apply`
80	def	`PaperXI.CliffordMultiplicity.operator`
100	theorem	`PaperXI.CliffordMultiplicity.operator_apply`
105	theorem	`PaperXI.CliffordMultiplicity.operator_symm_apply`
111	def	`PaperXI.CliffordMultiplicity.sectionOperator`
116	theorem	`PaperXI.CliffordMultiplicity.sectionOperator_one`
128	theorem	`PaperXI.CliffordMultiplicity.intertwiner_apply`
134	def	`PaperXI.CliffordMultiplicity.productIntertwiner`
145	theorem	`PaperXI.CliffordMultiplicity.productIntertwiner_apply`
152	def	`PaperXI.CliffordMultiplicity.actionEquiv`
165	theorem	`PaperXI.CliffordMultiplicity.actionEquiv_apply`
169	def	`PaperXI.CliffordMultiplicity.subgroupIntertwiner`
185	theorem	`PaperXI.CliffordMultiplicity.subgroupIntertwiner_apply`
191	def	`PaperXI.CliffordMultiplicity.defectIntertwiner`
210	theorem	`PaperXI.CliffordMultiplicity.defectIntertwiner_apply`
215	theorem	`PaperXI.CliffordMultiplicity.operator_product`
227	theorem	`PaperXI.CliffordMultiplicity.operator_subgroup_mul`

A.95 `PaperXI.CliffordPacket`

Selected layer: v0.44 parent. **Source lines:** 68.

SHA-256: `620bdf1b1eaca5bc5f86831a95fd342e72fe43e7fb697fd0590d56a03cc92533`.

```
Source 9.6: the actual Clifford action algebra map is an isomorphism,
and the actual multiplicity dimension is the square root of the quotient order.
```

Direct project imports: `PaperXI.CliffordSource`, `PaperXI.TwistedActionEquivalence`.

Line	Kind	Name
20	def	`PaperXI.CliffordPacket.action`
27	theorem	`PaperXI.CliffordPacket.action_basis`
38	theorem	`PaperXI.CliffordPacket.action_bijective`
44	def	`PaperXI.CliffordPacket.actionEquiv`
49	theorem	`PaperXI.CliffordPacket.actionEquiv_apply`
55	theorem	`PaperXI.CliffordPacket.finrank_sq`
65	theorem	`PaperXI.CliffordPacket.finrank_sqrt`

A.96 `PaperXI.CliffordSource`

Selected layer: v0.44 parent. **Source lines:** 48.

SHA-256: `f0b150418de3d56a1fcd7799169a9937e6b9746b2dc6a75c79de625bde828c88`.

```
The normalized cocycle and literal projective action for the source
Clifford datum, with all auxiliary dimensions and nontriviality derived.
```

Direct project imports: `PaperXI.CliffordIrreducible`, `PaperXI.CliffordCocycle`.

Line	Kind	Name
20	def	`PaperXI.CliffordSource.cocycle`
26	theorem	`PaperXI.CliffordSource.normalized`
32	theorem	`PaperXI.CliffordSource.operator_mul`
42	theorem	`PaperXI.CliffordSource.projective_irreducible`

A.97 `PaperXI.CliffordStabilizer`

Selected layer: v0.44 parent. **Source lines:** 216.

SHA-256: `6bc6cae8489f5025d88e5dfe8375db8a056c899365557a772e5188998ff689fc`.

```
The honest action induced on a normal quotient by ambient group
automorphisms. No coherent choices of representation intertwiners enter.
```

Direct project imports: `PaperXI.CliffordDatum`, `PaperXI.CliffordSource`, `PaperXI.CliffordEnvelope`, `PaperXI.CliffordTransport`, `PaperXI.ElementaryEnvelopeTransport`.

Line	Kind	Name
20	def	`PaperXI.CliffordStabilizer.quotientAction`
36	theorem	`PaperXI.CliffordStabilizer.quotientAction_mk`
41	theorem	`PaperXI.CliffordStabilizer.quotientAction_comp`
47	theorem	`PaperXI.CliffordStabilizer.quotientAction_unique`
67	def	`PaperXI.CliffordStabilizer.envelopeAction`
70	theorem	`PaperXI.CliffordStabilizer.envelopeAction_mk`
76	theorem	`PaperXI.CliffordStabilizer.envelopeAction_form`
89	theorem	`PaperXI.CliffordStabilizer.envelopeAction_equivariant`
137	theorem	`PaperXI.CliffordStabilizer.source_commutator_preserved`
151	def	`PaperXI.CliffordStabilizer.sourceAction`
157	theorem	`PaperXI.CliffordStabilizer.sourceAction_mk`
163	theorem	`PaperXI.CliffordStabilizer.sourceAction_form`
174	theorem	`PaperXI.CliffordStabilizer.sourceAction_independent`
200	theorem	`PaperXI.CliffordStabilizer.sourceAction_equivariant`

A.98 `PaperXI.CliffordTransport`

Selected layer: v0.44 parent. **Source lines:** 280.

SHA-256: `3b10aa513a5ba4f75eee3ca0e288cb4683b0a0044f3c22cf1bfaf031a568a995`.

```
Actual transport of subgroup intertwiners and multiplicity operators
along isomorphisms of the original representation data.
```

Direct project imports: `PaperXI.CliffordCocycle`.

Line	Kind	Name
15	def	PaperXI.CliffordTransport.subgroupEquiv
26	theorem	PaperXI.CliffordTransport.subgroupEquiv_coe
30	theorem	PaperXI.CliffordTransport.subgroupEquiv_conjugation
39	def	PaperXI.CliffordTransport.quotientEquiv
43	theorem	PaperXI.CliffordTransport.quotientEquiv_mk
50	theorem	PaperXI.CliffordTransport.quotientEquiv_refl
59	theorem	PaperXI.CliffordTransport.quotientEquiv_comp
76	theorem	PaperXI.CliffordTransport.source_apply
80	theorem	PaperXI.CliffordTransport.source_inverse_apply
88	theorem	PaperXI.CliffordTransport.target_apply
91	theorem	PaperXI.CliffordTransport.target_inverse_apply
99	def	PaperXI.CliffordTransport.homMap
113	theorem	PaperXI.CliffordTransport.homMap_apply
117	def	PaperXI.CliffordTransport.inverseHomMap
130	def	PaperXI.CliffordTransport.homEquiv
142	theorem	PaperXI.CliffordTransport.homEquiv_apply
145	theorem	PaperXI.CliffordTransport.homEquiv_symm_apply
151	def	PaperXI.CliffordTransport.transportedIntertwiner
162	theorem	PaperXI.CliffordTransport.transportedIntertwiner_apply
167	theorem	PaperXI.CliffordTransport.operator_transport
178	theorem	PaperXI.CliffordTransport.operator_transport_conj
189	theorem	PaperXI.CliffordTransport.transported_representative
198	def	PaperXI.CliffordTransport.cochain
206	theorem	PaperXI.CliffordTransport.sectionOperator_transport
230	theorem	PaperXI.CliffordTransport.cochain_one
243	theorem	PaperXI.CliffordTransport.cocycle_transport
265	theorem	PaperXI.CliffordTransport.commutator_preserved

A.99 PaperXI.CocycleBasis

Selected layer: v0.44 parent. **Source lines:** 33.

SHA-256: 5ba62f3c665922695e62725e35c969a9b223859a7dee0cad1163de238e51213e.

```
Paper XI, Proposition 8.1: the coefficient change in an actual algebra basis.
```

Direct project imports: PaperXI.CocycleCommutator.

Line	Kind	Name
15	theorem	PaperXI.CocycleCommutator.rescaled_basis_mul
29	theorem	PaperXI.CocycleCommutator.rescaled_basis_commutator

A.100 PaperXI.CocycleClass

Selected layer: v0.44 parent. **Source lines:** 168.

SHA-256: 393677b19c265df8ad023c34ede0aa8a8cb38f60e1bb050ef0b4282876a302bd.

```
The explicit quotient of normalized cocycles by normalized scalar coboundaries.
This is a concrete cohomological-class model; no identification with a separate
Mathlib group-cohomology construction is assumed.
```

Direct project imports: `PaperXI.CocycleCommutator`.

Line	Kind	Name
14	theorem	`PaperXI.CocycleClass.cocycle_ext`
21	theorem	`PaperXI.CocycleClass.rescale_one`
27	theorem	`PaperXI.CocycleClass.rescale_trans`
34	abbrev	`PaperXI.CocycleClass.NormalizedCocycle`
38	def	`PaperXI.CocycleClass.Related`
41	theorem	`PaperXI.CocycleClass.related_refl`
44	theorem	`PaperXI.CocycleClass.related_symm`
50	theorem	`PaperXI.CocycleClass.related_trans`
58	def	`PaperXI.CocycleClass.coboundarySetoid`
64	def	`PaperXI.CocycleClass.Class`
67	def	`PaperXI.CocycleClass.of`
71	theorem	`PaperXI.CocycleClass.of_eq_iff`
76	theorem	`PaperXI.CocycleClass.of_eq_of_rescale`
80	theorem	`PaperXI.CocycleClass.of_rescale`
86	def	`PaperXI.CocycleClass.commutator`
95	theorem	`PaperXI.CocycleClass.commutator_of`
99	def	`PaperXI.CocycleClass.radical`
107	theorem	`PaperXI.CocycleClass.radical_of`
111	def	`PaperXI.CocycleClass.pullbackCocycle`
115	theorem	`PaperXI.CocycleClass.normalized_pullback`
119	theorem	`PaperXI.CocycleClass.pullback_rescale`
126	def	`PaperXI.CocycleClass.pullback`
136	theorem	`PaperXI.CocycleClass.pullback_of`
140	def	`PaperXI.CocycleClass.congrEquiv`
164	theorem	`PaperXI.CocycleClass.congrEquiv_of`

A.101 `PaperXI.CocycleCommutator`

Selected layer: v0.44 parent. **Source lines:** 122.

SHA-256: `0b8b7f046bf6d69122f05e1838481020360574431927 62a7e89fa4e374f6d4df`.

```
Paper XI, Propositions 8.1–8.2. Multiplicative notation for the abelian
index group. Values may lie in any commutative group, including ℂˣ.
```

Direct project imports: none.

Line	Kind	Name
14	structure	`PaperXI.CocycleCommutator.Cocycle`
18	def	`PaperXI.CocycleCommutator.Normalized`
22	def	`PaperXI.CocycleCommutator.rescale`
35	def	`PaperXI.CocycleCommutator.commutator`
37	theorem	`PaperXI.CocycleCommutator.commutator_self`
40	theorem	`PaperXI.CocycleCommutator.commutator_swap`
43	theorem	`PaperXI.CocycleCommutator.commutator_rescale`
48	theorem	`PaperXI.CocycleCommutator.commutator_mul_left`
68	theorem	`PaperXI.CocycleCommutator.commutator_mul_right`
73	theorem	`PaperXI.CocycleCommutator.commutator_one_left`

Line	Kind	Name
78	theorem	`PaperXI.CocycleCommutator.commutator_one_right`
82	def	`PaperXI.CocycleCommutator.characterMap`
90	theorem	`PaperXI.CocycleCommutator.characterMap_apply`
94	def	`PaperXI.CocycleCommutator.radical`
96	theorem	`PaperXI.CocycleCommutator.mem_radical_iff`
103	theorem	`PaperXI.CocycleCommutator.radical_rescale`
110	theorem	`PaperXI.CocycleCommutator.normalized_rescale`
114	theorem	`PaperXI.CocycleCommutator.normalized_rescale_iff`

A.102 `PaperXI.CocycleQuotient`

Selected layer: v0.44 parent. **Source lines:** 85.

SHA-256: `e4fd2024e7a2b19c99fe508d778300fc161a53761c5ed2db8f0e5c0091101b51`.

```
Paper XI, Proposition 8.2: the actual quotient by the commutator radical.
```

Direct project imports: `PaperXI.CocycleCommutator`.

Line	Kind	Name
11	def	`PaperXI.CocycleCommutator.quotientCharacter`
14	theorem	`PaperXI.CocycleCommutator.quotientCharacter_mk`
17	theorem	`PaperXI.CocycleCommutator.radical_le_quotientCharacter_ker`
25	def	`PaperXI.CocycleCommutator.quotientPairing`
43	theorem	`PaperXI.CocycleCommutator.quotientPairing_mk`
47	theorem	`PaperXI.CocycleCommutator.quotientPairing_self`
52	theorem	`PaperXI.CocycleCommutator.quotientPairing_swap`
60	theorem	`PaperXI.CocycleCommutator.quotientPairing_nondegenerate_left`
74	theorem	`PaperXI.CocycleCommutator.quotientPairing_nondegenerate_right`
82	theorem	`PaperXI.CocycleCommutator.quotientCharacter_injective`

A.103 `PaperXI.CommutatorEnvelope`

Selected layer: v0.44 parent. **Source lines:** 160.

SHA-256: `bc6e6fc84b1c63fc1bb70568bc6eb1c3137c5f9f781783990c9c1bc51a6453b8`.

```
Exact universal properties of the quotient detected by the scalar
commutator. A pairing descends through A/L precisely when L lies in its
radical; that descended pairing is nondegenerate precisely at the radical.
```

Direct project imports: `PaperXI.CocycleQuotient`.

Line	Kind	Name
14	def	`PaperXI.CommutatorEnvelope.descendedCharacter`
17	theorem	`PaperXI.CommutatorEnvelope.descendedCharacter_mk`
21	theorem	`PaperXI.CommutatorEnvelope.le_descendedCharacter_ker`
29	def	`PaperXI.CommutatorEnvelope.descendedPairing`
48	theorem	`PaperXI.CommutatorEnvelope.descendedPairing_mk`
54	theorem	`PaperXI.CommutatorEnvelope.descends_iff`

Line	Kind	Name
68	theorem	`PaperXI.CommutatorEnvelope.descendedPairing_unique`
81	theorem	`PaperXI.CommutatorEnvelope.existsUnique_descendedPairing`
89	theorem	`PaperXI.CommutatorEnvelope.descendedPairing_self`
95	theorem	`PaperXI.CommutatorEnvelope.nondegenerate_iff`
116	def	`PaperXI.CommutatorEnvelope.quotientFactor`
120	theorem	`PaperXI.CommutatorEnvelope.quotientFactor_mk`
125	theorem	`PaperXI.CommutatorEnvelope.quotientFactor_surjective`
132	theorem	`PaperXI.CommutatorEnvelope.quotientFactor_pairing`
140	theorem	`PaperXI.CommutatorEnvelope.existsUnique_quotientFactor`
151	theorem	`PaperXI.CommutatorEnvelope.existsUnique_injective_characterFactor`

A.104 `PaperXI.CompactDegreeSeparation`

Selected layer: base108 arithmetic branch. **Source lines:** 37.

SHA-256: `c69f0838adb9db017a926ff1a66606634858bfb0aa89710e147b19b44a60d561`.

```
# Compact subgroups have zero integer degree

A continuous integer degree has finite image on a compact subgroup, because
the target has its usual discrete topology.  The image is a finite subgroup of
the torsion-free group `Multiplicative ℤ`, so it is trivial.  This supplies the
separation hypothesis for adjoining any element of nonzero degree.
```

Direct project imports: `PaperXI.CentralExtensionInducing`.

Line	Kind	Name
20	theorem	`PaperXI.CompactDegreeSeparation.compact_le_degree_ker`
32	theorem	`PaperXI.CompactDegreeSeparation.hfree_from_compact_degree`

A.105 `PaperXI.CompactIntersection108`

Selected layer: base108 arithmetic branch. **Source lines:** 55.

SHA-256: `318e23f5dfa82f3cf42743e82c7783ac44508ce10be1a856aa6fc60407b2fa37`.

```
The finite model of the actual conjugate-intersection Hom space exists for
any continuous embedded compact group with a smooth finite-dimensional action.
```

Direct project imports: `PaperXI.IntersectionModel108`, `PaperXI.FiniteJointQuotient108`.

Line	Kind	Name
17	theorem	`PaperXI.CompactIntersection108.compact_pairs`
26	theorem	`PaperXI.CompactIntersection108.smooth_comp`
35	theorem	`PaperXI.CompactIntersection108.finite_quotient`
46	theorem	`PaperXI.CompactIntersection108.finrank_quotient`

A.106 `PaperXI.CompactModCenterInduction68`

Selected layer: base108 arithmetic branch. **Source lines:** 126.

SHA-256: `b1765821b4681b903bdbaabe6a76b05200fba99f81d94cc260e9ecc3484a4065`.

```
# Full-subgroup Mackey irreducibility for compact-mod-central induction

The ambient induction and its coefficient maps are the actual Mathlib carriers.
No compact-restriction intertwining criterion is substituted for the given
full-subgroup criterion. Smoothness, scalar cancellation and compact quotient
supply the needed equivariant section by a finite orbit average.
```

Direct project imports: `PaperXI.EquivariantSplitting68`, `PaperXI.InducedTypeSpace68`.

Line	Kind	Name
25	theorem	`PaperXI.CompactModCenterInduction68.induced_central_scalar`
40	def	`PaperXI.CompactModCenterInduction68.subInclusion`
52	theorem	`PaperXI.CompactModCenterInduction68.irreducible_of_compact_mod_center_mackey`
111	theorem	`PaperXI.CompactModCenterInduction68.induction_package`

A.107 `PaperXI.CompactQuotientPeriod69`

Selected layer: base108 arithmetic branch. **Source lines:** 52.

SHA-256: `db609bc27e26e3964e4eaa9f6739b066d277b23dd9c0b595f4b51fc80baa975a`.

```
Compact fundamental domains from a finite period modulo a compact subgroup.
```

Direct project imports: none.

Line	Kind	Name
15	theorem	`PaperXI.CompactQuotientPeriod69.compact_quotient_of_period`

A.108 `PaperXI.CompactSmoothSemisimple`

Selected layer: base108 arithmetic branch. **Source lines:** 92.

SHA-256: `8f8329834876b5913b22197a22cc58d386a26429cb21fcb23d51947c33d93e74`.

```
Smooth representations of compact groups are locally finite dimensional.
This uses compactness of the actual group, not finiteness of the ambient group.
```

Direct project imports: `PaperXI.SmoothDegreeQuotient`, `PaperXI.LocallyFiniteSemisimple`.

Line	Kind	Name
19	def	`PaperXI.CompactSmoothSemisimple.orbitQuotient`
27	def	`PaperXI.CompactSmoothSemisimple.orbitSpan`
40	theorem	`PaperXI.CompactSmoothSemisimple.mem_orbitSpan`
48	theorem	`PaperXI.CompactSmoothSemisimple.subrepresentation_smooth`
61	theorem	`PaperXI.CompactSmoothSemisimple.orbit_finite`
70	theorem	`PaperXI.CompactSmoothSemisimple.orbitSpan_finiteDimensional`
75	theorem	`PaperXI.CompactSmoothSemisimple.subrepresentation_kernel_quotient_finite`
83	theorem	`PaperXI.CompactSmoothSemisimple.smooth_semisimple`

A.109 `PaperXI.CompactTypeTensor58`

Selected layer: base108 arithmetic branch. **Source lines:** 93.

SHA-256: `885a6eef6b9a66f86ddf833e1077fdd3de5fe94a004487d6b539d70cba73bc6b`.

```
# The prescribed compact factor with an actual finite quotient factor

An irreducible quotient representation is tensored with the supplied extension
of an irreducible subgroup action. The resulting compact inducing action is
constructed on the literal tensor space, is irreducible and smooth, and retains
the supplied subgroup action on the first factor exactly. No type-theoretic
compact-induction theorem or arithmetic comparison is asserted.
```

Direct project imports: `PaperXI.GallagherTensorIrreducibility58`, `PaperXI.SmoothDegreeQuotient`, `PaperXI.FiniteRepresentation`.

Line	Kind	Name
27	abbrev	`PaperXI.CompactTypeTensor58.action`
32	theorem	`PaperXI.CompactTypeTensor58.restriction`
39	theorem	`PaperXI.CompactTypeTensor58.subgroup_action`
48	theorem	`PaperXI.CompactTypeTensor58.irreducible`
55	theorem	`PaperXI.CompactTypeTensor58.finiteDimensional`
66	theorem	`PaperXI.CompactTypeTensor58.smooth`
82	theorem	`PaperXI.CompactTypeTensor58.completed`

A.110 `PaperXI.CompactTypeTensorNormalizer58`

Selected layer: base108 arithmetic branch. **Source lines:** 54.

SHA-256: `1058ac054e5f073f587c3012d307d04d2924924594cd5ec137685c0bfb3e10b5`.

```
# Lifting a prescribed normalizer intertwiner through the quotient tensor factor

When an actual normalizer automorphism acts trivially on the quotient, its
invertible intertwiner on the prescribed first factor supplies an invertible
intertwiner on the literal compact tensor action. The latter covariance is a
conclusion, not a hypothesis.
```

Direct project imports: `PaperXI.CompactTypeTensor58`.

Line	Kind	Name
26	def	`PaperXI.CompactTypeTensorNormalizer58.tensorOperator`
29	theorem	`PaperXI.CompactTypeTensorNormalizer58.tensorOperator_tmul`
34	theorem	`PaperXI.CompactTypeTensorNormalizer58.tensorOperator_covariance`
43	def	`PaperXI.CompactTypeTensorNormalizer58.normalizerEquiv`
51	theorem	`PaperXI.CompactTypeTensorNormalizer58.normalizerEquiv_linear`

A.111 `PaperXI.ConditionalAtomicComparison`

Selected layer: v0.44 parent. **Source lines:** 151.

SHA-256: `ce8d1ea0f8658d4d968aeb6665d5f1d3e4cc6bb1d472ebfbfbce6215f1103a58`.

```
The last comparison pipeline. Actual finite tensor intertwiners and global
parallel-section equivalences are constructed. Atom realization and normalized
pair-factor characterization remain explicit external interfaces. They are not
consequences of finite phases or of zero holonomy.
```

Direct project imports: `PaperXI.FiniteBreakAssembly`, `PaperXI.GluedPacketSections`, `PaperXI.LocalFactorIdentification`.

Line	Kind	Name
19	structure	`PaperXI.ConditionalAtomicComparison.Diagram`
27	structure	`PaperXI.ConditionalAtomicComparison.Atlas`
42	def	`PaperXI.ConditionalAtomicComparison.targetDiagram`
45	def	`PaperXI.ConditionalAtomicComparison.sourceDiagram`
49	def	`PaperXI.ConditionalAtomicComparison.atRoot`
56	theorem	`PaperXI.ConditionalAtomicComparison.atRoot_linear`
60	def	`PaperXI.ConditionalAtomicComparison.Holonomy`
65	def	`PaperXI.ConditionalAtomicComparison.leftSections`
67	def	`PaperXI.ConditionalAtomicComparison.rightSections`
70	def	`PaperXI.ConditionalAtomicComparison.glue`
75	theorem	`PaperXI.ConditionalAtomicComparison.glue_apply`
84	theorem	`PaperXI.ConditionalAtomicComparison.holonomy_iff_gauge`
119	def	`PaperXI.ConditionalAtomicComparison.finiteComparison`
127	theorem	`PaperXI.ConditionalAtomicComparison.source`

A.112 `PaperXI.ConstructedFieldFrobenius`

Selected layer: normalized Frobenius. **Source lines:** 65.

SHA-256: `37abb8f28a5e22530d0f0296eaef1880e5fca49629014ef7cd762d606916b5bc`.

```
# Specialization to Paper XI's constructed spectral local field

The base of the finite Galois extension is the literal UpperField(K,s) from the
verified predecessor. The base residue cardinality in geometric normalization
is identified with the residue cardinality of that same field, by equality of
integer subrings and the induced residue equivalence.
```

Direct project imports: `PaperXI.SpectralGeometricFrobenius`, `PaperXI.SpectralLocalField`.

Line	Kind	Name
22	abbrev	`PaperXI.ConstructedFieldFrobenius.BaseField`
28	theorem	`PaperXI.ConstructedFieldFrobenius.restricted_integer_subring`
36	def	`PaperXI.ConstructedFieldFrobenius.baseIntegerEquiv`
43	def	`PaperXI.ConstructedFieldFrobenius.baseResidueEquiv`
46	theorem	`PaperXI.ConstructedFieldFrobenius.baseResidue_card`
54	theorem	`PaperXI.ConstructedFieldFrobenius.exists_normalized_field_automorphism`

A.113 `PaperXI.ContinuousHolonomyResidual`

Selected layer: v0.44 parent. **Source lines:** 132.

SHA-256: `ef0ef5f7cdf0b952317f838c65b6f9e0229b32e424b9f384ab96634cd955f060`.

```
Continuous forms of the exact residual identity and its conditioning bound.
The maps are actual continuous linear equivalences. No isometry is assumed.
```

Direct project imports: `PaperXI.HolonomyResidual`.

Line	Kind	Name
18	def	`PaperXI.ContinuousHolonomyResidual.transported`

Line	Kind	Name
21	theorem	`PaperXI.ContinuousHolonomyResidual.transported_apply`
24	theorem	`PaperXI.ContinuousHolonomyResidual.transported_toLinearEquiv`
28	def	`PaperXI.ContinuousHolonomyResidual.cycleDefect`
37	def	`PaperXI.ContinuousHolonomyResidual.edgeResidual`
44	theorem	`PaperXI.ContinuousHolonomyResidual.cycleDefect_toLinearMap`
52	theorem	`PaperXI.ContinuousHolonomyResidual.edgeResidual_toLinearMap`
60	theorem	`PaperXI.ContinuousHolonomyResidual.residual_factorization`
72	theorem	`PaperXI.ContinuousHolonomyResidual.residual_zero_iff`
86	theorem	`PaperXI.ContinuousHolonomyResidual.residual_norm_bound`
103	theorem	`PaperXI.ContinuousHolonomyResidual.uniform_bound`
119	theorem	`PaperXI.ContinuousHolonomyResidual.product_conditioning_bound`

A.114 `PaperXI.CoordinateDeepIdeal111`

Selected layer: Continuation111. **Source lines:** 79.

SHA-256: `00a4e4aef33f78b2682431bc4370b2158cb1628f7d1c702f634861314e19cb09`.

```
A common open ideal in an actual coordinate integral matrix order.
The ideal lives in the integral order, not the full field matrix algebra.
```

Direct project imports: `PaperXI.OnePlusIntersection110`.

Line	Kind	Name
21	def	`PaperXI.CoordinateDeepIdeal111.uncurryHomeomorph`
31	theorem	`PaperXI.CoordinateDeepIdeal111.exists_open_matrix_ideal_subset`
45	theorem	`PaperXI.CoordinateDeepIdeal111.matrix_ideal_open`
64	theorem	`PaperXI.CoordinateDeepIdeal111.exists_common_ideal_finite`

A.115 `PaperXI.CuspidalAnnihilation`

Selected layer: v0.44 parent. **Source lines:** 87.

SHA-256: `e56bdeb22b0d449550cd009852c029feed4d614bdc02a53d49648784b4a6abd2`.

```
Subgroup averages and annihilation of an actual central character block.
The cuspidal input is the vanishing of genuine restricted invariants.
```

Direct project imports: `PaperXI.PrimitiveCharacterBlock`, `PaperXI.FiniteAverage`.

Line	Kind	Name
16	def	`PaperXI.CuspidalAnnihilation.subgroupAverage`
20	theorem	`PaperXI.CuspidalAnnihilation.average_action`
25	theorem	`PaperXI.CuspidalAnnihilation.average_range`
31	theorem	`PaperXI.CuspidalAnnihilation.average_idempotent`
38	theorem	`PaperXI.CuspidalAnnihilation.average_zero_iff`
45	theorem	`PaperXI.CuspidalAnnihilation.own_block_action`
51	theorem	`PaperXI.CuspidalAnnihilation.left_annihilation_iff`
66	theorem	`PaperXI.CuspidalAnnihilation.right_annihilation_iff`
73	theorem	`PaperXI.CuspidalAnnihilation.parabolic_annihilation`

Line	Kind	Name
79	theorem	`PaperXI.CuspidalAnnihilation.nonzero_of_fixed_vector`

A.116 `PaperXI.CuspidalCartan105`

Selected layer: base108 arithmetic branch. **Source lines:** 82.

SHA-256: `b2194352b6debd4ae75e3489078283c8f99b6c759d153356b9853900799a8087`.

```
An inflated finite cuspidal representation has no noncentral Cartan
intertwiners. The arithmetic Cartan decomposition remains a separate explicit
input; its cuspidal exclusion is proved on actual matrices.
```

Direct project imports: `PaperXI.CuspidalContraction105`, `PaperXI.IntertwiningAlong105`, `PaperXI.SplitRadical105`.

Line	Kind	Name
19	def	`PaperXI.CuspidalCartan105.Cuspidal`
23	theorem	`PaperXI.CuspidalCartan105.diagonal_intertwiner_zero`
48	def	`PaperXI.CuspidalCartan105.HasCartan`
56	theorem	`PaperXI.CuspidalCartan105.intertwining_support`

A.117 `PaperXI.CuspidalContraction105`

Selected layer: base108 arithmetic branch. **Source lines:** 37.

SHA-256: `cf19708af1422fb89f6f41404483d37fb54258b7a6b1b18428ac52a31995dcbd`.

```
Finite cuspidality excludes a map on which an actual radical contracts to
identity. The averaging sum and its vanishing are derived, not supplied.
```

Direct project imports: none.

Line	Kind	Name
14	def	`PaperXI.CuspidalContraction105.NoFixed`
17	theorem	`PaperXI.CuspidalContraction105.sum_fixed`
23	theorem	`PaperXI.CuspidalContraction105.sum_eq_zero`
26	theorem	`PaperXI.CuspidalContraction105.map_eq_zero`

A.118 `PaperXI.CycleCoordinates`

Selected layer: v0.44 parent. **Source lines:** 97.

SHA-256: `0e63d65fbd8eecf42832697e5cbb2b1c5e6c7fda9b165030352c34bb3ec19004`.

```
Transfer formulas in an actual basis, including independent rescalings.
```

Direct project imports: `PaperXI.CyclicPolynomial`.

Line	Kind	Name
13	theorem	`PaperXI.CycleCoordinates.basis_eq_powers`
28	theorem	`PaperXI.CycleCoordinates.closing_of_last_step`
45	theorem	`PaperXI.CycleCoordinates.cycle_formula`
55	theorem	`PaperXI.CycleCoordinates.rescaled_entry`
62	theorem	`PaperXI.CycleCoordinates.rescaled_charpoly`
68	theorem	`PaperXI.CycleCoordinates.rescaled_det`
74	theorem	`PaperXI.CycleCoordinates.cycle_orientation`
77	theorem	`PaperXI.CycleCoordinates.cycle_local_polynomial`
87	theorem	`PaperXI.CycleCoordinates.rescaled_cycle_formula`

A.119 `PaperXI.CycleFixed`

Selected layer: v0.44 parent. **Source lines:** 138.

SHA-256: `008caab8c32493eb96d7225cdc6ff9c2b44718f2cbf85360a0ebef91be48e3dd`.

```
Fixed vectors of an actual cyclic basis, valid in every characteristic.
```

Direct project imports: `PaperXI.CycleCoordinates`.

Line	Kind	Name
11	def	`PaperXI.CycleFixed.fixedSpace`
13	theorem	`PaperXI.CycleFixed.mem_fixedSpace`
16	theorem	`PaperXI.CycleFixed.coordinate_step`
42	theorem	`PaperXI.CycleFixed.fixed_coordinates_constant`
59	theorem	`PaperXI.CycleFixed.basis_sum_ne_zero`
65	theorem	`PaperXI.CycleFixed.fixed_eq_smul_sum`
75	theorem	`PaperXI.CycleFixed.map_basis_sum`
90	theorem	`PaperXI.CycleFixed.fixedSpace_eq`
126	theorem	`PaperXI.CycleFixed.fixedSpace_finrank`

A.120 `PaperXI.CycleSpan`

Selected layer: v0.44 parent. **Source lines:** 76.

SHA-256: `a96cff77ae41d6de99c41763e94a925f42820c3e7646ee438e6fe3bbb67f8054`.

```
The span of a finite Frobenius cycle with a nonzero closing scalar.
```

Direct project imports: `PaperXI.PacketReturn`.

Line	Kind	Name
10	def	`PaperXI.CycleSpan.cycleVector`
12	def	`PaperXI.CycleSpan.cycleSpan`
15	theorem	`PaperXI.CycleSpan.power_mem`
18	theorem	`PaperXI.CycleSpan.base_mem`
21	theorem	`PaperXI.CycleSpan.map_mem`
30	theorem	`PaperXI.CycleSpan.forward_power`
33	theorem	`PaperXI.CycleSpan.backward_power`

Line	Kind	Name
37	theorem	`PaperXI.CycleSpan.inverse_wrap`
43	theorem	`PaperXI.CycleSpan.forward_stable`
55	theorem	`PaperXI.CycleSpan.backward_stable`
69	theorem	`PaperXI.CycleSpan.span_ne_bot`

A.121 `PaperXI.CyclicAction`

Selected layer: v0.44 parent. **Source lines:** 55.

SHA-256: `581f94955c014edf5c55919aa1a8a85b4fc7ec5ec07b9110e81dea08ad0648cf`.

```
Actual representations of the additive finite cyclic group.
```

Direct project imports: `PaperXI.FourierCovariance`.

Line	Kind	Name
12	abbrev	`PaperXI.CyclicAction.Action`
14	def	`PaperXI.CyclicAction.operator`
17	theorem	`PaperXI.CyclicAction.operator_zero`
20	theorem	`PaperXI.CyclicAction.operator_add`
23	theorem	`PaperXI.CyclicAction.operator_nsmul`
31	theorem	`PaperXI.CyclicAction.operator_mul`
36	theorem	`PaperXI.CyclicAction.operator_eq_generator_pow`
40	theorem	`PaperXI.CyclicAction.generator_period`
46	theorem	`PaperXI.CyclicAction.conjugation_from_generator`

A.122 `PaperXI.CyclicBridge`

Selected layer: v0.44 parent. **Source lines:** 111.

SHA-256: `c24882a79bb6079e3ec2c68c8abfdd1db6c00634d6dd9045b2be41aa67cf601b`.

```
A trivial scalar discrepancy on one cyclic generator closes an actual
semidirect-product bridge. Phase means the scalar discrepancy, not trace.
```

Direct project imports: `PaperXI.DiscrepancyCharacter`.

Line	Kind	Name
12	theorem	`PaperXI.CyclicBridge.character_trivial`
18	theorem	`PaperXI.CyclicBridge.semidirect_intertwining`
28	theorem	`PaperXI.CyclicBridge.stabilizer_covariance`
41	theorem	`PaperXI.CyclicBridge.one_phase_bridge`
63	def	`PaperXI.CyclicBridge.semidirectRepresentation`
74	theorem	`PaperXI.CyclicBridge.semidirectRepresentation_inl`
80	theorem	`PaperXI.CyclicBridge.semidirectRepresentation_inr`
88	theorem	`PaperXI.CyclicBridge.one_phase_bridge_from_actions`

A.123 `PaperXI.CyclicDefect113`

Selected layer: Continuation113. **Source lines:** 339.

SHA-256: `b6fe373c44ebd8725be1683e7f851da03bcc708c896db2103e02a5a31fbb3a19`.

```
Explicit coprime correction of the character defect left by scalar
cocycle comparison. Arithmetic realization is not asserted here.
```

Direct project imports: `PaperXI.ScalarCocycle112`.

Line	Kind	Name
17	def	`PaperXI.CyclicDefect113.orbitNorm`
21	def	`PaperXI.CyclicDefect113.weightedCharacter`
24	theorem	`PaperXI.CyclicDefect113.orbitNorm_zero`
27	theorem	`PaperXI.CyclicDefect113.orbitNorm_succ`
31	theorem	`PaperXI.CyclicDefect113.weightedCharacter_zero`
34	theorem	`PaperXI.CyclicDefect113.weightedCharacter_succ`
39	theorem	`PaperXI.CyclicDefect113.power_apply_shift`
45	theorem	`PaperXI.CyclicDefect113.weightedCharacter_shift`
59	theorem	`PaperXI.CyclicDefect113.bezout_power`
71	def	`PaperXI.CyclicDefect113.correctingCharacter`
76	theorem	`PaperXI.CyclicDefect113.correctingCharacter_coboundary`
92	structure	`PaperXI.CyclicDefect113.ScalarLift`
102	def	`PaperXI.CyclicDefect113.ScalarLift.phase`
104	theorem	`PaperXI.CyclicDefect113.ScalarLift.phase_one`
108	theorem	`PaperXI.CyclicDefect113.ScalarLift.scalar_formula`
121	theorem	`PaperXI.CyclicDefect113.ScalarLift.phase_mul`
131	theorem	`PaperXI.CyclicDefect113.ScalarLift.power_base`
141	theorem	`PaperXI.CyclicDefect113.ScalarLift.power_scalar`
152	theorem	`PaperXI.CyclicDefect113.ScalarLift.norm_eq_one`
162	theorem	`PaperXI.CyclicDefect113.orbitNorm_coboundary`
172	theorem	`PaperXI.CyclicDefect113.orbitNorm_mul`
176	theorem	`PaperXI.CyclicDefect113.orbitNorm_div`
187	def	`PaperXI.CyclicDefect113.actionDefect`
190	theorem	`PaperXI.CyclicDefect113.actionDefect_one`
195	theorem	`PaperXI.CyclicDefect113.actionDefect_mul`
205	def	`PaperXI.CyclicDefect113.defectCharacter`
211	theorem	`PaperXI.CyclicDefect113.actionDefect_norm`
223	def	`PaperXI.CyclicDefect113.correctedCochain`
227	theorem	`PaperXI.CyclicDefect113.correctedCochain_one`
232	theorem	`PaperXI.CyclicDefect113.correctedCochain_cocycle`
243	theorem	`PaperXI.CyclicDefect113.correctedCochain_covariant`
264	def	`PaperXI.CyclicDefect113.scalarBridge`
279	theorem	`PaperXI.CyclicDefect113.scalarBridge_base`
284	theorem	`PaperXI.CyclicDefect113.scalarBridge_fixes_scalar`
290	theorem	`PaperXI.CyclicDefect113.scalarBridge_equivariant`
308	theorem	`PaperXI.CyclicDefect113.exists_equivariant_cochain`
325	theorem	`PaperXI.CyclicDefect113.exists_equivariant_scalarBridge`

A.124 `PaperXI.CyclicFrobeniusCount`

Selected layer: v0.44 parent. **Source lines:** 120.

SHA-256: `b97169de1bf48a86a58bf00236bad78341081a6db9ae46f6a8e49f95c87a4f82`.

```
Fixed points and regular orbit counts for the power map on a finite cyclic
 group. No fixed-point cardinality is assumed.
```

Direct project imports: `PaperXI.FrobeniusPeriodCount`.

Line	Kind	Name
16	theorem	`PaperXI.CyclicFrobeniusCount.iterate_power`
24	theorem	`PaperXI.CyclicFrobeniusCount.fixed_iff_kernel`
40	theorem	`PaperXI.CyclicFrobeniusCount.fixedCount_gcd`
48	theorem	`PaperXI.CyclicFrobeniusCount.fixedCount_pow_gcd`
54	theorem	`PaperXI.CyclicFrobeniusCount.fixedCount_of_dvd`
60	theorem	`PaperXI.CyclicFrobeniusCount.exactCount_formula`
72	theorem	`PaperXI.CyclicFrobeniusCount.card_coprime_q`
79	def	`PaperXI.CyclicFrobeniusCount.frobeniusPerm`
84	theorem	`PaperXI.CyclicFrobeniusCount.frobeniusPerm_apply`
89	theorem	`PaperXI.CyclicFrobeniusCount.orbitCount_mul_formula`
100	theorem	`PaperXI.CyclicFrobeniusCount.moebius_sum_dvd`
108	theorem	`PaperXI.CyclicFrobeniusCount.orbitCount_formula`

A.125 `PaperXI.CyclicInducedBasis`

Selected layer: v0.44 parent. **Source lines:** 83.

SHA-256: `a90ba613fe359b594d103fb19aa93529a4e4cb8584c38b335950bcb1ceaf0eb1`.

The Frobenius-indexed basis of an actual induction from a cyclic degree kernel.

Direct project imports: `PaperXI.CyclicInducedCoordinates`.

Line	Kind	Name
15	def	`PaperXI.CyclicInduced.cyclicRep`
19	theorem	`PaperXI.CyclicInduced.cyclicRep_degree`
25	theorem	`PaperXI.CyclicInduced.cyclicRep_one`
30	def	`PaperXI.CyclicInduced.cyclicBasis`
34	def	`PaperXI.CyclicInduced.cyclicWeight`
37	theorem	`PaperXI.CyclicInduced.cyclicBasis_apply`
43	theorem	`PaperXI.CyclicInduced.cyclicBasis_action`
51	def	`PaperXI.CyclicInduced.cyclicCharacter`
54	theorem	`PaperXI.CyclicInduced.cyclicCharacter_apply`
64	theorem	`PaperXI.CyclicInduced.cyclicBasis_action_kernel`
71	theorem	`PaperXI.CyclicInduced.cyclicBasis_action_zero`
78	theorem	`PaperXI.CyclicInduced.cyclicBasis_action_phi`

A.126 `PaperXI.CyclicInducedCharacter`

Selected layer: v0.44 parent. **Source lines:** 60.

SHA-256: `15707518268a187bd5d4f486516c842c236f5385de68af3fb7ef63eb83fcd4f5`.

Exact subgroup character values of the monomial coefficients of actual induction.

Direct project imports: `PaperXI.CyclicInducedDegree`.

Line	Kind	Name
13	theorem	`PaperXI.CyclicInduced.degreeWeight_kernel`
19	theorem	`PaperXI.CyclicInduced.degreeCharacter_apply`
28	theorem	`PaperXI.CyclicInduced.degreeCharacter_zero`
36	theorem	`PaperXI.CyclicInduced.degreeWeight_zero`
43	theorem	`PaperXI.CyclicInduced.degreeWeight_restrict`
54	theorem	`PaperXI.CyclicInduced.degreeWeight_restrict_zero`

A.127 `PaperXI.CyclicInducedCoordinates`

Selected layer: v0.44 parent. **Source lines:** 179.

SHA-256: `302c38129316e682653aa44bbff88450ec77341b005fbdbce392a31a475f2d90`.

```
Coordinates for genuine induction from the kernel of a finite quotient.
The representative indexed by `q` lies over `q⁻¹`, so the basis action is left translation.
```

Direct project imports: none.

Line	Kind	Name
17	def	`PaperXI.CyclicInduced.scalar`
22	theorem	`PaperXI.CyclicInduced.scalar_apply`
26	def	`PaperXI.CyclicInduced.residual`
29	theorem	`PaperXI.CyclicInduced.residual_rep`
33	theorem	`PaperXI.CyclicInduced.residual_mul`
40	abbrev	`PaperXI.CyclicInduced.CoindSpace`
41	abbrev	`PaperXI.CyclicInduced.coinduced`
42	abbrev	`PaperXI.CyclicInduced.IndSpace`
43	abbrev	`PaperXI.CyclicInduced.induced`
46	def	`PaperXI.CyclicInduced.extend`
52	theorem	`PaperXI.CyclicInduced.extend_rep`
57	def	`PaperXI.CyclicInduced.coordinates`
70	theorem	`PaperXI.CyclicInduced.coordinates_apply`
74	def	`PaperXI.CyclicInduced.weight`
80	def	`PaperXI.CyclicInduced.coindBasis`
83	theorem	`PaperXI.CyclicInduced.coindBasis_rep`
94	theorem	`PaperXI.CyclicInduced.coindBasis_action`
120	def	`PaperXI.CyclicInduced.indCoind`
125	def	`PaperXI.CyclicInduced.basis`
129	theorem	`PaperXI.CyclicInduced.basis_action`
142	def	`PaperXI.CyclicInduced.conjugate`
150	def	`PaperXI.CyclicInduced.conjugateCharacter`
154	theorem	`PaperXI.CyclicInduced.weight_kernel`
163	theorem	`PaperXI.CyclicInduced.basis_action_kernel`
170	theorem	`PaperXI.CyclicInduced.basis_action_origin`

A.128 `PaperXI.CyclicInducedDegree`

Selected layer: v0.44 parent. **Source lines:** 59.

SHA-256: `e60c559b41a83de4e58e4a01b0633216f888976f7748d01b1b05f7aa42aaf085`.

Source-facing degree-subgroup coordinates for tame cyclic induction.

Direct project imports: `PaperXI.CyclicInducedBasis`, `PaperXI.TameDegree`.

Line	Kind	Name
15	theorem	`PaperXI.CyclicInduced.degreeMod_phi`
20	abbrev	`PaperXI.CyclicInduced.degreeInduced`
21	abbrev	`PaperXI.CyclicInduced.DegreeIndSpace`
24	def	`PaperXI.CyclicInduced.degreeBasis`
27	def	`PaperXI.CyclicInduced.degreeWeight`
31	def	`PaperXI.CyclicInduced.degreeCharacter`
35	theorem	`PaperXI.CyclicInduced.degreeBasis_action`
42	theorem	`PaperXI.CyclicInduced.degreeBasis_action_kernel`
48	theorem	`PaperXI.CyclicInduced.degreeBasis_action_zero`
54	theorem	`PaperXI.CyclicInduced.degreeBasis_action_phi`

A.129 `PaperXI.CyclicInducedWrap`

Selected layer: v0.44 parent. **Source lines:** 119.

SHA-256: `4dec20ab681bfc9225f0bf157b406ac02ea6272612eb84376cb4b647af558bb9`.

Explicit wrap coefficients in the Frobenius-indexed induced basis.

Direct project imports: `PaperXI.CyclicInducedDegree`.

Line	Kind	Name
14	theorem	`PaperXI.CyclicInduced.cyclicWeight_phi_value`
38	def	`PaperXI.CyclicInduced.period`
44	theorem	`PaperXI.CyclicInduced.val_succ_of_lt`
50	theorem	`PaperXI.CyclicInduced.val_succ_of_last`
57	theorem	`PaperXI.CyclicInduced.cyclicWeight_phi_of_lt`
69	theorem	`PaperXI.CyclicInduced.cyclicWeight_phi_of_last`
85	def	`PaperXI.CyclicInduced.degreePeriod`
89	theorem	`PaperXI.CyclicInduced.degreePeriod_coe`
92	theorem	`PaperXI.CyclicInduced.degreeWeight_phi_of_lt`
97	theorem	`PaperXI.CyclicInduced.degreeWeight_phi_of_last`
102	theorem	`PaperXI.CyclicInduced.degreeBasis_phi_of_lt`
109	theorem	`PaperXI.CyclicInduced.degreeBasis_phi_of_last`

A.130 `PaperXI.CyclicLinearization`

Selected layer: v0.44 parent. **Source lines:** 83.

SHA-256: `77c69c57b811f5321306490bb3bb4aef92a7c262d5d875a613ef8d63e5170d0c`.

A genuine projective representation of a finite cyclic group lifts to GL.
The projective representation is an actual homomorphism to Mathlib's PGL,
and equality after projection is the conclusion, not a hypothesis.

Direct project imports: none.

Line	Kind	Name
16	theorem	`PaperXI.CyclicLinearization.unit_root`
29	theorem	`PaperXI.CyclicLinearization.rescale_generator`
50	theorem	`PaperXI.CyclicLinearization.lift_of_generator`
65	theorem	`PaperXI.CyclicLinearization.finite_cyclic_linearization`
73	theorem	`PaperXI.CyclicLinearization.finite_cyclic_scalar_equivalence`

A.131 `PaperXI.CyclicNormalization113`

Selected layer: Continuation113. **Source lines:** 386.

SHA-256: `607587d94a544d2c4e014d81b21e4c9241d852feddfaa826df57f9715d396ce7`.

```
Same-carrier existence and uniqueness of determinant-one cyclic
implementers. The full automorphism and prescribed central character are
retained. No arithmetic realization or identification of the original kappa
is asserted.
```

Direct project imports: `PaperXI.GlobalHeisenbergBridge112`.

Line	Kind	Name
18	def	`PaperXI.CyclicNormalization113.scale`
21	theorem	`PaperXI.CyclicNormalization113.scale_apply`
24	theorem	`PaperXI.CyclicNormalization113.scale_pow_apply`
33	theorem	`PaperXI.CyclicNormalization113.scalar_correction_exists`
48	theorem	`PaperXI.CyclicNormalization113.det_scale`
58	theorem	`PaperXI.CyclicNormalization113.normalize_honest`
78	theorem	`PaperXI.CyclicNormalization113.implements_pow`
90	theorem	`PaperXI.CyclicNormalization113.scale_implements`
101	theorem	`PaperXI.CyclicNormalization113.implementer_power_scalar`
111	theorem	`PaperXI.CyclicNormalization113.unit_nth_root`
124	theorem	`PaperXI.CyclicNormalization113.normalize_order`
140	theorem	`PaperXI.CyclicNormalization113.normalized_generator_exists`
153	theorem	`PaperXI.CyclicNormalization113.normalized_generator_unique`
167	theorem	`PaperXI.CyclicNormalization113.normalized_generator_existsUnique`
189	theorem	`PaperXI.CyclicNormalization113.implementer_of_actual_profile`
219	theorem	`PaperXI.CyclicNormalization113.canonical_generator_of_actual_profile`
239	theorem	`PaperXI.CyclicNormalization113.canonical_generator_of_actual_form`
268	def	`PaperXI.CyclicNormalization113.cyclicAction`
276	theorem	`PaperXI.CyclicNormalization113.cyclicAction_generator`
286	theorem	`PaperXI.CyclicNormalization113.canonical_action_of_implementer`
325	theorem	`PaperXI.CyclicNormalization113.canonical_action_of_actual_profile`
347	theorem	`PaperXI.CyclicNormalization113.canonical_equivariant_bridge_of_actual_profile`

A.132 `PaperXI.CyclicPolynomial`

Selected layer: v0.44 parent. **Source lines:** 81.

SHA-256: `0b7bf89b38ea2b0bc63675d4707494ec562ab9f04b8218c17fc092d9e82ea6ab`.

```
Characteristic polynomials from a proved cyclic basis.
```

Direct project imports: none.

Line	Kind	Name
13	theorem	`PaperXI.CyclicPolynomial.aeval_apply_eq_sum`
20	theorem	`PaperXI.CyclicPolynomial.no_short_annihilator`
31	theorem	`PaperXI.CyclicPolynomial.charpoly_of_orbit_basis`
45	theorem	`PaperXI.CyclicPolynomial.det_of_orbit_basis`
59	theorem	`PaperXI.CyclicPolynomial.eval_charpoly_of_orbit_basis`
67	theorem	`PaperXI.CyclicPolynomial.reverse_cycle_polynomial`
72	theorem	`PaperXI.CyclicPolynomial.local_polynomial_of_orbit_basis`

A.133 `PaperXI.DVRCartan106`

Selected layer: base108 arithmetic branch. **Source lines:** 68.

SHA-256: `450b28f31d92b480d5a52f75464a037a0aaf3ebd66ce7753b50f9e474951eeda`.

```
Normalized Cartan decomposition over the fraction field of a DVR.
```

Direct project imports: `PaperXI.FractionMatrix106`, `PaperXI.CuspidalCartan105`.

Line	Kind	Name
15	def	`PaperXI.DVRCartan106.unitsDiagonal`
21	theorem	`PaperXI.DVRCartan106.normalize_diagonal`
55	theorem	`PaperXI.DVRCartan106.hasCartan`

A.134 `PaperXI.DegreeIndex110`

Selected layer: Continuation110. **Source lines:** 432.

SHA-256: `0f87f63a5f0f5c28adbb1a04732fbe110b63a5f45cdde03e98830271c7c6a6c5`.

```
The degree-square identity from the original irreducible character profile,
and the positive-index cancellation that supplies the intersection Hom line.
Arithmetic trace support, character covariance, and the oriented lattice-index
identity remain explicit hypotheses. Schur is used only on the entire action.
```

Direct project imports: `PaperXI.FiniteProfileBridge109`, `PaperXI.CompactIntersection108`, `PaperXI.ActualIntersection108`, `PaperXI.PairIndex110`.

Line	Kind	Name
20	theorem	`PaperXI.DegreeIndex110.degree_square_of_finite_model`
44	theorem	`PaperXI.DegreeIndex110.degree_square_of_finite_kernel_quotient`
58	theorem	`PaperXI.DegreeIndex110.degree_square`
71	theorem	`PaperXI.DegreeIndex110.intersection_index_of_oriented_identity`
85	theorem	`PaperXI.DegreeIndex110.dimension_one_iff_original_index`
120	theorem	`PaperXI.DegreeIndex110.intersection_dimension_one_of_index`
154	theorem	`PaperXI.DegreeIndex110.intersection_dimension_one`
186	theorem	`PaperXI.DegreeIndex110.intersection_index_pos`
199	theorem	`PaperXI.DegreeIndex110.intersection_index_ambient`
205	theorem	`PaperXI.DegreeIndex110.scalarIntersection_index_ambient`

Line	Kind	Name
214	theorem	`PaperXI.DegreeIndex110.oriented_identity_of_ambient`
236	def	`PaperXI.DegreeIndex110.actualScalarSubgroup`
240	def	`PaperXI.DegreeIndex110.actualScalarCharacter`
251	theorem	`PaperXI.DegreeIndex110.actual_scalar`
264	theorem	`PaperXI.DegreeIndex110.actualScalarSubgroup_image`
275	theorem	`PaperXI.DegreeIndex110.actual_embedding_range_open`
295	theorem	`PaperXI.DegreeIndex110.actual_eta_dimension_one`
343	theorem	`PaperXI.DegreeIndex110.actual_eta_dimension`
390	theorem	`PaperXI.DegreeIndex110.actual_eta_dimension_ambient`

A.135 `PaperXI.DerivedCentralCharacter69`

Selected layer: base108 arithmetic branch. **Source lines:** 41.

SHA-256: `861257cc1f0b38c19e73ec7fec5cb12d5ee9e3affca53ab38d5287c502d56d0a`.

```
Construct the central character of the given irreducible finite-dimensional representation.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.DerivedCentralCharacter69.centralAction`
25	def	`PaperXI.DerivedCentralCharacter69.scalarEquiv`
30	def	`PaperXI.DerivedCentralCharacter69.character`
34	theorem	`PaperXI.DerivedCentralCharacter69.character_apply`

A.136 `PaperXI.DiscrepancyCharacter`

Selected layer: v0.44 parent. **Source lines:** 50.

SHA-256: `cf201fbd9c03f12816213b45d8e418ea5649df147dd993af122f826106f798af`.

```
Discrepancy of honest stabilizer actions is a character, derived from Schur.
```

Direct project imports: `PaperXI.SchurScalar`.

Line	Kind	Name
13	theorem	`PaperXI.DiscrepancyCharacter.exists_character`

A.137 `PaperXI.DualCharacterSums`

Selected layer: v0.44 parent. **Source lines:** 55.

SHA-256: `cd4cf6b9b43305d2e09f09566db4519b5821134b6fac03ac8e92c83fad7c0532`.

```
Literal finite character orthogonality on a vector space and its full dual.
```

Direct project imports: `PaperXI.SchrodingerTrace`, `PaperXI.SchrodingerSeparation`, `PaperXI.HeisenbergDimension`.

Line	Kind	Name
14	theorem	`PaperXI.DualCharacters.sum_over_points`
22	def	`PaperXI.DualCharacters.evaluationCharacter`
28	theorem	`PaperXI.DualCharacters.evaluationCharacter_apply`
31	theorem	`PaperXI.DualCharacters.evaluationCharacter_ne_one`
40	theorem	`PaperXI.DualCharacters.sum_over_dual`
50	theorem	`PaperXI.DualCharacters.card_dual`

A.138 `PaperXI.DualityLinear`

Selected layer: v0.44 parent. **Source lines:** 94.

SHA-256: `0f856e352f57db7831276fa70666d08465b1a79f9ac97e0b4a31265892c47e12`.

```
Actual dual representations, invariant spaces and tensor/Hom comparison.
No fixed-space dimension, invariant-space equivalence or dual irreducibility
is supplied as a hypothesis.
```

Direct project imports: `PaperXI.RepresentationStability`.

Line	Kind	Name
17	def	`PaperXI.DualityLinear.invariantsEquiv`
35	def	`PaperXI.DualityLinear.bidualEquiv`
43	theorem	`PaperXI.DualityLinear.coannihilator_stable`
54	theorem	`PaperXI.DualityLinear.dual_irreducible`
71	def	`PaperXI.DualityLinear.tensorHomEquiv`
82	def	`PaperXI.DualityLinear.fixedTensorHomEquiv`
88	theorem	`PaperXI.DualityLinear.hom_finrank_eq_tensor_invariants`

A.139 `PaperXI.DyadicAbstract`

Selected layer: v0.44 parent. **Source lines:** 159.

SHA-256: `1e70f115444ed1310c3bdae48b5dd98530fa9d8f2eae45684430b7cec5e836d0`.

```
Intrinsic Arf invariants and dyadic formulas under only the abstract
finite-dimensional alternating nondegenerate hypotheses.
```

Direct project imports: `PaperXI.SymplecticCoordinates`, `PaperXI.DyadicBasisConsequences`, `PaperXI.DyadicBasisCounts`.

Line	Kind	Name
15	def	`PaperXI.DyadicAbstract.coordinates`
20	theorem	`PaperXI.DyadicAbstract.coordinates_form`
25	def	`PaperXI.DyadicAbstract.arf`
32	theorem	`PaperXI.DyadicAbstract.arf_eq_basis`
43	def	`PaperXI.DyadicAbstract.gaussSum`
50	theorem	`PaperXI.DyadicAbstract.gaussSum_eq_sum`
58	theorem	`PaperXI.DyadicAbstract.gauss_sum`
68	def	`PaperXI.DyadicAbstract.normalizedGaussSign`
73	theorem	`PaperXI.DyadicAbstract.normalizedGaussSign_eq_sign`

Line	Kind	Name
83	theorem	`PaperXI.DyadicAbstract.arf_translation`
92	theorem	`PaperXI.DyadicAbstract.opposite_refinements`
113	theorem	`PaperXI.DyadicAbstract.even_count`
123	theorem	`PaperXI.DyadicAbstract.odd_count`
133	theorem	`PaperXI.DyadicAbstract.gauss_sum_of_finrank`
142	theorem	`PaperXI.DyadicAbstract.even_count_of_finrank`
151	theorem	`PaperXI.DyadicAbstract.odd_count_of_finrank`

A.140 `PaperXI.DyadicAbstractOrthogonal`

Selected layer: v0.44 parent. **Source lines:** 141.

SHA-256: `84c944337500a19941c2a2a1b839131b3c7ffd8b54f89c118ebb29f3416b811f`.

```
Intrinsic Arf and Gauss-sign laws for actual orthogonal sums.
Symplectic coordinates are constructed from alternation and nondegeneracy.
```

Direct project imports: `PaperXI.DyadicAbstract`, `PaperXI.DyadicIsometry`, `PaperXI.DyadicOrthogonal`.

Line	Kind	Name
17	theorem	`PaperXI.DyadicAbstractOrthogonal.form_alternating`
21	theorem	`PaperXI.DyadicAbstractOrthogonal.form_nondegenerate`
38	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_transport`
52	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_add`
71	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_sign_product`
84	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_of_decomposition`
107	theorem	`PaperXI.DyadicAbstractOrthogonal.arf_sign_of_decomposition`
116	theorem	`PaperXI.DyadicAbstractOrthogonal.normalized_sign_product`
129	theorem	`PaperXI.DyadicAbstractOrthogonal.normalized_sign_of_decomposition`

A.141 `PaperXI.DyadicAffine`

Selected layer: v0.44 parent. **Source lines:** 81.

SHA-256: `06fd5b5a57b17060abd22a11fb29592fcd53b06aa09388b92314fe2fd23692e3`.

```
Paper XI, Proposition 13.2: the affine space of refinements of a fixed
bilinear form over the field with two elements. No nondegeneracy is required.
```

Direct project imports: none.

Line	Kind	Name
13	structure	`PaperXI.DyadicAffine.Refinement`
19	theorem	`PaperXI.DyadicAffine.Refinement.ext`
25	theorem	`PaperXI.DyadicAffine.Refinement.value_zero`
31	theorem	`PaperXI.DyadicAffine.difference_add`
38	def	`PaperXI.DyadicAffine.linearDifference`
43	theorem	`PaperXI.DyadicAffine.linearDifference_apply`
47	def	`PaperXI.DyadicAffine.addLinear`
53	theorem	`PaperXI.DyadicAffine.addLinear_apply`

Line	Kind	Name
56	theorem	`PaperXI.DyadicAffine.addLinear_linearDifference`
61	theorem	`PaperXI.DyadicAffine.linearDifference_addLinear`
67	def	`PaperXI.DyadicAffine.refinementEquiv`
74	theorem	`PaperXI.DyadicAffine.exists_unique_linear_difference`

A.142 `PaperXI.DyadicBasis`

Selected layer: v0.44 parent. **Source lines:** 83.

SHA-256: `18c632203cbd0c023aa7aaf6fcc10af54acb0271d09db3fa9a38d74308c5546f`.

```
Transfer to arbitrary vector spaces with supplied symplectic coordinates.
The existence of those coordinates is explicitly separate.
```

Direct project imports: `PaperXI.DyadicCoordinates`.

Line	Kind	Name
14	def	`PaperXI.DyadicBasis.toCoordinates`
22	def	`PaperXI.DyadicBasis.fromCoordinates`
30	def	`PaperXI.DyadicBasis.refinementEquiv`
42	theorem	`PaperXI.DyadicBasis.toCoordinates_apply`
47	theorem	`PaperXI.DyadicBasis.fromCoordinates_apply`
53	def	`PaperXI.DyadicBasis.basisArf`
56	theorem	`PaperXI.DyadicBasis.basisArf_eq_coordinate_arf`
62	theorem	`PaperXI.DyadicBasis.gauss_sum`
74	theorem	`PaperXI.DyadicBasis.basisArf_independent`

A.143 `PaperXI.DyadicBasisConsequences`

Selected layer: v0.44 parent. **Source lines:** 68.

SHA-256: `d45fa0f60b8166b49c7962a6afc195f6e7cb92dccabb9ee3103ff488062a4031`.

```
Translation and opposite Arf phases on arbitrary spaces with supplied
symplectic coordinates.
```

Direct project imports: `PaperXI.DyadicBasis`, `PaperXI.DyadicRefinement`.

Line	Kind	Name
16	theorem	`PaperXI.DyadicBasisConsequences.toCoordinates_addLinear`
25	theorem	`PaperXI.DyadicBasisConsequences.arf_translation`
42	theorem	`PaperXI.DyadicBasisConsequences.basisArf_fromCoordinates`
48	theorem	`PaperXI.DyadicBasisConsequences.gauss_fromCoordinates`
57	theorem	`PaperXI.DyadicBasisConsequences.opposite_refinements`

A.144 `PaperXI.DyadicBasisCounts`

Selected layer: v0.44 parent. **Source lines:** 35.

SHA-256: `2b9d123b889d163bf29be6dfbab3e510195d98591db8098fb4d773a4a05b6093`.

Counts of every actual refinement on an arbitrary space, conditional on
supplied symplectic coordinates.

Direct project imports: `PaperXI.DyadicBasis`, `PaperXI.DyadicRefinementCounts`.

Line	Kind	Name
14	def	`PaperXI.DyadicBasisCounts.arfFiberEquiv`
21	theorem	`PaperXI.DyadicBasisCounts.even_count`
28	theorem	`PaperXI.DyadicBasisCounts.odd_count`

A.145 `PaperXI.DyadicCoordinates`

Selected layer: v0.44 parent. **Source lines:** 162.

SHA-256: `fc1a2a3a6569e264cfddcb51fd16e91c22ff91ed4a741634239798f18cd83004`.

Every quadratic refinement in symplectic coordinates has unique linear
coefficients. This identifies all refinements, not only a chosen family.

Direct project imports: `PaperXI.DyadicAffine`, `PaperXI.DyadicGauss`.

Line	Kind	Name
15	def	`PaperXI.DyadicCoordinates.symplectic`
56	theorem	`PaperXI.DyadicCoordinates.symplectic_apply`
59	theorem	`PaperXI.DyadicCoordinates.quadratic_polar`
67	def	`PaperXI.DyadicCoordinates.coordinateRefinement`
71	def	`PaperXI.DyadicCoordinates.firstAxis`
72	def	`PaperXI.DyadicCoordinates.secondAxis`
75	theorem	`PaperXI.DyadicCoordinates.single_axes`
79	def	`PaperXI.DyadicCoordinates.linearCoefficients`
82	theorem	`PaperXI.DyadicCoordinates.linear_coordinate_formula`
93	theorem	`PaperXI.DyadicCoordinates.quadratic_firstAxis`
102	theorem	`PaperXI.DyadicCoordinates.quadratic_secondAxis`
111	def	`PaperXI.DyadicCoordinates.refinementCoefficients`
114	theorem	`PaperXI.DyadicCoordinates.refinement_coordinate_formula`
125	theorem	`PaperXI.DyadicCoordinates.refinementCoefficients_coordinateRefinement`
130	def	`PaperXI.DyadicCoordinates.coordinateEquiv`
138	theorem	`PaperXI.DyadicCoordinates.refinementCoefficients_apply`
145	theorem	`PaperXI.DyadicCoordinates.arf_basis_formula`
149	theorem	`PaperXI.DyadicCoordinates.refinement_arf_basis_formula`
155	theorem	`PaperXI.DyadicCoordinates.refinement_gauss_sum`

A.146 `PaperXI.DyadicCounts`

Selected layer: v0.44 parent. **Source lines:** 135.

SHA-256: `b1155d1f3103b197ed16cea0c4c297e29da3f6e917e6006f486e47727af34b5d`.

Exact counts for binary quadratic coefficient spaces.

Direct project imports: `PaperXI.DyadicGauss`.

Line	Kind	Name
13	theorem	`PaperXI.DyadicCounts.signed_sum_eq_card_difference`
19	theorem	`PaperXI.DyadicCounts.card_partition`
27	def	`PaperXI.DyadicCounts.evenCount`
30	def	`PaperXI.DyadicCounts.oddCount`
33	theorem	`PaperXI.DyadicCounts.f2_ne_zero_iff_eq_one`
47	theorem	`PaperXI.DyadicCounts.parameter_card`
50	theorem	`PaperXI.DyadicCounts.even_add_odd`
54	theorem	`PaperXI.DyadicCounts.even_sub_odd`
63	theorem	`PaperXI.DyadicCounts.twice_even`
71	theorem	`PaperXI.DyadicCounts.twice_odd`
78	theorem	`PaperXI.DyadicCounts.even_factored`
90	theorem	`PaperXI.DyadicCounts.odd_factored`
102	theorem	`PaperXI.DyadicCounts.counts_rank_zero`
110	theorem	`PaperXI.DyadicCounts.even_factored_rational`
123	theorem	`PaperXI.DyadicCounts.odd_factored_rational`

A.147 `PaperXI.DyadicDimension`

Selected layer: v0.44 parent. **Source lines:** 49.

SHA-256: `295c12f1fe30abb310b8e54856cc51fb1f11bee0bd9f7d4258386bd30271944a`.

```
Dimensions and Arf comparison for arbitrary finite symplectic index types.
All coordinate maps in this module are explicit inputs.
```

Direct project imports: `PaperXI.DyadicBasis`.

Line	Kind	Name
15	theorem	`PaperXI.DyadicDimension.finrank_eq_twice_card`
20	theorem	`PaperXI.DyadicDimension.card_eq_finrank_div_two`
25	theorem	`PaperXI.DyadicDimension.card_pos`
32	theorem	`PaperXI.DyadicDimension.coordinate_card_eq`
38	theorem	`PaperXI.DyadicDimension.basisArf_independent`

A.148 `PaperXI.DyadicExtensionClass`

Selected layer: v0.44 parent. **Source lines:** 93.

SHA-256: `af6825148cbf10fa7e036416c6b158a32f9be04ed3a83f3767394273839b2656`.

```
A prescribed square profile determines the bilinear F2-extension up to an
actual center- and quotient-preserving isomorphism. We construct the coboundary
with the upper-triangular part in an ordinary basis.
```

Direct project imports: `PaperXI.DyadicHeisenbergLift`.

Line	Kind	Name
14	def	`PaperXI.DyadicExtensionClass.upper`
18	theorem	`PaperXI.DyadicExtensionClass.upper_basis`
24	theorem	`PaperXI.DyadicExtensionClass.upper_symmetrization`

Line	Kind	Name
45	theorem	`PaperXI.DyadicExtensionClass.exists_coboundary`
60	theorem	`PaperXI.DyadicExtensionClass.exists_equiv_of_diagonal`
74	theorem	`PaperXI.DyadicExtensionClass.exists_equiv_iff_diagonal`
85	theorem	`PaperXI.DyadicExtensionClass.refinement_model_independent`

A.149 `PaperXI.DyadicGauss`

Selected layer: v0.44 parent. **Source lines:** 60.

SHA-256: `634fe67a476f515f627455fa43a1ec870a4ccc66b87867cc5d40ee1227c91dea`.

```
Dyadic quadratic refinements in explicit symplectic coordinates.
```

Direct project imports: none.

Line	Kind	Name
10	abbrev	`PaperXI.Dyadic.F2`
12	def	`PaperXI.Dyadic.sign`
14	theorem	`PaperXI.Dyadic.sign_zero`
15	theorem	`PaperXI.Dyadic.sign_one`
17	theorem	`PaperXI.Dyadic.sign_add`
20	theorem	`PaperXI.Dyadic.sign_injective`
30	theorem	`PaperXI.Dyadic.sign_sum`
37	def	`PaperXI.Dyadic.plane`
39	theorem	`PaperXI.Dyadic.plane_gauss`
47	def	`PaperXI.Dyadic.quadratic`
49	def	`PaperXI.Dyadic.arf`
51	theorem	`PaperXI.Dyadic.gauss_sum`

A.150 `PaperXI.DyadicHeisenbergLift`

Selected layer: v0.44 parent. **Source lines:** 155.

SHA-256: `916986a499526e1343b2b63906fe133ac8db3cb95e6cbbbe31d6ada8e17229b1`.

```
A given quadratic refinement on an intrinsic finite F2-space supplies an
actual bilinear factor set, hence a finite Heisenberg extension. Coordinates are
constructed from the form; its square profile is the given refinement, not erased.
```

Direct project imports: `PaperXI.HeisenbergCochain`, `PaperXI.SymplecticCoordinates`.

Line	Kind	Name
15	def	`PaperXI.DyadicHeisenbergLift.coordinateFactor`
65	theorem	`PaperXI.DyadicHeisenbergLift.coordinateFactor_apply`
69	theorem	`PaperXI.DyadicHeisenbergLift.coordinateFactor_commutator`
78	theorem	`PaperXI.DyadicHeisenbergLift.coordinateFactor_diagonal`
91	def	`PaperXI.DyadicHeisenbergLift.coordinateRefinementOf`
100	def	`PaperXI.DyadicHeisenbergLift.factorOfCoordinates`
106	theorem	`PaperXI.DyadicHeisenbergLift.factorOfCoordinates_commutator`
113	theorem	`PaperXI.DyadicHeisenbergLift.factorOfCoordinates_diagonal`

Line	Kind	Name
125	theorem	`PaperXI.DyadicHeisenbergLift.exists_factor`
135	def	`PaperXI.DyadicHeisenbergLift.factor`
139	theorem	`PaperXI.DyadicHeisenbergLift.factor_commutator`
144	theorem	`PaperXI.DyadicHeisenbergLift.factor_diagonal`
150	theorem	`PaperXI.DyadicHeisenbergLift.square_profile`

A.151 `PaperXI.DyadicIsometry`

Selected layer: v0.44 parent. **Source lines:** 52.

SHA-256: `48d9678ba58d442aa451540502f01d5e1fb8a3257e8bdf4d64fef974305c651f`.

```
Quadratic refinements and their Arf values transport along actual isometries.
```

Direct project imports: `PaperXI.DyadicDimension`.

Line	Kind	Name
14	def	`PaperXI.DyadicIsometry.transportRefinement`
21	theorem	`PaperXI.DyadicIsometry.transportRefinement_apply`
25	def	`PaperXI.DyadicIsometry.transportRefinementEquiv`
33	theorem	`PaperXI.DyadicIsometry.gauss_sum_transport`
41	theorem	`PaperXI.DyadicIsometry.basisArf_transport`

A.152 `PaperXI.DyadicOrthogonal`

Selected layer: v0.44 parent. **Source lines:** 76.

SHA-256: `3ec83d3eeac7e69ef4986a573ba818eb9cfe33282530bb3e8f85d7ed4323d671`.

Direct project imports: `PaperXI.DyadicBasis`, `PaperXI.DyadicTranslation`.

Line	Kind	Name
14	def	`PaperXI.DyadicOrthogonal.form`
18	theorem	`PaperXI.DyadicOrthogonal.form_apply`
20	def	`PaperXI.DyadicOrthogonal.refinement`
28	theorem	`PaperXI.DyadicOrthogonal.gauss_product`
35	def	`PaperXI.DyadicOrthogonal.coordinates`
40	theorem	`PaperXI.DyadicOrthogonal.coordinates_form`
48	theorem	`PaperXI.DyadicOrthogonal.arf_add`
67	theorem	`PaperXI.DyadicOrthogonal.normalized_sign_product`

A.153 `PaperXI.DyadicRefinement`

Selected layer: v0.44 parent. **Source lines:** 48.

SHA-256: `4056c45a4a85c4fbaf44d651ce11f83f50656138b52fb6f18d95e9b8eafa01e5`.

```
Translation and opposite phases for all refinements of the actual
coordinate symplectic form, with the Arf value evaluated on basis vectors.
```

Direct project imports: `PaperXI.DyadicCoordinates`, `PaperXI.DyadicTranslation`.

Line	Kind	Name
13	def	`PaperXI.DyadicRefinement.basisArf`
16	theorem	`PaperXI.DyadicRefinement.basisArf_coordinateRefinement`
20	theorem	`PaperXI.DyadicRefinement.translation_refinement`
29	theorem	`PaperXI.DyadicRefinement.arf_translation`
36	theorem	`PaperXI.DyadicRefinement.opposite_refinements`
45	theorem	`PaperXI.DyadicRefinement.sign_eq_neg_one_pow`

A.154 `PaperXI.DyadicRefinementCounts`

Selected layer: v0.44 parent. **Source lines:** 55.

SHA-256: `5d9cdc46b7bde9d4f7ceed89b2a7a68ec38f36fee3142300f40b6a8d5301838b`.

```
Cardinalities of the complete spaces of quadratic refinements with each
Arf invariant, including dimension zero.
```

Direct project imports: `PaperXI.DyadicCounts`, `PaperXI.DyadicCoordinates`.

Line	Kind	Name
13	instance	`PaperXI.DyadicRefinementCounts.refinementFintype`
16	def	`PaperXI.DyadicRefinementCounts.basisArf`
19	theorem	`PaperXI.DyadicRefinementCounts.basisArf_coordinate`
24	def	`PaperXI.DyadicRefinementCounts.arfFiberEquiv`
29	theorem	`PaperXI.DyadicRefinementCounts.even_fiber_card`
35	theorem	`PaperXI.DyadicRefinementCounts.odd_fiber_card`
42	theorem	`PaperXI.DyadicRefinementCounts.even_refinement_count`
49	theorem	`PaperXI.DyadicRefinementCounts.odd_refinement_count`

A.155 `PaperXI.DyadicTorsor`

Selected layer: v0.44 parent. **Source lines:** 70.

SHA-256: `da8d7de85105943a07b9492568783dfc1df76d3789bb3444d344259b0ed8bb6b`.

```
The canonical affine-space structure on a nonempty family of refinements.
The action and subtraction are the actual addition and difference of functions.
```

Direct project imports: `PaperXI.DyadicAffine`.

Line	Kind	Name
14	theorem	`PaperXI.DyadicTorsor.addLinear_zero`
18	theorem	`PaperXI.DyadicTorsor.addLinear_add`
24	theorem	`PaperXI.DyadicTorsor.addLinear_injective`
30	theorem	`PaperXI.DyadicTorsor.addLinear_free`
39	theorem	`PaperXI.DyadicTorsor.addLinear_transitive`
44	instance	`PaperXI.DyadicTorsor.instAddAction`
49	instance	`PaperXI.DyadicTorsor.instVSub`
54	instance	`PaperXI.DyadicTorsor.instAddTorsor`

Line	Kind	Name
61	abbrev	`PaperXI.DyadicTorsor.canonicalTorsor`
64	theorem	`PaperXI.DyadicTorsor.vadd_apply`
67	theorem	`PaperXI.DyadicTorsor.vsub_apply`

A.156 `PaperXI.DyadicTranslation`

Selected layer: v0.44 parent. **Source lines:** 89.

SHA-256: `098a6b0ee464537aaea2cef020e535b28ba9572b8079134076fdd5ee6fb54969`.

Direct project imports: `PaperXI.DyadicGauss`.

Line	Kind	Name
9	def	`PaperXI.Dyadic.pairing`
12	def	`PaperXI.Dyadic.translate`
15	theorem	`PaperXI.Dyadic.quadratic_translate`
23	theorem	`PaperXI.Dyadic.arf_translate`
31	theorem	`PaperXI.Dyadic.gauss_translate`
37	theorem	`PaperXI.Dyadic.arf_orthogonal`
41	theorem	`PaperXI.Dyadic.gauss_orthogonal`
57	theorem	`PaperXI.Dyadic.sign_orthogonal`
61	theorem	`PaperXI.Dyadic.arf_zero`
64	def	`PaperXI.Dyadic.anisotropic`
66	theorem	`PaperXI.Dyadic.arf_anisotropic`
76	theorem	`PaperXI.Dyadic.opposite_gauss`
82	theorem	`PaperXI.Dyadic.both_arf`

A.157 `PaperXI.ElementaryEnvelope`

Selected layer: v0.44 parent. **Source lines:** 139.

SHA-256: `76c36ba11aabeef9351becc4f99f59db800649c73d1dc0f0ff57d59bc7e5d6c4`.

```
Paper XI, 8.4 and the elementary-abelian portion of 9.4–9.5.
The complex cocycle is arbitrary. Only its commutator is forced to have
values in the group of pth roots of unity.
```

Direct project imports: `PaperXI.CocycleQuotient`.

Line	Kind	Name
18	theorem	`PaperXI.ElementaryEnvelope.commutator_pow`
25	abbrev	`PaperXI.ElementaryEnvelope.groupModule`
28	theorem	`PaperXI.ElementaryEnvelope.finiteDimensional_group`
36	def	`PaperXI.ElementaryEnvelope.Space`
39	theorem	`PaperXI.ElementaryEnvelope.quotient_pow`
45	instance	`PaperXI.ElementaryEnvelope.spaceModule`
50	instance	`PaperXI.ElementaryEnvelope.finiteSpace`
53	instance	`PaperXI.ElementaryEnvelope.finiteDimensionalSpace`
58	def	`PaperXI.ElementaryEnvelope.rootCharacter`
69	def	`PaperXI.ElementaryEnvelope.rootEquiv`

Line	Kind	Name
76	theorem	`PaperXI.ElementaryEnvelope.rootEquiv_coe`
80	def	`PaperXI.ElementaryEnvelope.rootLog`
83	theorem	`PaperXI.ElementaryEnvelope.rootLog_character`
86	theorem	`PaperXI.ElementaryEnvelope.character_rootLog`
90	theorem	`PaperXI.ElementaryEnvelope.quotientPairing_pow`
97	def	`PaperXI.ElementaryEnvelope.rootPairing`
115	theorem	`PaperXI.ElementaryEnvelope.rootPairing_coe`
119	def	`PaperXI.ElementaryEnvelope.rootCommutator`
133	theorem	`PaperXI.ElementaryEnvelope.rootCommutator_coe`
136	theorem	`PaperXI.ElementaryEnvelope.rootCommutator_self`

A.158 `PaperXI.ElementaryEnvelopeForm`

Selected layer: v0.44 parent. **Source lines:** 137.

SHA-256: `8bfa652c46358cbe6ee15446b6d435521aeac0f75cc96a605a5fb41b3bf5bc6f`.

```
The actual alternating prime-field form on the commutator radical quotient.
The logarithm equivalence is explicit: the standard character supplies one,
and all alternative choices have a proved scalar comparison.
```

Direct project imports: `PaperXI.ElementaryEnvelope`.

Line	Kind	Name
15	def	`PaperXI.ElementaryEnvelope.logarithmicPairing`
37	def	`PaperXI.ElementaryEnvelope.formWith`
42	theorem	`PaperXI.ElementaryEnvelope.formWith_apply`
47	theorem	`PaperXI.ElementaryEnvelope.formWith_character`
54	theorem	`PaperXI.ElementaryEnvelope.formWith_smul_left`
59	theorem	`PaperXI.ElementaryEnvelope.formWith_smul_right`
64	theorem	`PaperXI.ElementaryEnvelope.formWith_isAlt`
73	theorem	`PaperXI.ElementaryEnvelope.formWith_eq_zero_iff`
80	theorem	`PaperXI.ElementaryEnvelope.formWith_nondegenerate`
93	def	`PaperXI.ElementaryEnvelope.form`
95	theorem	`PaperXI.ElementaryEnvelope.form_isAlt`
97	theorem	`PaperXI.ElementaryEnvelope.form_nondegenerate`
101	theorem	`PaperXI.ElementaryEnvelope.form_stdAddChar`
109	theorem	`PaperXI.ElementaryEnvelope.choice_scalar_ne_zero`
115	theorem	`PaperXI.ElementaryEnvelope.choice_map_eq_mul`
124	theorem	`PaperXI.ElementaryEnvelope.formWith_choice`
131	theorem	`PaperXI.ElementaryEnvelope.exists_choice_scalar`

A.159 `PaperXI.ElementaryEnvelopeTransport`

Selected layer: v0.44 parent. **Source lines:** 148.

SHA-256: `56b022844ede8e64c96a72cadf7393135d60c5c9601cd457d49f95322e3fcf18`.

```
Functoriality of the actual elementary commutator envelope.
All maps here descend the given group maps, and preserve the form for one
common chosen root coordinate.
```

Direct project imports: `PaperXI.ElementaryEnvelopeForm`.

Line	Kind	Name
15	theorem	`PaperXI.ElementaryEnvelope.radical_map`
34	def	`PaperXI.ElementaryEnvelope.quotientEquiv`
39	theorem	`PaperXI.ElementaryEnvelope.quotientEquiv_mk`
43	theorem	`PaperXI.ElementaryEnvelope.quotientEquiv_pairing`
56	def	`PaperXI.ElementaryEnvelope.linearEquiv`
62	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_mk`
68	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_formWith`
79	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_form`
85	theorem	`PaperXI.ElementaryEnvelope.space_induction_on`
89	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_refl`
98	theorem	`PaperXI.ElementaryEnvelope.linearEquiv_trans`
113	def	`PaperXI.ElementaryEnvelope.action`
135	theorem	`PaperXI.ElementaryEnvelope.action_mk`
142	theorem	`PaperXI.ElementaryEnvelope.action_form`

A.160 `PaperXI.EquivariantRepresentation113`

Selected layer: Continuation113. **Source lines:** 117.

SHA-256: `f1cf8ff3343f2647c0994270f6de5bc3478925a25406ea3a221861807b08b6c9`.

```
The corrected scalar cochain constructs an actual representation that
retains the original theta and intertwines the original packet generator.
Actual groups and their full automorphisms remain explicit inputs.
```

Direct project imports: `PaperXI.ActualScalarLift113`.

Line	Kind	Name
38	theorem	`PaperXI.EquivariantRepresentation113.scalarTransport_covariant`
67	theorem	`PaperXI.EquivariantRepresentation113.exists_equivariant_actual_action`

A.161 `PaperXI.EquivariantSplitting68`

Selected layer: base108 arithmetic branch. **Source lines:** 35.

SHA-256: `561e6ca7bb701f95ce183d25a1b39b3dbf2a5d389924851f78373fa3f52e4460`.

```
A compact-mod-scalar equivariant splitting, constructed by a finite orbit sum.
```

Direct project imports: `PaperXI.SmoothHomOrbit68`, `PaperXI.FiniteOrbitAverage68`.

Line	Kind	Name
20	theorem	`PaperXI.EquivariantSplitting68.exists_equivariant_section`

A.162 `PaperXI.FiniteAverage`

Selected layer: v0.44 parent. **Source lines:** 69.

SHA-256: `e960b8793c2a7e76fef815c456c4e61db736706716741679c3d423053722c977`.

```
Literal finite group averages and their invariant and orthogonal projection properties.
```

Direct project imports: none.

Line	Kind	Name
11	def	`PaperXI.FiniteAverage.average`
14	theorem	`PaperXI.FiniteAverage.average_apply`
18	theorem	`PaperXI.FiniteAverage.card_ne_zero`
21	theorem	`PaperXI.FiniteAverage.invariant`
27	theorem	`PaperXI.FiniteAverage.fixes`
34	theorem	`PaperXI.FiniteAverage.idempotent`
39	theorem	`PaperXI.FiniteAverage.range_eq`
51	theorem	`PaperXI.FiniteAverage.self_adjoint`
62	theorem	`PaperXI.FiniteAverage.residual_orthogonal`

A.163 `PaperXI.FiniteBreakAssembly`

Selected layer: v0.44 parent. **Source lines:** 175.

SHA-256: `ecca3693f7733c7ceef597b6e821b111610c82d612fd66abfe754af0e5be434c`.

```
Construct the initial tensor bridge and close its honest stabilizer action.
The central extensions (including dyadic square data), their actual lifts,
central characters and actions are inputs. No initial intertwiner, discrepancy
character, finite-dimensional certificate, or final comparison is an input.
```

Direct project imports: `PaperXI.BreakBridgeAlong`, `PaperXI.CentralExtensionComparison`, `PaperXI.TraceDiscrepancy`.

Line	Kind	Name
22	structure	`PaperXI.FiniteBreakAssembly.Data`
43	def	`PaperXI.FiniteBreakAssembly.factor`
51	theorem	`PaperXI.FiniteBreakAssembly.factor_intertwines`
60	def	`PaperXI.FiniteBreakAssembly.tensorBridge`
63	def	`PaperXI.FiniteBreakAssembly.sourceAction`
67	def	`PaperXI.FiniteBreakAssembly.targetAction`
71	theorem	`PaperXI.FiniteBreakAssembly.tensor_intertwines`
79	theorem	`PaperXI.FiniteBreakAssembly.source_finiteDimensional`
84	theorem	`PaperXI.FiniteBreakAssembly.source_irreducible`
96	def	`PaperXI.FiniteBreakAssembly.discrepancy`
102	theorem	`PaperXI.FiniteBreakAssembly.discrepancy_law`
111	def	`PaperXI.FiniteBreakAssembly.equivOfPhase`
132	theorem	`PaperXI.FiniteBreakAssembly.discrepancy_and_closure`
151	def	`PaperXI.FiniteBreakAssembly.equivOfTraces`
169	theorem	`PaperXI.FiniteBreakAssembly.equivOfTraces_map`

A.164 `PaperXI.FiniteFieldAverage`

Selected layer: v0.44 parent. **Source lines:** 186.

SHA-256: `da2bd3e2ade03fa988a3b50efe5b53ef4ed47f3636c86fc5f6841cc8b868f81d`.

```
Actual finite-order averages and their fixed/moving decomposition over a field.
The construction uses the invertibility of the order scalar, with no initial
complementary decomposition or finite-dimensionality assumption.
```

Direct project imports: none.

Line	Kind	Name
16	def	`PaperXI.FiniteFieldAverage.difference`
19	def	`PaperXI.FiniteFieldAverage.average`
23	def	`PaperXI.FiniteFieldAverage.companion`
26	theorem	`PaperXI.FiniteFieldAverage.difference_apply`
28	theorem	`PaperXI.FiniteFieldAverage.average_apply`
32	theorem	`PaperXI.FiniteFieldAverage.companion_apply`
37	theorem	`PaperXI.FiniteFieldAverage.weighted_identity`
50	theorem	`PaperXI.FiniteFieldAverage.power_eq_one`
58	theorem	`PaperXI.FiniteFieldAverage.average_mul_difference`
65	theorem	`PaperXI.FiniteFieldAverage.difference_mul_average`
72	theorem	`PaperXI.FiniteFieldAverage.moving_identity`
79	theorem	`PaperXI.FiniteFieldAverage.power_fixes`
86	theorem	`PaperXI.FiniteFieldAverage.average_fixes`
94	theorem	`PaperXI.FiniteFieldAverage.range_average`
104	theorem	`PaperXI.FiniteFieldAverage.ker_average`
117	theorem	`PaperXI.FiniteFieldAverage.idempotent`
124	theorem	`PaperXI.FiniteFieldAverage.isCompl`
131	def	`PaperXI.FiniteFieldAverage.decomposition`
136	theorem	`PaperXI.FiniteFieldAverage.decomposition_apply`
142	theorem	`PaperXI.FiniteFieldAverage.decomposition_symm_fst`
154	theorem	`PaperXI.FiniteFieldAverage.decomposition_symm_snd`
162	theorem	`PaperXI.FiniteFieldAverage.average_mul_action`
170	theorem	`PaperXI.FiniteFieldAverage.action_mul_average`
178	theorem	`PaperXI.FiniteFieldAverage.average_independent`

A.165 `PaperXI.FiniteFieldOrbitCount`

Selected layer: v0.44 parent. **Source lines:** 93.

SHA-256: `624a4a0234ebd63e8062471cb1439ed52dc73ed82e5d8f6400549c16196dde42`.

```
Proposition 3.6 on actual complex-valued multiplicative characters.
The cyclicity and cardinality of the character group are derived, not assumed.
```

Direct project imports: `PaperXI.CyclicFrobeniusCount`.

Line	Kind	Name
13	instance	`PaperXI.FiniteFieldOrbitCount.complexEnoughRoots`
19	theorem	`PaperXI.FiniteFieldOrbitCount.characterGroup_isCyclic`
25	theorem	`PaperXI.FiniteFieldOrbitCount.character_card`
28	def	`PaperXI.FiniteFieldOrbitCount.characterFrobenius`
33	theorem	`PaperXI.FiniteFieldOrbitCount.characterFrobenius_apply`
41	theorem	`PaperXI.FiniteFieldOrbitCount.fixed_character_count`
47	theorem	`PaperXI.FiniteFieldOrbitCount.exact_character_count`
54	theorem	`PaperXI.FiniteFieldOrbitCount.regular_character_orbits`

Line	Kind	Name
62	theorem	`PaperXI.FiniteFieldOrbitCount.regular_numerator_divisible`
70	def	`PaperXI.FiniteFieldOrbitCount.extensionFrobenius`
77	theorem	`PaperXI.FiniteFieldOrbitCount.extension_regular_orbits`
88	theorem	`PaperXI.FiniteFieldOrbitCount.extensionFrobenius_apply`

A.166 `PaperXI.FiniteFreeCompletion`

Selected layer: v0.44 parent. **Source lines:** 75.

SHA-256: `218a5b9e952143a9e26b36d2bcf93529baca8ae45f986c76903854d7c18aeead`.

```
Completeness is derived in finite free coordinates, not assumed of the
constructed root ring.
```

Direct project imports: `PaperXI.ResidueExtensionRealization`.

Line	Kind	Name
14	theorem	`PaperXI.FiniteFreeCompletion.linear_congruence`
25	theorem	`PaperXI.FiniteFreeCompletion.precomplete`
42	theorem	`PaperXI.FiniteFreeCompletion.hausdorff`
53	theorem	`PaperXI.FiniteFreeCompletion.complete`
65	theorem	`PaperXI.CompleteResidueRealization.complete`

A.167 `PaperXI.FiniteFunctionAverage`

Selected layer: v0.44 parent. **Source lines:** 54.

SHA-256: `a7d776b604ef43cade5a813a977a7ce80ebd96a2b7906759e3b9fc89c7420d01`.

```
Orthogonal averaging for the actual finite-sum Hermitian pairing on functions.
```

Direct project imports: `PaperXI.TranslationAverage`.

Line	Kind	Name
11	theorem	`PaperXI.FiniteFunctionAverage.pairing_average_left`
24	theorem	`PaperXI.FiniteFunctionAverage.pairing_average_right`
36	theorem	`PaperXI.FiniteFunctionAverage.self_adjoint`
46	theorem	`PaperXI.FiniteFunctionAverage.residual_orthogonal`

A.168 `PaperXI.FiniteIntersectionCriterion108`

Selected layer: base108 arithmetic branch. **Source lines:** 30.

SHA-256: `a428d271392d27755c20bb0b6dac61b71398d4cc6370fe0642c4e5ed41364f51`.

```
An exact criterion on a literal finite model. The scalar character profile,
compatibility and index identity must still be proved for the arithmetic data.
```

Direct project imports: `PaperXI.CompactIntersection108`.

Line	Kind	Name
14	theorem	`PaperXI.FiniteIntersectionCriterion108.dimension_one_iff`

A.169 `PaperXI.FiniteJointQuotient108`

Selected layer: base108 arithmetic branch. **Source lines:** 47.

SHA-256: `bd7174c5b31be9c6750454626e3e5627651dbea5f5465b49f2f3cbfbedc846df`.

```
A single finite quotient for two smooth actions, constructed from their
actual joint kernel. No faithful quotient or finite image is assumed.
```

Direct project imports: `PaperXI.SmoothDegreeQuotient`.

Line	Kind	Name
13	def	`PaperXI.FiniteJointQuotient108.kernel`
16	instance	`PaperXI.FiniteJointQuotient108.kernel_normal`
19	abbrev	`PaperXI.FiniteJointQuotient108.quotient`
21	def	`PaperXI.FiniteJointQuotient108.first`
25	def	`PaperXI.FiniteJointQuotient108.second`
29	theorem	`PaperXI.FiniteJointQuotient108.first_comp`
32	theorem	`PaperXI.FiniteJointQuotient108.second_comp`
40	theorem	`PaperXI.FiniteJointQuotient108.finite`

A.170 `PaperXI.FiniteOrbitAverage68`

Selected layer: base108 arithmetic branch. **Source lines:** 101.

SHA-256: `41b8b3e87046ef73167fbebd34c7c4a62a6bf13714ff572356c87c64a17b3de6`.

```
Finite-orbit averaging with preservation of a genuine affine constraint.
```

Direct project imports: none.

Line	Kind	Name
16	abbrev	`PaperXI.FiniteOrbitAverage68.Orbit`
20	def	`PaperXI.FiniteOrbitAverage68.orbitEquiv`
32	def	`PaperXI.FiniteOrbitAverage68.average`
40	theorem	`PaperXI.FiniteOrbitAverage68.average_invariant`
51	theorem	`PaperXI.FiniteOrbitAverage68.average_constraint`
68	def	`PaperXI.FiniteOrbitAverage68.invariantHom`
79	theorem	`PaperXI.FiniteOrbitAverage68.equivariant_section_of_finite_orbit`

A.171 `PaperXI.FiniteProductDeterminant`

Selected layer: v0.44 parent. **Source lines:** 64.

SHA-256: `07340f6176b2b67b41bba32f65bdb41e2805856a67877d7e70b73585b9c7b556`.

```
Determinants of actual finite products with varying coordinate spaces.
```

Direct project imports: none.

Line	Kind	Name
14	def	`PaperXI.FiniteProductDeterminant.map`
19	theorem	`PaperXI.FiniteProductDeterminant.map_apply`
24	theorem	`PaperXI.FiniteProductDeterminant.determinant`

A.172 `PaperXI.FiniteProfileBridge109`

Selected layer: Continuation110. **Source lines:** 195.

SHA-256: `d829f6afedc938f83bdb668850c4a684874773efe807983b4ae769844588cf0a`.

```
Bounded auxiliary results for the next finite-profile bridge. These prove
generic algebraic descent prerequisites, not the actual simple-character
profile or the BK lattice-index equality.
```

Direct project imports: `PaperXI.HeisenbergTrace108`, `PaperXI.CharacterPairing108`, `PaperXI.FiniteJointQuotient108`.

Line	Kind	Name
19	theorem	`PaperXI.FiniteProfileBridge109.kernel_le_of_trace_support`
33	theorem	`PaperXI.FiniteProfileBridge109.scalar_character_eq_one_on_kernel`
48	theorem	`PaperXI.FiniteProfileBridge109.joint_kernel_le`
60	def	`PaperXI.FiniteProfileBridge109.imageMap`
65	theorem	`PaperXI.FiniteProfileBridge109.imageMap_surjective`
73	theorem	`PaperXI.FiniteProfileBridge109.imageMap_kernel_le_character_kernel`
90	def	`PaperXI.FiniteProfileBridge109.imageCharacter`
94	theorem	`PaperXI.FiniteProfileBridge109.imageCharacter_apply`
102	theorem	`PaperXI.FiniteProfileBridge109.image_scalar`
113	theorem	`PaperXI.FiniteProfileBridge109.image_trace_zero`
120	theorem	`PaperXI.FiniteProfileBridge109.image_intersection`
130	theorem	`PaperXI.FiniteProfileBridge109.image_index`
134	theorem	`PaperXI.FiniteProfileBridge109.image_characters_compatible`
158	theorem	`PaperXI.FiniteProfileBridge109.quotient_kernel_le_action_kernel`
168	theorem	`PaperXI.FiniteProfileBridge109.dimension_one_iff_of_original_profiles`

A.173 `PaperXI.FiniteRepresentation`

Selected layer: v0.44 parent. **Source lines:** 43.

SHA-256: `b9ac153884bbf3b17a2701a8b50ee49c6f936899b7e807fa3a1cae5006e44d47`.

```
Finite dimension is derived from the actual finite group action.
The generic input is the cyclicity of a nonzero simple module, not a
semisimple decomposition or an assumed target dimension.
```

Direct project imports: `PaperXI.RepresentationStability`.

Line	Kind	Name
16	theorem	`PaperXI.FiniteRepresentation.simple_finiteDimensional`
26	theorem	`PaperXI.FiniteRepresentation.finiteDimensional`

Line	Kind	Name
35	theorem	PaperXI.FiniteRepresentation.finrank_pos

A.174 PaperXI.FiniteResidueBaseChange

Selected layer: v0.44 parent. **Source lines:** 91.

SHA-256: 559f2e07321f7991ab4790ffc84edb044865ea96de4a2c240148b8fb39aea272.

```
Genuine finite-residue Frobenius, base-field change, and period detection.
The chosen embeddings are inputs, not a constructed unramified extension of a
local field. Residue Frobenius is never replaced by an abstract cyclic label.
```

Direct project imports: PaperXI.ResidueDegreeBaseChange.

Line	Kind	Name
16	def	PaperXI.FiniteResidueBaseChange.arithmetic
20	def	PaperXI.FiniteResidueBaseChange.geometric
22	theorem	PaperXI.FiniteResidueBaseChange.arithmetic_apply
24	theorem	PaperXI.FiniteResidueBaseChange.arithmetic_pow_apply
33	theorem	PaperXI.FiniteResidueBaseChange.geometric_inverse_power
42	theorem	PaperXI.FiniteResidueBaseChange.arithmetic_tower
50	theorem	PaperXI.FiniteResidueBaseChange.geometric_tower
56	def	PaperXI.FiniteResidueBaseChange.pullbackEquiv
67	def	PaperXI.FiniteResidueBaseChange.forget
73	theorem	PaperXI.FiniteResidueBaseChange.forget_injective
80	theorem	PaperXI.FiniteResidueBaseChange.geometric_order
86	theorem	PaperXI.FiniteResidueBaseChange.geometric_zpow_eq_one

A.175 PaperXI.FiniteResidueSubgroup

Selected layer: v0.44 parent. **Source lines:** 128.

SHA-256: 2da76b0ad80ae46f7b05f39967fee2a9247b7c80f1a9e0ef61b0423a026f8a72.

```
A residue-field fixing subgroup, not an abstract subgroup assigned index f.
Its pullback for geometric |k'|-Frobenius is identified with the degree-f subgroup
of the |k|-Frobenius pullback. No local-field lifting theorem is asserted.
```

Direct project imports: PaperXI.FiniteResidueBaseChange.

Line	Kind	Name
14	theorem	PaperXI.FiniteResidueSubgroup.arithmetic_map
19	theorem	PaperXI.FiniteResidueSubgroup.geometric_map
26	theorem	PaperXI.FiniteResidueSubgroup.geometric_pow_map
38	theorem	PaperXI.FiniteResidueSubgroup.geometric_zpow_map
51	theorem	PaperXI.FiniteResidueSubgroup.fixes_subfield_iff
70	def	PaperXI.FiniteResidueSubgroup.fixing
87	theorem	PaperXI.FiniteResidueSubgroup.kernel_le_fixing
94	theorem	PaperXI.FiniteResidueSubgroup.projection_mem_iff
104	theorem	PaperXI.FiniteResidueSubgroup.projection_fixes

Line	Kind	Name
111	def	PaperXI.FiniteResidueSubgroup.fixingEquiv
122	def	PaperXI.FiniteResidueSubgroup.unramifiedEquiv

A.176 PaperXI.FiniteTameComparison

Selected layer: v0.44 parent. **Source lines:** 113.

SHA-256: 1c4357da32eaade5be0c88670eaa605f47fbc4269253888bbf9f72509d0e428d.

```
Proposition 16.1: an explicit finite comparison package on genuine induced
representations. The local-Weil and standard Artin/epsilon-factor identifications
are intentionally NOT fields of this finite record and are not proved here.
```

Direct project imports: PaperXI.InducedTameProfile, PaperXI.TameGaussNaturality, PaperXI.TameTensorSource.

Line	Kind	Name
17	structure	PaperXI.FiniteTameComparison.Observables
40	def	PaperXI.FiniteTameComparison.observed
53	def	PaperXI.FiniteTameComparison.fromCoordinates
64	theorem	PaperXI.FiniteTameComparison.observations_eq_coordinates
83	def	PaperXI.FiniteTameComparison.interactionEquiv
93	theorem	PaperXI.FiniteTameComparison.finite_comparison_package

A.177 PaperXI.FiniteTensor

Selected layer: v0.44 parent. **Source lines:** 69.

SHA-256: 8a628f3f2be9e4cbc308b13abbc71ef858a82cf897523b60554829ed81ac0782.

```
Actual tensor products indexed by an arbitrary finite family, with their
external product actions. The empty family retains its scalar tensor unit.
```

Direct project imports: none.

Line	Kind	Name
17	def	PaperXI.FiniteTensor.representation
25	theorem	PaperXI.FiniteTensor.representation_tprod
32	theorem	PaperXI.FiniteTensor.finrank
40	theorem	PaperXI.FiniteTensor.finrank_empty
46	theorem	PaperXI.FiniteTensor.nontrivial
53	theorem	PaperXI.FiniteTensor.empty_action
60	theorem	PaperXI.FiniteTensor.scalar_map

A.178 PaperXI.FiniteUnramifiedWeil

Selected layer: v0.44 parent. **Source lines:** 198.

SHA-256: 73d632d598c02e6eced406cf8bd8f48422a3b79634bfb718dea82cc440d29f40.

```
# Finite unramified Weil quotient of the constructed local field
```

Direct project imports: `PaperXI.ResidueFrobeniusAttachment`, `PaperXI.ResidueDegreeBaseChange`.

Line	Kind	Name
26	abbrev	`PaperXI.FiniteUnramifiedWeil.UnramifiedWeil`
30	def	`PaperXI.FiniteUnramifiedWeil.degreeEquiv`
44	theorem	`PaperXI.FiniteUnramifiedWeil.degreeEquiv_apply`
50	theorem	`PaperXI.FiniteUnramifiedWeil.projection_eq_geometric_zpow_degree`
57	theorem	`PaperXI.FiniteUnramifiedWeil.degree_bijective`
62	theorem	`PaperXI.FiniteUnramifiedWeil.inertia_eq_bot`
67	def	`PaperXI.FiniteUnramifiedWeil.geometricLift`
71	theorem	`PaperXI.FiniteUnramifiedWeil.degree_geometricLift`
77	theorem	`PaperXI.FiniteUnramifiedWeil.projection_geometricLift`
84	def	`PaperXI.FiniteUnramifiedWeil.arithmeticLift`
88	theorem	`PaperXI.FiniteUnramifiedWeil.degree_arithmeticLift`
95	theorem	`PaperXI.FiniteUnramifiedWeil.projection_arithmeticLift`
104	theorem	`PaperXI.FiniteUnramifiedWeil.eq_geometricLift_zpow_degree`
118	theorem	`PaperXI.FiniteUnramifiedWeil.geometricLift_zpowers_eq_top`
128	instance	`PaperXI.FiniteUnramifiedWeil.unramifiedWeil_isMulCommutative`
136	theorem	`PaperXI.FiniteUnramifiedWeil.irreducible_finrank_eq_one`
144	theorem	`PaperXI.FiniteUnramifiedWeil.projection_surjective`
156	theorem	`PaperXI.FiniteUnramifiedWeil.projection_ker_eq_period`
189	def	`PaperXI.FiniteUnramifiedWeil.finiteResidueQuotientEquiv`

A.179 `PaperXI.FixedMoving`

Selected layer: v0.44 parent. **Source lines:** 198.

SHA-256: `c5159f03721e3f1446641781dcf303a90ea160dafb3eed4edac5906238153c67`.

```
Actual fixed and moving subspaces, restricted symplectic forms and
the restricted moving automorphism.
```

Direct project imports: none.

Line	Kind	Name
13	def	`PaperXI.FixedMoving.fixed`
16	def	`PaperXI.FixedMoving.moving`
18	theorem	`PaperXI.FixedMoving.mem_fixed`
21	theorem	`PaperXI.FixedMoving.mem_moving`
25	theorem	`PaperXI.FixedMoving.orthogonal`
37	theorem	`PaperXI.FixedMoving.orthogonal_reverse`
43	def	`PaperXI.FixedMoving.fixedForm`
46	def	`PaperXI.FixedMoving.movingForm`
48	theorem	`PaperXI.FixedMoving.fixedForm_alt`
51	theorem	`PaperXI.FixedMoving.movingForm_alt`
55	theorem	`PaperXI.FixedMoving.restrict_nondegenerate`
77	theorem	`PaperXI.FixedMoving.movingForm_nondegenerate_of_isCompl`
84	theorem	`PaperXI.FixedMoving.fixedForm_nondegenerate_of_isCompl`
91	theorem	`PaperXI.FixedMoving.moving_stable`
97	theorem	`PaperXI.FixedMoving.moving_inverse_stable`
105	def	`PaperXI.FixedMoving.movingAction`

Line	Kind	Name
113	theorem	PaperXI.FixedMoving.movingAction_coe
116	theorem	PaperXI.FixedMoving.movingAction_symm_coe
119	theorem	PaperXI.FixedMoving.movingAction_preserves
124	theorem	PaperXI.FixedMoving.movingAction_fixedPointFree_of_isCompl
133	theorem	PaperXI.FixedMoving.moving_difference_injective
145	def	PaperXI.FixedMoving.movingDifferenceEquiv
151	theorem	PaperXI.FixedMoving.movingDifferenceEquiv_apply
159	def	PaperXI.FixedMoving.movingEquiv
179	theorem	PaperXI.FixedMoving.movingEquiv_coe
184	theorem	PaperXI.FixedMoving.movingEquiv_intertwines
191	theorem	PaperXI.FixedMoving.movingEquiv_preserves

A.180 PaperXI.FixedMovingSource

Selected layer: v0.44 parent. **Source lines:** 111.

SHA-256: 2591738676c9f4830f6e41ce811376e7b6b2c17f1afe99cfbf94cffebdb7aa22.

```
The source fixed/moving theorem, with the decomposition and restricted
invertibility derived from the actual finite order prime to the characteristic.
```

Direct project imports: PaperXI.FixedMoving, PaperXI.FiniteFieldAverage, PaperXI.OddPhase.

Line	Kind	Name
15	theorem	PaperXI.FixedMoving.isCompl_of_period
19	def	PaperXI.FixedMoving.decomposition
22	theorem	PaperXI.FixedMoving.decomposition_apply
27	theorem	PaperXI.FixedMoving.decomposition_fixed
33	theorem	PaperXI.FixedMoving.decomposition_moving
38	theorem	PaperXI.FixedMoving.movingForm_nondegenerate
44	theorem	PaperXI.FixedMoving.fixedForm_nondegenerate
55	theorem	PaperXI.FixedMoving.source_order_pos
63	theorem	PaperXI.FixedMoving.source_order_ne_zero
68	theorem	PaperXI.FixedMoving.source_isCompl
73	def	PaperXI.FixedMoving.sourceDecomposition
77	theorem	PaperXI.FixedMoving.sourceDecomposition_apply
81	theorem	PaperXI.FixedMoving.source_moving_fixedPointFree
86	def	PaperXI.FixedMoving.sourceMovingDifference
89	theorem	PaperXI.FixedMoving.sourceMovingDifference_apply
95	theorem	PaperXI.FixedMoving.source

A.181 PaperXI.FourierCovariance

Selected layer: v0.44 parent. **Source lines:** 40.

SHA-256: 83b273abc08207d3b23d39152863370e2345bc7a04f422578b880b39ffb6bb6c.

```
The corrected covariance half of Proposition 3.1 under its printed
relation F T(a) F⁻¹ = T(q*a). Idempotence and completeness are not asserted here.
```

Direct project imports: PaperXI.ProjectorTransport.

Line	Kind	Name
13	def	`PaperXI.FourierCovariance.mulPermutation`
19	def	`PaperXI.FourierCovariance.fourierOperator`
25	theorem	`PaperXI.FourierCovariance.inverse_exponent_covariance`

A.182 `PaperXI.FourierPackets`

Selected layer: v0.44 parent. **Source lines:** 50.

SHA-256: `3521693ff5ede2869607d8308db0f63ce2cb6a736c5d728fd43d9b18c942cade`.

```
The ranges of the computed projectors are precisely the declared character
spaces. This identifies the algebraic operators with the paper's tame packets.
```

Direct project imports: `PaperXI.FourierProjectors`.

Line	Kind	Name
13	def	`PaperXI.FourierPackets.packet`
20	theorem	`PaperXI.FourierPackets.projector_mem_packet`
27	theorem	`PaperXI.FourierPackets.projector_on_packet`
36	theorem	`PaperXI.FourierPackets.projector_range`
44	theorem	`PaperXI.FourierPackets.vector_reconstruction`

A.183 `PaperXI.FourierProjectors`

Selected layer: v0.44 parent. **Source lines:** 79.

SHA-256: `02af62a8c8e5753ef7a695629d44c71db407211d18a6bd6b6a48865ba731de52`.

```
Finite Fourier projectors derived from cyclic representation laws and a
primitive additive character. No idempotence or orthogonality is assumed.
```

Direct project imports: `PaperXI.CyclicAction`.

Line	Kind	Name
13	def	`PaperXI.FourierProjectors.projector`
17	theorem	`PaperXI.FourierProjectors.projector_formula`
21	theorem	`PaperXI.FourierProjectors.character_sum`
26	theorem	`PaperXI.FourierProjectors.action_projector`
44	theorem	`PaperXI.FourierProjectors.projector_product`
58	theorem	`PaperXI.FourierProjectors.projector_idempotent`
63	theorem	`PaperXI.FourierProjectors.projector_orthogonal`
68	theorem	`PaperXI.FourierProjectors.projector_complete`

A.184 `PaperXI.FractionMatrix106`

Selected layer: base108 arithmetic branch. **Source lines:** 57.

SHA-256: `37125ed47f602558238d6c6b2e9ef6abc4f83592ef58525b03697009a9ac54b6`.

```
Clear denominators and retain both integral basis changes.
```

Direct project imports: `PaperXI.SmithMatrix106`, `PaperXI.SplitRadical105`.

Line	Kind	Name
17	theorem	`PaperXI.FractionMatrix106.exists_integral_multiple`
38	theorem	`PaperXI.FractionMatrix106.exists_integral_diagonal`

A.185 `PaperXI.FreeLatticeCard110`

Selected layer: Continuation110. **Source lines:** 55.

SHA-256: `07d5f21b13996d68c3d71abd71dab1052fd72ae292beba461aee4f571f6bcd08`.

```
Cardinality of ideal quotients of an actual finite free module.
This supplies the numerical lattice leaf once its corner basis is constructed;
it does not assert that an arbitrary additive lattice has that basis.
```

Direct project imports: none.

Line	Kind	Name
16	theorem	`PaperXI.FreeLatticeCard110.ideal_smul_top_pi`
34	def	`PaperXI.FreeLatticeCard110.quotientEquiv`
41	theorem	`PaperXI.FreeLatticeCard110.card_quotient`
46	theorem	`PaperXI.FreeLatticeCard110.card_quotient_prime_pow`

A.186 `PaperXI.FrobeniusOrbit`

Selected layer: v0.44 parent. **Source lines:** 42.

SHA-256: `7d1f7d09d4d67afe3a6e5721cf265b76d2ff69a13ef63bb6a753133817f2f7c3`.

```
Frobenius orbits with integer powers, including both transport directions.
```

Direct project imports: `PaperXI.PacketTransport`.

Line	Kind	Name
8	def	`PaperXI.FrobeniusOrbit.Related`
11	theorem	`PaperXI.FrobeniusOrbit.refl`
14	theorem	`PaperXI.FrobeniusOrbit.trans`
20	theorem	`PaperXI.FrobeniusOrbit.symm`
26	theorem	`PaperXI.FrobeniusOrbit.step`
29	theorem	`PaperXI.FrobeniusOrbit.back`
32	theorem	`PaperXI.FrobeniusOrbit.step_mem`
35	theorem	`PaperXI.FrobeniusOrbit.back_mem`
38	theorem	`PaperXI.FrobeniusOrbit.same_orbit_iff`

A.187 `PaperXI.FrobeniusPeriod`

Selected layer: v0.44 parent. **Source lines:** 97.

SHA-256: `8e34bfa6b4636a8b392bdff1eb6f25eb5c9061f592ae1bff13c3179e48ec14fb`.

```
Minimal return period, exact finite orbit size, and stabilizing powers.
```

Direct project imports: `PaperXI.PacketSupport`.

Line	Kind	Name
12	def	`PaperXI.FrobeniusPeriod.permutation`
14	def	`PaperXI.FrobeniusPeriod.period`
16	theorem	`PaperXI.FrobeniusPeriod.iterate_label`
28	theorem	`PaperXI.FrobeniusPeriod.period_pos`
31	theorem	`PaperXI.FrobeniusPeriod.period_le`
35	theorem	`PaperXI.FrobeniusPeriod.return_label`
43	theorem	`PaperXI.FrobeniusPeriod.stabilizing_power_iff`
49	theorem	`PaperXI.FrobeniusPeriod.no_short_return`
56	theorem	`PaperXI.FrobeniusPeriod.related_iff_nat`
68	theorem	`PaperXI.FrobeniusPeriod.related_iff_bounded`
79	def	`PaperXI.FrobeniusPeriod.orbitEquiv`
93	theorem	`PaperXI.FrobeniusPeriod.orbit_card`

A.188 `PaperXI.FrobeniusPeriodCount`

Selected layer: v0.44 parent. **Source lines:** 125.

SHA-256: `53f45d0eb9fed4c2ad104695c3730de6b59560120f164630b865c30a3ccb0749`.

```
Exact-period counting for finite dynamical systems.  The count is the
cardinality of an actual subtype, not a recurrence defined to force inversion.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.FrobeniusPeriodCount.exactCount`
18	def	`PaperXI.FrobeniusPeriodCount.fixedCount`
22	lemma	`PaperXI.FrobeniusPeriodCount.exactCount_eq_filter`
28	lemma	`PaperXI.FrobeniusPeriodCount.fixedCount_eq_filter`
34	theorem	`PaperXI.FrobeniusPeriodCount.sum_exactCount`
59	theorem	`PaperXI.FrobeniusPeriodCount.moebius_exactCount`
75	def	`PaperXI.FrobeniusPeriodCount.regularPerm`
80	def	`PaperXI.FrobeniusPeriodCount.orbitCount`
84	theorem	`PaperXI.FrobeniusPeriodCount.regularPerm_minimalPeriod`
96	theorem	`PaperXI.FrobeniusPeriodCount.exactCount_eq_orbitCount_mul`
119	theorem	`PaperXI.FrobeniusPeriodCount.period_dvd_exactCount`

A.189 `PaperXI.FullBridgeTransport`

Selected layer: v0.44 parent. **Source lines:** 50.

SHA-256: `af018ebd737c4d182525a72c4eddc2561a4ca9a2a228aabd0bac83dae1a9d816`.

```
Actual semidirect and carrier transport of a finite bridge. These endpoints
connect the tensor model's normal and stabilizer equations to full operators.
```

Direct project imports: `PaperXI.CyclicBridge`.

Line	Kind	Name
14	theorem	`PaperXI.FullBridgeTransport.semidirect_along`
36	theorem	`PaperXI.FullBridgeTransport.through_factorizations`

A.190 `PaperXI.FullIntertwiningLocus69`

Selected layer: base108 arithmetic branch. **Source lines:** 95.

SHA-256: `d1f43873562d510630801d569413b3ad3b4d659e2b4403e1d74714196488c12a`.

```
Exact convention adapter for the full extended-type intertwining locus.
```

Direct project imports: `PaperXI.InductionIrreducibility`.

Line	Kind	Name
14	def	`PaperXI.FullIntertwiningLocus69.locus`
19	def	`PaperXI.FullIntertwiningLocus69.internalIntertwiner`
30	theorem	`PaperXI.FullIntertwiningLocus69.internalIntertwiner_ne_zero`
41	theorem	`PaperXI.FullIntertwiningLocus69.subgroup_subset`
48	theorem	`PaperXI.FullIntertwiningLocus69.locus_eq_iff`
71	def	`PaperXI.FullIntertwiningLocus69.standardLocus`
75	theorem	`PaperXI.FullIntertwiningLocus69.standardLocus_eq_iff`
88	theorem	`PaperXI.FullIntertwiningLocus69.off_subgroup_of_standard_locus`

A.191 `PaperXI.FullTypeIntertwining105`

Selected layer: base108 arithmetic branch. **Source lines:** 151.

SHA-256: `77dddac0fa33899fcac3fba052d3a5bf97fd3f7c3e9443409e6fd3be1e7f7be6`.

```
From the actual finite cuspidal factor to full extended-type Mackey vanishing.
The BK tensor-to-residue reduction and arithmetic Cartan decomposition remain
explicit inputs. The full-H intertwining bound and induced irreducibility do not.
```

Direct project imports: `PaperXI.CuspidalCartan105`, `PaperXI.FullIntertwiningLocus69`, `PaperXI.CompactModCenterInduction68`.

Line	Kind	Name
20	def	`PaperXI.FullTypeIntertwining105.mapOfAlong`
30	theorem	`PaperXI.FullTypeIntertwining105.alongOfMap`
47	theorem	`PaperXI.FullTypeIntertwining105.existsAt_iff_locus`
63	theorem	`PaperXI.FullTypeIntertwining105.restrict_existsAt`
72	theorem	`PaperXI.FullTypeIntertwining105.extension_irreducible`
92	theorem	`PaperXI.FullTypeIntertwining105.full_locus`
120	theorem	`PaperXI.FullTypeIntertwining105.induction_package`

A.192 `PaperXI.GallagherTensorIrreducibility58`

Selected layer: base108 arithmetic branch. **Source lines:** 165.

SHA-256: `8bf42d852c9fc0f54c67a1dbdfb96f9df652d86138597dd1db0006bcf90e552b`.

```
# Reverse Clifford tensor irreducibility

The prescribed extension factor is tensored with an irreducible quotient
representation. The resulting action is irreducible; no irreducibility of the
tensor product is assumed. The argument uses Jacobson density on the actual
normal-subgroup action.
```

Direct project imports: `PaperXI.SchurScalar`, `PaperXI.RepresentationStability`.

Line	Kind	Name
25	theorem	`PaperXI.GallagherTensorIrreducibility58.algebraAction_surjective`
59	theorem	`PaperXI.GallagherTensorIrreducibility58.stable_tensor_algebra`
73	theorem	`PaperXI.GallagherTensorIrreducibility58.stable_tensor_end`
83	theorem	`PaperXI.GallagherTensorIrreducibility58.map_rankOne`
95	def	`PaperXI.GallagherTensorIrreducibility58.multiplicitySubspace`
104	theorem	`PaperXI.GallagherTensorIrreducibility58.tensor_irreducible`
158	theorem	`PaperXI.GallagherTensorIrreducibility58.quotient_tensor_irreducible`

A.193 `PaperXI.GaussCharacters`

Selected layer: v0.44 parent. **Source lines:** 67.

SHA-256: `f49f317f384804ecff8614197f0acd360d1575046498c320b2027bd7ba16d8ca`.

```
Translation to the manuscript's multiplicative characters kˣ → ℂˣ.
No additional unitarity assumption is needed over a finite field.
```

Direct project imports: `PaperXI.GaussSums`.

Line	Kind	Name
12	def	`PaperXI.GaussCharacters.sum`
16	theorem	`PaperXI.GaussCharacters.sum_eq_unitSum`
22	theorem	`PaperXI.GaussCharacters.extension_ne_one`
33	theorem	`PaperXI.GaussCharacters.extension_inverse`
38	theorem	`PaperXI.GaussCharacters.magnitude`
45	def	`PaperXI.GaussCharacters.phase`
49	theorem	`PaperXI.GaussCharacters.phase_eq`
54	theorem	`PaperXI.GaussCharacters.phase_norm`
60	theorem	`PaperXI.GaussCharacters.phase_duality`

A.194 `PaperXI.GaussSums`

Selected layer: v0.44 parent. **Source lines:** 90.

SHA-256: `ace300219a85a3337ec026bb367507f5088f887306c309e17af938fdfca02d02`.

```
Finite-field Gauss sums with the manuscript's sum over the unit group.
The additive character is explicitly required to be nontrivial.
```

Direct project imports: none.

Line	Kind	Name
16	def	`PaperXI.GaussSums.unitSum`
20	theorem	`PaperXI.GaussSums.unitSum_eq_gaussSum`
28	theorem	`PaperXI.GaussSums.sum_mul_inverse`
35	theorem	`PaperXI.GaussSums.conjugate_sum`
40	theorem	`PaperXI.GaussSums.norm_sq`
50	theorem	`PaperXI.GaussSums.magnitude`
56	theorem	`PaperXI.GaussSums.sum_duality`
65	def	`PaperXI.GaussSums.phase`
69	theorem	`PaperXI.GaussSums.sqrt_card_pos`
73	theorem	`PaperXI.GaussSums.phase_norm`
80	theorem	`PaperXI.GaussSums.phase_duality`

A.195 `PaperXI.GeneralChain`

Selected layer: v0.44 parent. **Source lines:** 89.

SHA-256: `0e5a7f76765c2e3b753f9db36e3886906a9eac283a0583ebb7f07bcb404e33d7`.

```
Proposition 2.3 for any operator satisfying the stated block support.
The transfer model is constructed from the operator; it is not an extra hypothesis.
```

Direct project imports: `PaperXI.MonodromyChain`.

Line	Kind	Name
12	def	`PaperXI.MonodromyChain.ChainSupported`
15	theorem	`PaperXI.MonodromyChain.arbitrary_block_reconstruction`
29	theorem	`PaperXI.MonodromyChain.projector_eq_single`
36	def	`PaperXI.MonodromyChain.extractedTransfers`
42	theorem	`PaperXI.MonodromyChain.coordinate_block_zero`
49	theorem	`PaperXI.MonodromyChain.eq_extracted_shift`
75	theorem	`PaperXI.MonodromyChain.arbitrary_nilpotent_bound`
80	theorem	`PaperXI.MonodromyChain.arbitrary_power_paths`

A.196 `PaperXI.GeneratedIrreducibility`

Selected layer: v0.44 parent. **Source lines:** 55.

SHA-256: `3aca9c09ed2ae98d3df1627466cdadb860be6807be1e7555d2743c1b534bd5b2`.

```
Pass from stability under explicit group generators to Mathlib's actual
representation irreducibility. No orbit-transitivity assumption occurs here.
```

Direct project imports: `PaperXI.CyclicAction`.

Line	Kind	Name
11	theorem	`PaperXI.GeneratedIrreducibility.nontrivial_space`
20	def	`PaperXI.GeneratedIrreducibility.actionEquiv`
26	theorem	`PaperXI.GeneratedIrreducibility.actionEquiv_apply`
29	theorem	`PaperXI.GeneratedIrreducibility.actionEquiv_symm_apply`
32	theorem	`PaperXI.GeneratedIrreducibility.stable_of_generators`

Line	Kind	Name
47	theorem	PaperXI.GeneratedIrreducibility.eq_bot_or_top

A.197 PaperXI.GlobalHeisenbergBridge112

Selected layer: Continuation112. **Source lines:** 301.

SHA-256: b44f5fc5e576c356e04e33a27cfbcd1d0f70946a32dafba8389de23b83d83998.

A literal global central lift produces the Heisenberg representation bridge; the target representation equivalence is a conclusion. The actual quotient endpoint retains the original theta and eta on its original carrier. Existence of the canonical global quotient lift is not asserted here.

Direct project imports: PaperXI.HeisenbergCharacterProfile111, PaperXI.FiniteRepresentation, PaperXI.TypeSideDescent, PaperXI.SchurScalar.

Line	Kind	Name
23	theorem	PaperXI.GlobalHeisenbergBridge112.equiv_of_common_profile
54	theorem	PaperXI.GlobalHeisenbergBridge112.equiv_of_common_quotient_form
72	theorem	PaperXI.GlobalHeisenbergBridge112.equiv_of_global_central_lift
105	theorem	PaperXI.GlobalHeisenbergBridge112.actual_eta_bridge_of_global_lift
146	theorem	PaperXI.GlobalHeisenbergBridge112.generator_discrepancy
168	theorem	PaperXI.GlobalHeisenbergBridge112.generator_discrepancy_pow
179	theorem	PaperXI.GlobalHeisenbergBridge112.unit_eq_one_of_coprime_powers
188	theorem	PaperXI.GlobalHeisenbergBridge112.generator_phase_eq_one_of_det_coprime
224	theorem	PaperXI.GlobalHeisenbergBridge112.generator_equivariant_of_det_coprime
247	theorem	PaperXI.GlobalHeisenbergBridge112.cyclic_equivariant_of_generator
268	theorem	PaperXI.GlobalHeisenbergBridge112.equivariant_bridge_of_common_quotient_form

A.198 PaperXI.GluedPacketSections

Selected layer: v0.44 parent. **Source lines:** 170.

SHA-256: 5bf8047d59226ed80f93c4668b031aedab1df38bd5ee39456a9335f2685372bf.

Actual linear spaces of compatible sections, their representation, and the map obtained from root holonomy tests. The final section equivalence is built; a compatible family of comparison maps is not part of the input.

Direct project imports: PaperXI.RepresentationGraph.

Line	Kind	Name
22	def	PaperXI.GluedPacketSections.sections
28	theorem	PaperXI.GluedPacketSections.section_edge
37	theorem	PaperXI.GluedPacketSections.section_path
46	theorem	PaperXI.GluedPacketSections.root_evaluation_injective
56	def	PaperXI.GluedPacketSections.loopFixed
63	def	PaperXI.GluedPacketSections.extendRoot
71	def	PaperXI.GluedPacketSections.evaluationEquiv
90	def	PaperXI.GluedPacketSections.sectionRepresentation
101	theorem	PaperXI.GluedPacketSections.sectionRepresentation_apply

Line	Kind	Name
106	def	PaperXI.GluedPacketSections.ofGauge
123	def	PaperXI.GluedPacketSections.equivOfGauge
134	def	PaperXI.GluedPacketSections.gluedGauge
138	def	PaperXI.GluedPacketSections.fromChords
152	theorem	PaperXI.GluedPacketSections.fromChords_apply
163	theorem	PaperXI.GluedPacketSections.gauge_unique_at_root

A.199 PaperXI.GraphChordCharacters

Selected layer: v0.44 parent. **Source lines:** 144.

SHA-256: 037a0bd69fdb62bbe05ce9760a8f4bcf242c6c01d167c45f6d9bf591cf5bcca0.

```
# Independent scalar tests on fundamental graph cycles

Every assignment of unit scalars to the chords of the constructed rooted tree
is realized by actual one-dimensional edge transport: tree edges carry one and
each chord carries its prescribed scalar. Thus the scalar cycle tests can vary
independently. The same realization gives characters on the actual reduced loop
group and its abelianization.
```

Direct project imports: PaperXI.GraphTree, PaperXI.GraphLoopRepresentation, PaperXI.ScalarHolonomy.

Line	Kind	Name
24	def	PaperXI.GraphChordCharacters.weight
29	theorem	PaperXI.GraphChordCharacters.weight_tree
33	theorem	PaperXI.GraphChordCharacters.weight_chord
39	def	PaperXI.GraphChordCharacters.edgeSystem
43	theorem	PaperXI.GraphChordCharacters.edgeSystem_chord
49	theorem	PaperXI.GraphChordCharacters.traversal_tree
66	theorem	PaperXI.GraphChordCharacters.rootPath_transport
81	def	PaperXI.GraphChordCharacters.chordLoop
87	theorem	PaperXI.GraphChordCharacters.chordLoop_transport
94	theorem	PaperXI.GraphChordCharacters.chordLoop_holonomy
101	theorem	PaperXI.GraphChordCharacters.field_equiv_scalar
111	theorem	PaperXI.GraphChordCharacters.holonomy_isScalar
117	def	PaperXI.GraphChordCharacters.character
121	theorem	PaperXI.GraphChordCharacters.character_chord
128	theorem	PaperXI.GraphChordCharacters.character_tests_surjective
135	theorem	PaperXI.GraphChordCharacters.homology_tests_surjective

A.200 PaperXI.GraphGauge

Selected layer: v0.44 parent. **Source lines:** 190.

SHA-256: 1750d323d264d4760717c62ab3aa154cbebefcdd24e80cc76ef22d7d8f5a572b.

```
Actual graph gauges and the rooted matrix holonomy criterion, Paper XI 14.2–14.3.
The cycle tests are the based loops determined by a constructed rooted tree.
```

Direct project imports: PaperXI.GraphTransport, PaperXI.GraphTree, PaperXI.GraphLoopRepresentation, PaperXI.HolonomyResidual.

Line	Kind	Name
27	def	PaperXI.GraphGauge.cycle
31	def	PaperXI.GraphGauge.rootGauge
35	def	PaperXI.GraphGauge.EdgeCompatible
39	def	PaperXI.GraphGauge.LoopCompatible
43	def	PaperXI.GraphGauge.CycleCompatible
47	theorem	PaperXI.GraphGauge.rootGauge_root
53	theorem	PaperXI.GraphGauge.rootGauge_edge_iff
66	theorem	PaperXI.GraphGauge.edgeCompatible_loops
72	theorem	PaperXI.GraphGauge.rootGauge_compatible_of_loops
79	theorem	PaperXI.GraphGauge.gauge_unique
88	theorem	PaperXI.GraphGauge.based_iff_loops
101	theorem	PaperXI.GraphGauge.rootGauge_parent
110	theorem	PaperXI.GraphGauge.rootGauge_tree
127	theorem	PaperXI.GraphGauge.rootGauge_compatible_iff_chords
141	theorem	PaperXI.GraphGauge.based_iff_chords
155	theorem	PaperXI.GraphGauge.connected_based_iff_chords
163	theorem	PaperXI.GraphGauge.chords_iff_loops
170	theorem	PaperXI.GraphGauge.tree_independence
177	theorem	PaperXI.GraphGauge.unbased_iff_conjugates

A.201 PaperXI.GraphLoop

Selected layer: v0.44 parent. **Source lines:** 106.

SHA-256: f38f6d71925623d713432f41101158265cdfa0733f6bd3242f335f62bf01d198.

```
# Reduced graph loops and their abelianization

Edges belong to the original quiver. A path in its symmetrification can traverse
an edge in either direction. The free groupoid identifies paths under insertion
and deletion of an edge followed by its reverse. Based endomorphisms therefore
give the combinatorial fundamental group, with multiplication ordered as function
composition: `x * y` traverses `y` before `x`.

`H1` is the abelianization of this actual path quotient. No comparison with a
topological realization or singular homology is asserted here.
```

Direct project imports: none.

Line	Kind	Name
27	abbrev	PaperXI.GraphLoop.vertex
31	def	PaperXI.GraphLoop.pathClass
35	theorem	PaperXI.GraphLoop.pathClass_nil
38	theorem	PaperXI.GraphLoop.pathClass_comp
43	theorem	PaperXI.GraphLoop.pathClass_reverse
48	theorem	PaperXI.GraphLoop.pathClass_surjective
53	abbrev	PaperXI.GraphLoop.Loop
56	def	PaperXI.GraphLoop.loopClass
59	theorem	PaperXI.GraphLoop.loopClass_nil
63	theorem	PaperXI.GraphLoop.loopClass_comp
67	theorem	PaperXI.GraphLoop.loopClass_reverse
71	theorem	PaperXI.GraphLoop.loopClass_surjective
75	abbrev	PaperXI.GraphLoop.H1

Line	Kind	Name
78	def	`PaperXI.GraphLoop.homologyClass`
80	theorem	`PaperXI.GraphLoop.homologyClass_mul`
84	def	`PaperXI.GraphLoop.factorCharacter`
87	theorem	`PaperXI.GraphLoop.factorCharacter_apply_class`
92	theorem	`PaperXI.GraphLoop.factorCharacter_unique`
99	theorem	`PaperXI.GraphLoop.character_ext_paths`

A.202 `PaperXI.GraphLoopRepresentation`

Selected layer: v0.44 parent. **Source lines:** 138.

SHA-256: `8e6bb58b55ba016228d5ae3cd85b19e40ccde0abd116f4a13101a8d8939a4e95`.

```
# Holonomy as a homomorphism on actual reduced graph loops

The free-groupoid extension of the edge system agrees with the recursively
defined path transport on every represented path. Surjectivity of path classes
then identifies intertwining all based paths with conjugacy of the resulting
honest group homomorphisms.
```

Direct project imports: `PaperXI.GraphLoop`, `PaperXI.GraphTransport`.

Line	Kind	Name
25	structure	`PaperXI.GraphLoop.Fiber`
28	instance	`PaperXI.GraphLoop.fiberGroupoid`
40	def	`PaperXI.GraphLoop.edgePrefunctor`
46	def	`PaperXI.GraphLoop.transportFunctor`
51	def	`PaperXI.GraphLoop.transport`
55	theorem	`PaperXI.GraphLoop.transport_id`
60	theorem	`PaperXI.GraphLoop.transport_comp`
66	theorem	`PaperXI.GraphLoop.transport_edge`
73	theorem	`PaperXI.GraphLoop.transport_pathClass`
90	def	`PaperXI.GraphLoop.holonomy`
96	theorem	`PaperXI.GraphLoop.holonomy_loopClass`
105	def	`PaperXI.GraphLoop.Conjugates`
112	theorem	`PaperXI.GraphLoop.conjugates_iff_paths`
129	theorem	`PaperXI.GraphLoop.exists_conjugates_iff_paths`

A.203 `PaperXI.GraphResidual`

Selected layer: v0.44 parent. **Source lines:** 274.

SHA-256: `ffc35ff3cd92cf45144e94c501a36af8567e4bb7531e9dc21eb81401385adbf8`.

```
Paper XI, 14.6 and 14.7 on the actual rooted graph paths. Finite dimensionality
makes the given edge equivalences and their actual inverses continuous for the
specified norms. Parallel edges and self-loops keep their identities.
```

Direct project imports: `PaperXI.ContinuousHolonomyResidual`, `PaperXI.GraphTransport`, `PaperXI.GraphTree`.

Line	Kind	Name
24	def	PaperXI.GraphResidual.pathL
28	theorem	PaperXI.GraphResidual.pathL_toLinearEquiv
33	theorem	PaperXI.GraphResidual.pathL_apply
38	theorem	PaperXI.GraphResidual.pathL_symm_apply
43	theorem	PaperXI.GraphResidual.pathL_reverse
50	theorem	PaperXI.GraphResidual.pathL_comp
60	def	PaperXI.GraphResidual.gauge
67	theorem	PaperXI.GraphResidual.gauge_toLinearEquiv
74	theorem	PaperXI.GraphResidual.gauge_apply
81	theorem	PaperXI.GraphResidual.gauge_root
90	theorem	PaperXI.GraphResidual.gauge_parent
106	def	PaperXI.GraphResidual.edgeResidual
116	def	PaperXI.GraphResidual.cycleDefect
125	theorem	PaperXI.GraphResidual.cycleDefect_toLinearMap
134	theorem	PaperXI.GraphResidual.edgeResidual_toLinearMap
144	theorem	PaperXI.GraphResidual.cycleDefect_loop
162	theorem	PaperXI.GraphResidual.residual_factorization
173	theorem	PaperXI.GraphResidual.residual_zero_iff
184	theorem	PaperXI.GraphResidual.residual_norm_bound
196	theorem	PaperXI.GraphResidual.uniform_bound
212	theorem	PaperXI.GraphResidual.uniform_product_bound
228	theorem	PaperXI.GraphResidual.uniform_chord_bound
246	theorem	PaperXI.GraphResidual.uniform_chord_product_bound
264	theorem	PaperXI.GraphResidual.connected_residual_bound

A.204 PaperXI.GraphTransport

Selected layer: v0.44 parent. **Source lines:** 150.

SHA-256: `72c2feeee0c9f9e98eeed01a1ff511d439bdfdac6c3bdb2732290dcc1f8151fb`.

```
Paper XI, Section 14. Actual typed path transport on the symmetrification
of a quiver. An original arrow is one chosen orientation of an underlying edge;
parallel arrows and self-loops keep their identities. Concatenation traverses
its left argument first.
```

Direct project imports: none.

Line	Kind	Name
17	abbrev	PaperXI.GraphTransport.Traversal
20	abbrev	PaperXI.GraphTransport.Walk
25	def	PaperXI.GraphTransport.symmetrifyAdd
29	def	PaperXI.GraphTransport.symmetrifyModule
33	def	PaperXI.GraphTransport.edge
39	theorem	PaperXI.GraphTransport.edge_inl
42	theorem	PaperXI.GraphTransport.edge_inr
45	theorem	PaperXI.GraphTransport.edge_reverse
51	def	PaperXI.GraphTransport.path
56	theorem	PaperXI.GraphTransport.nil
59	theorem	PaperXI.GraphTransport.cons
63	theorem	PaperXI.GraphTransport.toPath

Line	Kind	Name
68	theorem	`PaperXI.GraphTransport.comp`
77	theorem	`PaperXI.GraphTransport.reverse`
85	theorem	`PaperXI.GraphTransport.path_reverse_cancel`
92	theorem	`PaperXI.GraphTransport.reverse_path_cancel`
100	theorem	`PaperXI.GraphTransport.cancel`
110	theorem	`PaperXI.GraphTransport.gauge_edge`
125	theorem	`PaperXI.GraphTransport.gauge_path`
139	theorem	`PaperXI.GraphTransport.gauge_path_equiv`

A.205 `PaperXI.GraphTree`

Selected layer: v0.44 parent. **Source lines:** 245.

SHA-256: `37a1cea3da4966b2d631d1f44a997095037c88ca15a4b3001a6b8efa607dd486`.

```
A rooted spanning tree of a connected quiver, with the identities of the
original edges retained. The construction selects a predecessor along a
shortest path in the symmetrified quiver.
```

Direct project imports: none.

Line	Kind	Name
18	abbrev	`PaperXI.GraphTree.Edge`
21	abbrev	`PaperXI.GraphTree.Traversal`
24	abbrev	`PaperXI.GraphTree.Path`
27	def	`PaperXI.GraphTree.Connected`
30	structure	`PaperXI.GraphTree.Rooted`
40	theorem	`PaperXI.GraphTree.exists_length`
45	def	`PaperXI.GraphTree.distance`
49	theorem	`PaperXI.GraphTree.distance_le`
53	theorem	`PaperXI.GraphTree.distance_root`
57	theorem	`PaperXI.GraphTree.exists_predecessor`
71	def	`PaperXI.GraphTree.ofConnected`
83	def	`PaperXI.GraphTree.rootPath`
92	theorem	`PaperXI.GraphTree.rootPath_root`
96	theorem	`PaperXI.GraphTree.rootPath_parent`
102	theorem	`PaperXI.GraphTree.rooted_induction`
112	def	`PaperXI.GraphTree.underlying`
118	def	`PaperXI.GraphTree.selected`
120	theorem	`PaperXI.GraphTree.underlying_endpoints`
125	theorem	`PaperXI.GraphTree.selected_injective`
152	def	`PaperXI.GraphTree.treeEdges`
155	def	`PaperXI.GraphTree.Chord`
157	theorem	`PaperXI.GraphTree.mem_treeEdges`
160	theorem	`PaperXI.GraphTree.treeEdges_induction`
167	theorem	`PaperXI.GraphTree.selected_not_loop`
183	def	`PaperXI.GraphTree.Supported`
188	theorem	`PaperXI.GraphTree.rootPath_supported`
196	theorem	`PaperXI.GraphTree.treeEdges_no_loop`
200	theorem	`PaperXI.GraphTree.tree_card`
211	theorem	`PaperXI.GraphTree.chord_count`

Line	Kind	Name
226	theorem	`PaperXI.GraphTree.edge_count_lower_bound`
233	theorem	`PaperXI.GraphTree.chord_card`
239	theorem	`PaperXI.GraphTree.chord_card_int`

A.206 `PaperXI.GraphTreePaths`

Selected layer: v0.44 parent. **Source lines:** 87.

SHA-256: `732095eebf2b794f7d31d503a7928bfde9b7d40da777fd5c99b3d1cccd1e2e18`.

```
Actual tree paths intertwine every reversible system of edge transports.
Tree membership refers to the chosen original edge identity.
```

Direct project imports: `PaperXI.GraphTree`, `PaperXI.GraphTransport`.

Line	Kind	Name
20	theorem	`PaperXI.GraphTreePaths.parent_transport`
27	theorem	`PaperXI.GraphTreePaths.tree_transport`
55	theorem	`PaperXI.GraphTreePaths.traversal_transport`
70	theorem	`PaperXI.GraphTreePaths.supported_transport`
82	theorem	`PaperXI.GraphTreePaths.tree_loop_identity`

A.207 `PaperXI.GroupFixed`

Selected layer: v0.44 parent. **Source lines:** 85.

SHA-256: `32a4e3c21abaf0e27169d70976a894f79172987de4954913d8fc607fd925e51b`.

```
Fixedness under an actual group, including supplied inertia quotient data.
```

Direct project imports: `PaperXI.CycleFixed`, `PaperXI.InertiaQuotient`.

Line	Kind	Name
12	theorem	`PaperXI.GroupFixed.fixed_of_generators`
22	theorem	`PaperXI.GroupFixed.invariants_eq_common`
46	theorem	`PaperXI.GroupFixed.orbit_basis_steps`
63	theorem	`PaperXI.GroupFixed.packet_finrank`

A.208 `PaperXI.HeisenbergCardinality`

Selected layer: v0.44 parent. **Source lines:** 65.

SHA-256: `3325cc5f2651ba031ec6ab41f813ec4f9e63afea875a3e4cac9827997fef3225`.

```
Dimension and cardinality of a maximal isotropic subspace in the standard
Heisenberg symplectic space. No cardinality or self-orthogonality conclusion is
assumed as an input.
```

Direct project imports: `PaperXI.HeisenbergForm`, `PaperXI.HeisenbergGeometry`.

Line	Kind	Name
17	theorem	`PaperXI.HeisenbergCardinality.form_isAlt`
19	theorem	`PaperXI.HeisenbergCardinality.form_separatingLeft`
29	theorem	`PaperXI.HeisenbergCardinality.form_nondegenerate`
37	theorem	`PaperXI.HeisenbergCardinality.form_forall_eq_zero_iff`
41	theorem	`PaperXI.HeisenbergCardinality.maximal_orthogonal_eq`
48	theorem	`PaperXI.HeisenbergCardinality.finrank_eq`
54	theorem	`PaperXI.HeisenbergCardinality.natCard_eq_pow`
59	theorem	`PaperXI.HeisenbergCardinality.natCard_eq`

A.209 `PaperXI.HeisenbergCenter`

Selected layer: v0.44 parent. **Source lines:** 64.

SHA-256: `f2e8ae8af020e8557e7b08bb957b7145733776c6b5c83c37218c1e0e184f4966`.

```
The center is proved to consist exactly of the central-coordinate elements.
```

Direct project imports: `PaperXI.HeisenbergGroup`, `PaperXI.HeisenbergCardinality`.

Line	Kind	Name
15	theorem	`PaperXI.HeisenbergCenter.commute_iff`
35	theorem	`PaperXI.HeisenbergCenter.mem_center_iff`
51	theorem	`PaperXI.HeisenbergCenter.center_eq_centralElement`

A.210 `PaperXI.HeisenbergCharacterProfile111`

Selected layer: Continuation111. **Source lines:** 317.

SHA-256: `1b222ac6ed229dac1c3c2682d5340cc880a2699c3c0ca3c9cbc80ae11bcb712e`.

```
The original type-side quotient/form datum derives trace support through
actual commutators. Character intertwining of the original one-dimensional
theta is equivalent to its covariance on the literal intersection. Neither
result assumes the desired eta intertwiner or eta intersection dimension.
```

Direct project imports: `PaperXI.DegreeIndex110`.

Line	Kind	Name
21	def	`PaperXI.HeisenbergCharacterProfile111.coordinate`
24	theorem	`PaperXI.HeisenbergCharacterProfile111.coordinate_surjective`
31	theorem	`PaperXI.HeisenbergCharacterProfile111.coordinate_eq_zero_iff`
44	theorem	`PaperXI.HeisenbergCharacterProfile111.commutator_mem`
52	def	`PaperXI.HeisenbergCharacterProfile111.characterCommutator`
59	theorem	`PaperXI.HeisenbergCharacterProfile111.commutator_detection_of_nondegenerate_form`
94	theorem	`PaperXI.HeisenbergCharacterProfile111.trace_support_of_nondegenerate_form`
116	def	`PaperXI.HeisenbergCharacterProfile111.transportedQuotient`
120	theorem	`PaperXI.HeisenbergCharacterProfile111.coordinate_transport`
124	def	`PaperXI.HeisenbergCharacterProfile111.pulledBackCharacter`
128	theorem	`PaperXI.HeisenbergCharacterProfile111.characterCommutator_transport`

Line	Kind	Name
140	theorem	`PaperXI.HeisenbergCharacterProfile111.form_transport`
160	theorem	`PaperXI.HeisenbergCharacterProfile111.character_intertwinedAt_iff_covariance`
187	theorem	`PaperXI.HeisenbergCharacterProfile111.character_values_of_intertwinedAt`
216	instance	`PaperXI.HeisenbergCharacterProfile111.actualScalarSubgroup_normal`
237	theorem	`PaperXI.HeisenbergCharacterProfile111.actual_trace_support_of_quotient_form`
261	theorem	`PaperXI.HeisenbergCharacterProfile111.actual_covariance_of_character_intertwining`
283	theorem	`PaperXI.HeisenbergCharacterProfile111.actual_eta_dimension_from_quotient_form`

A.211 `PaperXI.HeisenbergCochain`

Selected layer: v0.44 parent. **Source lines:** 69.

SHA-256: `ba7c5f4ea39d1e68eadd6f852cf159e67a2b0c454eb6083d92f7d48ca9e92f8f`.

```
Actual changes of central section and the square obstruction in characteristic
two. These statements concern the middle groups, not just their commutator forms.
```

Direct project imports: `PaperXI.BilinearHeisenberg`.

Line	Kind	Name
14	def	`PaperXI.HeisenbergCochain.equiv`
28	theorem	`PaperXI.HeisenbergCochain.equiv_center`
33	theorem	`PaperXI.HeisenbergCochain.equiv_vector`
41	theorem	`PaperXI.HeisenbergCochain.square`
51	theorem	`PaperXI.HeisenbergCochain.diagonal_preserved`
62	theorem	`PaperXI.HeisenbergCochain.no_equiv_of_diagonal_mismatch`

A.212 `PaperXI.HeisenbergComparison`

Selected layer: v0.44 parent. **Source lines:** 92.

SHA-256: `545124ccd81c40a4378e2d854940a0ddb3402c6081775d92751fa13fa0a7f96b`.

```
The first actual intertwiner between arbitrary irreducible actions along
a group equivalence from the odd-characteristic coordinate Heisenberg group.
The coordinate isometry specialization constructs that group equivalence.
Abstract symplectic coordinates, source quotient actions, and the dyadic
extension branch are not supplied by this module.
```

Direct project imports: `PaperXI.HeisenbergTransport`, `PaperXI.SchrodingerUniversal`.

Line	Kind	Name
21	def	`PaperXI.HeisenbergComparison.equivAlong`
32	theorem	`PaperXI.HeisenbergComparison.equivAlong_intertwines`
46	theorem	`PaperXI.HeisenbergComparison.exists_unique_along`
75	theorem	`PaperXI.HeisenbergComparison.exists_unique_of_isometry`

A.213 `PaperXI.HeisenbergDimension`

Selected layer: v0.44 parent. **Source lines:** 41.

SHA-256: `08109693bdd53c46678bff188224863dda5f276ce70dee638238a08c28317a02`.

Dimension and square-root cardinality in the finite Schrödinger model.

Direct project imports: `PaperXI.HeisenbergCardinality`.

Line	Kind	Name
16	theorem	`PaperXI.HeisenbergDimension.natCard_dual`
21	theorem	`PaperXI.HeisenbergDimension.natCard_symplectic`
24	theorem	`PaperXI.HeisenbergDimension.sqrt_natCard_symplectic`
31	theorem	`PaperXI.HeisenbergDimension.finrank_function`
34	theorem	`PaperXI.HeisenbergDimension.finrank_function_pow`
37	theorem	`PaperXI.HeisenbergDimension.finrank_function_sqrt`

A.214 `PaperXI.HeisenbergFinite`

Selected layer: v0.44 parent. **Source lines:** 32.

SHA-256: `d3703379c8db33d8675a342dc898f778aa75b1599529098f11b2af1df156e553`.

Finiteness and order of the actual Heisenberg group.

Direct project imports: `PaperXI.HeisenbergGroup`, `PaperXI.HeisenbergDimension`.

Line	Kind	Name
16	def	`PaperXI.HeisenbergFinite.coordinates`
22	instance	`PaperXI.HeisenbergFinite.instFinite`
27	theorem	`PaperXI.HeisenbergFinite.natCard_eq`

A.215 `PaperXI.HeisenbergForm`

Selected layer: v0.44 parent. **Source lines:** 50.

SHA-256: `b9d055bf58ee5326c45c49e9267dcd9b113d6b1d2b3f0f82d5e7cd909890230c`.

The manuscript's standard alternating form on X plus its linear dual.

Direct project imports: none.

Line	Kind	Name
13	def	`PaperXI.HeisenbergForm.form`
41	theorem	`PaperXI.HeisenbergForm.form_apply`
44	theorem	`PaperXI.HeisenbergForm.form_self`
47	theorem	`PaperXI.HeisenbergForm.form_skew`

A.216 `PaperXI.HeisenbergFrequency`

Selected layer: v0.44 parent. **Source lines:** 197.

SHA-256: `511658b8ac1ca7f93b8002ee2a626418b076f2cd34875bd57bbad47a4deb6b59`.

Frequency projectors in an arbitrary representation of the actual Heisenberg group.
Completeness and translation produce a nonzero modulation-fixed vector without any
assumption on the dimension of the target representation.

Direct project imports: `PaperXI.DualCharacterSums`, `PaperXI.HeisenbergNormalForm`, `PaperXI.HeisenbergRepresentation`.

Line	Kind	Name
19	def	`PaperXI.HeisenbergFrequency.projector`
24	theorem	`PaperXI.HeisenbergFrequency.projector_apply`
31	theorem	`PaperXI.HeisenbergFrequency.eigen`
63	theorem	`PaperXI.HeisenbergFrequency.complete`
92	theorem	`PaperXI.HeisenbergFrequency.projector_of_eigen`
114	theorem	`PaperXI.HeisenbergFrequency.orthogonal_idempotents`
127	theorem	`PaperXI.HeisenbergFrequency.modulation_translation_action`
147	theorem	`PaperXI.HeisenbergFrequency.translation_covariance`
169	theorem	`PaperXI.HeisenbergFrequency.nonzero_zero_fiber`

A.217 `PaperXI.HeisenbergGeometry`

Selected layer: v0.44 parent. **Source lines:** 82.

SHA-256: `df27cc30cbdcf1f4f8ac3ac85c90e2266dcc866bcd81ea3fbdc18d9ac3022c7c`.

```
# Maximal isotropic subspaces

Maximality is by inclusion among subspaces on which the alternating form vanishes.
The equality with the orthogonal complement is derived from that hypothesis.
```

Direct project imports: none.

Line	Kind	Name
21	def	`PaperXI.HeisenbergGeometry.IsIsotropic`
25	def	`PaperXI.HeisenbergGeometry.IsMaximalIsotropic`
28	theorem	`PaperXI.HeisenbergGeometry.isotropic_iff_le_orthogonal`
37	theorem	`PaperXI.HeisenbergGeometry.isotropic_sup_span`
50	theorem	`PaperXI.HeisenbergGeometry.orthogonal_eq`
60	theorem	`PaperXI.HeisenbergGeometry.maximal_iff_orthogonal_eq`
72	theorem	`PaperXI.HeisenbergGeometry.two_mul_finrank`

A.218 `PaperXI.HeisenbergGroup`

Selected layer: v0.44 parent. **Source lines:** 109.

SHA-256: `1a146672c602d8cacb7184a8bf364365676d1766a0ffd7af7a1e1c56652776dd`.

```
The Heisenberg group in position, frequency, and central coordinates.
The alternating bilinear cocycle defines a group in every characteristic.
Identifying its commutator with the symplectic pairing additionally needs 2 ≠ 0.
```

Direct project imports: none.

Line	Kind	Name
12	structure	`PaperXI.Heisenberg.Element`
19	instance	`PaperXI.Heisenberg.instMul`
23	instance	`PaperXI.Heisenberg.instOne`

Line	Kind	Name
26	instance	`PaperXI.Heisenberg.instInv`
29	theorem	`PaperXI.Heisenberg.mul_position`
32	theorem	`PaperXI.Heisenberg.mul_frequency`
35	theorem	`PaperXI.Heisenberg.mul_central`
39	theorem	`PaperXI.Heisenberg.one_position`
40	theorem	`PaperXI.Heisenberg.one_frequency`
41	theorem	`PaperXI.Heisenberg.one_central`
42	theorem	`PaperXI.Heisenberg.inv_position`
43	theorem	`PaperXI.Heisenberg.inv_frequency`
44	theorem	`PaperXI.Heisenberg.inv_central`
47	instance	`PaperXI.Heisenberg.instGroup`
59	def	`PaperXI.Heisenberg.centralElement`
62	def	`PaperXI.Heisenberg.translationElement`
65	def	`PaperXI.Heisenberg.modulationElement`
67	theorem	`PaperXI.Heisenberg.centralElement_position`
69	theorem	`PaperXI.Heisenberg.centralElement_frequency`
71	theorem	`PaperXI.Heisenberg.centralElement_central`
73	theorem	`PaperXI.Heisenberg.translationElement_position`
75	theorem	`PaperXI.Heisenberg.translationElement_frequency`
77	theorem	`PaperXI.Heisenberg.translationElement_central`
79	theorem	`PaperXI.Heisenberg.modulationElement_position`
81	theorem	`PaperXI.Heisenberg.modulationElement_frequency`
83	theorem	`PaperXI.Heisenberg.modulationElement_central`
86	theorem	`PaperXI.Heisenberg.centralElement_zero`
87	theorem	`PaperXI.Heisenberg.translationElement_zero`
88	theorem	`PaperXI.Heisenberg.modulationElement_zero`
91	theorem	`PaperXI.Heisenberg.centralElement_mul`
95	theorem	`PaperXI.Heisenberg.translationElement_mul`
99	theorem	`PaperXI.Heisenberg.modulationElement_mul`
104	theorem	`PaperXI.Heisenberg.centralElement_commute`

A.219 `PaperXI.HeisenbergNormalForm`

Selected layer: v0.44 parent. **Source lines:** 33.

SHA-256: `61b84ce5c3a491de98e96f9f6edee89760c3d6516081c9372b8cc2d4cb98c1b2`.

Normal forms and signed commutation in the actual coordinate Heisenberg group.

Direct project imports: `PaperXI.HeisenbergGroup`.

Line	Kind	Name
10	theorem	`PaperXI.Heisenberg.normal_form`
16	theorem	`PaperXI.Heisenberg.modulation_translation`
25	theorem	`PaperXI.Heisenberg.translation_modulation`

A.220 `PaperXI.HeisenbergPrime`

Selected layer: v0.44 parent. **Source lines:** 29.

SHA-256: `5f481efb90ff5aa531e502311d9145cf59b1546638b05905f8dbce01b4be4dc9`.

```
The odd-prime specialization uses exactly the manuscript's exponential
additive character, with nontriviality and invertibility of two proved.
```

Direct project imports: `PaperXI.SchrodingerOperator`.

Line	Kind	Name
12	theorem	`PaperXI.HeisenbergPrime.two_ne_zero`
17	theorem	`PaperXI.HeisenbergPrime.character_nontrivial`
24	theorem	`PaperXI.HeisenbergPrime.character_formula`

A.221 `PaperXI.HeisenbergProjector`

Selected layer: v0.44 parent. **Source lines:** 230.

SHA-256: `8d41365d8376b7d75e5d8bd5fed0875e6d650020b0341126c480b93f5bfe0d6a`.

```
Actual zero-central isotropic lifts and their orthogonal averaging projectors.
```

Direct project imports: `PaperXI.HeisenbergRepresentation`, `PaperXI.HeisenbergForm`, `PaperXI.HeisenbergGeometry`, `PaperXI.SchrodingerUnitary`, `PaperXI.FiniteFunctionAverage`.

Line	Kind	Name
16	def	`PaperXI.HeisenbergProjector.liftHom`
25	theorem	`PaperXI.HeisenbergProjector.lift_position`
29	theorem	`PaperXI.HeisenbergProjector.lift_frequency`
33	theorem	`PaperXI.HeisenbergProjector.lift_central`
37	theorem	`PaperXI.HeisenbergProjector.lift_injective`
45	def	`PaperXI.HeisenbergProjector.liftedSubgroup`
48	theorem	`PaperXI.HeisenbergProjector.mem_liftedSubgroup_iff`
58	def	`PaperXI.HeisenbergProjector.liftEquiv`
67	instance	`PaperXI.HeisenbergProjector.liftedFintype`
71	theorem	`PaperXI.HeisenbergProjector.liftEquiv_coe`
75	def	`PaperXI.HeisenbergProjector.restriction`
80	theorem	`PaperXI.HeisenbergProjector.restriction_apply`
85	def	`PaperXI.HeisenbergProjector.projector`
89	theorem	`PaperXI.HeisenbergProjector.projector_apply`
101	def	`PaperXI.HeisenbergProjector.subgroupProjector`
106	theorem	`PaperXI.HeisenbergProjector.projector_eq_subgroupProjector`
121	theorem	`PaperXI.HeisenbergProjector.idempotent`
127	theorem	`PaperXI.HeisenbergProjector.range_eq_invariants`
136	theorem	`PaperXI.HeisenbergProjector.pairing_invariant`
143	theorem	`PaperXI.HeisenbergProjector.self_adjoint`
155	theorem	`PaperXI.HeisenbergProjector.residual_orthogonal`
173	def	`PaperXI.HeisenbergProjector.positionSpace`
179	def	`PaperXI.HeisenbergProjector.positionEquiv`
187	instance	`PaperXI.HeisenbergProjector.positionFintype`
190	theorem	`PaperXI.HeisenbergProjector.position_isotropic`
196	theorem	`PaperXI.HeisenbergProjector.position_projector_apply`
214	theorem	`PaperXI.HeisenbergProjector.position_projector_eq`
220	theorem	`PaperXI.HeisenbergProjector.position_range_eq`
225	theorem	`PaperXI.HeisenbergProjector.position_rank_one`

A.222 PaperXI.HeisenbergReduction107

Selected layer: base108 arithmetic branch. **Source lines:** 61.

SHA-256: 7bb3cef0f423eb861a688bbf662334632a9bfeaf015ddd1b429d554fc2c19934.

```
Assemble the operator proof and two-sided compact transport. The only
arithmetic inputs are support and line structure for the restricted first factor.
```

Direct project imports: PaperXI.TensorIntertwining107.

Line	Kind	Name
15	theorem	PaperXI.HeisenbergReduction107.pullback
27	theorem	PaperXI.HeisenbergReduction107.double_coset_reduction
43	theorem	PaperXI.HeisenbergReduction107.integral_reduction

A.223 PaperXI.HeisenbergRepresentation

Selected layer: v0.44 parent. **Source lines:** 72.

SHA-256: 386317cf254a374627df755dc02465986fc602151303b4e2b3413f2b82b46492.

```
The actual finite Schrödinger representation of the Heisenberg group,
its invertible operators, central character, and irreducibility.
```

Direct project imports: PaperXI.HeisenbergGroup, PaperXI.SchrodingerOperator, PaperXI.SchrodingerIrreducible.

Line	Kind	Name
14	def	PaperXI.Heisenberg.representation
21	theorem	PaperXI.Heisenberg.representation_apply
28	def	PaperXI.Heisenberg.representationGL
32	theorem	PaperXI.Heisenberg.representationGL_coe
38	theorem	PaperXI.Heisenberg.representation_isUnit
43	theorem	PaperXI.Heisenberg.central_action
47	theorem	PaperXI.Heisenberg.translation_action
51	theorem	PaperXI.Heisenberg.modulation_action
58	theorem	PaperXI.Heisenberg.irreducible

A.224 PaperXI.HeisenbergSelector

Selected layer: v0.44 parent. **Source lines:** 109.

SHA-256: 89d25c89f133dcf4a56a48864b73092bcf942c81c09adf27fcd7b8f8149c6643.

```
Trace and rank of the literal maximal isotropic averaging projector.
```

Direct project imports: PaperXI.FiniteAverage, PaperXI.SchrodingerTrace, PaperXI.HeisenbergCardinality, PaperXI.HeisenbergProjector.

Line	Kind	Name
17	theorem	PaperXI.HeisenbergSelector.trace_average

Line	Kind	Name
23	theorem	PaperXI.HeisenbergSelector.rank_average
30	theorem	PaperXI.HeisenbergSelector.invariants_rank_one_of_trace_one
42	theorem	PaperXI.HeisenbergSelector.coordinates_zero_iff
57	theorem	PaperXI.HeisenbergSelector.trace_restriction
66	theorem	PaperXI.HeisenbergSelector.trace_projector
77	theorem	PaperXI.HeisenbergSelector.maximal_trace_one
88	theorem	PaperXI.HeisenbergSelector.maximal_rank_one
100	theorem	PaperXI.HeisenbergSelector.maximal_fixed_line

A.225 PaperXI.HeisenbergSource

Selected layer: v0.44 parent. **Source lines:** 63.

SHA-256: ffb52975ff4f261a1791da759b032fd5c736abf36257053961a5b5b84c963686.

```
Source propositions 7.2 and 7.3 for odd prime p and the canonical
exponential character, with the manuscript's dimension parameter.
```

Direct project imports: PaperXI.HeisenbergPrime, PaperXI.HeisenbergRepresentation, PaperXI.HeisenbergDimension, PaperXI.HeisenbergFinite, PaperXI.HeisenbergCenter, PaperXI.SchrodingerTrace.

Line	Kind	Name
19	def	PaperXI.HeisenbergSource.model
22	theorem	PaperXI.HeisenbergSource.model_formula
27	theorem	PaperXI.HeisenbergSource.center_character
33	theorem	PaperXI.HeisenbergSource.model_invertible
38	theorem	PaperXI.HeisenbergSource.model_irreducible
44	theorem	PaperXI.HeisenbergSource.dimension_power
48	theorem	PaperXI.HeisenbergSource.dimension_sqrt
52	theorem	PaperXI.HeisenbergSource.character_support

A.226 PaperXI.HeisenbergSourceProjector

Selected layer: v0.44 parent. **Source lines:** 118.

SHA-256: 5e8ee76ad84809af2c219cb594f6ae86e0538557cd45997e05b8df888155396e.

```
Source propositions 7.5 and 7.6 for the canonical odd-prime model.
The finite enumeration of each isotropic subspace is constructed here.
```

Direct project imports: PaperXI.HeisenbergSource, PaperXI.HeisenbergSelector, PaperXI.HeisenbergProjector.

Line	Kind	Name
17	def	PaperXI.HeisenbergSourceProjector.projector
24	theorem	PaperXI.HeisenbergSourceProjector.idempotent
34	theorem	PaperXI.HeisenbergSourceProjector.range_eq_invariants
46	theorem	PaperXI.HeisenbergSourceProjector.self_adjoint
57	theorem	PaperXI.HeisenbergSourceProjector.residual_orthogonal
68	theorem	PaperXI.HeisenbergSourceProjector.lift_central_iff_identity
76	theorem	PaperXI.HeisenbergSourceProjector.position_range

Line	Kind	Name
87	theorem	`PaperXI.HeisenbergSourceProjector.position_rank_one`
97	theorem	`PaperXI.HeisenbergSourceProjector.maximal_rank_one`
107	theorem	`PaperXI.HeisenbergSourceProjector.maximal_fixed_line`

A.227 `PaperXI.HeisenbergTensor`

Selected layer: v0.44 parent. **Source lines:** 99.

SHA-256: `c869c46b15622f6c072689647c34c28046d5f9ff18bbc8191683f0588727d4e7`.

```
The actual tensor action descends from the direct product, with one retained
central factor, to the balanced central product. Its central character occurs once.
```

Direct project imports: `PaperXI.FiniteTensor`, `PaperXI.CentralProduct`.

Line	Kind	Name
17	def	`PaperXI.HeisenbergTensor.ambient`
27	theorem	`PaperXI.HeisenbergTensor.ambient_apply`
33	theorem	`PaperXI.HeisenbergTensor.ambient_tprod`
41	theorem	`PaperXI.HeisenbergTensor.ambient_balanced`
55	def	`PaperXI.HeisenbergTensor.descend`
61	theorem	`PaperXI.HeisenbergTensor.descend_mk`
68	theorem	`PaperXI.HeisenbergTensor.descend_tprod`
77	theorem	`PaperXI.HeisenbergTensor.central`
88	theorem	`PaperXI.HeisenbergTensor.empty_central`

A.228 `PaperXI.HeisenbergTensorComparison`

Selected layer: v0.44 parent. **Source lines:** 70.

SHA-256: `cc8aa01b6a54bea28296549299ad41b7873e08c01703c5aca621d700c4469661`.

```
Tensor factorization transported along an actual equivalence to a declared
full central extension. The comparisons intertwine actual group actions, not
only vector-space dimensions.
```

Direct project imports: `PaperXI.HeisenbergTensorFactorization`, `PaperXI.CentralExtensionComparison`.

Line	Kind	Name
23	def	`PaperXI.HeisenbergTensor.factorizationAlong`
33	theorem	`PaperXI.HeisenbergTensor.factorizationAlong_intertwines`
43	def	`PaperXI.HeisenbergTensor.onFullEnvelope`
48	def	`PaperXI.HeisenbergTensor.fullEnvelopeFactorization`
60	theorem	`PaperXI.HeisenbergTensor.fullEnvelopeFactorization_intertwines`

A.229 `PaperXI.HeisenbergTensorFactorization`

Selected layer: v0.44 parent. **Source lines:** 87.

SHA-256: `03fec0d85721da724851da3e533aa76b616b5ba39ddfc3fe9a3ab5b513399b16`.

```
Finite Heisenberg tensor factorization for the actual central product.
All factor spaces are actual irreducible representation spaces. Their finite
dimensions and nontriviality are derived, including zero quotient factors and
the empty family. The descended tensor action is proved irreducible before
Stone--von Neumann uniqueness is used.
```

Direct project imports: `PaperXI.HeisenbergTensor`, `PaperXI.CentralProductCommutator`, `PaperXI.TensorIrreducible`.

Line	Kind	Name
27	theorem	`PaperXI.HeisenbergTensor.factor_finiteDimensional`
31	theorem	`PaperXI.HeisenbergTensor.finiteDimensional`
36	theorem	`PaperXI.HeisenbergTensor.nontrivial`
42	theorem	`PaperXI.HeisenbergTensor.finrank_sq`
51	theorem	`PaperXI.HeisenbergTensor.irreducible`
61	theorem	`PaperXI.HeisenbergTensor.exists_factorization`
71	def	`PaperXI.HeisenbergTensor.factorization`
78	theorem	`PaperXI.HeisenbergTensor.factorization_intertwines`

A.230 `PaperXI.HeisenbergTrace108`

Selected layer: base108 arithmetic branch. **Source lines:** 47.

SHA-256: `045edb4a7a7672920cc0a4fdfc44e0b7b64d78563fe3cf3c625335b1e09240c8`.

```
The finite Heisenberg character formula, derived from scalar restriction
and detection by actual commutators. No trace-support hypothesis is supplied.
```

Direct project imports: none.

Line	Kind	Name
16	theorem	`PaperXI.HeisenbergTrace108.character_on_subgroup`
23	theorem	`PaperXI.HeisenbergTrace108.character_zero_of_commutator`
40	theorem	`PaperXI.HeisenbergTrace108.character_off_subgroup`

A.231 `PaperXI.HeisenbergTransport`

Selected layer: v0.44 parent. **Source lines:** 180.

SHA-256: `7be38801f777c48277b8229987482c3b7e8c656c351a2e5201c7a067740572e2`.

```
Transport of the coordinate Heisenberg group and actual representations.
The lift preserves the alternating cocycle in every characteristic. Its use
as the nondegenerate Heisenberg extension requires odd characteristic.
No coordinates on an abstract symplectic space are asserted here.
```

Direct project imports: `PaperXI.HeisenbergRepresentation`, `PaperXI.HeisenbergForm`.

Line	Kind	Name
20	def	`PaperXI.HeisenbergTransport.vector`
22	theorem	`PaperXI.HeisenbergTransport.vector_mul`
25	def	`PaperXI.HeisenbergTransport.PreservesForm`
28	theorem	`PaperXI.HeisenbergTransport.preservesForm_refl`

Line	Kind	Name
31	theorem	PaperXI.HeisenbergTransport.PreservesForm.symm
37	theorem	PaperXI.HeisenbergTransport.PreservesForm.trans
46	def	PaperXI.HeisenbergTransport.liftIsometry
69	theorem	PaperXI.HeisenbergTransport.liftIsometry_vector
74	theorem	PaperXI.HeisenbergTransport.liftIsometry_central
79	theorem	PaperXI.HeisenbergTransport.liftIsometry_centralElement
90	theorem	PaperXI.HeisenbergTransport.liftIsometry_refl
95	theorem	PaperXI.HeisenbergTransport.liftIsometry_symm
101	theorem	PaperXI.HeisenbergTransport.liftIsometry_trans
115	def	PaperXI.HeisenbergTransport.pullback
118	theorem	PaperXI.HeisenbergTransport.pullback_apply
121	theorem	PaperXI.HeisenbergTransport.pullback_refl
126	theorem	PaperXI.HeisenbergTransport.pullback_trans
130	theorem	PaperXI.HeisenbergTransport.pullback_symm
136	def	PaperXI.HeisenbergTransport.pullbackSubrepresentationEquiv
145	theorem	PaperXI.HeisenbergTransport.pullback_irreducible_iff
149	theorem	PaperXI.HeisenbergTransport.pullback_irreducible
155	theorem	PaperXI.HeisenbergTransport.pullback_scalar_action
170	theorem	PaperXI.HeisenbergTransport.pullback_central_action

A.232 PaperXI.HenselianConductorTransport

Selected layer: v0.44 parent. **Source lines:** 120.

SHA-256: 4195ca7a0238b26ebca438814c59c9712c09d21fd211f8468e3738fadebcb48a.

```
Conductor invariance for the actual base change to the Hensel-generated field.
The valuation ring is unchanged upstairs. This is not ramified quotient invariance.
```

Direct project imports: PaperXI.HenselianWeilAttachment.

Line	Kind	Name
17	def	PaperXI.HenselianConductorTransport.localLowerEquiv
31	def	PaperXI.HenselianConductorTransport.lowerEquiv
39	theorem	PaperXI.HenselianConductorTransport.lower_card
45	theorem	PaperXI.HenselianConductorTransport.embedding_lift
58	theorem	PaperXI.HenselianConductorTransport.fixed_eq
78	theorem	PaperXI.HenselianConductorTransport.summand_eq
87	theorem	PaperXI.HenselianConductorTransport.swan_eq
98	theorem	PaperXI.HenselianConductorTransport.artin_eq
108	theorem	PaperXI.HenselianConductorTransport.summable_source

A.233 PaperXI.HenselianFieldLift

Selected layer: v0.44 parent. **Source lines:** 95.

SHA-256: af945609f76e7f115f367f158515f7c5053cb2b0290dbc8baf21c20ea5f12e09.

```
Construct an actual finite intermediate field from finite residue-field
Hensel lifts. The fixed-field statement concerns literal automorphisms of L.
```

Direct project imports: `PaperXI.HenselianResidueLift`.

Line	Kind	Name
16	def	`PaperXI.HenselianFieldLift.element`
19	def	`PaperXI.HenselianFieldLift.field`
21	theorem	`PaperXI.HenselianFieldLift.element_mem`
24	theorem	`PaperXI.HenselianFieldLift.element_integral`
31	instance	`PaperXI.HenselianFieldLift.finiteDimensional`
38	theorem	`PaperXI.HenselianFieldLift.residue_element`
43	def	`PaperXI.HenselianFieldLift.fixing`
47	theorem	`PaperXI.HenselianFieldLift.mem_fixing_iff`
63	theorem	`PaperXI.HenselianFieldLift.fixing_iff_residue`
81	theorem	`PaperXI.HenselianFieldLift.inertia_le_fixing`
89	theorem	`PaperXI.HenselianFieldLift.contained_in_inertia_fixed`

A.234 `PaperXI.HenselianGaloisResidue`

Selected layer: v0.44 parent. **Source lines:** 75.

SHA-256: `4a35a55e8f7a231bdd6f1f2affbe1d44e00c6423dea5d53811260120032d96a0`.

```
Derive residue-automorphism lifting in a finite Galois extension with invariant
Henselian valuation and finite residue field. No surjectivity is assumed.
```

Direct project imports: `PaperXI.ValuationResidueEmbedding`.

Line	Kind	Name
13	def	`PaperXI.HenselianGaloisResidue.decompositionMap`
17	def	`PaperXI.HenselianGaloisResidue.residueAction`
20	theorem	`PaperXI.HenselianGaloisResidue.residueAction_residue`
23	theorem	`PaperXI.HenselianGaloisResidue.fixes_base`
35	theorem	`PaperXI.HenselianGaloisResidue.fixed_iff_base`
59	theorem	`PaperXI.HenselianGaloisResidue.lift_automorphism`

A.235 `PaperXI.HenselianLiftTower`

Selected layer: v0.44 parent. **Source lines:** 71.

SHA-256: `3ba24216ac193a2da00a39d6d731e8e750f4d8932c2ea761c52b6de0657f9cc7`.

```
Completion implies the Hensel hypothesis, and the unique lifts respect
finite residue-field towers. Multiplicativity is not additivity.
```

Direct project imports: `PaperXI.HenselianFieldLift`.

Line	Kind	Name
12	theorem	`PaperXI.HenselianLiftTower.henselian_of_complete`
24	theorem	`PaperXI.HenselianLiftTower.root_power_tower`
32	theorem	`PaperXI.HenselianLiftTower.lift_tower`
43	theorem	`PaperXI.HenselianLiftTower.lift_equiv`

Line	Kind	Name
58	theorem	PaperXI.HenselianLiftTower.field_tower

A.236 PaperXI.HenselianResidueLift

Selected layer: v0.44 parent. **Source lines:** 115.

SHA-256: 52a3d30f9b5773234105b18f548446e88910a055cc08896c493da1c5bed39370.

Unique multiplicative lifts of an embedded finite residue field. These are
actual Hensel roots in the given local ring, not an assumed ring section.

Direct project imports: none.

Line	Kind	Name
15	def	PaperXI.HenselianResidueLift.poly
17	theorem	PaperXI.HenselianResidueLift.poly_monic
22	theorem	PaperXI.HenselianResidueLift.card_residue_zero
26	theorem	PaperXI.HenselianResidueLift.derivative_unit
31	theorem	PaperXI.HenselianResidueLift.exists_lift
43	theorem	PaperXI.HenselianResidueLift.unique_lift
53	def	PaperXI.HenselianResidueLift.lift
55	theorem	PaperXI.HenselianResidueLift.lift_pow
58	theorem	PaperXI.HenselianResidueLift.residue_lift
61	theorem	PaperXI.HenselianResidueLift.lift_injective
66	theorem	PaperXI.HenselianResidueLift.lift_zero
71	theorem	PaperXI.HenselianResidueLift.lift_one
76	theorem	PaperXI.HenselianResidueLift.lift_mul
82	def	PaperXI.HenselianResidueLift.sectionMonoid
88	theorem	PaperXI.HenselianResidueLift.lift_nat_pow
91	theorem	PaperXI.HenselianResidueLift.lift_unit
96	theorem	PaperXI.HenselianResidueLift.automorphism_fix_iff
107	theorem	PaperXI.HenselianResidueLift.automorphism_transport

A.237 PaperXI.HenselianWeilAttachment

Selected layer: v0.44 parent. **Source lines:** 140.

SHA-256: af3f6d521f61881ee03e54dcc204f5911df3f47dc1b0201fa77efd687608ba2d.

Replace the previous residue stabilizer by the decomposition group over an
actual, constructed finite intermediate field. This does not assert its full
local-field unramified classification or an absolute Weil identification.

Direct project imports: PaperXI.HenselianLiftTower, PaperXI.ValuationFixingField, PaperXI.UnramifiedValuation.

Line	Kind	Name
19	abbrev	PaperXI.HenselianWeilAttachment.liftedField
23	theorem	PaperXI.HenselianWeilAttachment.field_fixing_eq
30	def	PaperXI.HenselianWeilAttachment.decompositionEquiv

Line	Kind	Name
37	theorem	PaperXI.HenselianWeilAttachment.ring_action
42	theorem	PaperXI.HenselianWeilAttachment.residue_action
50	theorem	PaperXI.HenselianWeilAttachment.residue_hom
58	def	PaperXI.HenselianWeilAttachment.pullbackEquiv
84	def	PaperXI.HenselianWeilAttachment.sourceEquiv
92	def	PaperXI.HenselianWeilAttachment.embedding
100	theorem	PaperXI.HenselianWeilAttachment.embedding_injective
104	theorem	PaperXI.HenselianWeilAttachment.degree_formula
111	theorem	PaperXI.HenselianWeilAttachment.embedding_range
130	theorem	PaperXI.HenselianWeilAttachment.fixes_liftedField_iff

A.238 PaperXI.HolonomyResidual

Selected layer: v0.44 parent. **Source lines:** 57.

SHA-256: d3e383de6f107b77fc33648b64a84751caa375bf3e468908ec513f79b1e11d52.

```
Paper XI, Proposition 14.6. The supplied equivalences are actual
root-to-vertex transports. Constructing the graph/tree is a separate obligation.
```

Direct project imports: none.

Line	Kind	Name
14	def	PaperXI.HolonomyResidual.transported
17	theorem	PaperXI.HolonomyResidual.transported_apply
20	def	PaperXI.HolonomyResidual.cycleDefect
27	def	PaperXI.HolonomyResidual.edgeResidual
34	theorem	PaperXI.HolonomyResidual.residual_factorization
43	theorem	PaperXI.HolonomyResidual.residual_zero_iff

A.239 PaperXI.HyperbolicDeterminant

Selected layer: v0.44 parent. **Source lines:** 66.

SHA-256: 36feef061c87d2e39afe6bfc3e8a246b942f70904fff07a77ab5dd5cff6fd275.

```
Determinants of the literal inverse-dual and hyperbolic difference maps.
All identities are established for the actual endomorphisms.
```

Direct project imports: PaperXI.HyperbolicSpace.

Line	Kind	Name
13	theorem	PaperXI.HyperbolicDeterminant.inverse_difference
21	theorem	PaperXI.HyperbolicDeterminant.dual_difference
33	theorem	PaperXI.HyperbolicDeterminant.dual_difference_det
43	theorem	PaperXI.HyperbolicDeterminant.action_difference
51	theorem	PaperXI.HyperbolicDeterminant.total_difference_det
59	theorem	PaperXI.HyperbolicDeterminant.action_det

A.240 PaperXI.HyperbolicPhase

Selected layer: v0.44 parent. **Source lines:** 56.

SHA-256: 419bca65c3efe41e14cf7793f6d4defd5c05e15201eb00e4161a916f3cb975f6.

```
Source 12.8 for the literal hyperbolic action. The calculation also holds
for the defined determinant phase in characteristic two; no Weil trace
interpretation or normalization is asserted here.
```

Direct project imports: PaperXI.HyperbolicDeterminant, PaperXI.OddPhase.

Line	Kind	Name
16	theorem	PaperXI.HyperbolicPhase.fixedPointFree
22	theorem	PaperXI.HyperbolicPhase.source
47	theorem	PaperXI.HyperbolicPhase.symplectic_fixedPointFree

A.241 PaperXI.HyperbolicSpace

Selected layer: v0.44 parent. **Source lines:** 120.

SHA-256: 1532ea2542009051b6a09e99d9353d460de621615cf408b4dff48af6a072c501.

```
The actual hyperbolic space U × U*, its evaluation form, and its
contragredient action. No bilinear form or action identities are assumed.
```

Direct project imports: none.

Line	Kind	Name
16	abbrev	PaperXI.HyperbolicSpace.Space
19	def	PaperXI.HyperbolicSpace.form
39	theorem	PaperXI.HyperbolicSpace.form_apply
43	theorem	PaperXI.HyperbolicSpace.form_isAlt
48	theorem	PaperXI.HyperbolicSpace.form_nondegenerate
72	def	PaperXI.HyperbolicSpace.action
75	theorem	PaperXI.HyperbolicSpace.action_apply
79	theorem	PaperXI.HyperbolicSpace.action_form
85	theorem	PaperXI.HyperbolicSpace.dual_fixedPointFree
107	theorem	PaperXI.HyperbolicSpace.action_fixedPointFree
115	theorem	PaperXI.HyperbolicSpace.finrank

A.242 PaperXI.InducedClassification

Selected layer: v0.44 parent. **Source lines:** 136.

SHA-256: 5700629a23c61b9f8d38d3ca69ba94c5efdaf0c847404c0c162aa3c7e54321a0.

```
Classification uses the full subgroup characters, including return values.
No assertion identifies an abstract degree subgroup with a local Weil group.
```

Direct project imports: PaperXI.InducedDual, PaperXI.CyclicInducedWrap.

Line	Kind	Name
16	theorem	`PaperXI.InducedClassification.index_transitive`
22	theorem	`PaperXI.InducedClassification.index_stabilizer`
29	def	`PaperXI.InducedClassification.conjugateEquiv`
38	theorem	`PaperXI.InducedClassification.coefficient_action`
56	theorem	`PaperXI.InducedClassification.character_of_equiv`
84	theorem	`PaperXI.InducedClassification.equiv_iff`
93	theorem	`PaperXI.InducedClassification.selfDual_iff`
110	def	`PaperXI.InducedClassification.degreeDualEquiv`
120	theorem	`PaperXI.InducedClassification.degreeSelfDual_iff`

A.243 `PaperXI.InducedCycleInvariants`

Selected layer: v0.44 parent. **Source lines:** 107.

SHA-256: `2faaeb89211dca9d88df443a27ad69ab7e310783d4e07a3ad54d85c3c5bcae1a`.

```
Frobenius invariants of genuine degree-kernel induction. The basis and its
closing scalar are constructed from the inducing character. No cyclic matrix,
characteristic polynomial or return-map certificate is assumed.
```

Direct project imports: `PaperXI.CyclicInducedWrap`, `PaperXI.CycleCoordinates`.

Line	Kind	Name
18	def	`PaperXI.InducedCycleInvariants.indexEquiv`
25	def	`PaperXI.InducedCycleInvariants.basis`
28	theorem	`PaperXI.InducedCycleInvariants.basis_apply`
33	def	`PaperXI.InducedCycleInvariants.returnScalar`
37	theorem	`PaperXI.InducedCycleInvariants.dimension`
41	theorem	`PaperXI.InducedCycleInvariants.step`
51	theorem	`PaperXI.InducedCycleInvariants.closing_step`
61	theorem	`PaperXI.InducedCycleInvariants.basis_powers`
69	theorem	`PaperXI.InducedCycleInvariants.finiteSpace`
75	theorem	`PaperXI.InducedCycleInvariants.cycle_invariants`
83	theorem	`PaperXI.InducedCycleInvariants.returnScalar_from_charpoly`
89	theorem	`PaperXI.InducedCycleInvariants.ambient_polynomial`
99	theorem	`PaperXI.InducedCycleInvariants.return_operator`

A.244 `PaperXI.InducedDual`

Selected layer: v0.44 parent. **Source lines:** 42.

SHA-256: `e915fa1d7389f923ae824272ca73b059424964f89c7e46b0aa503af596137995`.

```
Duality and the full-character classification for induction from a finite normal quotient.
```

Direct project imports: `PaperXI.MonomialDual`, `PaperXI.CyclicInducedCoordinates`, `PaperXI.MonomialInductionSubgroup`.

Line	Kind	Name
15	def	`PaperXI.InducedDual.indexAction`

Line	Kind	Name
21	theorem	`PaperXI.InducedDual.indexAction_apply`
26	def	`PaperXI.InducedDual.inverseCharacterEquiv`

A.245 `PaperXI.InducedTameProfile`

Selected layer: v0.44 parent. **Source lines:** 187.

SHA-256: `76a61c7e77722f469f734ee21b80e55786a2aec4a1916c23c4dc48a92249d576`.

```
All inertia and determinant data are read from the actual induced space.
The finite ramification expression is kept distinct from an arithmetically
identified Artin conductor. No irreducibility or regularity is needed here.
```

Direct project imports: `PaperXI.InducedCycleInvariants`, `PaperXI.TameConductor`, `PaperXI.InvariantPolynomial`.

Line	Kind	Name
18	theorem	`PaperXI.InducedTameProfile.inertia_degree`
33	def	`PaperXI.InducedTameProfile.inertiaInDegree`
51	theorem	`PaperXI.InducedTameProfile.origin_mem`
63	theorem	`PaperXI.InducedTameProfile.basis_packet`
77	theorem	`PaperXI.InducedTameProfile.support_of_basis`
104	theorem	`PaperXI.InducedTameProfile.inertia_support`
112	theorem	`PaperXI.InducedTameProfile.inertia_invariants`
128	theorem	`PaperXI.InducedTameProfile.invariant_dimension`
149	theorem	`PaperXI.InducedTameProfile.local_polynomial`
173	theorem	`PaperXI.InducedTameProfile.finite_conductors`

A.246 `PaperXI.InducedTwist`

Selected layer: v0.44 parent. **Source lines:** 103.

SHA-256: `53e7034ad1f3dacf431f99ae0d0334e94296550244011390bb7f137f09a4ebec`.

```
The projection formula on actual induced spaces, with inertia and return scalars.
```

Direct project imports: `PaperXI.CharacterTwist`, `PaperXI.CyclicInducedWrap`.

Line	Kind	Name
14	def	`PaperXI.InducedTwist.character`
16	theorem	`PaperXI.InducedTwist.character_apply`
20	def	`PaperXI.InducedTwist.forward`
31	def	`PaperXI.InducedTwist.backward`
43	def	`PaperXI.InducedTwist.coindLinearEquiv`
52	def	`PaperXI.InducedTwist.coindEquiv`
62	def	`PaperXI.InducedTwist.inducedEquiv`
74	def	`PaperXI.InducedTwist.degreeEquiv`
81	theorem	`PaperXI.InducedTwist.period_value`
88	theorem	`PaperXI.InducedTwist.closing_basis`
97	theorem	`PaperXI.InducedTwist.inertia_unchanged`

A.247 PaperXI.InducedTypeSpace68

Selected layer: base108 arithmetic branch. **Source lines:** 142.

SHA-256: `fe12e8899453c20460a4e73b84995d88a3551bb34ad86ab9b31fc3b83a89b409`.

```
# The canonical inducing-type subspace of actual induction

The full off-H Mackey condition gives factorization of every H-map through the
literal identity-coset inclusion.  No topological or semisimplicity hypothesis
is needed for this factorization.
```

Direct project imports: `PaperXI.InductionIrreducibility`.

Line	Kind	Name
21	theorem	`PaperXI.InducedTypeSpace68.factorization`
35	theorem	`PaperXI.InducedTypeSpace68.factorization_apply`
45	def	`PaperXI.InducedTypeSpace68.homEquiv`
60	theorem	`PaperXI.InducedTypeSpace68.range_eq_identity`
85	def	`PaperXI.InducedTypeSpace68.typeSpace`
89	def	`PaperXI.InducedTypeSpace68.typeEquiv`
96	def	`PaperXI.InducedTypeSpace68.typeOperators`
99	theorem	`PaperXI.InducedTypeSpace68.typeOperators_mul`
103	theorem	`PaperXI.InducedTypeSpace68.typeOperators_add`
107	theorem	`PaperXI.InducedTypeSpace68.typeOperators_eq_zero_iff`
112	theorem	`PaperXI.InducedTypeSpace68.typeOperators_idempotent_iff`
119	theorem	`PaperXI.InducedTypeSpace68.type_map_scalar`
134	theorem	`PaperXI.InducedTypeSpace68.conjugation_eq_of_scalar`

A.248 PaperXI.InductionIrreducibility

Selected layer: base108 arithmetic branch. **Source lines:** 237.

SHA-256: `b099fd6ad458401e85a83ae0289bec6be2de9575eb7b73c496fafb98d13088ed`.

```
# Irreducibility of actual subgroup induction from a local coefficient projector

All spaces and actions below are Mathlib's actual `IndV` and `ind`, at arbitrary
subgroup index.  The projection hypothesis is a local group-algebra condition:
each vector's identity-coset coefficient can be isolated by a finite linear
combination of subgroup operators.  The conclusion is actual irreducibility,
not scalar endomorphisms or an assumed simple subrepresentation lattice.
```

Direct project imports: `PaperXI.OpenSubgroupSmoothInduction`, `PaperXI.RepresentationStability`, `PaperXI.SemisimpleProjection`, `PaperXI.CompactSmoothSemisimple`.

Line	Kind	Name
26	def	`PaperXI.InductionIrreducibility.identityInclusion`
36	theorem	`PaperXI.InductionIrreducibility.evaluate_translate`
42	def	`PaperXI.InductionIrreducibility.identityEvaluation`
53	theorem	`PaperXI.InductionIrreducibility.identityEvaluation_inclusion`
57	def	`PaperXI.InductionIrreducibility.identityProjection`
61	theorem	`PaperXI.InductionIrreducibility.exists_nonzero_coefficient`
72	theorem	`PaperXI.InductionIrreducibility.eq_top_of_contains_identity`
88	theorem	`PaperXI.InductionIrreducibility.irreducible_of_projection`

Line	Kind	Name
126	theorem	`PaperXI.InductionIrreducibility.algebra_action_mem`
137	theorem	`PaperXI.InductionIrreducibility.irreducible_of_local_groupAlgebra`
150	def	`PaperXI.InductionIrreducibility.intersectionSubgroup`
154	def	`PaperXI.InductionIrreducibility.intersectionConjugation`
161	def	`PaperXI.InductionIrreducibility.coefficientIntertwiner`
182	theorem	`PaperXI.InductionIrreducibility.intertwiner_eq_zero_of_mackey`
205	theorem	`PaperXI.InductionIrreducibility.irreducible_of_mackey`
221	theorem	`PaperXI.InductionIrreducibility.irreducible_of_compact_open_mackey`

A.249 `PaperXI.InertiaFixed`

Selected layer: v0.44 parent. **Source lines:** 90.

SHA-256: `ac85a017c6c2462073de65b7304984cdc44ecebf7b201d52cad8f4a5e916e967`.

```
Actual inertia invariants of a cyclic packet, without irreducibility or
an assumption that its inertia labels are distinct.
```

Direct project imports: `PaperXI.PacketReturn`.

Line	Kind	Name
13	theorem	`PaperXI.InertiaFixed.invariants_eq_packet_zero`
23	theorem	`PaperXI.InertiaFixed.orbit_label_eq_zero_iff`
32	theorem	`PaperXI.InertiaFixed.invariants_eq_top_of_basis`
40	theorem	`PaperXI.InertiaFixed.invariants_eq_bot_of_basis`
56	theorem	`PaperXI.InertiaFixed.fixed_forward`
63	theorem	`PaperXI.InertiaFixed.fixed_backward`
70	theorem	`PaperXI.InertiaFixed.invariants_of_cycle_basis`

A.250 `PaperXI.InertiaQuotient`

Selected layer: v0.44 parent. **Source lines:** 98.

SHA-256: `6864c2943a2417a4883fae946d64fcb26273f3e590700e9ae16b296e8d5361bf`.

```
Invariant-space and restricted-operator comparison for a supplied finite
inertia quotient. No finite quotient is embedded into the original group.
```

Direct project imports: `PaperXI.InteractionPolynomial`.

Line	Kind	Name
13	theorem	`PaperXI.InertiaQuotient.invariants_comp_of_surjective`
23	def	`PaperXI.InertiaQuotient.invariantEquiv`
27	theorem	`PaperXI.InertiaQuotient.invariantEquiv_apply`
34	def	`PaperXI.InertiaQuotient.factoredFrobenius`
42	theorem	`PaperXI.InertiaQuotient.factoredFrobenius_apply`
49	theorem	`PaperXI.InertiaQuotient.factored_polynomial_eq`
58	theorem	`PaperXI.InertiaQuotient.factored_local_polynomial`
70	theorem	`PaperXI.InertiaQuotient.factored_character_trivial_iff`

Line	Kind	Name
83	theorem	`PaperXI.InertiaQuotient.factored_determinant`

A.251 `PaperXI.IntegerRingComparison`

Selected layer: v0.44 parent. **Source lines:** 230.

SHA-256: `382149600512912093535bbc83e0f1dcd500caa3db84e5340ac368157c74b06d`.

```
# Comparison of the algebraic and spectral integer rings

For the unramified residue extension constructed in `LocalFieldResidueExtension`,
the finite-extension spectral norm has unit ball equal to the image of the
algebraically constructed complete DVR.  The proof proceeds through the exact
integrality criterion over the base local-field integers.
```

Direct project imports: `PaperXI.SpectralLocalField`.

Line	Kind	Name
45	abbrev	`PaperXI.IntegerRingComparison.UpperRing`
46	abbrev	`PaperXI.IntegerRingComparison.UpperField`
50	theorem	`PaperXI.IntegerRingComparison.norm_le_one_iff_isIntegral`
95	theorem	`PaperXI.IntegerRingComparison.isIntegral_iff_exists_upper_lift`
108	theorem	`PaperXI.IntegerRingComparison.upper_integerRing_eq_spectral`
127	def	`PaperXI.IntegerRingComparison.upperRingEquivSpectralIntegers`
137	instance	`PaperXI.IntegerRingComparison.spectralIntegerAlgebra`
143	instance	`PaperXI.IntegerRingComparison.spectralIntegerAlgebra_isLocal`
155	def	`PaperXI.IntegerRingComparison.upperRingAlgEquivSpectralIntegers`
159	instance	`PaperXI.IntegerRingComparison.spectralIntegersEtale`
166	theorem	`PaperXI.IntegerRingComparison.upperRingEquivSpectralIntegers_coe`
173	theorem	`PaperXI.IntegerRingComparison.upper_maximalIdeal_map`
182	theorem	`PaperXI.IntegerRingComparison.spectral_ramificationIdx_one`
188	def	`PaperXI.IntegerRingComparison.upperResidueFieldEquivSpectral`
193	def	`PaperXI.IntegerRingComparison.upperResidueFieldAlgEquivSpectral`
200	def	`PaperXI.IntegerRingComparison.spectralResidueAlgEquiv`
207	def	`PaperXI.IntegerRingComparison.spectralResidueEquiv`
214	theorem	`PaperXI.IntegerRingComparison.spectral_residue_finrank`
220	theorem	`PaperXI.IntegerRingComparison.upper_degree_eq_spectral_residue_finrank`
226	theorem	`PaperXI.IntegerRingComparison.natCard_spectralResidue`

A.252 `PaperXI.InteractionPolynomial`

Selected layer: v0.44 parent. **Source lines:** 77.

SHA-256: `e9129207250ae783e05f622a71a16a8654e22680c10c252bdb028abccec69ec2`.

```
Both local-polynomial branches for a cyclic interaction packet.
```

Direct project imports: `PaperXI.InvariantPolynomial`, `PaperXI.PacketCycle`.

Line	Kind	Name
14	theorem	`PaperXI.InteractionPolynomial.character_trivial_iff`

Line	Kind	Name
24	theorem	`PaperXI.InteractionPolynomial.local_polynomial`
47	theorem	`PaperXI.InteractionPolynomial.local_polynomial_character_form`
62	theorem	`PaperXI.InteractionPolynomial.group_local_polynomial`

A.253 `PaperXI.IntersectionModel108`

Selected layer: base108 arithmetic branch. **Source lines:** 62.

SHA-256: `8f2f518c8309162145d80c2ae8d403d2c1eed20ac545a995985c18af108d6343`.

```
The literal conjugate-intersection operator space and its exact finite
quotient model. No identification of distinct Hom spaces is assumed.
```

Direct project imports: `PaperXI.IntertwiningSpace107`, `PaperXI.QuotientRepresentation`, `PaperXI.CharacterPairing108`.

Line	Kind	Name
16	def	`PaperXI.IntersectionModel108.pairs`
20	def	`PaperXI.IntersectionModel108.left`
23	def	`PaperXI.IntersectionModel108.right`
27	def	`PaperXI.IntersectionModel108.spaceEquiv`
39	def	`PaperXI.IntersectionModel108.pullbackEquiv`
53	theorem	`PaperXI.IntersectionModel108.finrank_model`

A.254 `PaperXI.IntertwiningAlong105`

Selected layer: base108 arithmetic branch. **Source lines:** 78.

SHA-256: `e1af42743d7ea4d6cbdbb96cb3d355e1f6502d69367e55041090370ccbea5478`.

```
Intertwining on a literal embedded subgroup, with exact carrier maps and
transport through both compact factors.
```

Direct project imports: none.

Line	Kind	Name
13	theorem	`PaperXI.IntertwiningAlong105.act_mul`
16	theorem	`PaperXI.IntertwiningAlong105.act_inv_cancel`
20	theorem	`PaperXI.IntertwiningAlong105.act_mul_inv_cancel`
25	def	`PaperXI.IntertwiningAlong105.Along`
29	def	`PaperXI.IntertwiningAlong105.ExistsAt`
32	theorem	`PaperXI.IntertwiningAlong105.left_right`
46	theorem	`PaperXI.IntertwiningAlong105.left_right_ne_zero`
56	theorem	`PaperXI.IntertwiningAlong105.existsAt_of_left_right`
62	theorem	`PaperXI.IntertwiningAlong105.remove_central`
72	theorem	`PaperXI.IntertwiningAlong105.existsAt_remove_central`

A.255 `PaperXI.IntertwiningSpace107`

Selected layer: base108 arithmetic branch. **Source lines:** 81.

SHA-256: `2c3ea887cc1868b2c539bf9d466778b39567a4ea10aa245876ba0dfe9bef8aa7`.

A linear equivalence of actual intertwining spaces on the same embedded
compact group. Identification with a smaller integral subgroup is separate.

Direct project imports: `PaperXI.TensorIntertwining107`.

Line	Kind	Name
16	def	`PaperXI.IntertwiningSpace107.space`
33	def	`PaperXI.IntertwiningSpace107.tensorMap`
43	def	`PaperXI.IntertwiningSpace107.equivalence`
54	theorem	`PaperXI.IntertwiningSpace107.finrank_eq`
61	theorem	`PaperXI.IntertwiningSpace107.line_of_dimension`

A.256 `PaperXI.InvariantAssembly`

Selected layer: v0.44 parent. **Source lines:** 48.

SHA-256: `2fb39cacd47c40f2a53037ffb5b416bb5f6311285265e3f57aed422baa40e5b4`.

Actual invariant spaces under a supplied equivariant finite decomposition.

Direct project imports: `PaperXI.GroupFixed`.

Line	Kind	Name
13	def	`PaperXI.InvariantAssembly.invariantsEquiv`
28	theorem	`PaperXI.InvariantAssembly.invariantsEquiv_apply`
32	theorem	`PaperXI.InvariantAssembly.finrank_invariants`
39	theorem	`PaperXI.InvariantAssembly.count_invariants`

A.257 `PaperXI.InvariantPolynomial`

Selected layer: v0.44 parent. **Source lines:** 90.

SHA-256: `cd9db934f06df326729d69a24446a44a9b6d00c9e5dbf48c6ff8c666e2b36990`.

The local polynomial is attached to Frobenius restricted to the actual
invariant subspace. Its determinant meaning is proved before its evaluation.

Direct project imports: `PaperXI.InertiaFixed`, `PaperXI.CyclicPolynomial`.

Line	Kind	Name
13	def	`PaperXI.InvariantPolynomial.operatorPolynomial`
16	theorem	`PaperXI.InvariantPolynomial.operatorPolynomial_eq_det`
22	theorem	`PaperXI.InvariantPolynomial.operatorPolynomial_of_subsingleton`
29	theorem	`PaperXI.InvariantPolynomial.operatorPolynomial_conj`
38	def	`PaperXI.InvariantPolynomial.fixedFrobenius`
49	theorem	`PaperXI.InvariantPolynomial.fixedFrobenius_apply`
55	def	`PaperXI.InvariantPolynomial.localPolynomial`
60	theorem	`PaperXI.InvariantPolynomial.localPolynomial_eq_det`
68	theorem	`PaperXI.InvariantPolynomial.fixed_conj_of_top`
76	theorem	`PaperXI.InvariantPolynomial.localPolynomial_of_top`

Line	Kind	Name
83	theorem	`PaperXI.InvariantPolynomial.localPolynomial_of_bot`

A.258 `PaperXI.IsotypicEvaluation`

Selected layer: v0.44 parent. **Source lines:** 97.

SHA-256: `9fa381127f5578659025a1972b5e641f6f3000ff3e6a0d17c4f402b6ec08beaa`.

Direct project imports: none.

Line	Kind	Name
19	def	`PaperXI.IsotypicEvaluation.map`
28	theorem	`PaperXI.IsotypicEvaluation.map_tmul`
31	theorem	`PaperXI.IsotypicEvaluation.bijective_of_copies`
80	theorem	`PaperXI.IsotypicEvaluation.bijective`
87	def	`PaperXI.IsotypicEvaluation.equiv`
92	theorem	`PaperXI.IsotypicEvaluation.equiv_tmul`

A.259 `PaperXI.LatticeCongruenceTopology60`

Selected layer: base108 arithmetic branch. **Source lines:** 144.

SHA-256: `2bdb8a0315e6b0015b8c69df7a1ef382ca5455f3976d5d040c507a94e2736fd0`.

```
# Topology of the full periodic-lattice congruence group

The infinite definition is preserved literally. A base-field scalar period
proves openness by finite reduction, while compactness follows from the zero
lattice stabilizer. The ambient group is actual `GL_d(K)`.
```

Direct project imports: `PaperXI.LatticeMatrixTopology60`, `PaperXI.PeriodicLatticeReduction60`.

Line	Kind	Name
21	def	`PaperXI.LatticeCongruenceTopology60.coordinateAut`
28	theorem	`PaperXI.LatticeCongruenceTopology60.coordinateAut_apply`
33	def	`PaperXI.LatticeCongruenceTopology60.congruence`
38	theorem	`PaperXI.LatticeCongruenceTopology60.mem_stabilizer_iff`
60	def	`PaperXI.LatticeCongruenceTopology60.levelSet`
65	theorem	`PaperXI.LatticeCongruenceTopology60.levelSet_eq`
72	theorem	`PaperXI.LatticeCongruenceTopology60.levelSet_closed`
89	theorem	`PaperXI.LatticeCongruenceTopology60.levelSet_open`
105	theorem	`PaperXI.LatticeCongruenceTopology60.congruence_closed`
120	theorem	`PaperXI.LatticeCongruenceTopology60.congruence_compact`
130	theorem	`PaperXI.LatticeCongruenceTopology60.congruence_open`

A.260 `PaperXI.LatticeMatrixTopology60`

Selected layer: base108 arithmetic branch. **Source lines:** 152.

SHA-256: `c924df503533e0221a502da1cdc8e886d4290c90602f3ee33287cdfb9ef98274`.

```
# Compact open lattice stabilizers in actual matrix coordinates

A finite continuous basis transports a literal compact open additive lattice
into the given local field's matrix group. Compactness uses a basis contained
in the lattice. No equality with a simple-character group is asserted.
```

Direct project imports: `PaperXI.ActualELatticeQuotient59`.

Line	Kind	Name
22	def	`PaperXI.LatticeMatrixTopology60.matrixAction`
26	theorem	`PaperXI.LatticeMatrixTopology60.matrixAction_continuous`
36	def	`PaperXI.LatticeMatrixTopology60.preserving`
45	def	`PaperXI.LatticeMatrixTopology60.stabilizer`
50	theorem	`PaperXI.LatticeMatrixTopology60.preserving_closed`
62	theorem	`PaperXI.LatticeMatrixTopology60.preserving_open`
73	theorem	`PaperXI.LatticeMatrixTopology60.preserving_compact`
92	theorem	`PaperXI.LatticeMatrixTopology60.exists_scaled_basis`
110	theorem	`PaperXI.LatticeMatrixTopology60.preserving_compact_of_open`
131	theorem	`PaperXI.LatticeMatrixTopology60.stabilizer_compact`
139	theorem	`PaperXI.LatticeMatrixTopology60.stabilizer_open`
145	theorem	`PaperXI.LatticeMatrixTopology60.stabilizer_compact_of_open`

A.261 `PaperXI.LiftedEtale`

Selected layer: v0.44 parent. **Source lines:** 95.

SHA-256: `2fa34a8ba6c429ab131c13111dc85621648e62f7ef6edf662e0367ebbddce8bf`.

```
Actual finite-etale and ramification-index endpoints for the Hensel-generated
field over a DVR base. All ring/module properties are derived.
```

Direct project imports: `PaperXI.LiftedValuationFinite`, `PaperXI.LocalResidueRank`.

Line	Kind	Name
20	theorem	`PaperXI.LiftedEtale.free`
28	def	`PaperXI.LiftedEtale.canonicalResidueLinearEquiv`
38	theorem	`PaperXI.LiftedEtale.rank_residue`
45	theorem	`PaperXI.LiftedEtale.maximalIdeal_map`
54	theorem	`PaperXI.LiftedEtale.etale`
65	theorem	`PaperXI.LiftedEtale.ramificationIdx_one`
72	theorem	`PaperXI.LiftedEtale.uniformizer_ideal`
80	theorem	`PaperXI.LiftedEtale.monogenic`

A.262 `PaperXI.LiftedFullResidue`

Selected layer: v0.44 parent. **Source lines:** 115.

SHA-256: `88447e2315204103283102ff48fb70a06eb201867cbbeceaaba3f94325b44618`.

```
The whole residue field of the lifted intermediate field, not merely an
embedded copy of the requested finite field.
```

Direct project imports: `PaperXI.LiftedResidueDegree`.

Line	Kind	Name
15	def	`PaperXI.LiftedFullResidue.valuation`
17	def	`PaperXI.LiftedFullResidue.residueMap`
21	theorem	`PaperXI.LiftedFullResidue.residueMap_contains`
35	theorem	`PaperXI.LiftedFullResidue.residueMap_mem`
56	theorem	`PaperXI.LiftedFullResidue.residueMap_range`
63	def	`PaperXI.LiftedFullResidue.toSubfield`
71	def	`PaperXI.LiftedFullResidue.residueEquiv`
77	theorem	`PaperXI.LiftedFullResidue.residueEquiv_val`
81	theorem	`PaperXI.LiftedFullResidue.residue_card`
85	def	`PaperXI.LiftedFullResidue.baseResidueMap`
89	theorem	`PaperXI.LiftedFullResidue.base_composition`
95	instance	`PaperXI.LiftedFullResidue.actualBaseAlgebra`
99	def	`PaperXI.LiftedFullResidue.residueAlgEquiv`
109	theorem	`PaperXI.LiftedFullResidue.residue_degree`

A.263 `PaperXI.LiftedResidueDegree`

Selected layer: v0.44 parent. **Source lines:** 82.

SHA-256: `d49acefdc58604537e7f2afe7c3ab0f46eba16fb8096a2dc28f1f8a655103bf7`.

```
Exact degree of the Hensel-generated field in a finite Galois ambient extension.
The base is the full residue field of the restricted valuation.
```

Direct project imports: `PaperXI.HenselianGaloisResidue`.

Line	Kind	Name
13	abbrev	`PaperXI.LiftedResidueDegree.BaseResidue`
15	instance	`PaperXI.LiftedResidueDegree.baseResidueAlgebra`
20	def	`PaperXI.LiftedResidueDegree.residueAut`
26	theorem	`PaperXI.LiftedResidueDegree.residueAut_apply`
32	theorem	`PaperXI.LiftedResidueDegree.residueAut_surjective`
41	def	`PaperXI.LiftedResidueDegree.field`
46	theorem	`PaperXI.LiftedResidueDegree.fixingSubgroup_eq`
67	theorem	`PaperXI.LiftedResidueDegree.degree`
75	theorem	`PaperXI.LiftedResidueDegree.galois`

A.264 `PaperXI.LiftedResidueFrobenius`

Selected layer: v0.44 parent. **Source lines:** 67.

SHA-256: `45e5ccfb6f245b6318a6d628168f8eba77834ea08b554b9f74ba407b49a9f45c`.

```
Frobenius lifting and the exact degree quotient with the actual base residue.
The degree-one lift is derived from finite Galois residue surjectivity.
```

Direct project imports: `PaperXI.LiftedFullResidue`, `PaperXI.FiniteResidueBaseChange`.

Line	Kind	Name
14	instance	`PaperXI.LiftedResidueFrobenius.baseFinite`

Line	Kind	Name
20	def	`PaperXI.LiftedResidueFrobenius.frobenius`
22	theorem	`PaperXI.LiftedResidueFrobenius.frobenius_inverse_power`
30	theorem	`PaperXI.LiftedResidueFrobenius.frobenius_in_range`
38	theorem	`PaperXI.LiftedResidueFrobenius.decomposition_frobenius_lift`
45	theorem	`PaperXI.LiftedResidueFrobenius.full_degree`
50	def	`PaperXI.LiftedResidueFrobenius.cyclicQuotient`
59	theorem	`PaperXI.LiftedResidueFrobenius.quotient_order`

A.265 `PaperXI.LiftedValuationFinite`

Selected layer: v0.44 parent. **Source lines:** 78.

SHA-256: `b4d29f529bdede481493a48e735a4998dd0ff3f401796cc652dbc870a82c2b0e`.

```
The Hensel-generated field has the actual finite integral valuation ring.
```

Direct project imports: `PaperXI.ValuationIntegralModel`, `PaperXI.LiftedFullResidue`.

Line	Kind	Name
17	instance	`PaperXI.LiftedValuationFinite.baseAlgebra`
20	instance	`PaperXI.LiftedValuationFinite.baseTower`
23	instance	`PaperXI.LiftedValuationFinite.baseFieldTower`
26	instance	`PaperXI.LiftedValuationFinite.baseLocal`
34	theorem	`PaperXI.LiftedValuationFinite.stable_valuation`
51	theorem	`PaperXI.LiftedValuationFinite.integralClosure`
58	theorem	`PaperXI.LiftedValuationFinite.finite`
68	theorem	`PaperXI.LiftedValuationFinite.rank`

A.266 `PaperXI.LiftedValuationInertia`

Selected layer: v0.44 parent. **Source lines:** 51.

SHA-256: `84e9166787ac68126346acfa087e68e5b5ad325b43fe0b761ac2b001f1ba310b`.

```
The actual residue-action inertia of the generated field is trivial.
This is not an assumed ramification-index certificate.
```

Direct project imports: `PaperXI.LiftedFullResidue`.

Line	Kind	Name
16	theorem	`PaperXI.LiftedValuationInertia.inertia_trivial`

A.267 `PaperXI.LineCharacter`

Selected layer: v0.44 parent. **Source lines:** 172.

SHA-256: `fe421a06982cd72d4f80a09db6f3ee52a249a6be9ded4bcd4aa927800c4ed007`.

```
The actual scalar character on the stabilizer of a nonzero line.
```

Direct project imports: `PaperXI.PacketBasis`.

Line	Kind	Name
11	theorem	`PaperXI.LineCharacter.scalar_unique`
16	theorem	`PaperXI.LineCharacter.scalar_ne_zero`
22	def	`PaperXI.LineCharacter.lineStabilizer`
39	theorem	`PaperXI.LineCharacter.mem_iff_preserves_line`
70	def	`PaperXI.LineCharacter.scalar`
73	theorem	`PaperXI.LineCharacter.scalar_spec`
77	def	`PaperXI.LineCharacter.character`
94	theorem	`PaperXI.LineCharacter.character_action`
97	theorem	`PaperXI.LineCharacter.power_action`
109	theorem	`PaperXI.LineCharacter.inertia_mem`
116	theorem	`PaperXI.LineCharacter.return_mem`
123	theorem	`PaperXI.LineCharacter.character_on_inertia`
133	theorem	`PaperXI.LineCharacter.character_on_return`
143	theorem	`PaperXI.LineCharacter.packet_stabilizer_iff`
159	theorem	`PaperXI.LineCharacter.expected_stabilizer_le`

A.268 `PaperXI.LocalFactorIdentification`

Selected layer: v0.44 parent. **Source lines:** 150.

SHA-256: `3e81e1ce1c7b50d2771432e73d46f02de2a2ce4910e0a3420aa67e6b02a514f3`.

```
Explicit realization and factor-characterization interfaces for the last
arithmetic step. Realizers need not be unique. Their isomorphism class becomes
unique by equality on a genuinely characterizing test family. No reference
comparison map or final equality is stored in either input interface.
```

Direct project imports: `PaperXI.AtomicReconstruction`, `PaperXI.CharacterProjector`.

Line	Kind	Name
14	structure	`PaperXI.LocalFactorIdentification.FactorSystem`
25	structure	`PaperXI.LocalFactorIdentification.Realization`
32	def	`PaperXI.LocalFactorIdentification.chosenAtom`
34	theorem	`PaperXI.LocalFactorIdentification.chosenAtom_realizes`
37	def	`PaperXI.LocalFactorIdentification.candidate`
40	theorem	`PaperXI.LocalFactorIdentification.class_eq_iff`
49	theorem	`PaperXI.LocalFactorIdentification.candidate_eq_reference`
59	theorem	`PaperXI.LocalFactorIdentification.unique_realized_class`
70	theorem	`PaperXI.LocalFactorIdentification.realization_independence`
81	theorem	`PaperXI.LocalFactorIdentification.candidate_congr`
89	def	`PaperXI.LocalFactorIdentification.onClasses`
95	theorem	`PaperXI.LocalFactorIdentification.constructed_comparison`
105	theorem	`PaperXI.LocalFactorIdentification.reconstructed_comparison`
117	def	`PaperXI.LocalFactorIdentification.representationSetoid`
124	theorem	`PaperXI.LocalFactorIdentification.finite_character_separates`
140	def	`PaperXI.LocalFactorIdentification.finiteCharacterSystem`

A.269 `PaperXI.LocalFieldResidueExtension`

Selected layer: v0.44 parent. **Source lines:** 63.

SHA-256: `e7438b3803d39699e4ac020c70f4f4d2377d99e32af405cdea45fa2eb1c9cb9e`.

```
Specialization at the actual ring of integers of a nonarchimedean local field.
The output ring, its residue and degree are constructed for every positive degree.
Absolute Weil groups and their smooth induction remain separate attachments.
```

Direct project imports: `PaperXI.FiniteFreeCompletion`.

Line	Kind	Name
16	abbrev	`PaperXI.LocalFieldResidueExtension.ring`
17	abbrev	`PaperXI.LocalFieldResidueExtension.field`
19	instance	`PaperXI.LocalFieldResidueExtension.baseAlgebra`
22	instance	`PaperXI.LocalFieldResidueExtension.baseTower`
28	theorem	`PaperXI.LocalFieldResidueExtension.degree`
32	instance	`PaperXI.LocalFieldResidueExtension.finiteExtension`
35	theorem	`PaperXI.LocalFieldResidueExtension.source`
50	abbrev	`PaperXI.LocalFieldResidueExtension.residueExtension`
52	theorem	`PaperXI.LocalFieldResidueExtension.every_positive_degree`

A.270 `PaperXI.LocalResidueRank`

Selected layer: v0.44 parent. **Source lines:** 83.

SHA-256: `04aa1ea55d4707f48f22f293dff3a05669d750c0ecaf1dca52bce090c7553fd3`.

```
An equal-rank residue fiber criterion for actual finite etale local algebras.
```

Direct project imports: none.

Line	Kind	Name
19	theorem	`PaperXI.LocalResidueRank.map_maximal_of_rank`
51	theorem	`PaperXI.LocalResidueRank.formallyUnramified`
57	theorem	`PaperXI.LocalResidueRank.etale`
66	theorem	`PaperXI.LocalResidueRank.etale_iff_rank_residue`
77	theorem	`PaperXI.LocalResidueRank.ramificationIdx_one`

A.271 `PaperXI.LocallyFiniteSemisimple`

Selected layer: base108 arithmetic branch. **Source lines:** 108.

SHA-256: `9d51826538fb9c038407e4a3d741ccbb3ad1c1266fc6642f1e29df6b81cefdd8`.

```
Semisimplicity from finite quotient actions on local invariant subspaces.
The representation may have arbitrary dimension; the local pieces supply the
semisimple submodules whose supremum is the whole group-algebra module.
```

Direct project imports: `PaperXI.QuotientRepresentation`.

Line	Kind	Name
21	theorem	`PaperXI.LocallyFiniteSemisimple.module_of_locally_semisimple`
39	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_comp_iff`
45	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_finite_pullback`

Line	Kind	Name
52	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_of_finite_quotient`
59	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_asSubmodule`
81	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_of_locally_semisimple`
94	theorem	`PaperXI.LocallyFiniteSemisimple.semisimple_of_locally_finite_quotient`

A.272 `PaperXI.MonicResidueAlgebra`

Selected layer: v0.44 parent. **Source lines:** 87.

SHA-256: `61e77b723a6fc1939e003a5cfd834cb152674a48112bee181e683903cdccdca5`.

```
Monic lifts with irreducible residue polynomial give finite free local
algebras with their literal special fiber. No ambient field is supplied.
```

Direct project imports: none.

Line	Kind	Name
17	abbrev	`PaperXI.MonicResidueAlgebra.specialIdeal`
20	instance	`PaperXI.MonicResidueAlgebra.finite`
21	instance	`PaperXI.MonicResidueAlgebra.free`
24	def	`PaperXI.MonicResidueAlgebra.specialFiberEquiv`
29	theorem	`PaperXI.MonicResidueAlgebra.specialIdeal_maximal`
34	instance	`PaperXI.MonicResidueAlgebra.localRing`
46	theorem	`PaperXI.MonicResidueAlgebra.maximalIdeal_eq`
49	def	`PaperXI.MonicResidueAlgebra.residueEquiv`
53	theorem	`PaperXI.MonicResidueAlgebra.rank`
56	theorem	`PaperXI.MonicResidueAlgebra.degree_pos`
60	instance	`PaperXI.MonicResidueAlgebra.localHom`
63	theorem	`PaperXI.MonicResidueAlgebra.residueEquiv_mk`
72	def	`PaperXI.MonicResidueAlgebra.residueAlgebraEquiv`
81	theorem	`PaperXI.MonicResidueAlgebra.residue_rank`

A.273 `PaperXI.MonodromyAtom`

Selected layer: v0.44 parent. **Source lines:** 117.

SHA-256: `f53d79ddc74c43995955bc45c8bce600d821bb6d118e9426d85e8478c0829bf2`.

```
Literal monodromy blocks underlying the formal reconstruction contract.
For ν(Φ)=q the action on the i-th copy is ν(g)^i ρ(g); the identity
R(g) N R(g)⁻¹ = ν(g)⁻¹ N fixes the normalization explicitly.
No assertion of irreducibility or a decomposition classification is assumed here.
```

Direct project imports: `PaperXI.MonodromyChain`.

Line	Kind	Name
18	structure	`PaperXI.MonodromyAtom.Datum`
25	def	`PaperXI.MonodromyAtom.action`
40	theorem	`PaperXI.MonodromyAtom.action_apply`
44	def	`PaperXI.MonodromyAtom.monodromy`

Line	Kind	Name
46	theorem	`PaperXI.MonodromyAtom.nilpotence`
50	theorem	`PaperXI.MonodromyAtom.commutation`
64	theorem	`PaperXI.MonodromyAtom.covariance`
70	def	`PaperXI.MonodromyAtom.sp`
78	theorem	`PaperXI.MonodromyAtom.dimension`
83	def	`PaperXI.MonodromyAtom.mapEquiv`
97	theorem	`PaperXI.MonodromyAtom.mapEquiv_monodromy`
108	theorem	`PaperXI.MonodromyAtom.scalar_twist`

A.274 `PaperXI.MonodromyChain`

Selected layer: v0.44 parent. **Source lines:** 122.

SHA-256: `3cf19949976f59a32001fee38e472b4bf8f9fda9d4d7c67e3819ff2e27ab101c`.

```
Paper XI, Proposition 2.3. Actual operators on a finite chain of copies of W.
The transfers A i are arbitrary linear maps. Nilpotence is a conclusion.
`path` lists all composable transfers in their matrix multiplication order.
```

Direct project imports: none.

Line	Kind	Name
16	def	`PaperXI.MonodromyChain.shift`
21	theorem	`PaperXI.MonodromyChain.shift_apply`
24	def	`PaperXI.MonodromyChain.path`
28	theorem	`PaperXI.MonodromyChain.path_zero`
29	theorem	`PaperXI.MonodromyChain.path_succ`
33	theorem	`PaperXI.MonodromyChain.pow_apply`
51	theorem	`PaperXI.MonodromyChain.nilpotent_bound`
59	def	`PaperXI.MonodromyChain.projector`
64	theorem	`PaperXI.MonodromyChain.projector_apply`
67	theorem	`PaperXI.MonodromyChain.projector_idempotent`
74	theorem	`PaperXI.MonodromyChain.projector_orthogonal`
83	theorem	`PaperXI.MonodromyChain.projector_sum`
90	theorem	`PaperXI.MonodromyChain.block_vanishes`
109	theorem	`PaperXI.MonodromyChain.block_reconstruction`

A.275 `PaperXI.MonomialAction`

Selected layer: v0.44 parent. **Source lines:** 92.

SHA-256: `70169fa35998c99c6d9830b345ee188e42778dace1f244f1181413f956a8961b`.

```
Actual monomial actions on a supplied basis. The scalar cocycle laws are
proved from an honest representation, then reused to construct orbit actions.
```

Direct project imports: none.

Line	Kind	Name
16	def	`PaperXI.MonomialAction.operator`

Line	Kind	Name
18	theorem	`PaperXI.MonomialAction.operator_basis`
22	def	`PaperXI.MonomialAction.representation`
37	theorem	`PaperXI.MonomialAction.representation_basis`
42	theorem	`PaperXI.MonomialAction.scalar_injective`
47	theorem	`PaperXI.MonomialAction.weight_one`
54	theorem	`PaperXI.MonomialAction.weight_mul`
63	theorem	`PaperXI.MonomialAction.representation_eq`
76	def	`PaperXI.MonomialAction.tensorPermutation`
82	theorem	`PaperXI.MonomialAction.tensor_basis`

A.276 `PaperXI.MonomialDual`

Selected layer: v0.44 parent. **Source lines:** 46.

SHA-256: `1f8a72858be61f32ca3a9daa195e4e3045e31db32c4be366e1a73c604753e8b8`.

```
The contragredient of a genuine monomial representation on the dual basis.
```

Direct project imports: `PaperXI.MonomialAction`.

Line	Kind	Name
18	theorem	`PaperXI.MonomialDual.inverse_weight`
26	theorem	`PaperXI.MonomialDual.dual_basis_action`

A.277 `PaperXI.MonomialInduction`

Selected layer: v0.44 parent. **Source lines:** 152.

SHA-256: `9406f172392d78a75931d9a416f9e701e1dc43b326140a53ab238c582b35813b`.

```
Actual induction from the stabilizer of a nonzero monomial basis line.
```

Direct project imports: `PaperXI.LineCharacter`.

Line	Kind	Name
12	def	`PaperXI.MonomialInduction.scalarRepresentation`
17	theorem	`PaperXI.MonomialInduction.scalarRepresentation_apply`
22	abbrev	`PaperXI.MonomialInduction.stabilizer`
23	abbrev	`PaperXI.MonomialInduction.character`
24	abbrev	`PaperXI.MonomialInduction.lineRepresentation`
25	abbrev	`PaperXI.MonomialInduction.induced`
26	abbrev	`PaperXI.MonomialInduction.InducedSpace`
29	def	`PaperXI.MonomialInduction.lineMap`
40	def	`PaperXI.MonomialInduction.evaluation`
52	theorem	`PaperXI.MonomialInduction.evaluation_mk`
58	theorem	`PaperXI.MonomialInduction.mk_stabilizer`
71	def	`PaperXI.MonomialInduction.reconstruction`
75	theorem	`PaperXI.MonomialInduction.reconstruction_basis`
83	theorem	`PaperXI.MonomialInduction.evaluation_reconstruction`

Line	Kind	Name
92	theorem	`PaperXI.MonomialInduction.representative_inverse`
100	theorem	`PaperXI.MonomialInduction.reconstruction_evaluation`
133	theorem	`PaperXI.MonomialInduction.evaluation_bijective`
144	def	`PaperXI.MonomialInduction.equivalence`
147	theorem	`PaperXI.MonomialInduction.equivalence_mk`

A.278 `PaperXI.MonomialInductionSubgroup`

Selected layer: v0.44 parent. **Source lines:** 69.

SHA-256: `392f827d63ce2b3a4c1ede29d1d61d70de0f705e596b32f42c617d03a6a96910`.

```
Induction from a specified subgroup and its actual scalar character.
```

Direct project imports: `PaperXI.MonomialInductionTransitive`.

Line	Kind	Name
12	def	`PaperXI.MonomialInduction.equivalenceOfSubgroup`
31	def	`PaperXI.MonomialInduction.transitiveEquivalenceOfSubgroupAction`
50	def	`PaperXI.MonomialInduction.transitiveEquivalenceOfSubgroup`

A.279 `PaperXI.MonomialInductionTransitive`

Selected layer: v0.44 parent. **Source lines:** 69.

SHA-256: `371cd6e7cd417a0f86606695b9bfa11c152a0a424eb0bf642fcdcb4a0ac04140`.

```
The transitive weighted permutation criterion for actual induced representations.
```

Direct project imports: `PaperXI.MonomialInduction`.

Line	Kind	Name
15	theorem	`PaperXI.MonomialInduction.mem_lineStabilizer_iff`
28	theorem	`PaperXI.MonomialInduction.character_eq_weight`
38	def	`PaperXI.MonomialInduction.representative`
40	theorem	`PaperXI.MonomialInduction.representative_spec`
44	theorem	`PaperXI.MonomialInduction.representative_action`
50	theorem	`PaperXI.MonomialInduction.orbit_is_monomial`
56	def	`PaperXI.MonomialInduction.transitiveEquivalence`
63	theorem	`PaperXI.MonomialInduction.transitiveEquivalence_mk`

A.280 `PaperXI.MonomialOrbit`

Selected layer: v0.44 parent. **Source lines:** 82.

SHA-256: `7552a5e8d7825191e603a7e2b8ad25bcf97b785a4fc4d3dd7e48cb7cdc0c1734`.

```
Decomposition of an actual monomial representation along actual invariant
basis orbits. The direct sum and its representation equivalence are constructed.
```

Direct project imports: `PaperXI.MonomialAction`.

Line	Kind	Name
23	def	`PaperXI.MonomialOrbit.block`
29	theorem	`PaperXI.MonomialOrbit.block_basis`
34	theorem	`PaperXI.MonomialOrbit.block_single`
40	def	`PaperXI.MonomialOrbit.sumBasis`
43	theorem	`PaperXI.MonomialOrbit.sumBasis_apply`
50	def	`PaperXI.MonomialOrbit.linearEquiv`
53	theorem	`PaperXI.MonomialOrbit.linearEquiv_basis`
59	theorem	`PaperXI.MonomialOrbit.intertwines`
72	def	`PaperXI.MonomialOrbit.representationEquiv`
79	theorem	`PaperXI.MonomialOrbit.block_finrank`

A.281 `PaperXI.MoritaActualCartan110`

Selected layer: Continuation110. **Source lines:** 266.

SHA-256: `dd18d3a60333f0e737ef4ff22c34db295f49460de701a9fddf9fcd3dc22d293a`.

```
Actual centralizer matrices act in the noncommutative endomorphism ring
through explicit ring homomorphisms.
```

Direct project imports: `PaperXI.MoritaIndex110`, `PaperXI.ActualLocalCartan106`, `PaperXI.MoritaCartanTransport110`, `PaperXI.MoritaScalarUpgrade110`.

Line	Kind	Name
21	def	`PaperXI.MoritaActualCartan110.centralizerEmbedding`
24	def	`PaperXI.MoritaActualCartan110.scalarEmbedding`
27	def	`PaperXI.MoritaActualCartan110.uniformizerEndUnit`
32	theorem	`PaperXI.MoritaActualCartan110.centralizerEmbedding_integral`
39	theorem	`PaperXI.MoritaActualCartan110.scalarEmbedding_integralUnit`
54	theorem	`PaperXI.MoritaActualCartan110.scalarEmbedding_uniformizer`
72	theorem	`PaperXI.MoritaActualCartan110.scalarEmbedding_normalizes`
93	theorem	`PaperXI.MoritaActualCartan110.centralizerEmbedding_cartan`
127	def	`PaperXI.MoritaActualCartan110.RadicalCommutes`
137	theorem	`PaperXI.MoritaActualCartan110.relIndex_of_actual_cartan`
182	theorem	`PaperXI.MoritaActualCartan110.exists_actual_cartan_index`
206	theorem	`PaperXI.MoritaActualCartan110.exists_actual_cartan_index_of_base_lattice`
232	theorem	`PaperXI.MoritaActualCartan110.relIndex_isEmpty`
246	theorem	`PaperXI.MoritaActualCartan110.relIndex_subsingleton`

A.282 `PaperXI.MoritaCartanTransport110`

Selected layer: Continuation110. **Source lines:** 121.

SHA-256: `61b021d7f9fd0fecaebb1a9dc3c89a7cbe30760ebecf6120c0af691d1e0e0645`.

```
Actual lattice-index transport under normalizing outer Cartan factors.
The intersection is transported by the additive conjugation equivalence;
relative index is preserved by its proved injectivity.
```

Direct project imports: `PaperXI.MoritaLattice110`.

Line	Kind	Name
16	def	`PaperXI.MoritaCartanTransport110.conjugate`
20	theorem	`PaperXI.MoritaCartanTransport110.unitConjugation_mul`
26	theorem	`PaperXI.MoritaCartanTransport110.unitConjugation_inv_apply_apply`
30	theorem	`PaperXI.MoritaCartanTransport110.conjugate_mul`
36	theorem	`PaperXI.MoritaCartanTransport110.conjugate_eq_of_normalizing`
44	theorem	`PaperXI.MoritaCartanTransport110.conjugate_inf`
50	theorem	`PaperXI.MoritaCartanTransport110.intersection_outer_factors`
58	theorem	`PaperXI.MoritaCartanTransport110.intersection_relIndex_outer_factors`
77	def	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit`
80	theorem	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit_val`
83	theorem	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit_inv`
89	theorem	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit_conjugation_mem`
98	theorem	`PaperXI.MoritaCartanTransport110.mappedMatrixUnit_mem_normalizingUnits`
110	theorem	`PaperXI.MoritaCartanTransport110.matrix_intersection_relIndex_outer_factors`

A.283 `PaperXI.MoritaCornerRank110`

Selected layer: Continuation110. **Source lines:** 132.

SHA-256: `8b3f281926d04dbaa4b40d47b74138cda2b422b7f8c4fd68785b65d37586aed2`.

```
Genuine finite/full submodules of the endomorphism ring acquire their
rank from the ambient field extension. No numerical index is a hypothesis.
```

Direct project imports: `PaperXI.MoritaLattice110`.

Line	Kind	Name
22	theorem	`PaperXI.MoritaCornerRank110.exists_basis_full_submodule`
63	def	`PaperXI.MoritaCornerRank110.coefficientMap`
67	def	`PaperXI.MoritaCornerRank110.cornerSubmodule`
79	theorem	`PaperXI.MoritaCornerRank110.cornerSubmodule_eq_map`
92	theorem	`PaperXI.MoritaCornerRank110.corner_fg`
100	theorem	`PaperXI.MoritaCornerRank110.corner_full`
120	theorem	`PaperXI.MoritaCornerRank110.exists_corner_basis_degree`

A.284 `PaperXI.MoritaEndomorphismCoordinates111`

Selected layer: Continuation111. **Source lines:** 176.

SHA-256: `d0023761aee69e81f53222a6e334e6ac57122a15fd64ce9017ee0bc0e27663c0`.

```
Natural block-endomorphism coordinates for the literal Morita carrier.
No basis of E over F is chosen: the block decomposition uses the finite
product's coordinate inclusions and projections.
```

Direct project imports: `PaperXI.MoritaActualCartan110`, `PaperXI.ActualCuspidalCandidate105`.

Line	Kind	Name
20	def	PaperXI.MoritaEndomorphismCoordinates111.blockLinearEquiv
24	theorem	PaperXI.MoritaEndomorphismCoordinates111.blockLinearEquiv_apply
31	theorem	PaperXI.MoritaEndomorphismCoordinates111.blockLinearEquiv_mul
41	def	PaperXI.MoritaEndomorphismCoordinates111.blockRingEquiv
45	theorem	PaperXI.MoritaEndomorphismCoordinates111.blockRingEquiv_apply
58	def	PaperXI.MoritaEndomorphismCoordinates111.coordinateRingEquiv
65	def	PaperXI.MoritaEndomorphismCoordinates111.coordinateUnitMap
71	theorem	PaperXI.MoritaEndomorphismCoordinates111.blockRingEquiv_centralizerEmbedding
82	theorem	PaperXI.MoritaEndomorphismCoordinates111.coordinateAut_coordinateUnitMap
96	theorem	PaperXI.MoritaEndomorphismCoordinates111.coordinateUnitMap_centralizer
115	theorem	PaperXI.MoritaEndomorphismCoordinates111.coordinateRingEquiv_unitConjugation
128	theorem	PaperXI.MoritaEndomorphismCoordinates111.coordinate_map_conjugate
142	theorem	PaperXI.MoritaEndomorphismCoordinates111.coordinate_intersection_relIndex
164	theorem	PaperXI.MoritaEndomorphismCoordinates111.coordinateUnitMap_integral

A.285 PaperXI.MoritaIndex110

Selected layer: Continuation110. **Source lines:** 144.

SHA-256: 30251babfb354b36cc6f28c150e892b0e244e9046f24d606dd8921245105a9d2.

```
Actual corner power quotients and the diagonal matrix-lattice index.
```

Direct project imports: PaperXI.MoritaCornerRank110, PaperXI.MoritaQuotient110, PaperXI.FreeLatticeCard110, PaperXI.MoritaUniformizer110.

Line	Kind	Name
20	theorem	PaperXI.MoritaIndex110.leftPower_addSubgroupOf_eq
44	theorem	PaperXI.MoritaIndex110.card_leftPower_quotient
69	def	PaperXI.MoritaIndex110.coefficientUnit
73	theorem	PaperXI.MoritaIndex110.coefficientUnit_coe
78	theorem	PaperXI.MoritaIndex110.card_diagonal_intersection
122	theorem	PaperXI.MoritaIndex110.diagonal_relIndex_of_radical_commutation

A.286 PaperXI.MoritaLattice110

Selected layer: Continuation110. **Source lines:** 354.

SHA-256: abdbc4e8506fb153a12c66728668a5c6ff97d94d2f2655dd1ebf49d43f4008d0.

```
The common-corner and diagonal-transport part of the maximal-order
Morita calculation. The coefficient ring acts through an arbitrary ring
homomorphism into a possibly noncommutative endomorphism ring. In particular,
no false centrality assumption on O_E inside End_F(E) is imposed.
```

Direct project imports: none.

Line	Kind	Name
18	def	PaperXI.MoritaLattice110.IsMatrixBimodule
23	def	PaperXI.MoritaLattice110.cornerEmbedding

Line	Kind	Name
29	def	PaperXI.MoritaLattice110.corner
33	theorem	PaperXI.MoritaLattice110.mem_corner
37	theorem	PaperXI.MoritaLattice110.entry_mem_corner
45	theorem	PaperXI.MoritaLattice110.single_mem_of_corner
53	theorem	PaperXI.MoritaLattice110.eq_matrix_corner
68	theorem	PaperXI.MoritaLattice110.corner_bimodule
82	def	PaperXI.MoritaLattice110.unitConjugation
89	theorem	PaperXI.MoritaLattice110.unitConjugation_apply
93	def	PaperXI.MoritaLattice110.normalizingUnits
106	theorem	PaperXI.MoritaLattice110.mem_normalizingUnits
110	theorem	PaperXI.MoritaLattice110.normalizes_of_left_right_map_eq
137	theorem	PaperXI.MoritaLattice110.zpow_normalizes
142	theorem	PaperXI.MoritaLattice110.two_sided_mem_iff_left
159	def	PaperXI.MoritaLattice110.leftPower
162	theorem	PaperXI.MoritaLattice110.mem_leftPower
177	def	PaperXI.MoritaLattice110.diagonalUnit
194	theorem	PaperXI.MoritaLattice110.diagonal_conjugation_apply
200	theorem	PaperXI.MoritaLattice110.scalar_conjugation_single
208	theorem	PaperXI.MoritaLattice110.corner_normalizes_of_scalar_commutation
221	theorem	PaperXI.MoritaLattice110.mem_conjugate
232	def	PaperXI.MoritaLattice110.entrywise
239	theorem	PaperXI.MoritaLattice110.mem_entrywise
244	theorem	PaperXI.MoritaLattice110.conjugate_eq_entrywise_leftPower
276	theorem	PaperXI.MoritaLattice110.intersection_eq_entrywise
297	theorem	PaperXI.MoritaLattice110.pow_mul_mem
304	theorem	PaperXI.MoritaLattice110.leftPower_zero
310	theorem	PaperXI.MoritaLattice110.inf_leftPower_eq_toNat
331	theorem	PaperXI.MoritaLattice110.corner_scalar_stable
341	theorem	PaperXI.MoritaLattice110.intersection_eq_nonnegative_entrywise

A.287 PaperXI.MoritaPairIndex110

Selected layer: Continuation110. **Source lines:** 39.

SHA-256: 1a7eeb3145c18b69770ec5d2b83acd8a4b2d23bb966e4cc47a44655a47472727.

```
The same actual Cartan witnesses compare two actual lattices. This is an
additive assertion; no canonical H1/J1 or multiplicative attachment is assumed.
```

Direct project imports: PaperXI.MoritaActualCartan110.

Line	Kind	Name
20	theorem	PaperXI.MoritaPairIndex110.actual_pair_index

A.288 PaperXI.MoritaQuotient110

Selected layer: Continuation110. **Source lines:** 170.

SHA-256: 33d3a455e6a8a202c2ce041e07c1c48e5d82d632a4f4dd114458be95809369e9.

```
Actual product quotients for the entrywise Morita calculation.
The additive equivalences are constructed from coordinate quotient maps,
```

```
their proved kernels, and proved surjectivity. The cardinality conclusions
are consequences of those equivalences, with no supplied index formula.
```

Direct project imports: `PaperXI.MoritaLattice110`.

Line	Kind	Name
16	def	`PaperXI.MoritaQuotient110.coordinateQuotientMap`
22	theorem	`PaperXI.MoritaQuotient110.coordinateQuotientMap_apply`
27	theorem	`PaperXI.MoritaQuotient110.coordinateQuotientMap_surjective`
35	theorem	`PaperXI.MoritaQuotient110.coordinateQuotientMap_ker`
43	def	`PaperXI.MoritaQuotient110.piQuotientEquiv`
48	theorem	`PaperXI.MoritaQuotient110.piQuotientEquiv_mk`
54	theorem	`PaperXI.MoritaQuotient110.card_pi_quotient`
60	def	`PaperXI.MoritaQuotient110.subgroupCoordinateQuotientMap`
66	theorem	`PaperXI.MoritaQuotient110.subgroupCoordinateQuotientMap_apply`
72	theorem	`PaperXI.MoritaQuotient110.subgroupCoordinateQuotientMap_surjective`
82	theorem	`PaperXI.MoritaQuotient110.subgroupCoordinateQuotientMap_ker`
93	def	`PaperXI.MoritaQuotient110.subgroupPiQuotientEquiv`
102	theorem	`PaperXI.MoritaQuotient110.subgroupPiQuotientEquiv_mk`
110	theorem	`PaperXI.MoritaQuotient110.card_subgroup_pi_quotient`
121	def	`PaperXI.MoritaQuotient110.matrixCoordinateQuotientMap`
127	theorem	`PaperXI.MoritaQuotient110.matrixCoordinateQuotientMap_apply`
132	theorem	`PaperXI.MoritaQuotient110.matrixCoordinateQuotientMap_surjective`
141	theorem	`PaperXI.MoritaQuotient110.matrixCoordinateQuotientMap_ker`
149	def	`PaperXI.MoritaQuotient110.matrixQuotientEquiv`
155	theorem	`PaperXI.MoritaQuotient110.matrixQuotientEquiv_mk`
162	theorem	`PaperXI.MoritaQuotient110.card_matrix_quotient`

A.289 `PaperXI.MoritaRadicalDictionary110`

Selected layer: Continuation110. **Source lines:** 132.

SHA-256: `caf45f046cafea555ed4871a05c2baece3d153318b1fced87d701d20622f781f`.

```
The literal additive product with the actual mapped maximal-order radical.
Only mapped coefficient matrices commute with its scalar; no centrality in the
whole endomorphism ring is asserted.
```

Direct project imports: `PaperXI.MoritaActualCartan110`.

Line	Kind	Name
17	def	`PaperXI.MoritaRadicalDictionary110.additiveProduct`
20	theorem	`PaperXI.MoritaRadicalDictionary110.mul_mem_additiveProduct`
30	def	`PaperXI.MoritaRadicalDictionary110.maximalOrderRadical`
34	theorem	`PaperXI.MoritaRadicalDictionary110.scalar_mem_maximalOrderRadical`
42	theorem	`PaperXI.MoritaRadicalDictionary110.scalar_commutes_mapped`
49	theorem	`PaperXI.MoritaRadicalDictionary110.radical_product_eq_left_image`
69	theorem	`PaperXI.MoritaRadicalDictionary110.product_radical_eq_right_image`
90	theorem	`PaperXI.MoritaRadicalDictionary110.scalar_images_eq_of_radical_products_eq`
109	def	`PaperXI.MoritaRadicalDictionary110.actualRadical`
115	theorem	`PaperXI.MoritaRadicalDictionary110.RadicalCommutes_of_actual_radical_products`

A.290 PaperXI.MoritaScalarUpgrade110

Selected layer: Continuation110. **Source lines:** 55.

SHA-256: `0a05325327c85f7a0de2f900d68524b1a168fd2264a2b5d8565ba75c64356291`.

```
Finite generation passes from a genuine base-integer lattice to the same
carrier with extension-integer scalars.
```

Direct project imports: `PaperXI.MoritaLattice110`.

Line	Kind	Name
17	def	`PaperXI.MoritaScalarUpgrade110.upgrade`
24	theorem	`PaperXI.MoritaScalarUpgrade110.upgrade_restrictScalars`
30	theorem	`PaperXI.MoritaScalarUpgrade110.upgrade_fg`
43	theorem	`PaperXI.MoritaScalarUpgrade110.scalar_stable_of_matrix_bimodule`

A.291 PaperXI.MoritaUniformizer110

Selected layer: Continuation110. **Source lines:** 62.

SHA-256: `b8f0b8cf79c2e58d57e03edd7ec29f81f51c91f1a0a837045514ee4ccf1c126e`.

```
Uniformizer arithmetic in the actual integer ring of a local field.
An irreducible integer-ring element generates the actual maximal ideal,
so primality, nonzeroness, and residue size are derived rather than supplied.
```

Direct project imports: none.

Line	Kind	Name
17	def	`PaperXI.MoritaUniformizer110.uniformizerIdeal`
20	theorem	`PaperXI.MoritaUniformizer110.exists_uniformizer`
24	theorem	`PaperXI.MoritaUniformizer110.uniformizerIdeal_eq_maximalIdeal`
27	theorem	`PaperXI.MoritaUniformizer110.uniformizerIdeal_ne_bot`
32	theorem	`PaperXI.MoritaUniformizer110.uniformizerIdeal_isMaximal`
38	theorem	`PaperXI.MoritaUniformizer110.uniformizerIdeal_isPrime`
43	def	`PaperXI.MoritaUniformizer110.uniformizerResidueEquiv`
47	theorem	`PaperXI.MoritaUniformizer110.uniformizerResidueEquiv_mk`
53	theorem	`PaperXI.MoritaUniformizer110.uniformizer_quotient_finite`
58	theorem	`PaperXI.MoritaUniformizer110.card_uniformizer_quotient`

A.292 PaperXI.MovingPhase

Selected layer: v0.44 parent. **Source lines:** 87.

SHA-256: `c01ee25e1c3e1f60b027ae5a5f09f85341892753a59e2eae1257490b80f8ec52`.

```
The determinant phase on the actual moving image, with its source
validity, conjugacy invariance and square-class comparison.
```

Direct project imports: `PaperXI.FixedMovingSource`, `PaperXI.OddPhaseTransport`.

Line	Kind	Name
16	def	PaperXI.MovingPhase.phase
18	theorem	PaperXI.MovingPhase.phase_formula
23	theorem	PaperXI.MovingPhase.determinant_ne_zero
28	theorem	PaperXI.MovingPhase.phase_sq_one
31	theorem	PaperXI.MovingPhase.phase_sign
37	theorem	PaperXI.MovingPhase.source_valid
51	theorem	PaperXI.MovingPhase.conjugacy
58	theorem	PaperXI.MovingPhase.square_class
66	theorem	PaperXI.MovingPhase.source_square_class
80	theorem	PaperXI.MovingPhase.phase_identity

A.293 PaperXI.NormalInertia

Selected layer: v0.44 parent. **Source lines:** 85.

SHA-256: bc52e7a907e034f8583c73eb69d3de4b3f0575f0ce1174605d50cce3864ab80a.

```
Normal-inertia fixed spaces and abelian quotient descent.
No cyclic finite-inertia generator, Frobenius orbit basis, or tame hypothesis
is required for the fixed-space vanishing theorem.
```

Direct project imports: PaperXI.QuotientRepresentation.

Line	Kind	Name
16	def	PaperXI.NormalInertia.fixedSubrepresentation
22	theorem	PaperXI.NormalInertia.fixed_eq_bot_or_top
30	theorem	PaperXI.NormalInertia.fixed_top_iff_le_kernel
46	theorem	PaperXI.NormalInertia.dimension_one_of_inertia_trivial
57	theorem	PaperXI.NormalInertia.fixed_eq_bot_of_dimension_gt_one
68	theorem	PaperXI.NormalInertia.fixed_eq_bot_of_not_le_kernel
76	theorem	PaperXI.NormalInertia.fixed_codimension

A.294 PaperXI.NormalizedFrobeniusQuotient

Selected layer: normalized Frobenius. **Source lines:** 120.

SHA-256: 917cfa62d1b914d7d47a3c31f682a52d3338dc040a1606450f1cff71ca075b1f.

```
# The field-normalized Frobenius pullback and its inertia quotient

All valuation data below come from the same finite Galois spectral extension.
The degree-one element and surjectivity of degree are derived, not certificate
fields. The quotient remains a bounded pullback model, not an absolute Weil group.
```

Direct project imports: PaperXI.SpectralGeometricFrobenius, PaperXI.RamifiedUnramifiedQuotient.

Line	Kind	Name
22	abbrev	PaperXI.NormalizedFrobeniusQuotient.Decomposition
23	abbrev	PaperXI.NormalizedFrobeniusQuotient.residueHom
25	abbrev	PaperXI.NormalizedFrobeniusQuotient.Weil
26	abbrev	PaperXI.NormalizedFrobeniusQuotient.degree

Line	Kind	Name
28	abbrev	PaperXI.NormalizedFrobeniusQuotient.Inertia
31	def	PaperXI.NormalizedFrobeniusQuotient.inertiaEquiv
37	def	PaperXI.NormalizedFrobeniusQuotient.decompositionLift
40	theorem	PaperXI.NormalizedFrobeniusQuotient.residue_decompositionLift
45	theorem	PaperXI.NormalizedFrobeniusQuotient.frobenius_mem_range
49	def	PaperXI.NormalizedFrobeniusQuotient.weilGeometricLift
55	theorem	PaperXI.NormalizedFrobeniusQuotient.degree_geometricLift
59	theorem	PaperXI.NormalizedFrobeniusQuotient.lift_inverse_power
65	def	PaperXI.NormalizedFrobeniusQuotient.weilArithmeticLift
68	theorem	PaperXI.NormalizedFrobeniusQuotient.degree_arithmeticLift
74	theorem	PaperXI.NormalizedFrobeniusQuotient.degree_surjective
79	def	PaperXI.NormalizedFrobeniusQuotient.quotientEquivIntegerDegree
84	theorem	PaperXI.NormalizedFrobeniusQuotient.quotientEquivIntegerDegree_mk
90	theorem	PaperXI.NormalizedFrobeniusQuotient.quotient_geometricLift
102	theorem	PaperXI.NormalizedFrobeniusQuotient.degree_one_action_eq_on_invariants

A.295 PaperXI.NormalizerDictionary110

Selected layer: Continuation110. **Source lines:** 432.

SHA-256: 19839adda13ba916ac5dbe60982bf13e65ce22a746ae2b2a63ca3cc54e4fe9e4.

```
Conditional character-normalizer dictionary, followed by literal instances.
No canonical simple-stratum group is introduced or identified by assumption.
```

Direct project imports: PaperXI.ActualCharacterIdentification61, PaperXI.ActualCharacterResidue61, PaperXI.ActualCuspidalCandidate105.

Line	Kind	Name
16	theorem	PaperXI.NormalizerDictionary110.inf_eq_of_support
37	theorem	PaperXI.NormalizerDictionary110.compact_eq_sup_of_support
61	theorem	PaperXI.NormalizerDictionary110.compact_le_normalizer_inf
66	theorem	PaperXI.NormalizerDictionary110.compact_le_normalizer_of_inf_eq
73	theorem	PaperXI.NormalizerDictionary110.mem_compact_iff_product
123	theorem	PaperXI.NormalizerDictionary110.chainGroup_le_normalizer_congruence
136	theorem	PaperXI.NormalizerDictionary110.firstSubgroup_ambient_image
148	theorem	PaperXI.NormalizerDictionary110.firstSubgroup_embedding_range
157	theorem	PaperXI.NormalizerDictionary110.firstSubgroup_normal
169	def	PaperXI.NormalizerDictionary110.integralCentralizer
174	theorem	PaperXI.NormalizerDictionary110.matrixType_eq_integral_sup
182	theorem	PaperXI.NormalizerDictionary110.integralCentralizer_le_chainGroup
193	theorem	PaperXI.NormalizerDictionary110.integralCentralizer_le_character_normalizer
209	theorem	PaperXI.NormalizerDictionary110.restrict_mem_chain_iff
234	theorem	PaperXI.NormalizerDictionary110.centralizer_inf_chain
271	theorem	PaperXI.NormalizerDictionary110.centralizer_inf_congruence
318	theorem	PaperXI.NormalizerDictionary110.actual_first_eq
329	theorem	PaperXI.NormalizerDictionary110.actual_first_subgroup_eq
340	theorem	PaperXI.NormalizerDictionary110.actual_compact_eq_sup
351	theorem	PaperXI.NormalizerDictionary110.actual_compact_product
367	theorem	PaperXI.NormalizerDictionary110.actual_model_le

Line	Kind	Name
378	theorem	PaperXI.NormalizerDictionary110.actual_model_from_centralizer
395	theorem	PaperXI.NormalizerDictionary110.actual_first_from_residue
411	theorem	PaperXI.NormalizerDictionary110.character_support_compact

A.296 PaperXI.NormalizerExtensionInducing58

Selected layer: base108 arithmetic branch. **Source lines:** 272.

SHA-256: e30d543d65b9008eaf4b66affdcd66e745a09db8651881df7678fda5e84c4b79.

```
# Extension across a noncentral normalizing cyclic direction

A single conjugation intertwiner propagates to all integer powers. Unique
semidirect coordinates then define an action on the literal generated subgroup.
No centrality assumption is made on the adjoined ambient group element.
```

Direct project imports: PaperXI.CentralExtensionInducing, PaperXI.QuotientRepresentation.

Line	Kind	Name
22	def	PaperXI.NormalizerExtensionInducing58.compatiblePairs
40	theorem	PaperXI.NormalizerExtensionInducing58.covariance_zpow
53	def	PaperXI.NormalizerExtensionInducing58.conjugation
56	def	PaperXI.NormalizerExtensionInducing58.conjugationAction
59	theorem	PaperXI.NormalizerExtensionInducing58.conjugation_coe
62	theorem	PaperXI.NormalizerExtensionInducing58.conjugationAction_coe
69	def	PaperXI.NormalizerExtensionInducing58.normalFormHom
75	theorem	PaperXI.NormalizerExtensionInducing58.normalFormHom_apply
80	def	PaperXI.NormalizerExtensionInducing58.extendedSubgroup
82	theorem	PaperXI.NormalizerExtensionInducing58.le_extendedSubgroup
86	theorem	PaperXI.NormalizerExtensionInducing58.generator_mem
89	theorem	PaperXI.NormalizerExtensionInducing58.extendedSubgroup_eq_sup
97	theorem	PaperXI.NormalizerExtensionInducing58.normalFormHom_injective
112	def	PaperXI.NormalizerExtensionInducing58.normalFormEquiv
115	theorem	PaperXI.NormalizerExtensionInducing58.normalFormEquiv_coe
122	def	PaperXI.NormalizerExtensionInducing58.subgroupInclusion
126	def	PaperXI.NormalizerExtensionInducing58.generator
133	def	PaperXI.NormalizerExtensionInducing58.semidirectRepresentation
143	def	PaperXI.NormalizerExtensionInducing58.extendedRepresentation
147	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_normalForm
155	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_restrict
163	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_generator
172	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_irreducible
183	theorem	PaperXI.NormalizerExtensionInducing58.equiv_unit_covariance
200	theorem	PaperXI.NormalizerExtensionInducing58.intertwiner_bijective
209	def	PaperXI.NormalizerExtensionInducing58.extensionOfEquiv
216	theorem	PaperXI.NormalizerExtensionInducing58.extensionOfEquiv_restrict
222	theorem	PaperXI.NormalizerExtensionInducing58.extensionOfEquiv_generator
231	theorem	PaperXI.NormalizerExtensionInducing58.exists_extension_of_nonzero
249	theorem	PaperXI.NormalizerExtensionInducing58.extendedSubgroup_isOpen
254	theorem	PaperXI.NormalizerExtensionInducing58.extendedRepresentation_smooth

A.297 PaperXI.NormalizerIntertwining60

Selected layer: base108 arithmetic branch. **Source lines:** 111.

SHA-256: 5993f730521e96961b4fed40e9a77796d990e2a0cccd720874c20680873c2c7f.

```
# Actual intersection intertwining at a normalizer

The intertwining locus is defined by nonzero maps on the literal intersection
subgroup. At a normalizing element such a map is a whole-subgroup intertwiner.
An inclusion of the actual character intertwining locus in that of a prescribed
representation supplies the required map, without a determinant or residue-size
condition. The inclusion is an input; no simple-character theorem is asserted.
```

Direct project imports: PaperXI.InductionIrreducibility, PaperXI.NormalizerExtensionInducing58, PaperXI.MonomialInduction.

Line	Kind	Name
23	def	PaperXI.NormalizerIntertwining60.IntertwinedAt
31	def	PaperXI.NormalizerIntertwining60.intersectionInclusion
37	def	PaperXI.NormalizerIntertwining60.normalizerIntertwiner
44	theorem	PaperXI.NormalizerIntertwining60.normalizerIntertwiner_apply
49	theorem	PaperXI.NormalizerIntertwining60.normalizerIntertwiner_ne_zero
60	theorem	PaperXI.NormalizerIntertwining60.exists_normalizerIntertwiner
67	theorem	PaperXI.NormalizerIntertwining60.exists_normalizerEquiv
76	theorem	PaperXI.NormalizerIntertwining60.character_intertwinedAt
101	theorem	PaperXI.NormalizerIntertwining60.exists_normalizerIntertwiner_of_character

A.298 PaperXI.OddEnvelopeHeisenberg

Selected layer: v0.44 parent. **Source lines:** 52.

SHA-256: 0ff24e111c1739d834a0e471bac34b82c9da25eb040e12f9cea2f9cf3b1d1465.

```
Canonical odd-characteristic extension, constructed directly on the given
alternating space. No polarization, symplectic basis or pre-existing group is input.
```

Direct project imports: PaperXI.BilinearHeisenberg, PaperXI.HeisenbergPrime.

Line	Kind	Name
13	def	PaperXI.OddEnvelopeHeisenberg.factor
15	theorem	PaperXI.OddEnvelopeHeisenberg.factor_apply
19	theorem	PaperXI.OddEnvelopeHeisenberg.factor_commutator
33	def	PaperXI.OddEnvelopeHeisenberg.isometryEquiv
46	theorem	PaperXI.OddEnvelopeHeisenberg.isometryEquiv_center

A.299 PaperXI.OddPhase

Selected layer: v0.44 parent. **Source lines:** 89.

SHA-256: a78d2e7c2db5c75599339479441e4e0b95181ec11ced910642a239377291249c.

```
Determinant-defined finite-field phases. These declarations do not identify
the phase with a Weil trace or assume any Weil normalization theorem.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.OddPhase.difference`
17	theorem	`PaperXI.OddPhase.difference_apply`
21	def	`PaperXI.OddPhase.FixedPointFree`
23	theorem	`PaperXI.OddPhase.fixedPointFree_iff_ker`
28	theorem	`PaperXI.OddPhase.fixedPointFree_iff_injective`
35	theorem	`PaperXI.OddPhase.fixedPointFree_iff_bijective`
40	theorem	`PaperXI.OddPhase.fixedPointFree_iff_det_ne_zero`
45	def	`PaperXI.OddPhase.differenceEquiv`
48	theorem	`PaperXI.OddPhase.differenceEquiv_apply`
52	def	`PaperXI.OddPhase.phase`
56	theorem	`PaperXI.OddPhase.phase_sq_one`
61	theorem	`PaperXI.OddPhase.phase_sign`
64	theorem	`PaperXI.OddPhase.phase_ne_zero`
69	theorem	`PaperXI.OddPhase.phase_zero_space`
74	theorem	`PaperXI.OddPhase.fixedPointFree_zero_space`
80	theorem	`PaperXI.OddPhase.phase_eq_zero_iff`

A.300 `PaperXI.OddPhaseCertificate`

Selected layer: v0.44 parent. **Source lines:** 61.

SHA-256: `783d84147c4f09cbca432ad799b1e57d9bd47da13f3841e92d4f071d49709f2e`.

```
The actual characteristic-polynomial certificate for the determinant phase.
The source specialization derives even dimension from the supplied form.
```

Direct project imports: `PaperXI.OddPhase`, `PaperXI.OddSymplectic`.

Line	Kind	Name
16	theorem	`PaperXI.OddPhase.charpoly_eval_one`
24	theorem	`PaperXI.OddPhase.determinant_neg_difference`
30	theorem	`PaperXI.OddPhase.characteristic_determinant`
36	theorem	`PaperXI.OddPhase.characteristic_certificate_of_even`
46	theorem	`PaperXI.OddPhase.characteristic_certificate`
55	theorem	`PaperXI.OddPhase.characteristic_eval_one_ne_zero`

A.301 `PaperXI.OddPhaseSum`

Selected layer: v0.44 parent. **Source lines:** 92.

SHA-256: `d97a3ea60f4bcb92aedcc1ccd4ba6b44c70ccfdf1ab43161b9c4194ad26867d1`.

```
Finite-product laws for the actual determinant phases.
```

Direct project imports: `PaperXI.FiniteProductDeterminant`, `PaperXI.OddPhaseTransport`, `PaperXI.OddSymplectic`.

Line	Kind	Name
16	def	`PaperXI.OddPhaseSum.action`
20	theorem	`PaperXI.OddPhaseSum.action_apply`
24	theorem	`PaperXI.OddPhaseSum.difference`
31	theorem	`PaperXI.OddPhaseSum.determinant`
36	theorem	`PaperXI.OddPhaseSum.fixedPointFree`
42	theorem	`PaperXI.OddPhaseSum.half_dimension`
58	theorem	`PaperXI.OddPhaseSum.phase_product`
65	theorem	`PaperXI.OddPhaseSum.phase_product_symplectic`
75	theorem	`PaperXI.OddPhaseSum.of_decomposition`
86	theorem	`PaperXI.OddPhaseSum.fixedPointFree_of_decomposition`

A.302 `PaperXI.OddPhaseTransport`

Selected layer: v0.44 parent. **Source lines:** 72.

SHA-256: `97c339e43d2c59d8b62995b29c986d88df5214c3df3ae472fd5249ad2ef9ba19`.

```
Actual conjugacy and determinant-square comparisons of the determinant phase.
```

Direct project imports: `PaperXI.OddPhase`.

Line	Kind	Name
13	theorem	`PaperXI.OddPhase.difference_intertwines`
18	theorem	`PaperXI.OddPhase.difference_conjugacy`
26	theorem	`PaperXI.OddPhase.fixedPointFree_conjugacy`
42	theorem	`PaperXI.OddPhase.determinant_conjugacy`
50	theorem	`PaperXI.OddPhase.conjugacy`
56	theorem	`PaperXI.OddPhase.square_class`
64	theorem	`PaperXI.OddPhase.square_class_of_exists`

A.303 `PaperXI.OddSymplectic`

Selected layer: v0.44 parent. **Source lines:** 70.

SHA-256: `e1b15050070b85893ab0d094b3ffaf7aaed326673ec11418c39d52487354e063`.

```
Even dimension of the actual alternating nondegenerate form in odd
characteristic. The determinant argument uses its matrix in a constructed
finite basis and does not assume symplectic coordinates or evenness.
```

Direct project imports: none.

Line	Kind	Name
17	theorem	`PaperXI.OddSymplectic.transpose_toMatrix`
26	theorem	`PaperXI.OddSymplectic.even_finrank`
44	theorem	`PaperXI.OddSymplectic.finrank_eq_two_mul_half`
51	theorem	`PaperXI.OddSymplectic.two_ne_zero`
59	theorem	`PaperXI.OddSymplectic.even_finrank_zmod`
64	theorem	`PaperXI.OddSymplectic.finrank_eq_two_mul_half_zmod`

A.304 PaperXI.OnePlusIndex110

Selected layer: Continuation110. **Source lines:** 87.

SHA-256: 733e4888e34cb2eec83dd8b0d29ea395d269e9caa209f00d1fb751c61563ef49.

```
Exact multiplicative/additive coset comparison. The named unit subgroup K
must really consist of all units 1+x for x in L. This does not pretend that
an arbitrary full lattice contained in a radical automatically is a group.
```

Direct project imports: none.

Line	Kind	Name
15	def	PaperXI.OnePlusIndex110.congruence
35	def	PaperXI.OnePlusIndex110.offsetEquiv
52	theorem	PaperXI.OnePlusIndex110.inv_mul_sub_one_mem_iff
67	def	PaperXI.OnePlusIndex110.cosetEquiv
81	theorem	PaperXI.OnePlusIndex110.index_eq_additive_relIndex

A.305 PaperXI.OnePlusIntersection110

Selected layer: Continuation110. **Source lines:** 197.

SHA-256: 512e3b7a894d7db7474f8586c4c652dddd45bfd98197f637f7be8e5a8fec3e7e.

```
Common-ideal quotient towers for exact one-plus groups. The primary
statement stays in the original integral-order ring and only needs nested
unit groups. It does not assume that a conjugate lattice is an ideal in the
full endomorphism algebra, or that a common integral order exists.
```

Direct project imports: PaperXI.OnePlusIndex110.

Line	Kind	Name
15	def	PaperXI.OnePlusIntersection110.idealUnits
18	theorem	PaperXI.OnePlusIntersection110.mem_idealUnits
21	theorem	PaperXI.OnePlusIntersection110.mem_of_offset
32	theorem	PaperXI.OnePlusIntersection110.idealUnits_le
38	theorem	PaperXI.OnePlusIntersection110.idealUnits_relIndex
50	theorem	PaperXI.OnePlusIntersection110.offset_le_of_group_le
62	theorem	PaperXI.OnePlusIntersection110.nested_relIndex_eq
90	def	PaperXI.OnePlusIntersection110.innerRing
99	theorem	PaperXI.OnePlusIntersection110.mapped_offsets
111	theorem	PaperXI.OnePlusIntersection110.mapped_onePlus
124	def	PaperXI.OnePlusIntersection110.pullbackGroup
127	def	PaperXI.OnePlusIntersection110.pullbackLattice
130	theorem	PaperXI.OnePlusIntersection110.pullback_offsets
143	theorem	PaperXI.OnePlusIntersection110.pullback_onePlus
161	theorem	PaperXI.OnePlusIntersection110.pullback_intersection_relIndex_eq
179	theorem	PaperXI.OnePlusIntersection110.conjugate_intersection_relIndex_eq

A.306 PaperXI.OpenSubgroupSmoothInduction

Selected layer: base108 arithmetic branch. **Source lines:** 409.

SHA-256: eec2e1f81ea1ed3ed0161aa0df8518115a4605b351d08127ef4282943d5978f7.

```
# Smooth induction from an open subgroup, with no finite-index hypothesis

The target is the actual algebraic induced representation `Representation.ind`.
For an open subgroup and a smooth inducing action, its finite-sum vectors are
smooth even when the subgroup has infinite index. This is an existence step
towards compact induction, not an irreducibility or local-factor theorem.
```

Direct project imports: `PaperXI.SmoothDegreeQuotient`.

Line	Kind	Name
25	def	`PaperXI.OpenSubgroupSmoothInduction.smoothSubmodule`
48	theorem	`PaperXI.OpenSubgroupSmoothInduction.mem_smoothSubmodule`
57	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_subgroup`
65	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_smooth`
84	theorem	`PaperXI.OpenSubgroupSmoothInduction.ind_isSmooth`
100	def	`PaperXI.OpenSubgroupSmoothInduction.functionModel`
105	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_mk`
114	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_mk_self`
122	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_injective`
129	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_ne_zero`
136	theorem	`PaperXI.OpenSubgroupSmoothInduction.ind_nontrivial`
140	def	`PaperXI.OpenSubgroupSmoothInduction.finiteCosetSupport`
172	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_finite_support`
194	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_finite_quotient_support`
205	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_compact_quotient_support`
212	def	`PaperXI.OpenSubgroupSmoothInduction.functionIntertwiner`
225	def	`PaperXI.OpenSubgroupSmoothInduction.evaluate`
229	theorem	`PaperXI.OpenSubgroupSmoothInduction.evaluate_mk_self`
233	theorem	`PaperXI.OpenSubgroupSmoothInduction.evaluate_mk_of_not_rel`
243	theorem	`PaperXI.OpenSubgroupSmoothInduction.mk_linearIndependent`
264	def	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent`
274	theorem	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent_mk`
279	theorem	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent_add`
286	theorem	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent_smul`
293	theorem	`PaperXI.OpenSubgroupSmoothInduction.cosetComponent_finite`
309	def	`PaperXI.OpenSubgroupSmoothInduction.reconstruct`
322	def	`PaperXI.OpenSubgroupSmoothInduction.compactModel`
326	theorem	`PaperXI.OpenSubgroupSmoothInduction.reconstruct_compactModel`
342	theorem	`PaperXI.OpenSubgroupSmoothInduction.compactModel_reconstruct`
358	def	`PaperXI.OpenSubgroupSmoothInduction.compactModelEquiv`
364	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_injective`
371	theorem	`PaperXI.OpenSubgroupSmoothInduction.functionModel_range`
382	theorem	`PaperXI.OpenSubgroupSmoothInduction.finite_functions_smooth`

A.307 `PaperXI.OrbitSpan`

Selected layer: v0.44 parent. **Source lines:** 86.

SHA-256: `1c23d6d51d10af04e9788052c0601a726639e8d6938f47b653949b20a8bdd0c3`.

```
The span of a Frobenius orbit of actual character spaces. Its stability and
separation from other packets are proved, not assumed.
```

Direct project imports: `PaperXI.FrobeniusOrbit`.

Line	Kind	Name
12	def	`PaperXI.OrbitSpan.orbitSpan`
16	theorem	`PaperXI.OrbitSpan.packet_le`
22	theorem	`PaperXI.OrbitSpan.self_packet_le`
26	theorem	`PaperXI.OrbitSpan.map_mem`
36	theorem	`PaperXI.OrbitSpan.inertia_stable`
42	theorem	`PaperXI.OrbitSpan.forward_stable`
49	theorem	`PaperXI.OrbitSpan.backward_stable`
56	theorem	`PaperXI.OrbitSpan.outside_projector_zero`
66	theorem	`PaperXI.OrbitSpan.span_ne_bot`
71	theorem	`PaperXI.OrbitSpan.outside_packet_eq_bot`
80	theorem	`PaperXI.OrbitSpan.related_of_span_top`

A.308 `PaperXI.OrthogonalPhase`

Selected layer: v0.44 parent. **Source lines:** 111.

SHA-256: `f822d2a61741660dd23d656ef3748309c17ca6e982ee0b05bb69f6b06e8edeb0`.

```
Source 12.4 for an actual finite internal orthogonal decomposition.
```

Direct project imports: `PaperXI.OddPhaseSum`, `PaperXI.OrthogonalSummands`.

Line	Kind	Name
14	def	`PaperXI.OrthogonalPhase.decomposition`
19	theorem	`PaperXI.OrthogonalPhase.decomposition_sum`
47	def	`PaperXI.OrthogonalPhase.restricted`
52	theorem	`PaperXI.OrthogonalPhase.restricted_coe`
56	theorem	`PaperXI.OrthogonalPhase.intertwines`
68	theorem	`PaperXI.OrthogonalPhase.phase_product`
81	theorem	`PaperXI.OrthogonalPhase.fixedPointFree`
91	theorem	`PaperXI.OrthogonalPhase.source`

A.309 `PaperXI.OrthogonalSummands`

Selected layer: v0.44 parent. **Source lines:** 120.

SHA-256: `fa9f0303b80e64cd56907157d0dc0c86d1f3e34d1ebd579bf0614a9c89116898`.

```
Actual restricted forms and automorphisms on an internal orthogonal
family. Nondegeneracy is derived from orthogonality and spanning. Inverse
invariance is derived from finite-dimensionality and forward invariance.
```

Direct project imports: `PaperXI.OddSymplectic`.

Line	Kind	Name
17	theorem	`PaperXI.OrthogonalSummands.restricted_isAlt`
22	theorem	`PaperXI.OrthogonalSummands.restricted_form_nondegenerate`
58	theorem	`PaperXI.OrthogonalSummands.iSup_eq_top_of_equiv_sum`
71	def	`PaperXI.OrthogonalSummands.restrictionEquiv`
78	theorem	`PaperXI.OrthogonalSummands.restrictionEquiv_coe`

Line	Kind	Name
82	theorem	`PaperXI.OrthogonalSummands.restrictionEquiv_symm_coe`
91	theorem	`PaperXI.OrthogonalSummands.inverse_mem`
97	theorem	`PaperXI.OrthogonalSummands.restrictionEquiv_form`
104	theorem	`PaperXI.OrthogonalSummands.restrictionEquiv_fixed_eq_zero`
112	theorem	`PaperXI.OrthogonalSummands.restricted_even_finrank`

A.310 `PaperXI.PacketBasis`

Selected layer: v0.44 parent. **Source lines:** 97.

SHA-256: `c5f5d83a01759c8aa3984c2b11462888159937d7a2fd4c3f407cc7495a5aee45`.

```
The finite Frobenius orbit is a proved basis, giving the exact ambient
dimension and the action of inertia and Frobenius on the basis vectors.
```

Direct project imports: `PaperXI.PacketLines`.

Line	Kind	Name
16	theorem	`PaperXI.PacketBasis.cycleVector_ne_zero`
22	theorem	`PaperXI.PacketBasis.cycle_linearIndependent`
39	def	`PaperXI.PacketBasis.orbitBasis`
52	theorem	`PaperXI.PacketBasis.orbitBasis_apply`
63	theorem	`PaperXI.PacketBasis.finrank_eq_period_of_return`
74	theorem	`PaperXI.PacketBasis.finrank_eq_period`
86	theorem	`PaperXI.PacketBasis.finrank_eq_orbit_card`

A.311 `PaperXI.PacketCycle`

Selected layer: v0.44 parent. **Source lines:** 76.

SHA-256: `55a96ecf51c156847ccfe2269e6caecfdfb45dbb4792983af8cb809e3a438bc8`.

```
Cycle invariants of the actual irreducible tame algebraic group action.
```

Direct project imports: `PaperXI.CycleCoordinates`, `PaperXI.PacketBasis`.

Line	Kind	Name
13	theorem	`PaperXI.PacketCycle.linear_power`
32	theorem	`PaperXI.PacketCycle.charpoly`
43	theorem	`PaperXI.PacketCycle.det`
54	theorem	`PaperXI.PacketCycle.ambient_polynomial`
66	theorem	`PaperXI.PacketCycle.exists_cycle_invariants`

A.312 `PaperXI.PacketLines`

Selected layer: v0.44 parent. **Source lines:** 147.

SHA-256: `a7c34f355ec41735f4fe274cde83c52d9d91c22db3eabb38d6d7de1c7222a4dd`.

```
A return eigenline exhausts its packet in an actual irreducible
representation. No packet dimension or spanning conclusion is assumed.
```

Direct project imports: `PaperXI.CycleSpan`.

Line	Kind	Name
13	def	`PaperXI.PacketLines.orbitLabel`
16	theorem	`PaperXI.PacketLines.orbitLabel_injective`
22	theorem	`PaperXI.PacketLines.cycle_mem_packet`
29	theorem	`PaperXI.PacketLines.inertia_cycle`
38	theorem	`PaperXI.PacketLines.inertia_stable`
49	theorem	`PaperXI.PacketLines.projector_cycle`
66	theorem	`PaperXI.PacketLines.projection_to_line`
83	theorem	`PaperXI.PacketLines.packet_eq_line_of_span_top`
98	theorem	`PaperXI.PacketLines.generators_preserve_cycleSpan`
118	theorem	`PaperXI.PacketLines.cycleSpan_eq_top`
131	theorem	`PaperXI.PacketLines.nonzero_packet_finrank_one`

A.313 `PaperXI.PacketReturn`

Selected layer: v0.44 parent. **Source lines:** 61.

SHA-256: `54669679788cd0095e0d26b16343de68a5196641db0e60dad21ec97090f52942`.

```
The actual return operator on a packet and its nonzero eigenline. The
claim that this line is the whole packet is deliberately not assumed.
```

Direct project imports: `PaperXI.FrobeniusPeriod`.

Line	Kind	Name
13	theorem	`PaperXI.PacketReturn.power_mem`
23	theorem	`PaperXI.PacketReturn.return_mem`
30	def	`PaperXI.PacketReturn.closingOperator`
40	theorem	`PaperXI.PacketReturn.exists_closing_eigenvector`

A.314 `PaperXI.PacketSupport`

Selected layer: v0.44 parent. **Source lines:** 105.

SHA-256: `3265ba26afef8a86567d0e4ecc83f6448bf3458790ff26d376ae84f7834b78f1`.

```
Exact support, and constant packet dimension along the Frobenius orbit.
```

Direct project imports: `PaperXI.TameOrbitTransitivity`.

Line	Kind	Name
12	theorem	`PaperXI.PacketSupport.nonzero_step_iff`
19	theorem	`PaperXI.PacketSupport.nonzero_back_iff`
26	theorem	`PaperXI.PacketSupport.nonzero_zpow_iff`
39	theorem	`PaperXI.PacketSupport.finrank_zpow`
68	theorem	`PaperXI.PacketSupport.support_iff_orbit`
82	theorem	`PaperXI.PacketSupport.support_is_one_orbit`
92	theorem	`PaperXI.PacketSupport.equal_packet_dimensions`

A.315 PaperXI.PacketTransport

Selected layer: v0.44 parent. **Source lines:** 81.

SHA-256: `dc7282e77c935ca61b8807fe87f4fe2fa372d720a28d0a1c8982c6c749e1a910`.

```
Actual Frobenius transport between the ranges of the computed projectors.
```

Direct project imports: `PaperXI.TameProjectors`.

Line	Kind	Name
11	abbrev	`PaperXI.PacketTransport.packet`
14	abbrev	`PaperXI.PacketTransport.projector`
17	theorem	`PaperXI.PacketTransport.mem_iff_projector`
25	theorem	`PaperXI.PacketTransport.projector_on_other_packet`
32	theorem	`PaperXI.PacketTransport.inertia_mem`
38	theorem	`PaperXI.PacketTransport.transport_of_conjugation`
46	theorem	`PaperXI.PacketTransport.forward_mem`
53	theorem	`PaperXI.PacketTransport.backward_mem`
62	def	`PaperXI.PacketTransport.packetEquiv`
75	theorem	`PaperXI.PacketTransport.packet_finrank_step`

A.316 PaperXI.PacketTypeAttachment70

Selected layer: base108 arithmetic branch. **Source lines:** 85.

SHA-256: `ed82a000e951e6dca19a854165a2e9f92be60a3bcc1ed876f0fae8a9978b8a79`.

```
The finite packet action is the restriction of the SAME actual induced
representation on its literal type image. Full operator relations and graph
edge residuals are preserved and reflected. No arithmetic factor theory is used.
```

Direct project imports: `PaperXI.TensorTypeOperators70`.

Line	Kind	Name
17	def	`PaperXI.PacketTypeAttachment70.typeAction`
21	theorem	`PaperXI.PacketTypeAttachment70.ambient_action`
32	theorem	`PaperXI.PacketTypeAttachment70.first_factor_action`
47	theorem	`PaperXI.PacketTypeAttachment70.first_factor_average`
61	theorem	`PaperXI.PacketTypeAttachment70.generator_action`
71	theorem	`PaperXI.PacketTypeAttachment70.edge_compatibility_iff`
79	theorem	`PaperXI.PacketTypeAttachment70.edge_residual_iff`

A.317 PaperXI.PairIndex110

Selected layer: Continuation110. **Source lines:** 122.

SHA-256: `453d18235bd27cd82e48a7cf367e7b3433288906412d7e7154e80b955025408b`.

```
The pair group is identified with the first-coordinate intersection.
The orientation is literal: `ι b = g * ι a * g⁻¹`. The intersection in `K`
therefore contains the `a` whose conjugated image remains in `ι(K)`.
All results are generic group and index results; no arithmetic identity is assumed.
```

Direct project imports: `PaperXI.IntersectionModel108`.

Line	Kind	Name
19	def	`PaperXI.PairIndex110.intersection`
23	def	`PaperXI.PairIndex110.scalarIntersection`
26	theorem	`PaperXI.PairIndex110.scalarIntersection_le`
29	theorem	`PaperXI.PairIndex110.scalarIntersection_le_intersection`
35	theorem	`PaperXI.PairIndex110.intersection_image`
47	theorem	`PaperXI.PairIndex110.scalarIntersection_image`
58	theorem	`PaperXI.PairIndex110.left_mem_intersection`
62	theorem	`PaperXI.PairIndex110.left_range`
71	theorem	`PaperXI.PairIndex110.left_injective`
82	def	`PaperXI.PairIndex110.pairEquivIntersection`
91	theorem	`PaperXI.PairIndex110.pairSupport_eq_comap`
106	theorem	`PaperXI.PairIndex110.pairSupport_index`
113	theorem	`PaperXI.PairIndex110.pairSupport_index_ambient`

A.318 `PaperXI.PeriodicLatticeReduction60`

Selected layer: base108 arithmetic branch. **Source lines:** 86.

SHA-256: `f70d40f2ca38dfa0635a4232d830d16dbbdc2f58c4472d1c82c4a25950e93a02`.

```
# Finite reduction of the literal lattice-chain conditions

A positive period given by a base-field scalar reduces the entire integer
chain to finitely many levels for every base-field linear automorphism.
```

Direct project imports: `PaperXI.ActualELatticeNormalizer59`.

Line	Kind	Name
17	def	`PaperXI.PeriodicLatticeReduction60.levelCondition`
21	theorem	`PaperXI.PeriodicLatticeReduction60.levelCondition_shift`
53	theorem	`PaperXI.PeriodicLatticeReduction60.levelCondition_periodic`
60	theorem	`PaperXI.PeriodicLatticeReduction60.mem_firstCongruence_iff_fin`

A.319 `PaperXI.PiPolynomial`

Selected layer: v0.44 parent. **Source lines:** 58.

SHA-256: `3aaecfa5d206cd6d476431f7e9f3f823ec7b57ec488a1edb5d6170a3c973496e`.

```
Characteristic and local polynomials of finite families of varying spaces.
```

Direct project imports: `PaperXI.InvariantPolynomial`.

Line	Kind	Name
13	theorem	`PaperXI.PiPolynomial.charpoly_pi`
46	theorem	`PaperXI.PiPolynomial.operatorPolynomial_pi`

A.320 PaperXI.PolynomialAssembly

Selected layer: v0.44 parent. **Source lines:** 83.

SHA-256: 538480496e4eb1bceb57fd376c890da010064e5ca4d64edc62ceb4394e77663f.

Restricted Frobenius and determinant products under equivariant decompositions.

Direct project imports: PaperXI.InvariantAssembly, PaperXI.PiPolynomial.

Line	Kind	Name
14	def	PaperXI.PolynomialAssembly.restricted
20	theorem	PaperXI.PolynomialAssembly.restricted_apply
24	theorem	PaperXI.PolynomialAssembly.stable_of_decomposition
37	theorem	PaperXI.PolynomialAssembly.restricted_conj
53	theorem	PaperXI.PolynomialAssembly.polynomial_product
67	def	PaperXI.PolynomialAssembly.localFactor
71	theorem	PaperXI.PolynomialAssembly.localFactor_product

A.321 PaperXI.PrescribedFieldType70

Selected layer: base108 arithmetic branch. **Source lines:** 87.

SHA-256: 6cd430fa9358f4bcd875eaede181bed47500c5d2d71f94e4d937f10231a5d88f.

Construction of the full inducing representation from the same compact
first factor and actual finite quotient. The full-H locus is a later condition
on this exact constructed extension, not on an unrelated supplied Lambda.

Direct project imports: PaperXI.ActualCompactTypeCompletion58, PaperXI.ActualFieldTypeQuotient69.

Line	Kind	Name
28	def	PaperXI.PrescribedFieldType70.compactInclusion
31	def	PaperXI.PrescribedFieldType70.fieldGenerator
37	def	PaperXI.PrescribedFieldType70.domainEquiv
44	theorem	PaperXI.PrescribedFieldType70.exists_prescribed_extension

A.322 PaperXI.PrimitiveCharacterBlock

Selected layer: v0.44 parent. **Source lines:** 128.

SHA-256: 5950f0f65ec50d3dc6a4c36c40237c087ccc4a4c0eeb35b1fe536e897bcdf6e5.

Primitive central character blocks in the literal finite group algebra.
Primitivity means absence of nontrivial central sub-idempotents, not rank one.

Direct project imports: PaperXI.SemisimpleDetection.

Line	Kind	Name
17	theorem	PaperXI.PrimitiveCharacterBlock.equiv_algebra_action
26	theorem	PaperXI.PrimitiveCharacterBlock.nonzero
35	theorem	PaperXI.PrimitiveCharacterBlock.idempotent

Line	Kind	Name
43	theorem	`PaperXI.PrimitiveCharacterBlock.supported_eq_zero`
60	theorem	`PaperXI.PrimitiveCharacterBlock.block_action_injective`
69	def	`PaperXI.PrimitiveCharacterBlock.IsPrimitiveCentral`
74	theorem	`PaperXI.PrimitiveCharacterBlock.primitive_central`
99	theorem	`PaperXI.PrimitiveCharacterBlock.no_nontrivial_central_split`
112	theorem	`PaperXI.PrimitiveCharacterBlock.orthogonal`
123	theorem	`PaperXI.PrimitiveCharacterBlock.presentation_invariant`

A.323 `PaperXI.ProductNormalizerQuotient59`

Selected layer: base108 arithmetic branch. **Source lines:** 95.

SHA-256: `253347996905c91f7605f14539fa7abaf854e23c86fc22008272369cf3bd30bd`.

```
# Product subgroups and the action of a centralizing normalizer

For actual subgroups `H,N` of an ambient group, with `H` normalizing `N`,
the quotient of `H ⊔ N` by `N` is the quotient of `H` by its intersection
with `N`. An ambient element centralizing `H` and normalizing `N` acts
trivially on this quotient. No identification with a simple-character group
is asserted here.
```

Direct project imports: `PaperXI.NormalizerExtensionInducing58`.

Line	Kind	Name
25	theorem	`PaperXI.ProductNormalizerQuotient59.normalizes_sup`
32	def	`PaperXI.ProductNormalizerQuotient59.conjugation`
35	theorem	`PaperXI.ProductNormalizerQuotient59.conjugation_coe`
40	theorem	`PaperXI.ProductNormalizerQuotient59.map_conjugation`
63	theorem	`PaperXI.ProductNormalizerQuotient59.quotient_conjugation`
73	def	`PaperXI.ProductNormalizerQuotient59.productQuotientEquiv`
83	def	`PaperXI.ProductNormalizerQuotient59.quotientEquivOfKernel`

A.324 `PaperXI.ProjectiveCocycle`

Selected layer: v0.44 parent. **Source lines:** 171.

SHA-256: `d2eb7d728fef4b48932df07f3dc13a003f39995cc477c3728f0677302794dcf2`.

```
Scalar cocycles extracted from the multiplication of actual nonzero operators.
Associativity and normalization are conclusions of operator identities.
```

Direct project imports: `PaperXI.CocycleCommutator`, `PaperXI.SchurScalar`.

Line	Kind	Name
14	theorem	`PaperXI.ProjectiveCocycle.scalar_injective`
21	def	`PaperXI.ProjectiveCocycle.cocycle`
32	theorem	`PaperXI.ProjectiveCocycle.normalized`
45	theorem	`PaperXI.ProjectiveCocycle.coefficient_unique`
53	theorem	`PaperXI.ProjectiveCocycle.rescale_coefficients`
72	theorem	`PaperXI.ProjectiveCocycle.commutator_rescale_operators`

Line	Kind	Name
89	theorem	PaperXI.ProjectiveCocycle.cocycle_rescale_operators
101	theorem	PaperXI.ProjectiveCocycle.cochain_one
109	theorem	PaperXI.ProjectiveCocycle.rescaledFamily_mul
123	theorem	PaperXI.ProjectiveCocycle.conjugate_mul
133	theorem	PaperXI.ProjectiveCocycle.conjugate_rescale_coefficients
149	theorem	PaperXI.ProjectiveCocycle.conjugate_rescale_commutator

A.325 PaperXI.ProjectiveHolonomy

Selected layer: v0.44 parent. **Source lines:** 173.

SHA-256: dbc14aa36dc916019628345b811c2d1904c4652d23cddbe71792a93751a8db05.

```
Vertex rescaling of actual graph transport and scalar ratio characters.
```

Direct project imports: PaperXI.ScalarHolonomy, PaperXI.GraphTransport, PaperXI.GraphLoopRepresentation.

Line	Kind	Name
20	def	PaperXI.ProjectiveHolonomy.rescale
24	theorem	PaperXI.ProjectiveHolonomy.rescale_apply
31	theorem	PaperXI.ProjectiveHolonomy.rescale_gauge
39	theorem	PaperXI.ProjectiveHolonomy.rescale_path
50	theorem	PaperXI.ProjectiveHolonomy.rescale_path_apply
59	theorem	PaperXI.ProjectiveHolonomy.rescale_loop
66	theorem	PaperXI.ProjectiveHolonomy.rescale_holonomy
75	theorem	PaperXI.ProjectiveHolonomy.rescale_isScalar
81	theorem	PaperXI.ProjectiveHolonomy.isScalar_iff_paths
93	def	PaperXI.ProjectiveHolonomy.loopCharacter
97	theorem	PaperXI.ProjectiveHolonomy.loopCharacter_spec
105	theorem	PaperXI.ProjectiveHolonomy.rescale_character
116	def	PaperXI.ProjectiveHolonomy.ratioH1
119	theorem	PaperXI.ProjectiveHolonomy.ratioH1_class
123	theorem	PaperXI.ProjectiveHolonomy.ratioH1_unique
129	theorem	PaperXI.ProjectiveHolonomy.ratioH1_eq_one_iff
145	def	PaperXI.ProjectiveHolonomy.transportRatioH1
152	theorem	PaperXI.ProjectiveHolonomy.transportRatioH1_class
162	theorem	PaperXI.ProjectiveHolonomy.transportRatioH1_rescale

A.326 PaperXI.ProjectorTransport

Selected layer: v0.44 parent. **Source lines:** 27.

SHA-256: c88eba65d49511600eaef67165160b9de7d6f1dcb5c6f56476feb8a5259b3592.

```
Exact reindexing law underlying the Frobenius covariance in Proposition 3.1.
This does not yet prove Fourier idempotence or the Weil-group realization.
The character weight transforms by the inverse permutation.
```

Direct project imports: none.

Line	Kind	Name
14	theorem	`PaperXI.ProjectorTransport.weighted_sum_conjugation`
21	theorem	`PaperXI.ProjectorTransport.normalized_sum_conjugation`

A.327 `PaperXI.QuotientRepresentation`

Selected layer: v0.44 parent. **Source lines:** 72.

SHA-256: `4439d25c13ba1ce6c73f18c0d67c0a8ce5f96caa0c1befb335168c02a01fffe5`.

```
Actual representation descent through a quotient group and preservation
of irreducibility and intertwining equations under a surjective group map.
```

Direct project imports: none.

Line	Kind	Name
14	def	`PaperXI.QuotientRepresentation.subrepresentationEquiv`
25	theorem	`PaperXI.QuotientRepresentation.irreducible_comp_iff`
30	theorem	`PaperXI.QuotientRepresentation.intertwines_comp_iff`
42	def	`PaperXI.QuotientRepresentation.descend`
46	theorem	`PaperXI.QuotientRepresentation.descend_apply`
49	theorem	`PaperXI.QuotientRepresentation.descend_comp`
52	theorem	`PaperXI.QuotientRepresentation.descend_irreducible_iff`
58	theorem	`PaperXI.QuotientRepresentation.descend_irreducible`
64	theorem	`PaperXI.QuotientRepresentation.descend_unique`

A.328 `PaperXI.RadicalOnePlus111`

Selected layer: Continuation111. **Source lines:** 58.

SHA-256: `3ebe8185adccd13a0c9d9102e437f22fb910637ef074bfb50331602f7eac41a7`.

```
Exact one-plus groups from actual radical ideals in the original order.
The order may be noncommutative. No arbitrary additive lattice is declared
to be a group or an ideal of the full field endomorphism algebra.
```

Direct project imports: `PaperXI.OnePlusIntersection110`.

Line	Kind	Name
16	theorem	`PaperXI.RadicalOnePlus111.isUnit_of_sub_one_mem_jacobson`
34	def	`PaperXI.RadicalOnePlus111.onePlus`
36	theorem	`PaperXI.RadicalOnePlus111.onePlus_offsets`
40	theorem	`PaperXI.RadicalOnePlus111.onePlus_surjective`
53	def	`PaperXI.RadicalOnePlus111.offsetEquiv`

A.329 `PaperXI.RamificationPolyphase`

Selected layer: v0.44 parent. **Source lines:** 326.

SHA-256: `2f34bd694def457221d6f00b874c8b54547dcc95ee249e3afcf34d71b5b64549`.

```
# Ramification polyphase projectors

The lower ramification filtration gives a nested family of literal finite-group
averaging projectors.  Successive differences are the depth-detail channels.
This is a representation-theoretic multiresolution theorem; it does not assert
existence of an intrinsic wild-break decomposition beyond the supplied lower
filtration.
```

Direct project imports: `PaperXI.ValuationConductor`, `PaperXI.FiniteAverage`.

Line	Kind	Name
29	theorem	`PaperXI.RamificationPolyphase.average_comp_operator`
40	theorem	`PaperXI.RamificationPolyphase.average_comp_restricted_average`
53	theorem	`PaperXI.RamificationPolyphase.restricted_average_comp_average`
69	def	`PaperXI.RamificationPolyphase.lowerInclusion`
75	theorem	`PaperXI.RamificationPolyphase.lowerInclusion_val`
80	theorem	`PaperXI.RamificationPolyphase.lift_lowerInclusion`
84	def	`PaperXI.RamificationPolyphase.depthProjector`
88	theorem	`PaperXI.RamificationPolyphase.depthProjector_range`
93	theorem	`PaperXI.RamificationPolyphase.depthProjector_idempotent`
100	theorem	`PaperXI.RamificationPolyphase.depthProjector_comp_of_le`
114	theorem	`PaperXI.RamificationPolyphase.depthProjector_comp_of_le_rev`
127	def	`PaperXI.RamificationPolyphase.detailProjector`
130	theorem	`PaperXI.RamificationPolyphase.detailProjector_apply`
135	theorem	`PaperXI.RamificationPolyphase.detailProjector_idempotent`
167	theorem	`PaperXI.RamificationPolyphase.depthProjector_detailProjector`
186	theorem	`PaperXI.RamificationPolyphase.depthProjector_succ_detailProjector`
206	theorem	`PaperXI.RamificationPolyphase.detailProjector_mem_next_fixed`
215	theorem	`PaperXI.RamificationPolyphase.detailProjector_range`
232	theorem	`PaperXI.RamificationPolyphase.detailProjector_comp_eq_zero`
265	theorem	`PaperXI.RamificationPolyphase.detailProjector_comp_eq_zero_rev`
299	theorem	`PaperXI.RamificationPolyphase.depthProjector_eq_zero_add_sum`
311	theorem	`PaperXI.RamificationPolyphase.identity_eq_inertia_add_details`

A.330 `PaperXI.RamificationPolyphaseBaseChange`

Selected layer: v0.44 parent. **Source lines:** 115.

SHA-256: `eb07cacbdaafae8606ca94ef6790ab0e3c7df865ff062943f1757f3ca826daa2`.

```
# Unramified transport of ramification polyphase channels

For an inertia-containing subgroup and the corresponding Frobenius power
change, every literal lower-depth averaging projector is preserved. Hence every
successive ramification-detail projector is preserved as an actual endomorphism,
not only at the level of dimensions or conductor sums.
```

Direct project imports: `PaperXI.RamificationPolyphase`, `PaperXI.UnramifiedConductor`.

Line	Kind	Name
36	theorem	`PaperXI.RamificationPolyphaseBaseChange.depthProjector_eq`
59	theorem	`PaperXI.RamificationPolyphaseBaseChange.detailProjector_eq`
69	theorem	`PaperXI.RamificationPolyphaseBaseChange.detailRange_eq`
78	theorem	`PaperXI.RamificationPolyphaseBaseChange.finiteReconstruction_eq`
96	theorem	`PaperXI.RamificationPolyphaseBaseChange.conductorSummand_eq`

Line	Kind	Name
103	theorem	`PaperXI.RamificationPolyphaseBaseChange.swan_eq`
110	theorem	`PaperXI.RamificationPolyphaseBaseChange.artin_eq`

A.331 `PaperXI.RamificationPolyphaseDimensions`

Selected layer: v0.44 parent. **Source lines:** 180.

SHA-256: `f38214cac70955d55434afad4ca9162cd9145d6a627a88311444bac1906e871d`.

```
# Dimension and conductor accounting for ramification details

The literal depth projectors split every successive fixed space into its previous
coarse part and one newly visible detail range.  In finite dimension, the detail
ranks telescope to the whole representation and give a finite tail formula for
every lower-numbering codimension.
```

Direct project imports: `PaperXI.RamificationPolyphase`, `PaperXI.RamificationSeparation`.

Line	Kind	Name
31	def	`PaperXI.RamificationPolyphaseDimensions.detailSpace`
37	theorem	`PaperXI.RamificationPolyphaseDimensions.fixed_mono`
47	theorem	`PaperXI.RamificationPolyphaseDimensions.fixed_succ_eq_sup_detail`
76	theorem	`PaperXI.RamificationPolyphaseDimensions.fixed_disjoint_detail`
97	theorem	`PaperXI.RamificationPolyphaseDimensions.finrank_fixed_succ`
108	theorem	`PaperXI.RamificationPolyphaseDimensions.finrank_fixed_eq_base_add_details`
119	theorem	`PaperXI.RamificationPolyphaseDimensions.finrank_eq_inertia_add_details`
135	theorem	`PaperXI.RamificationPolyphaseDimensions.exists_identity_eq_inertia_add_details`
146	theorem	`PaperXI.RamificationPolyphaseDimensions.exists_finrank_eq_inertia_add_details`
157	theorem	`PaperXI.RamificationPolyphaseDimensions.codimension_eq_detail_tail`
170	theorem	`PaperXI.RamificationPolyphaseDimensions.summand_eq_weight_mul_detail_tail`

A.332 `PaperXI.RamificationSeparation`

Selected layer: v0.44 parent. **Source lines:** 78.

SHA-256: `f4b0880d3b76829d7b9274fe0f6817a408aa233aed0a8530687b2ec95f7c6094`.

```
Separated local rings and finite faithful automorphism groups have a terminating
lower congruence filtration. This justifies finite support of the conductor sum;
eventual triviality is a theorem, not an assumed cutoff.
```

Direct project imports: `PaperXI.ValuationRamification`.

Line	Kind	Name
17	theorem	`PaperXI.RamificationSeparation.mem_all_lower_iff`
37	theorem	`PaperXI.RamificationSeparation.eventually_trivial`
59	instance	`PaperXI.RamificationSeparation.valuation_faithful`
72	theorem	`PaperXI.RamificationSeparation.valuation_eventually_trivial`

A.333 PaperXI.RamifiedUnramifiedQuotient

Selected layer: v0.44 parent. **Source lines:** 322.

SHA-256: `755d457b0f11dfbac94eaee10326fa290bd5b6a331d8883bb8ce26811f00a747`.

```
# Ramified pullbacks and their inertia-free degree quotient

The residue--Frobenius pullback retains a decomposition-group coordinate and an
integer degree.  Its degree kernel is inertia.  This file proves the precise
first-isomorphism statement and compares the quotient with the integer-degree
unramified model only under an explicit degree-one Frobenius-lift hypothesis.

It also proves the exact representation-theoretic information loss: a
representation pulled back only through integer degree kills every lower
ramification subgroup, so every ramification depth projector is the identity
and every positive detail projector is zero.
```

Direct project imports: `PaperXI.RamificationPolyphase`, `PaperXI.FiniteUnramifiedWeil`.

Line	Kind	Name
38	abbrev	`PaperXI.RamifiedUnramifiedQuotient.Inertia`
42	def	`PaperXI.RamifiedUnramifiedQuotient.quotientEquivDegreeRange`
47	theorem	`PaperXI.RamifiedUnramifiedQuotient.degree_surjective_of_degree_one`
59	theorem	`PaperXI.RamifiedUnramifiedQuotient.degree_surjective_of_frobenius_mem_range`
66	def	`PaperXI.RamifiedUnramifiedQuotient.quotientEquivIntegerDegree`
72	theorem	`PaperXI.RamifiedUnramifiedQuotient.quotientEquivIntegerDegree_mk`
79	theorem	`PaperXI.RamifiedUnramifiedQuotient.frobenius_lift_ratio_mem_inertia`
102	def	`PaperXI.RamifiedUnramifiedQuotient.degreeModelMap`
107	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeModelMap_degree`
112	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeModelMap_ker`
119	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeModelMap_surjective`
130	def	`PaperXI.RamifiedUnramifiedQuotient.quotientEquivUnramified`
136	theorem	`PaperXI.RamifiedUnramifiedQuotient.quotientEquivUnramified_mk_degree`
145	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeModelMap_eq_geometricLift_of_degree_one`
158	def	`PaperXI.RamifiedUnramifiedQuotient.pullbackDegreeRepresentation`
164	theorem	`PaperXI.RamifiedUnramifiedQuotient.inertia_le_pullbackDegreeRepresentation_ker`
175	def	`PaperXI.RamifiedUnramifiedQuotient.quotientRepresentation`
183	def	`PaperXI.RamifiedUnramifiedQuotient.descendRepresentation`
192	theorem	`PaperXI.RamifiedUnramifiedQuotient.descendRepresentation_comp_degreeModelMap`
215	theorem	`PaperXI.RamifiedUnramifiedQuotient.descendRepresentation_unique`
250	abbrev	`PaperXI.RamifiedUnramifiedQuotient.degreeOnlyRepresentation`
257	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeOnlyRepresentation_lift_eq_one`
271	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeOnly_fixed_eq_top`
280	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeOnly_depthProjector_eq_id`
292	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeOnly_detailProjector_eq_zero`
301	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeOnly_summand_eq_zero`
308	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeOnly_swan_eq_zero`
314	theorem	`PaperXI.RamifiedUnramifiedQuotient.degreeOnly_artin_eq_zero`

A.334 PaperXI.RankOneEulerFactor

Selected layer: v0.44 parent. **Source lines:** 113.

SHA-256: `a512466f456be1c2a7b5c0f49b6b7cfad6bbda69ec1bc75283bb51b8140b95c3`.

```
# Rank-one Euler factors for irreducible unramified representations
```

Direct project imports: `PaperXI.UnramifiedConductorZero`.

Line	Kind	Name
27	theorem	`PaperXI.FiniteUnramifiedWeil.reverse_X_sub_C`
38	theorem	`PaperXI.FiniteUnramifiedWeil.reverse_charpoly_smul_id`
59	def	`PaperXI.FiniteUnramifiedWeil.frobeniusEigenvalue`
65	theorem	`PaperXI.FiniteUnramifiedWeil.geometricFrobenius_action_eq_smul_id`
73	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusEigenvalue_ne_zero`
92	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_eq_linear`
100	def	`PaperXI.FiniteUnramifiedWeil.rankOneEulerFactor`
105	theorem	`PaperXI.FiniteUnramifiedWeil.unramifiedLocalFactor_eq_rankOneEulerFactor`

A.335 `PaperXI.RankOneSourceCandidate110`

Selected layer: Continuation110. **Source lines:** 77.

SHA-256: `7f57ede141636d625efea3d9b67e905b1f34bcd5dd6c475f31a7e2ec2a7b2928`.

```
Residue rank one from the lower-level simple-character support dictionary.
The model containment, compact support, integral normalization and finite
residue factor are derived. Canonical character data are not manufactured.
```

Direct project imports: `PaperXI.ArithmeticAttachment110`, `PaperXI.NormalizerDictionary110`.

Line	Kind	Name
29	theorem	`PaperXI.RankOneSourceCandidate110.candidate_exists`

A.336 `PaperXI.RegularCyclic`

Selected layer: v0.44 parent. **Source lines:** 90.

SHA-256: `2ed6fa5d0f002e4a087b67badeca9c39702efe1de6d6feade2a7eaf34fc5bccf`.

```
A concrete regular cyclic model. Every character packet is nonzero, and
the erroneous forward-label rule has an actual complex representation counterexample.
```

Direct project imports: `PaperXI.TameProjectors`.

Line	Kind	Name
12	def	`PaperXI.RegularCyclic.translation`
17	def	`PaperXI.RegularCyclic.regular`
30	theorem	`PaperXI.RegularCyclic.operator_apply`
33	def	`PaperXI.RegularCyclic.frobenius`
41	theorem	`PaperXI.RegularCyclic.generator_relation`
50	theorem	`PaperXI.RegularCyclic.character_vector_mem`
59	theorem	`PaperXI.RegularCyclic.projector_nonzero`
67	theorem	`PaperXI.RegularCyclic.projectors_distinct`
76	def	`PaperXI.RegularCyclic.twoModFive`
79	theorem	`PaperXI.RegularCyclic.wrong_forward_label`

A.337 PaperXI.RepresentationGraph

Selected layer: v0.44 parent. **Source lines:** 106.

SHA-256: 5919599a901246a3def38a1fc9ae5480fabe1dc0a2f1c09be274d25317788d6d.

```
Gluing within actual representation intertwiners. Edge covariance ensures
that tree transport of a root bridge remains a local bridge at every vertex.
```

Direct project imports: PaperXI.GraphGauge.

Line	Kind	Name
19	def	PaperXI.RepresentationGraph.EdgeEquivariant
25	def	PaperXI.RepresentationGraph.Intertwines
34	theorem	PaperXI.RepresentationGraph.path_equivariant
42	theorem	PaperXI.RepresentationGraph.rootGauge_intertwines
55	theorem	PaperXI.RepresentationGraph.based_iff_chords
71	theorem	PaperXI.RepresentationGraph.connected_iff_chords
81	theorem	PaperXI.RepresentationGraph.based_iff_loops
91	theorem	PaperXI.RepresentationGraph.unbased_iff_conjugates

A.338 PaperXI.RepresentationStability

Selected layer: v0.44 parent. **Source lines:** 47.

SHA-256: ea48bcbf4e471e1d436ca33befd6c1aa3dcec5b7e765b15b407087b7c26cd33c.

```
An explicit invariant-subspace criterion for actual irreducibility. This
interface keeps the nonzero-space condition when comparing group actions and
algebra actions.
```

Direct project imports: PaperXI.GeneratedIrreducibility.

Line	Kind	Name
13	def	PaperXI.RepresentationStability.Stable
17	theorem	PaperXI.RepresentationStability.eq_bot_or_top
25	theorem	PaperXI.RepresentationStability.irreducible_of_stable
36	theorem	PaperXI.RepresentationStability.irreducible_iff

A.339 PaperXI.ResidueDegreeBaseChange

Selected layer: v0.44 parent. **Source lines:** 148.

SHA-256: c581b1a747917d5c92d930d8b817fd79fb5bf2c3de951682b786377850705bee.

```
Integer-degree base change for residue-action pullbacks.
The map (g,m) ↦ (g,n*m) is constructed; neither subgroup equality nor index is
assumed. This does not identify a pullback with an absolute local Weil group.
```

Direct project imports: PaperXI.ResidueWeil.

Line	Kind	Name
15	def	PaperXI.ResidueDegreeBaseChange.degreeMod
24	def	PaperXI.ResidueDegreeBaseChange.periodSubgroup
27	instance	PaperXI.ResidueDegreeBaseChange.period_normal
30	theorem	PaperXI.ResidueDegreeBaseChange.mem_periodSubgroup
36	def	PaperXI.ResidueDegreeBaseChange.inclusion
50	theorem	PaperXI.ResidueDegreeBaseChange.inclusion_projection
53	theorem	PaperXI.ResidueDegreeBaseChange.inclusion_degree
57	theorem	PaperXI.ResidueDegreeBaseChange.inclusion_injective
69	theorem	PaperXI.ResidueDegreeBaseChange.inclusion_range
89	def	PaperXI.ResidueDegreeBaseChange.periodEquiv
95	theorem	PaperXI.ResidueDegreeBaseChange.degree_surjective
105	theorem	PaperXI.ResidueDegreeBaseChange.degreeMod_surjective
116	def	PaperXI.ResidueDegreeBaseChange.quotientEquiv
121	theorem	PaperXI.ResidueDegreeBaseChange.quotient_card
127	theorem	PaperXI.ResidueDegreeBaseChange.inertia_preimage
141	theorem	PaperXI.ResidueDegreeBaseChange.tower_degree

A.340 PaperXI.ResidueExtensionRealization

Selected layer: v0.44 parent. **Source lines:** 97.

SHA-256: b24752381c6a6353645a53ffe2613d10e783ecf9086a2cdfb5876dba34bc9ccf.

```
Realization of any finite separable extension of the actual base residue field.
The monic lift and its ring/field are constructed; no ambient Galois extension,
residue embedding into an existing extension, or degree certificate is supplied.
```

Direct project imports: PaperXI.UnramifiedPolynomial.

Line	Kind	Name
16	def	PaperXI.ResidueExtensionRealization.residueBasis
19	def	PaperXI.ResidueExtensionRealization.residuePolynomial
22	theorem	PaperXI.ResidueExtensionRealization.residuePolynomial_monic
25	theorem	PaperXI.ResidueExtensionRealization.residuePolynomial_irreducible
28	theorem	PaperXI.ResidueExtensionRealization.residuePolynomial_degree
33	theorem	PaperXI.ResidueExtensionRealization.exists_monic_lift
43	def	PaperXI.ResidueExtensionRealization.liftedPolynomial
45	theorem	PaperXI.ResidueExtensionRealization.liftedPolynomial_map
49	theorem	PaperXI.ResidueExtensionRealization.liftedPolynomial_degree
53	instance	PaperXI.ResidueExtensionRealization.monic
56	instance	PaperXI.ResidueExtensionRealization.residueIrreducible
60	abbrev	PaperXI.ResidueExtensionRealization.ring
61	abbrev	PaperXI.ResidueExtensionRealization.field
64	def	PaperXI.ResidueExtensionRealization.residueEquiv
72	theorem	PaperXI.ResidueExtensionRealization.rank
76	theorem	PaperXI.ResidueExtensionRealization.field_degree
82	theorem	PaperXI.ResidueExtensionRealization.etale
85	theorem	PaperXI.ResidueExtensionRealization.ramificationIdx_one
88	theorem	PaperXI.ResidueExtensionRealization.source

A.341 PaperXI.ResidueFrobeniusAttachment

Selected layer: v0.44 parent. **Source lines:** 144.

SHA-256: `f8bdcb103727ce29ab4fa9647716c21624920c75aff28d23c3a398cee0f59f7c`.

```
# Frobenius attachment to the constructed residue field
```

Direct project imports: `PaperXI.IntegerRingComparison`.

Line	Kind	Name
23	abbrev	`PaperXI.FiniteUnramifiedWeil.UpperField`
24	abbrev	`PaperXI.FiniteUnramifiedWeil.BaseResidue`
25	abbrev	`PaperXI.FiniteUnramifiedWeil.UpperResidue`
26	abbrev	`PaperXI.FiniteUnramifiedWeil.ResidueGalois`
27	abbrev	`PaperXI.FiniteUnramifiedWeil.ResidueDegree`
36	def	`PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius`
40	def	`PaperXI.FiniteUnramifiedWeil.geometricFrobenius`
44	theorem	`PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_apply`
49	theorem	`PaperXI.FiniteUnramifiedWeil.geometricFrobenius_inverse_power`
55	theorem	`PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_order`
61	theorem	`PaperXI.FiniteUnramifiedWeil.geometricFrobenius_order`
67	theorem	`PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_order_eq_fieldDegree`
73	theorem	`PaperXI.FiniteUnramifiedWeil.geometricFrobenius_order_eq_fieldDegree`
79	theorem	`PaperXI.FiniteUnramifiedWeil.natCard_residueGalois`
85	theorem	`PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_mem_zpowers`
95	theorem	`PaperXI.FiniteUnramifiedWeil.arithmeticFrobenius_zpowers_eq_top`
101	theorem	`PaperXI.FiniteUnramifiedWeil.geometricFrobenius_zpowers_eq_top`
107	def	`PaperXI.FiniteUnramifiedWeil.geometricCycleEquiv`
116	theorem	`PaperXI.FiniteUnramifiedWeil.geometricCycleEquiv_one`
122	def	`PaperXI.FiniteUnramifiedWeil.prescribedResidueGaloisEquiv`
127	theorem	`PaperXI.FiniteUnramifiedWeil.prescribedResidueGaloisEquiv_arithmetic`
138	theorem	`PaperXI.FiniteUnramifiedWeil.prescribedResidueGaloisEquiv_geometric`

A.342 PaperXI.ResidueRestriction107

Selected layer: base108 arithmetic branch. **Source lines:** 75.

SHA-256: `d1008a38fcb166fa8167e9591321de444cc9d86d3ae176b13511e4639d0ab405`.

```
The previously constructed residue map agrees with literal entrywise
reduction on integral centralizer matrices. This identity is proved, not supplied.
```

Direct project imports: `PaperXI.ActualCuspidalCandidate105`.

Line	Kind	Name
12	theorem	`PaperXI.ResidueRestriction107.quotient_inclusion`
44	def	`PaperXI.ResidueRestriction107.integralInclusion`
49	theorem	`PaperXI.ResidueRestriction107.integralInclusion_coe`
56	theorem	`PaperXI.ResidueRestriction107.reduction_integral`

A.343 PaperXI.ResidueWeil

Selected layer: v0.44 parent. **Source lines:** 114.

SHA-256: `1eee4bb854ff6f99731ba944d6c0540ef7595df6fb8fa862ae030132fcd38166`.

```
A pullback of an actual residue-field action along the integer powers of a
chosen residue automorphism. No equivalence with the absolute local Weil group
is postulated. The inertia identification below is with Mathlib's valuation
inertia subgroup, not with an independently named abstract kernel.
```

Direct project imports: `PaperXI.NormalInertia`.

Line	Kind	Name
18	def	`PaperXI.ResidueWeil.subgroup`
30	abbrev	`PaperXI.ResidueWeil.Pullback`
32	def	`PaperXI.ResidueWeil.projection`
35	def	`PaperXI.ResidueWeil.degree`
38	theorem	`PaperXI.ResidueWeil.projection_apply`
41	theorem	`PaperXI.ResidueWeil.degree_apply`
45	def	`PaperXI.ResidueWeil.inertiaEquiv`
65	instance	`PaperXI.ResidueWeil.degreeQuotientCommutative`
73	theorem	`PaperXI.ResidueWeil.exists_degree_one_iff`
90	abbrev	`PaperXI.ResidueWeil.residueAction`
94	abbrev	`PaperXI.ResidueWeil.valuationGroup`
98	def	`PaperXI.ResidueWeil.valuationInertiaEquiv`
106	theorem	`PaperXI.ResidueWeil.valuation_invariants_eq_bot`

A.344 PaperXI.RestrictedHeisenbergBridge

Selected layer: v0.44 parent. **Source lines:** 84.

SHA-256: `4385767418858294f763e1ef3b8a60700679a6399d58874349e83ce8533e5af6`.

```
Source 11.2: the initial bridge between the supplied Heisenberg envelopes.
The odd branch constructs the lift on arbitrary abstract symplectic spaces.
The dyadic branch retains the supplied isomorphism of actual central
extensions. Neither branch assumes an initial intertwiner, scalar-cocycle
nondegeneracy, or finite-dimensionality of either representation space.
```

Direct project imports: `PaperXI.CentralExtensionComparison`, `PaperXI.AbstractHeisenbergExtension`, `PaperXI.SymplecticCentralExtension`.

Line	Kind	Name
24	theorem	`PaperXI.RestrictedHeisenbergBridge.odd_source`
58	theorem	`PaperXI.RestrictedHeisenbergBridge.dyadic_source`

A.345 PaperXI.ScalarCocycle112

Selected layer: Continuation112. **Source lines:** 195.

SHA-256: `351ac3d1e21afeb0a8897f6eea1ba99f83bb63eaeee390eebefa44c1dfbb0f4e`.

```
Scalar cocycle classification for complex projective comparison.
No identification of the full finite centers or dyadic group lift is assumed.
```

Direct project imports: none.

Line	Kind	Name
16	structure	`PaperXI.ScalarCocycle112.Cocycle`
23	def	`PaperXI.ScalarCocycle112.commutator`
26	structure	`PaperXI.ScalarCocycle112.Extension`
33	theorem	`PaperXI.ScalarCocycle112.Extension.ext`
42	theorem	`PaperXI.ScalarCocycle112.Extension.mul_scalar`
44	theorem	`PaperXI.ScalarCocycle112.Extension.mul_base`
45	theorem	`PaperXI.ScalarCocycle112.Extension.one_scalar`
46	theorem	`PaperXI.ScalarCocycle112.Extension.one_base`
69	def	`PaperXI.ScalarCocycle112.Extension.commGroup`
78	def	`PaperXI.ScalarCocycle112.Extension.scalarInclusion`
83	theorem	`PaperXI.ScalarCocycle112.Extension.scalarInclusion_scalar`
85	theorem	`PaperXI.ScalarCocycle112.Extension.scalarInclusion_base`
88	theorem	`PaperXI.ScalarCocycle112.Extension.scalarInclusion_injective`
92	def	`PaperXI.ScalarCocycle112.Extension.sectionOf`
94	theorem	`PaperXI.ScalarCocycle112.Extension.sectionOf_one`
96	theorem	`PaperXI.ScalarCocycle112.Extension.sectionOf_mul`
104	theorem	`PaperXI.ScalarCocycle112.symmetric_exists_cochain`
123	def	`PaperXI.ScalarCocycle112.ratio`
131	theorem	`PaperXI.ScalarCocycle112.ratio_symmetric`
141	theorem	`PaperXI.ScalarCocycle112.same_commutator_exists_cochain_of_divisible`
150	theorem	`PaperXI.ScalarCocycle112.complex_units_exists_pow`
162	def	`PaperXI.ScalarCocycle112.complexUnitsDivisible`
171	theorem	`PaperXI.ScalarCocycle112.same_commutator_exists_cochain`
181	theorem	`PaperXI.ScalarCocycle112.same_commutator_exists_cochain_raw`

A.346 `PaperXI.ScalarCovariance`

Selected layer: v0.44 parent. **Source lines:** 52.

SHA-256: `4859f25f802820daa9b4d9f45b627eb9f51c9be6756a872134890a6a94da4e06`.

```
Paper XI, Proposition 2.4. F = diag(1,q,...,q^(m-1)) as an actual
linear automorphism. q is a unit. The covariance factor is q⁻¹.
```

Direct project imports: `PaperXI.MonodromyChain`.

Line	Kind	Name
13	def	`PaperXI.ScalarCovariance.frobenius`
25	theorem	`PaperXI.ScalarCovariance.frobenius_apply`
29	theorem	`PaperXI.ScalarCovariance.covariance`
41	theorem	`PaperXI.ScalarCovariance.conjugation`
48	theorem	`PaperXI.ScalarCovariance.scalar_model`

A.347 `PaperXI.ScalarGluing`

Selected layer: v0.44 parent. **Source lines:** 167.

SHA-256: `f32ed302e2af8ac3b71237f1ac455195516c0932bb67a8d9f2c363d241b8321c`.

Scalar gluing on actual graph cycles, with the zero-dimensional branch retained.

Direct project imports: `PaperXI.GraphGauge`, `PaperXI.ProjectiveHolonomy`.

Line	Kind	Name
25	theorem	`PaperXI.ScalarGluing.cycle_operator_iff`
37	theorem	`PaperXI.ScalarGluing.cycle_independent`
46	theorem	`PaperXI.ScalarGluing.based_iff_scalar_operators`
58	theorem	`PaperXI.ScalarGluing.based_root_independent`
66	theorem	`PaperXI.ScalarGluing.cycle_units_iff`
78	theorem	`PaperXI.ScalarGluing.based_iff_scalar_units`
89	theorem	`PaperXI.ScalarGluing.unbased_iff_scalar_units`
104	theorem	`PaperXI.ScalarGluing.loops_iff_characters`
117	theorem	`PaperXI.ScalarGluing.based_iff_ratioH1_trivial`
127	theorem	`PaperXI.ScalarGluing.zero_root_gluing`
135	theorem	`PaperXI.ScalarGluing.zero_finrank_gluing`
144	theorem	`PaperXI.ScalarGluing.connected_scalar_criterion`
159	theorem	`PaperXI.ScalarGluing.connected_zero_dimensional`

A.348 `PaperXI.ScalarHolonomy`

Selected layer: v0.44 parent. **Source lines:** 152.

SHA-256: `f3309937e8a28dc3211c81306280c5e8a7c74c5c2cd944bfc884b0ad5e8cdb8a`.

Scalar loop operators and the faithfulness boundary at rank zero.

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.ScalarHolonomy.scalarHom`
20	theorem	`PaperXI.ScalarHolonomy.scalarHom_apply`
24	theorem	`PaperXI.ScalarHolonomy.scalar_conjugate`
30	theorem	`PaperXI.ScalarHolonomy.scalar_conjugate_independent`
37	theorem	`PaperXI.ScalarHolonomy.scalarHom_injective`
45	def	`PaperXI.ScalarHolonomy.IsScalar`
49	def	`PaperXI.ScalarHolonomy.character`
59	theorem	`PaperXI.ScalarHolonomy.character_spec`
63	theorem	`PaperXI.ScalarHolonomy.character_apply`
69	theorem	`PaperXI.ScalarHolonomy.character_unique`
78	theorem	`PaperXI.ScalarHolonomy.intertwines_iff`
92	theorem	`PaperXI.ScalarHolonomy.character_intertwines_iff`
107	def	`PaperXI.ScalarHolonomy.ratio`
109	theorem	`PaperXI.ScalarHolonomy.ratio_apply`
112	theorem	`PaperXI.ScalarHolonomy.ratio_eq_one_iff`
124	theorem	`PaperXI.ScalarHolonomy.ratio_apply_eq_one_iff`
131	theorem	`PaperXI.ScalarHolonomy.zero_space_operator_eq`
136	theorem	`PaperXI.ScalarHolonomy.zero_space_intertwines`
141	theorem	`PaperXI.ScalarHolonomy.zero_space_scalar_eq`
145	theorem	`PaperXI.ScalarHolonomy.zero_space_complex_not_injective`

A.349 PaperXI.ScalarRepresentation112

Selected layer: Continuation112. **Source lines:** 373.

SHA-256: `f8bad889bfb84cf03242cd2fd13dbd620d15080c126ade4b5f9bbfdb2ae7807c`.

```
Construct an action of the actual group from labelled packet operators
and an explicit scalar cochain. The section multiplication and original theta
restriction are derived from the literal quotient kernel, not supplied as
representation or lambda hypotheses. Existence of a compatible quotient map
and preservation of any required dyadic lift remain separate obligations.
```

Direct project imports: `PaperXI.GlobalHeisenbergBridge112`, `PaperXI.RepresentationStability`, `PaperXI.ScalarCocycle112`.

Line	Kind	Name
23	def	`PaperXI.ScalarRepresentation112.sectionResidue`
30	def	`PaperXI.ScalarRepresentation112.sectionScalar`
33	def	`PaperXI.ScalarRepresentation112.sectionCocycle`
36	theorem	`PaperXI.ScalarRepresentation112.sectionResidue_mul`
47	theorem	`PaperXI.ScalarRepresentation112.sectionScalar_mul`
56	theorem	`PaperXI.ScalarRepresentation112.sectionResidue_section`
61	theorem	`PaperXI.ScalarRepresentation112.sectionScalar_section`
66	theorem	`PaperXI.ScalarRepresentation112.sectionScalar_one`
71	theorem	`PaperXI.ScalarRepresentation112.sectionCocycle_one_left`
76	theorem	`PaperXI.ScalarRepresentation112.sectionCocycle_one_right`
82	theorem	`PaperXI.ScalarRepresentation112.sectionCocycle_assoc`
91	theorem	`PaperXI.ScalarRepresentation112.sectionScalar_on_subgroup`
111	def	`PaperXI.ScalarRepresentation112.packetOperator`
117	theorem	`PaperXI.ScalarRepresentation112.packetOperator_mul`
150	def	`PaperXI.ScalarRepresentation112.scalarTransport`
178	theorem	`PaperXI.ScalarRepresentation112.scalarTransport_on_subgroup`
193	theorem	`PaperXI.ScalarRepresentation112.scalarTransport_section`
204	theorem	`PaperXI.ScalarRepresentation112.packet_normalForm`
219	theorem	`PaperXI.ScalarRepresentation112.scalarTransport_irreducible`
238	theorem	`PaperXI.ScalarRepresentation112.scalarTransport_contains`
254	def	`PaperXI.ScalarRepresentation112.actualScalarCocycle`
266	def	`PaperXI.ScalarRepresentation112.sectionCommutator`
277	theorem	`PaperXI.ScalarRepresentation112.sectionCocycle_commutator`
297	theorem	`PaperXI.ScalarRepresentation112.exists_actual_action_of_equal_scalar_commutators`
334	theorem	`PaperXI.ScalarRepresentation112.bridge_to_prescribed_eta_of_equal_commutators`

A.350 PaperXI.SchrodingerConsequences

Selected layer: v0.44 parent. **Source lines:** 151.

SHA-256: `f0f3bb3edc8a1eeb75ebeb2d70adc089b5e324fb2647841978788425a34c3c19`.

```
Character support and actual isotropic selectors in every irreducible
representation with the prescribed nontrivial central character.
```

Direct project imports: `PaperXI.SchrodingerUniversal`, `PaperXI.HeisenbergSelector`.

Line	Kind	Name
15	def	`PaperXI.SchrodingerConsequences.restriction`
21	def	`PaperXI.SchrodingerConsequences.projector`
26	theorem	`PaperXI.SchrodingerConsequences.projector_idempotent`
33	theorem	`PaperXI.SchrodingerConsequences.range_eq_invariants`
41	def	`PaperXI.SchrodingerConsequences.subgroupProjector`
47	theorem	`PaperXI.SchrodingerConsequences.projector_eq_subgroupProjector`
68	theorem	`PaperXI.SchrodingerConsequences.conjugates`
79	theorem	`PaperXI.SchrodingerConsequences.trace_eq_model`
85	theorem	`PaperXI.SchrodingerConsequences.character_support`
93	theorem	`PaperXI.SchrodingerConsequences.conjugates_projector`
106	theorem	`PaperXI.SchrodingerConsequences.trace_projector`
114	theorem	`PaperXI.SchrodingerConsequences.rank_projector`
124	theorem	`PaperXI.SchrodingerConsequences.maximal_trace_one`
132	theorem	`PaperXI.SchrodingerConsequences.maximal_rank_one`
144	theorem	`PaperXI.SchrodingerConsequences.maximal_fixed_line`

A.351 `PaperXI.SchrodingerFourier`

Selected layer: v0.44 parent. **Source lines:** 77.

SHA-256: `1c9a3b9a2264e75afda601089ef6b0665e2e1f3533f7146575d747946f361125`.

```
Exact coordinate projections from a separating family of diagonal operators.
The finite elimination argument supplies the projector; orthogonality is not a hypothesis.
```

Direct project imports: none.

Line	Kind	Name
16	theorem	`PaperXI.SchrodingerFourier.finite_elimination`
49	theorem	`PaperXI.SchrodingerFourier.coordinate_projection_mem`
65	theorem	`PaperXI.SchrodingerFourier.coordinate_unit_mem`

A.352 `PaperXI.SchrodingerIrreducible`

Selected layer: v0.44 parent. **Source lines:** 88.

SHA-256: `b45be66566d924a65606b03dc54678b2bb795070e1099de79205c24a0bff555c`.

```
Irreducibility of the finite Schrödinger action from its actual translation
and dual-character multiplication operators. The proof uses coordinate
projections and spans all functions, including the zero-dimensional case.
```

Direct project imports: `PaperXI.SchrodingerFourier`, `PaperXI.SchrodingerSeparation`.

Line	Kind	Name
16	theorem	`PaperXI.SchrodingerIrreducible.invariant_eq_bot_or_top`
61	theorem	`PaperXI.SchrodingerIrreducible.irreducible_of_actions`

A.353 PaperXI.SchrodingerOperator

Selected layer: v0.44 parent. **Source lines:** 93.

SHA-256: `2d1478c4d0de8e6ea366020e41530e8967ad47f25516f5cd4d8cbb544d08616d`.

```
The literal Schrödinger operators in the manuscript's sign convention.
```

Direct project imports: none.

Line	Kind	Name
13	def	`PaperXI.Schrodinger.operator`
25	theorem	`PaperXI.Schrodinger.operator_apply`
29	def	`PaperXI.Schrodinger.translation`
34	theorem	`PaperXI.Schrodinger.translation_apply`
37	def	`PaperXI.Schrodinger.modulation`
49	theorem	`PaperXI.Schrodinger.modulation_apply`
52	theorem	`PaperXI.Schrodinger.operator_translation`
57	theorem	`PaperXI.Schrodinger.operator_modulation`
62	theorem	`PaperXI.Schrodinger.operator_central`
67	theorem	`PaperXI.Schrodinger.operator_one`
73	theorem	`PaperXI.Schrodinger.phase_add`
83	theorem	`PaperXI.Schrodinger.operator_mul`

A.354 PaperXI.SchrodingerSeparation

Selected layer: v0.44 parent. **Source lines:** 32.

SHA-256: `ea554103afc7f9badccc857e5421b5b4749f1bea3df4043f0aae400783cb8b8b`.

```
A nontrivial scalar additive character, composed with the full linear dual,
separates points of a vector space. No finiteness assumption is required.
```

Direct project imports: none.

Line	Kind	Name
15	theorem	`PaperXI.SchrodingerSeparation.exists_dual_character_ne`

A.355 PaperXI.SchrodingerTrace

Selected layer: v0.44 parent. **Source lines:** 88.

SHA-256: `0c523e1824a16155d9072d81a47308e828eef34cdf178595eff2ccbd8dbe7498`.

```
Character support for the literal finite function-space Schrödinger operators.
```

Direct project imports: `PaperXI.SchrodingerOperator`.

Line	Kind	Name
16	theorem	`PaperXI.SchrodingerTrace.functional_surjective`
25	theorem	`PaperXI.SchrodingerTrace.functional_character_ne_one`

Line	Kind	Name
38	theorem	PaperXI.SchrodingerTrace.sum_functional_character
43	theorem	PaperXI.SchrodingerTrace.operator_diagonal
58	theorem	PaperXI.SchrodingerTrace.trace_of_translation_ne_zero
66	theorem	PaperXI.SchrodingerTrace.trace_of_translation_zero
74	theorem	PaperXI.SchrodingerTrace.character_support

A.356 PaperXI.SchrodingerUnitary

Selected layer: v0.44 parent. **Source lines:** 45.

SHA-256: 90bc68223666d16ea0fd539caf1719dc4d7a73e4774eadd64d5136507b4e40cb.

The literal Schrödinger operators preserve the usual finite sum Hermitian pairing.

Direct project imports: PaperXI.SchrodingerOperator.

Line	Kind	Name
15	theorem	PaperXI.SchrodingerUnitary.conjugate_character
21	theorem	PaperXI.SchrodingerUnitary.conjugate_mul_character
29	theorem	PaperXI.SchrodingerUnitary.pairing_invariant

A.357 PaperXI.SchrodingerUniversal

Selected layer: v0.44 parent. **Source lines:** 217.

SHA-256: 3a45fcb361ce2045666670d9fe67cd2b7406dc7840b50f9a9cbe11c00f77c608.

The actual map from the Schrödinger model, built from a nonzero vector
fixed by all modulations, and its representation equivalence.

Direct project imports: PaperXI.HeisenbergNormalForm, PaperXI.HeisenbergFrequency, PaperXI.HeisenbergDimension, PaperXI.HeisenbergRepresentation, PaperXI.SchurScalar.

Line	Kind	Name
18	def	PaperXI.SchrodingerUniversal.basis_map
22	theorem	PaperXI.SchrodingerUniversal.basis_map_basis
28	theorem	PaperXI.SchrodingerUniversal.translation_basis
34	theorem	PaperXI.SchrodingerUniversal.modulation_basis
43	theorem	PaperXI.SchrodingerUniversal.basis_map_translation
56	theorem	PaperXI.SchrodingerUniversal.basis_map_modulation
72	theorem	PaperXI.SchrodingerUniversal.basis_map_central
83	theorem	PaperXI.SchrodingerUniversal.intertwines
103	def	PaperXI.SchrodingerUniversal.intertwiningMap
111	theorem	PaperXI.SchrodingerUniversal.intertwiningMap_ne_zero
122	def	PaperXI.SchrodingerUniversal.equiv_of_fixed
138	theorem	PaperXI.SchrodingerUniversal.exists_equiv
148	def	PaperXI.SchrodingerUniversal.equiv
155	theorem	PaperXI.SchrodingerUniversal.equiv_intertwines
166	theorem	PaperXI.SchrodingerUniversal.finiteDimensional
174	theorem	PaperXI.SchrodingerUniversal.finrank_eq

Line	Kind	Name
182	theorem	`PaperXI.SchrodingerUniversal.finrank_eq_sqrt`
191	theorem	`PaperXI.SchrodingerUniversal.unique_scalar`
206	theorem	`PaperXI.SchrodingerUniversal.exists_equiv_finiteField`

A.358 `PaperXI.SchurHom`

Selected layer: v0.44 parent. **Source lines:** 68.

SHA-256: `b86b71d721ec121989b4959337263985da723a313f102af82e8b05a8775c35bf`.

```
Exact Hom dimensions for actual irreducible complex representations.
```

Direct project imports: `PaperXI.DualityLinear`, `PaperXI.SchurScalar`.

Line	Kind	Name
14	def	`PaperXI.SchurHom.postcomposeEquiv`
25	theorem	`PaperXI.SchurHom.finrank_eq_indicator`
40	theorem	`PaperXI.SchurHom.finrank_zero_or_one`
49	theorem	`PaperXI.SchurHom.finrank_one_iff_equiv`
59	theorem	`PaperXI.SchurHom.tensor_invariants_zero_or_one`

A.359 `PaperXI.SchurScalar`

Selected layer: v0.44 parent. **Source lines:** 66.

SHA-256: `ab78b7e45ea049accb02eacca96ad12dd74812d23ea41c9186645a9b5700ee92`.

```
Scalar commutants and uniqueness of actual irreducible representation bridges.
```

Direct project imports: `PaperXI.GeneratedIrreducibility`.

Line	Kind	Name
13	theorem	`PaperXI.SchurScalar.commuting_scalar`
25	theorem	`PaperXI.SchurScalar.scalar_unique`
33	theorem	`PaperXI.SchurScalar.commuting_unit`
46	theorem	`PaperXI.SchurScalar.bridge_unique`

A.360 `PaperXI.SemisimpleDetection`

Selected layer: v0.44 parent. **Source lines:** 71.

SHA-256: `b2713fc088d1ea21bffebfd8dd05a2b93962db392825add54d953db52b00c8c8`.

```
Maschke separation: literal elements of C[G] are detected on simple modules.
No choice of a complete irreducible character table is assumed.
```

Direct project imports: `PaperXI.CharacterProjector`.

Line	Kind	Name
14	theorem	`PaperXI.SemisimpleDetection.eq_zero_of_simple_actions`
41	theorem	`PaperXI.SemisimpleDetection.eq_zero_of_irreducible_actions`
62	theorem	`PaperXI.SemisimpleDetection.eq_of_irreducible_actions`

A.361 `PaperXI.SemisimpleProjection`

Selected layer: base108 arithmetic branch. **Source lines:** 99.

SHA-256: `dab8a3dd87a59b725c83dd9c32b853b2de9db98afc7461311017ed27ec4b21ea`.

```
A multiplicity-one summand of a semisimple representation is preserved by
every invariant subspace.  The splitting used here is obtained from semisimplicity,
not imposed as an additional assumption on the invariant subspace.
```

Direct project imports: `PaperXI.RepresentationStability`.

Line	Kind	Name
19	theorem	`PaperXI.SemisimpleProjection.module_range_le_of_nonzero`
43	theorem	`PaperXI.SemisimpleProjection.module_projection_mem`
66	theorem	`PaperXI.SemisimpleProjection.range_le_of_nonzero`
87	theorem	`PaperXI.SemisimpleProjection.projection_mem`

A.362 `PaperXI.SimpleModuleClassification`

Selected layer: v0.44 parent. **Source lines:** 106.

SHA-256: `b28f26a821af96bffc99660b774cfedfbafa942365b515c8a346a771ce3b4bf3`.

```
Actual simple modules over a finite-dimensional simple complex algebra:
uniqueness, automatic finite dimensionality, and the square-root dimension.
```

Direct project imports: none.

Line	Kind	Name
17	theorem	`PaperXI.SimpleModuleClassification.equivalent`
27	theorem	`PaperXI.SimpleModuleClassification.regular_decomposition`
37	theorem	`PaperXI.SimpleModuleClassification.exists_simple_ideal`
46	theorem	`PaperXI.SimpleModuleClassification.equivalent_complex`
55	theorem	`PaperXI.SimpleModuleClassification.finiteDimensional`
64	theorem	`PaperXI.SimpleModuleClassification.finrank_pos`
70	def	`PaperXI.SimpleModuleClassification.scalarEndEquiv`
76	theorem	`PaperXI.SimpleModuleClassification.finrank_end`
81	theorem	`PaperXI.SimpleModuleClassification.finrank_sq`
103	theorem	`PaperXI.SimpleModuleClassification.finrank_sqrt`

A.363 `PaperXI.SmithMatrix106`

Selected layer: base108 arithmetic branch. **Source lines:** 57.

SHA-256: `9104a6467adaf608eeb4b3700c5d827828b603184ed83c55516445df5f23a948`.

```
Integral left and right basis changes for an injective square matrix over
a PID. The changes of basis are retained as actual general linear matrices.
```

Direct project imports: none.

Line	Kind	Name
16	def	`PaperXI.SmithMatrix106.basisGL`
23	theorem	`PaperXI.SmithMatrix106.basisGL_apply`
27	theorem	`PaperXI.SmithMatrix106.exists_diagonal`

A.364 `PaperXI.SmoothDegreeQuotient`

Selected layer: base108 arithmetic branch. **Source lines:** 192.

SHA-256: `77138b22ce63047224ba2ad5c14fc85e25acce77ad9ea79b7e3404e4fb684a3d`.

```
# A smooth representation quotient which retains every integer degree

The quotient kernel is the intersection of the representation kernel and
integer-degree kernel.  No finiteness of the original inertia group and no
identification with an absolute local-field Weil group is assumed here.
```

Direct project imports: `PaperXI.QuotientRepresentation`, `PaperXI.TameDegree`.

Line	Kind	Name
21	def	`PaperXI.SmoothDegreeQuotient.vectorStabilizer`
33	theorem	`PaperXI.SmoothDegreeQuotient.mem_vectorStabilizer`
40	def	`PaperXI.SmoothDegreeQuotient.IsSmooth`
44	theorem	`PaperXI.SmoothDegreeQuotient.isOpen_kernel_of_smooth`
67	def	`PaperXI.SmoothDegreeQuotient.jointKernel`
69	instance	`PaperXI.SmoothDegreeQuotient.jointKernel_normal`
74	abbrev	`PaperXI.SmoothDegreeQuotient.DegreeQuotient`
77	abbrev	`PaperXI.SmoothDegreeQuotient.quotientMap`
81	def	`PaperXI.SmoothDegreeQuotient.descendedDegree`
85	def	`PaperXI.SmoothDegreeQuotient.descendedRepresentation`
88	theorem	`PaperXI.SmoothDegreeQuotient.descendedDegree_mk`
91	theorem	`PaperXI.SmoothDegreeQuotient.descendedRepresentation_mk`
94	theorem	`PaperXI.SmoothDegreeQuotient.descendedDegree_comp`
97	theorem	`PaperXI.SmoothDegreeQuotient.descendedRepresentation_comp`
101	theorem	`PaperXI.SmoothDegreeQuotient.descended_irreducible`
106	theorem	`PaperXI.SmoothDegreeQuotient.degree_surjective_iff`
118	theorem	`PaperXI.SmoothDegreeQuotient.degreeSubgroup_comap`
126	theorem	`PaperXI.SmoothDegreeQuotient.jointKernel_descended`
135	theorem	`PaperXI.SmoothDegreeQuotient.degree_descends_iff`
150	theorem	`PaperXI.SmoothDegreeQuotient.inertia_image`
165	theorem	`PaperXI.SmoothDegreeQuotient.isOpen_jointKernel`
171	theorem	`PaperXI.SmoothDegreeQuotient.quotient_discrete`
177	theorem	`PaperXI.SmoothDegreeQuotient.quotientMap_continuous`
182	theorem	`PaperXI.SmoothDegreeQuotient.finite_descended_inertia`

A.365 `PaperXI.SmoothHomOrbit68`

Selected layer: base108 arithmetic branch. **Source lines:** 95.

SHA-256: `87d24ea9fbb25f0394cace54953a216a8158ddded33b1f0e68f620c6a0e5c2d6`.

```
# Finite orbits on actual linear-map spaces modulo a scalar subgroup

The subgroup and quotient are literal.  Compactness is imposed on G / N, not
on G.  Finiteness of the Hom orbit is derived from smoothness and equal scalar
actions, rather than supplied as a splitting certificate.
```

Direct project imports: `PaperXI.CompactSmoothSemisimple`.

Line	Kind	Name
22	theorem	`PaperXI.SmoothHomOrbit68.linHom_smooth`
48	theorem	`PaperXI.SmoothHomOrbit68.descend_smooth`
56	theorem	`PaperXI.SmoothHomOrbit68.orbit_finite_of_compact_quotient`
69	theorem	`PaperXI.SmoothHomOrbit68.scalar_subgroup_le_linHom_kernel`
84	theorem	`PaperXI.SmoothHomOrbit68.linHom_orbit_finite`

A.366 `PaperXI.SpectralGeometricFrobenius`

Selected layer: normalized Frobenius. **Source lines:** 184.

SHA-256: `e437a81f9af9b976a3c272dbb0dbb4824e47f138852a60adbb5dc5ac9236685e`.

```
# Geometric Frobenius for a finite Galois spectral extension

The integer ring is constructed from the spectral norm of the actual extension.
Valuation stability, the Hensel property, and residue finiteness are derived.
The Frobenius-lift premise is then discharged by the inherited Galois/Hensel
surjectivity theorem. No full absolute Weil group or automorphic object is asserted.
```

Direct project imports: `PaperXI.HenselianGaloisResidue`, `PaperXI.HenselianLiftTower`, `PaperXI.FiniteResidueBaseChange`.

Line	Kind	Name
28	def	`PaperXI.SpectralGeometricFrobenius.SpectralExtension`
30	instance	`PaperXI.SpectralGeometricFrobenius.spectralField`
31	instance	`PaperXI.SpectralGeometricFrobenius.spectralAlgebra`
32	instance	`PaperXI.SpectralGeometricFrobenius.spectralFinite`
34	instance	`PaperXI.SpectralGeometricFrobenius.spectralGalois`
38	def	`PaperXI.SpectralGeometricFrobenius.originalFieldEquiv`
73	def	`PaperXI.SpectralGeometricFrobenius.integers`
76	def	`PaperXI.SpectralGeometricFrobenius.baseIntegers`
80	theorem	`PaperXI.SpectralGeometricFrobenius.mem_integers`
92	theorem	`PaperXI.SpectralGeometricFrobenius.integers_comap`
102	theorem	`PaperXI.SpectralGeometricFrobenius.automorphism_stable`
116	instance	`PaperXI.SpectralGeometricFrobenius.integersDVR`
120	instance	`PaperXI.SpectralGeometricFrobenius.integersFiniteResidue`
124	instance	`PaperXI.SpectralGeometricFrobenius.integersHenselian`
128	abbrev	`PaperXI.SpectralGeometricFrobenius.BaseResidue`
129	abbrev	`PaperXI.SpectralGeometricFrobenius.ExtensionResidue`
140	def	`PaperXI.SpectralGeometricFrobenius.geometricFrobenius`
144	def	`PaperXI.SpectralGeometricFrobenius.arithmeticFrobenius`
147	theorem	`PaperXI.SpectralGeometricFrobenius.arithmetic_apply`
152	theorem	`PaperXI.SpectralGeometricFrobenius.geometric_inverse_power`

Line	Kind	Name
158	def	PaperXI.SpectralGeometricFrobenius.residueAction
164	theorem	PaperXI.SpectralGeometricFrobenius.exists_geometric_lift
172	def	PaperXI.SpectralGeometricFrobenius.geometricLift
174	theorem	PaperXI.SpectralGeometricFrobenius.residue_geometricLift
179	theorem	PaperXI.SpectralGeometricFrobenius.geometricLift_normalized

A.367 PaperXI.SpectralLocalField

Selected layer: v0.44 parent. **Source lines:** 175.

SHA-256: 96cc4d62e4fc37f3550c6800e3715c60d191bf454547ea9a67d4aeae6376f9ab.

```
# Spectral local-field structure on the constructed unramified fraction field

The upper field is finite over the given nonarchimedean local field. This file
installs its spectral norm, proves completeness and local compactness, and
packages the norm valuation as an actual nonarchimedean local-field structure.
The canonical DVR valuation remains a separate explicitly named valuation; the
comparison of the two valued structures is not hidden in an instance.
```

Direct project imports: PaperXI.CanonicalDVRValuation.

Line	Kind	Name
45	abbrev	PaperXI.SpectralLocalField.UpperField
48	instance	PaperXI.SpectralLocalField.upperNormedField
52	instance	PaperXI.SpectralLocalField.upperNontriviallyNormedField
57	instance	PaperXI.SpectralLocalField.upperNormedSpace
61	instance	PaperXI.SpectralLocalField.upperNormedAlgebra
65	theorem	PaperXI.SpectralLocalField.norm_algebraMap
70	theorem	PaperXI.SpectralLocalField.norm_eq_spectralNorm
74	theorem	PaperXI.SpectralLocalField.algebraMap_isometry
81	instance	PaperXI.SpectralLocalField.upperUltrametric
93	instance	PaperXI.SpectralLocalField.upperComplete
97	instance	PaperXI.SpectralLocalField.upperProper
101	instance	PaperXI.SpectralLocalField.upperLocallyCompact
105	instance	PaperXI.SpectralLocalField.upperNormValued
109	instance	PaperXI.SpectralLocalField.upperValuativeRel
112	instance	PaperXI.SpectralLocalField.upperNormValuationCompatible
117	instance	PaperXI.SpectralLocalField.upperIsValuativeTopology
124	instance	PaperXI.SpectralLocalField.upperValuativeNontrivial
131	theorem	PaperXI.SpectralLocalField.relation_integer_eq_norm_integer
143	theorem	PaperXI.SpectralLocalField.mem_integer_iff_norm_le_one
150	instance	PaperXI.SpectralLocalField.upperIsNonarchimedeanLocalField
157	instance	PaperXI.SpectralLocalField.upperFiniteResidue
161	instance	PaperXI.SpectralLocalField.upperIntegerIsDVR
166	def	PaperXI.SpectralLocalField.upperValueGroupIsoInt
171	theorem	PaperXI.SpectralLocalField.finrank_eq_residue_finrank

A.368 PaperXI.SplitRadical105

Selected layer: base108 arithmetic branch. **Source lines:** 152.

SHA-256: bbe4cd3ed2985df636cc326b7fceac83a5c48fec8b393ba3ae4a8da6e7fd5ca7.

```
Literal square-zero coordinate radicals, entrywise lifting, and contraction
by a normalized Cartan diagonal. Neither a contracting subgroup nor its lifts
are supplied to the matrix theorem.
```

Direct project imports: none.

Line	Kind	Name
17	def	`PaperXI.SplitRadical105.rect`
20	theorem	`PaperXI.SplitRadical105.rect_mul`
28	theorem	`PaperXI.SplitRadical105.rect_add`
34	def	`PaperXI.SplitRadical105.unip`
40	theorem	`PaperXI.SplitRadical105.unip_add`
48	theorem	`PaperXI.SplitRadical105.unip_zero`
53	def	`PaperXI.SplitRadical105.radicalMap`
58	theorem	`PaperXI.SplitRadical105.map_unip`
67	def	`PaperXI.SplitRadical105.diagonalGL`
79	def	`PaperXI.SplitRadical105.cartan`
82	def	`PaperXI.SplitRadical105.zeroSet`
85	theorem	`PaperXI.SplitRadical105.mem_zeroSet`
89	theorem	`PaperXI.SplitRadical105.cartan_conjugation`
115	theorem	`PaperXI.SplitRadical105.contracted_reduction`
129	theorem	`PaperXI.SplitRadical105.exists_contracted_lift`
146	theorem	`PaperXI.SplitRadical105.cartan_zero`

A.369 `PaperXI.SymplecticAssembly`

Selected layer: v0.44 parent. **Source lines:** 51.

SHA-256: `67819fec0a0969ee2375d7ee93287a9f1e91d60e5da87e7a12aee6013e3e0d94`.

```
Adding one hyperbolic plane to an existing symplectic coordinate system.
```

Direct project imports: `PaperXI.DyadicCoordinates`.

Line	Kind	Name
11	def	`PaperXI.SymplecticAssembly.prepend`
26	theorem	`PaperXI.SymplecticAssembly.prepend_form`
35	def	`PaperXI.SymplecticAssembly.assemble`
40	theorem	`PaperXI.SymplecticAssembly.assemble_form`

A.370 `PaperXI.SymplecticCentralExtension`

Selected layer: v0.44 parent. **Source lines:** 75.

SHA-256: `b54d1f3d499676466c42f50e7b1fff895c0dd532d855652a51347f5bff12c61d`.

```
A nontrivial additive character detects every nonzero vector through a
left-nondegenerate field-valued bilinear commutator. This supplies scalar
nondegeneracy for actual symplectic central extensions in every characteristic.
```

Direct project imports: `PaperXI.CentralExtensionStructure`.

Line	Kind	Name
17	def	PaperXI.SymplecticCentralExtension.scalarCharacter
20	theorem	PaperXI.SymplecticCentralExtension.scalarCharacter_apply
23	theorem	PaperXI.SymplecticCentralExtension.scalarCharacter_ofAdd
28	theorem	PaperXI.SymplecticCentralExtension.character_separating_left
45	theorem	PaperXI.SymplecticCentralExtension.radical_eq_bot
66	theorem	PaperXI.SymplecticCentralExtension.radical_eq_bot_scalarCharacter

A.371 PaperXI.SymplecticCoordinates

Selected layer: v0.44 parent. **Source lines:** 70.

SHA-256: b9b5ab88baf7b682acac5698d8cd558de3acf406eaec991f76a8b16e63500ecf.

```
Symplectic coordinates constructed from alternation and nondegeneracy,
by strong induction on the dimension.
```

Direct project imports: PaperXI.SymplecticSplit, PaperXI.SymplecticAssembly, PaperXI.DyadicDimension.

Line	Kind	Name
17	theorem	PaperXI.SymplecticCoordinates.exists_coordinates
45	theorem	PaperXI.SymplecticCoordinates.exists_coordinates_dimension
54	theorem	PaperXI.SymplecticCoordinates.even_finrank
61	theorem	PaperXI.SymplecticCoordinates.exists_half_coordinates

A.372 PaperXI.SymplecticHeisenbergUniqueness

Selected layer: v0.44 parent. **Source lines:** 85.

SHA-256: e54d8758376e52f844dd630abfdd19a6f940709729fabf3449bfa5f755195104.

```
Finite Stone–von Neumann from an actual field-valued nondegenerate
commutator and a nontrivial additive character. This applies to supplied actual
central extensions in every characteristic, without assuming scalar
nondegeneracy, target dimensions, or a first intertwiner.
```

Direct project imports: PaperXI.SymplecticCentralExtension, PaperXI.CentralExtensionExistence.

Line	Kind	Name
19	theorem	PaperXI.SymplecticHeisenbergUniqueness.central_action
35	theorem	PaperXI.SymplecticHeisenbergUniqueness.finiteDimensional
42	theorem	PaperXI.SymplecticHeisenbergUniqueness.finrank_sqrt
51	theorem	PaperXI.SymplecticHeisenbergUniqueness.exists_equiv
61	theorem	PaperXI.SymplecticHeisenbergUniqueness.card_eq_square
69	theorem	PaperXI.SymplecticHeisenbergUniqueness.exists_irreducible

A.373 PaperXI.SymplecticSplit

Selected layer: v0.44 parent. **Source lines:** 126.

SHA-256: 66521ce6aa9e9ca4cbf9bc05aad9112ad95e36069f2d3d718f214beb6515bc05.

A constructive hyperbolic-plane splitting of a nondegenerate alternating form over F_2. The complement and both directions of the coordinate map are built explicitly from one hyperbolic pair.

Direct project imports: `PaperXI.DyadicGauss`.

Line	Kind	Name
17	theorem	`PaperXI.SymplecticSplit.alternating_symmetric`
24	theorem	`PaperXI.SymplecticSplit.exists_pair`
35	theorem	`PaperXI.SymplecticSplit.split_of_pair`
113	theorem	`PaperXI.SymplecticSplit.exists_split`

A.374 `PaperXI.TameConductor`

Selected layer: v0.44 parent. **Source lines:** 113.

SHA-256: `808eb443d2a292fc79cd70ce323c369acd62c38e9393ec4c19269dc06d575dcf`.

The nontrivial tame conductor calculation, with its finite-quotient attachment explicit. These definitions encode finite ramification sums. They do not construct local fields, Weil groups or ramification filtrations.

Direct project imports: `PaperXI.PacketBasis`, `PaperXI.InertiaFixed`, `PaperXI.InertiaQuotient`.

Line	Kind	Name
21	theorem	`PaperXI.TameConductor.zero_label_period`
26	theorem	`PaperXI.TameConductor.inertia_invariants_eq_bot`
40	theorem	`PaperXI.TameConductor.inertia_finrank_zero`
51	def	`PaperXI.TameConductor.fixedCodimension`
56	def	`PaperXI.TameConductor.finiteSwan`
61	theorem	`PaperXI.TameConductor.invariants_eq_top_of_trivial_on`
68	theorem	`PaperXI.TameConductor.fixedCodimension_zero_of_trivial_on`
76	theorem	`PaperXI.TameConductor.finiteSwan_zero`
88	def	`PaperXI.TameConductor.finiteArtin`
95	theorem	`PaperXI.TameConductor.conductor_of_nontrivial_tame_atom`

A.375 `PaperXI.TameDegree`

Selected layer: v0.44 parent. **Source lines:** 170.

SHA-256: `868e2f79b22a754bc070c23805f852b2c90a71cb802703bea2eaad248f39ad92`.

Degree subgroups and their actual cyclic translation actions.

Direct project imports: `PaperXI.TameTensorOrbits`.

Line	Kind	Name
13	def	`PaperXI.TameDegree.degree`
15	theorem	`PaperXI.TameDegree.degree_one`
17	theorem	`PaperXI.TameDegree.degree_mul`
20	theorem	`PaperXI.TameDegree.degree_zpow`

Line	Kind	Name
23	theorem	`PaperXI.TameDegree.degree_inv`
26	theorem	`PaperXI.TameDegree.degree_conjugate`
30	def	`PaperXI.TameDegree.degreeMod`
35	theorem	`PaperXI.TameDegree.degreeMod_apply`
39	def	`PaperXI.TameDegree.degreeSubgroup`
41	theorem	`PaperXI.TameDegree.mem_degreeSubgroup`
46	theorem	`PaperXI.TameDegree.degreeSubgroup_inter`
54	theorem	`PaperXI.TameDegree.degreeSubgroup_le`
61	def	`PaperXI.TameDegree.conjugationInto`
69	theorem	`PaperXI.TameDegree.conjugationInto_apply`
73	def	`PaperXI.TameDegree.translationAction`
83	theorem	`PaperXI.TameDegree.translationAction_apply`
86	theorem	`PaperXI.TameDegree.translationAction_fixed_iff`
92	def	`PaperXI.TameDegree.pairAction`
98	theorem	`PaperXI.TameDegree.pairAction_apply`
102	theorem	`PaperXI.TameDegree.pairAction_fixed_iff`
107	def	`PaperXI.TameDegree.blockAction`
113	theorem	`PaperXI.TameDegree.blockAction_apply`
117	theorem	`PaperXI.TameDegree.orbitMap_intertwines`
123	theorem	`PaperXI.TameDegree.orbitEquiv_intertwines`
132	theorem	`PaperXI.TameDegree.degree_phi`
135	theorem	`PaperXI.TameDegree.degree_phi_zpow`
139	theorem	`PaperXI.TameDegree.degreeMod_phi_zpow`
144	theorem	`PaperXI.TameDegree.degreeMod_surjective`
152	theorem	`PaperXI.TameDegree.translationAction_phi_zpow`
157	theorem	`PaperXI.TameDegree.pairAction_phi_zpow`
163	theorem	`PaperXI.TameDegree.fiber_transitive`

A.376 `PaperXI.TameExponent`

Selected layer: v0.44 parent. **Source lines:** 117.

SHA-256: `738e3f3fe21ccb970422c7dc2b0de7f7780e588ba208153af7e04e70415b767e`.

`Faithful exponent coordinates for genuine finite-field unit characters.`

Direct project imports: `PaperXI.TameNorm`, `PaperXI.ElementaryEnvelope`.

Line	Kind	Name
11	def	`PaperXI.TameExponent.rootEmbedding`
22	theorem	`PaperXI.TameExponent.rootEmbedding_injective`
32	def	`PaperXI.TameExponent.faithfulCharacter`
36	theorem	`PaperXI.TameExponent.faithfulCharacter_injective`
44	theorem	`PaperXI.TameExponent.pow_eq_one_iff`
63	theorem	`PaperXI.TameExponent.pow_add_card`
73	theorem	`PaperXI.TameExponent.product`
77	theorem	`PaperXI.TameExponent.exists_exponent`
92	theorem	`PaperXI.TameExponent.pow_eq_pow_iff`
102	def	`PaperXI.TameExponent.coordinate`

Line	Kind	Name
105	theorem	`PaperXI.TameExponent.coordinate_bijective`

A.377 `PaperXI.TameGaussNaturality`

Selected layer: v0.44 parent. **Source lines:** 79.

SHA-256: `85bd64090e77888f06f233d6ea04a3cee2a539e3341988dd95273b8cdf5c1f7b`.

```
Additive-character normalization is part of the finite comparison data.
These statements concern the literal finite Gauss sum, not a full local epsilon
factor. A nontrivial residue character is required for a unit-norm phase.
```

Direct project imports: `PaperXI.GaussCharacters`.

Line	Kind	Name
14	def	`PaperXI.TameGaussNaturality.pullMul`
18	def	`PaperXI.TameGaussNaturality.pullAdd`
22	theorem	`PaperXI.TameGaussNaturality.sum_transport`
27	theorem	`PaperXI.TameGaussNaturality.phase_transport`
34	theorem	`PaperXI.TameGaussNaturality.phase_of_stable_additive`
40	theorem	`PaperXI.TameGaussNaturality.phase_mulShift`
49	theorem	`PaperXI.TameGaussNaturality.phase_trivial`
59	theorem	`PaperXI.TameGaussNaturality.phase_norm_cases`
71	theorem	`PaperXI.TameGaussNaturality.phase_ne_zero`

A.378 `PaperXI.TameHomCriterion`

Selected layer: v0.44 parent. **Source lines:** 66.

SHA-256: `d125abe9165991cb809454306c36d0fa17a486c198612b7ddb05b4dd3cb35d35`.

```
Corollary 5.7: the actual tensor fixed space is the actual Hom space.
The refined packet data are precisely the inherited Proposition 5.6 contract.
The 0/1 bound is derived from irreducibility, not supplied with the packets.
```

Direct project imports: `PaperXI.SchurHom`, `PaperXI.TensorFixed`.

Line	Kind	Name
20	theorem	`PaperXI.TameHomCriterion.hom_finrank_eq_trivial_count`
40	theorem	`PaperXI.TameHomCriterion.tame_hom_test`

A.379 `PaperXI.TameNorm`

Selected layer: v0.44 parent. **Source lines:** 102.

SHA-256: `249939314ef4f5eff217c783fa013cb8fe51f659337e9f7db4ec9a22d2a11c05`.

```
Paper XI, 5.3: norm lifting and Frobenius on actual finite-field unit characters.
The characters on subfields are restrictions of one character of the common field.
This is the compatible choice of exponent coordinates used by the source formula.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.TameNorm.embedding`
18	theorem	`PaperXI.TameNorm.embedding_injective`
24	def	`PaperXI.TameNorm.norm`
27	def	`PaperXI.TameNorm.normExponent`
29	theorem	`PaperXI.TameNorm.embedding_norm`
35	theorem	`PaperXI.TameNorm.norm_lift`
44	theorem	`PaperXI.TameNorm.norm_lift_eq_one_iff`
58	def	`PaperXI.TameNorm.frobenius`
61	theorem	`PaperXI.TameNorm.frobenius_apply`
68	theorem	`PaperXI.TameNorm.frobenius_character`
76	theorem	`PaperXI.TameNorm.shifted_norm_lift`
88	theorem	`PaperXI.TameNorm.norm_frobenius`
93	theorem	`PaperXI.TameNorm.shifted_lower_norm_lift`

A.380 `PaperXI.TameNormPacket`

Selected layer: v0.44 parent. **Source lines:** 118.

SHA-256: `d19c32f222a6fd3f565bce086da6baf95a562f7266ec7a0105afefeb54a3a071`.

```
Paper XI, Proposition 5.3. The interaction character is built from actual
finite-field norms, the actual Frobenius, and multiplication of characters.
Exponent coordinates on the two subfields come from one faithful common-field
character. Every pair of characters admits these compatible coordinates.
```

Direct project imports: `PaperXI.TameExponent`.

Line	Kind	Name
17	def	`PaperXI.TameNormPacket.interaction`
23	theorem	`PaperXI.TameNormPacket.refined_exponent`
35	theorem	`PaperXI.TameNormPacket.refined_trivial_iff`
46	theorem	`PaperXI.TameNormPacket.source_formula`
58	theorem	`PaperXI.TameNormPacket.source_trivial_iff`
71	theorem	`PaperXI.TameNormPacket.restricted_faithful`
80	theorem	`PaperXI.TameNormPacket.exists_compatible_exponents`
93	theorem	`PaperXI.TameNormPacket.source`

A.381 `PaperXI.TameOrbitTransitivity`

Selected layer: v0.44 parent. **Source lines:** 110.

SHA-256: `e2f06470f27d1f2d54ee8573c71fdb9e4dd3a4f5526cf551c225e087f94f7971`.

```
The orbit-transitivity part of Theorem 3.2 for an actual irreducible group
representation with explicitly generating tame inertia and Frobenius.
```

Direct project imports: `PaperXI.OrbitSpan`, `PaperXI.GeneratedIrreducibility`.

Line	Kind	Name
15	abbrev	`PaperXI.TameOrbitTransitivity.tameRestriction`
18	def	`PaperXI.TameOrbitTransitivity.generators`
23	theorem	`PaperXI.TameOrbitTransitivity.geometric_relation`
38	theorem	`PaperXI.TameOrbitTransitivity.generators_preserve_orbitSpan`
58	def	`PaperXI.TameOrbitTransitivity.orbitSubrepresentation`
68	theorem	`PaperXI.TameOrbitTransitivity.orbitSpan_eq_top`
80	theorem	`PaperXI.TameOrbitTransitivity.nonzero_packets_one_orbit`
92	theorem	`PaperXI.TameOrbitTransitivity.exists_nonzero_packet`

A.382 `PaperXI.TameProjectors`

Selected layer: v0.44 parent. **Source lines:** 86.

SHA-256: `d5881299706cbe60a5cbefc9dc60bb9bdffbc1365228529c181b9a5120b71d8e`.

```
Proposition 3.1 over C, with primitivity and the nonzero denominator proved.
Both Frobenius conventions are stated explicitly, without changing the operators.
```

Direct project imports: `PaperXI.FourierPackets`.

Line	Kind	Name
14	def	`PaperXI.TameProjectors.projector`
17	theorem	`PaperXI.TameProjectors.idempotent`
22	theorem	`PaperXI.TameProjectors.orthogonal`
27	theorem	`PaperXI.TameProjectors.complete`
32	theorem	`PaperXI.TameProjectors.range_eq_character_space`
37	theorem	`PaperXI.TameProjectors.forward_generator_covariance`
45	theorem	`PaperXI.TameProjectors.geometric_covariance`
54	theorem	`PaperXI.TameProjectors.character_weight`
61	theorem	`PaperXI.TameProjectors.standard_root`
66	theorem	`PaperXI.TameProjectors.manuscript_formula`
77	theorem	`PaperXI.TameProjectors.tame_projector_theorem`

A.383 `PaperXI.TameTensorInduction`

Selected layer: v0.44 parent. **Source lines:** 111.

SHA-256: `148b5e2d70afcc38a07c5b9fe8167fd690d2f687def771b465c1a8b81fbb799e`.

```
Actual induction of the interaction packets and the tensor decomposition.
```

Direct project imports: `PaperXI.TameTensorPackets`, `PaperXI.MonomialInductionSubgroup`.

Line	Kind	Name
20	def	`PaperXI.TameTensorInduction.character`
39	theorem	`PaperXI.TameTensorInduction.character_apply`
48	theorem	`PaperXI.TameTensorInduction.cycle_transitive`
55	def	`PaperXI.TameTensorInduction.packetEquivalence`
70	def	`PaperXI.TameTensorInduction.directSumEquiv`

Line	Kind	Name
86	def	PaperXI.TameTensorInduction.decomposition
95	def	PaperXI.TameTensorInduction.decompositionWithCharacters

A.384 PaperXI.TameTensorOrbits

Selected layer: v0.44 parent. **Source lines:** 210.

SHA-256: 9485733d69ff7a1333197a690fb0ae38421460f342371510df530db8047753ec.

Actual simultaneous-translation orbits used in the tame tensor decomposition.

Direct project imports: none.

Line	Kind	Name
13	def	PaperXI.TameTensorOrbits.diagonal
16	theorem	PaperXI.TameTensorOrbits.diagonal_injective
19	theorem	PaperXI.TameTensorOrbits.diagonal_intCast
24	theorem	PaperXI.TameTensorOrbits.diagonal_natCast
30	def	PaperXI.TameTensorOrbits.difference
34	theorem	PaperXI.TameTensorOrbits.difference_diagonal
39	theorem	PaperXI.TameTensorOrbits.difference_add
45	def	PaperXI.TameTensorOrbits.orbitMap
48	theorem	PaperXI.TameTensorOrbits.orbitMap_apply
53	theorem	PaperXI.TameTensorOrbits.orbitMap_zero
57	theorem	PaperXI.TameTensorOrbits.difference_orbitMap
62	theorem	PaperXI.TameTensorOrbits.orbitMap_injective
77	instance	PaperXI.TameTensorOrbits.lcm_neZero
80	instance	PaperXI.TameTensorOrbits.gcd_neZero
83	theorem	PaperXI.TameTensorOrbits.orbitMap_bijective
90	def	PaperXI.TameTensorOrbits.orbitEquiv
93	theorem	PaperXI.TameTensorOrbits.orbitEquiv_apply
97	theorem	PaperXI.TameTensorOrbits.orbitMap_translate
103	def	PaperXI.TameTensorOrbits.translate
107	theorem	PaperXI.TameTensorOrbits.orbitMap_translate_int
116	theorem	PaperXI.TameTensorOrbits.translate_eq_iff
124	theorem	PaperXI.TameTensorOrbits.interaction_count
127	theorem	PaperXI.TameTensorOrbits.interaction_length
131	theorem	PaperXI.TameTensorOrbits.total_dimension
135	theorem	PaperXI.TameTensorOrbits.difference_translate
140	theorem	PaperXI.TameTensorOrbits.sameOrbit_iff
159	theorem	PaperXI.TameTensorOrbits.representatives_sameOrbit_iff
173	def	PaperXI.TameTensorOrbits.fiberEquiv
191	theorem	PaperXI.TameTensorOrbits.fiber_card
196	theorem	PaperXI.TameTensorOrbits.fiberEquiv_apply
200	theorem	PaperXI.TameTensorOrbits.orbitEquiv_symm_translate

A.385 PaperXI.TameTensorPackets

Selected layer: v0.44 parent. **Source lines:** 81.

SHA-256: 507e433f56fdae07efb7824aa8823c182dd07364260ea7b9d594f003e2514334.

```
Actual gcd/lcm decomposition of tensor representations with cyclic monomial
bases. The bases may belong to nonregular or reducible induced representations.
```

Direct project imports: `PaperXI.MonomialOrbit`, `PaperXI.TameDegree`.

Line	Kind	Name
21	def	`PaperXI.TameTensorPackets.coefficient`
26	theorem	`PaperXI.TameTensorPackets.tensor_action`
33	theorem	`PaperXI.TameTensorPackets.orbit_covariance`
39	def	`PaperXI.TameTensorPackets.packet`
46	def	`PaperXI.TameTensorPackets.decomposition`
54	theorem	`PaperXI.TameTensorPackets.packet_basis`
61	theorem	`PaperXI.TameTensorPackets.packet_zero`
70	theorem	`PaperXI.TameTensorPackets.packet_count`
73	theorem	`PaperXI.TameTensorPackets.packet_dimension`
76	theorem	`PaperXI.TameTensorPackets.dimension_sum`

A.386 `PaperXI.TameTensorSource`

Selected layer: v0.44 parent. **Source lines:** 122.

SHA-256: `b52fef0aac236edae7a22103dcf267e147fa13606783529fa64e3c4c4e954d3e`.

```
Source-facing tame tensor decomposition of genuine induced representations.
The Frobenius convention uses representatives `φ^(-i)` and hence the shifted
character `h ↦ η(φ^(-i) h φ^i)`.
```

Direct project imports: `PaperXI.CyclicInducedCharacter`, `PaperXI.TameTensorInduction`.

Line	Kind	Name
18	def	`PaperXI.TameTensorSource.interactionCharacter`
25	theorem	`PaperXI.TameTensorSource.interactionCharacter_apply`
33	theorem	`PaperXI.TameTensorSource.interactionCharacter_literal`
51	theorem	`PaperXI.TameTensorSource.representative_val`
56	theorem	`PaperXI.TameTensorSource.interactionCharacter_weight`
66	def	`PaperXI.TameTensorSource.decomposition`
81	theorem	`PaperXI.TameTensorSource.decomposition_intertwines`
93	theorem	`PaperXI.TameTensorSource.interaction_dimension`
101	theorem	`PaperXI.TameTensorSource.interaction_count`
106	theorem	`PaperXI.TameTensorSource.interaction_dimension_sum`
116	theorem	`PaperXI.TameTensorSource.tensor_dimension`

A.387 `PaperXI.TensorFixed`

Selected layer: v0.44 parent. **Source lines:** 34.

SHA-256: `26c3d415523844b3389e5d7054e76b719d6d67574ac4fda6f67934a3be0c1f41`.

```
Proposition 5.6 for an actual representation with supplied refined packets.
```

Direct project imports: `PaperXI.InvariantAssembly`.

Line	Kind	Name
15	theorem	`PaperXI.TensorFixed.count_trivial_packets`

A.388 `PaperXI.TensorIntertwining107`

Selected layer: base108 arithmetic branch. **Source lines:** 106.

SHA-256: `4bb9b42f6f58022b82ae78cc0fe2d83d0aa6a153c7fc03aed66dd2ba7ea6f095`.

```
The operator argument underlying BK 5.3.2. The arithmetic support and
one-dimensional Heisenberg-intertwining results are kept as explicit inputs.
```

Direct project imports: `PaperXI.TensorSlices107`, `PaperXI.IntertwiningAlong105`.

Line	Kind	Name
20	theorem	`PaperXI.TensorIntertwining107.slice_along`
36	theorem	`PaperXI.TensorIntertwining107.restrict_support`
45	theorem	`PaperXI.TensorIntertwining107.second_along`
62	theorem	`PaperXI.TensorIntertwining107.map_along`
76	theorem	`PaperXI.TensorIntertwining107.factor_along`
88	theorem	`PaperXI.TensorIntertwining107.existsAt_iff`

A.389 `PaperXI.TensorIrreducible`

Selected layer: v0.44 parent. **Source lines:** 78.

SHA-256: `f75b3acfdfeba7e36d31c8d0085caffec072b5c712c4b76f407d8a0b65d5f721`.

```
A dimension criterion for irreducibility of an actual central-character
representation. Simplicity proves injectivity of its actual algebra action;
the dimension identity then proves surjectivity and irreducibility.
```

Direct project imports: `PaperXI.CentralExtensionUniqueness`.

Line	Kind	Name
18	theorem	`PaperXI.TensorIrreducible.finrank_sq_of_irreducible`
32	theorem	`PaperXI.TensorIrreducible.action_injective`
42	theorem	`PaperXI.TensorIrreducible.action_surjective`
58	theorem	`PaperXI.TensorIrreducible.irreducible`

A.390 `PaperXI.TensorPolynomial`

Selected layer: v0.44 parent. **Source lines:** 89.

SHA-256: `45345b4d384aa9e69de8e139f5f1cd75cc18a97728cf3ba6d5d6f53413956ca7`.

```
Corollary 5.5: actual inertia determinant and rational local-factor products.
```

Direct project imports: `PaperXI.PolynomialAssembly`.

Line	Kind	Name
13	theorem	`PaperXI.TensorPolynomial.quotient_stable`
22	theorem	`PaperXI.TensorPolynomial.restricted_eq_factored`
33	theorem	`PaperXI.TensorPolynomial.cyclePolynomial_ne_zero`
56	theorem	`PaperXI.TensorPolynomial.total_stable`
63	theorem	`PaperXI.TensorPolynomial.polynomial_packets`
75	theorem	`PaperXI.TensorPolynomial.determinant_packets`
82	theorem	`PaperXI.TensorPolynomial.localFactor_packets`

A.391 `PaperXI.TensorSlices107`

Selected layer: base108 arithmetic branch. **Source lines:** 115.

SHA-256: `0176d323e1f5d380a8aac25c784710e8b877b7d4aca3fa718eb098f4f08b954b`.

```
Slice an actual tensor operator and reconstruct it when all its first-factor
slices belong to one nonzero line. No invertibility of that line's generator,
and no finite-dimensionality of either carrier, is assumed.
```

Direct project imports: none.

Line	Kind	Name
14	def	`PaperXI.TensorSlices107.right`
17	def	`PaperXI.TensorSlices107.left`
20	theorem	`PaperXI.TensorSlices107.right_tmul`
23	theorem	`PaperXI.TensorSlices107.left_tmul`
26	theorem	`PaperXI.TensorSlices107.right_ext`
35	theorem	`PaperXI.TensorSlices107.contractions_commute`
42	theorem	`PaperXI.TensorSlices107.right_map`
49	def	`PaperXI.TensorSlices107.slice`
53	theorem	`PaperXI.TensorSlices107.slice_apply`
57	theorem	`PaperXI.TensorSlices107.exists_nonzero_slice`
71	theorem	`PaperXI.TensorSlices107.factor_of_slice_line`
97	theorem	`PaperXI.TensorSlices107.tmul_left_injective`
104	theorem	`PaperXI.TensorSlices107.map_left_injective`

A.392 `PaperXI.TensorSubspace`

Selected layer: v0.44 parent. **Source lines:** 76.

SHA-256: `e65d7a7aea5e82ccccba7c606a3a314c89834d52f16441aa2d64c184362a5038`.

```
Tensoring a multiplicity subspace with a nonzero finite-dimensional
factor preserves whether it is zero or the whole space. The actual image
is transported through a specified, independently proved tensor equivalence.
```

Direct project imports: none.

Line	Kind	Name
18	def	`PaperXI.TensorSubspace.image`
21	theorem	`PaperXI.TensorSubspace.inclusion_injective`

Line	Kind	Name
25	theorem	`PaperXI.TensorSubspace.map_injective`
29	theorem	`PaperXI.TensorSubspace.tmul_mem`
34	theorem	`PaperXI.TensorSubspace.image_stable`
48	theorem	`PaperXI.TensorSubspace.finrank_image`
55	theorem	`PaperXI.TensorSubspace.image_eq_bot_iff`
61	theorem	`PaperXI.TensorSubspace.image_eq_top_iff`

A.393 `PaperXI.TensorTypeOperators70`

Selected layer: base108 arithmetic branch. **Source lines:** 94.

SHA-256: `e82f0e46bab6ff2d1096ab07053553ddb6218bf13ff00afc296a84e2af2b270c`.

```
Faithful first-factor operator realization in the literal inducing-type image.
The multiplicity space must be nonzero. Neither tensor amplification nor the
ambient induced space is silently identified with the first-factor carrier.
```

Direct project imports: `PaperXI.InducedTypeSpace68`.

Line	Kind	Name
17	def	`PaperXI.TensorTypeOperators70.amplify`
20	theorem	`PaperXI.TensorTypeOperators70.amplify_tmul`
24	theorem	`PaperXI.TensorTypeOperators70.amplify_injective`
37	def	`PaperXI.TensorTypeOperators70.realize`
41	theorem	`PaperXI.TensorTypeOperators70.realize_tmul`
50	theorem	`PaperXI.TensorTypeOperators70.realize_injective`
53	theorem	`PaperXI.TensorTypeOperators70.realize_eq_iff`
56	theorem	`PaperXI.TensorTypeOperators70.realize_zero_iff`
61	theorem	`PaperXI.TensorTypeOperators70.residual_zero_iff`
66	theorem	`PaperXI.TensorTypeOperators70.projector_iff`
71	theorem	`PaperXI.TensorTypeOperators70.intertwining_iff`
77	theorem	`PaperXI.TensorTypeOperators70.word_product`
84	theorem	`PaperXI.TensorTypeOperators70.holonomy_iff`
89	theorem	`PaperXI.TensorTypeOperators70.weighted_sum`

A.394 `PaperXI.TraceDiscrepancy`

Selected layer: v0.44 parent. **Source lines:** 84.

SHA-256: `0bdba459861ec52105d2a28d2acdd6d4f0f5ac5d7a55cd93ddff6186ed822e15`.

```
A checked normalization route from actual operator traces to discrepancy.
Nonzero normalizing traces are essential and are retained in every cancellation.
```

Direct project imports: `PaperXI.CharacterGenerators`.

Line	Kind	Name
14	theorem	`PaperXI.TraceDiscrepancy.conjugate_eq`
20	theorem	`PaperXI.TraceDiscrepancy.trace_eq`
26	theorem	`PaperXI.TraceDiscrepancy.scalar_eq_one`

Line	Kind	Name
35	theorem	`PaperXI.TraceDiscrepancy.normalized_trace_eq`
41	theorem	`PaperXI.TraceDiscrepancy.scalar_eq_one_of_normalized`
51	theorem	`PaperXI.TraceDiscrepancy.character_trivial`
63	theorem	`PaperXI.TraceDiscrepancy.zero_trace_cannot_detect_scalar`
67	theorem	`PaperXI.TraceDiscrepancy.bridge_from_traces`

A.395 `PaperXI.TranslationAverage`

Selected layer: v0.44 parent. **Source lines:** 102.

SHA-256: `7b04c80800e59fc6ced892041eb4fe9306e01e08e18e61e79d27c07acefe9d1e`.

```
The translation representation and its rank-one constant-function projector.
```

Direct project imports: `PaperXI.FiniteAverage`.

Line	Kind	Name
13	def	`PaperXI.TranslationAverage.translation`
22	theorem	`PaperXI.TranslationAverage.translation_apply`
25	def	`PaperXI.TranslationAverage.mean`
27	def	`PaperXI.TranslationAverage.constantSubmodule`
31	theorem	`PaperXI.TranslationAverage.mem_constantSubmodule`
39	theorem	`PaperXI.TranslationAverage.average_apply`
48	theorem	`PaperXI.TranslationAverage.range_eq`
61	theorem	`PaperXI.TranslationAverage.finrank_range`
71	def	`PaperXI.TranslationAverage.pairing`
73	theorem	`PaperXI.TranslationAverage.pairing_translation`
79	theorem	`PaperXI.TranslationAverage.pairing_self_adjoint`
91	theorem	`PaperXI.TranslationAverage.residual_orthogonal`

A.396 `PaperXI.TwistedAction`

Selected layer: v0.44 parent. **Source lines:** 79.

SHA-256: `97ed5ee03bec3fc432018d895c823299a7cc73ddf65cb1b712417d9948a5eb71`.

```
Universal extension of actual projective operators to an action of the
constructed twisted group algebra.
```

Direct project imports: `PaperXI.TwistedAlgebra`.

Line	Kind	Name
15	def	`PaperXI.TwistedAction.linearExtension`
19	theorem	`PaperXI.TwistedAction.linearExtension_basis`
24	theorem	`PaperXI.TwistedAction.linearExtension_one`
29	theorem	`PaperXI.TwistedAction.linearExtension_mul`
44	def	`PaperXI.TwistedAction.action`
51	theorem	`PaperXI.TwistedAction.action_basis`
58	theorem	`PaperXI.TwistedAction.action_unique`

Line	Kind	Name
70	theorem	`PaperXI.TwistedAction.existsUnique_action`

A.397 `PaperXI.TwistedActionEquivalence`

Selected layer: v0.44 parent. **Source lines:** 107.

SHA-256: `6a14cac5426c7fb2c44d091685da2d07440c4b883f7409f7caff5c2c38919e74`.

```
The actual algebra action of an irreducible projective operator family is
an isomorphism when its scalar commutator is nondegenerate.
```

Direct project imports: `PaperXI.TwistedAction`, `PaperXI.TwistedModules`, `PaperXI.AlgebraActionModule`.

Line	Kind	Name
21	theorem	`PaperXI.TwistedActionEquivalence.stable_iff`
34	theorem	`PaperXI.TwistedActionEquivalence.simpleModule`
47	theorem	`PaperXI.TwistedActionEquivalence.finiteDimensional`
58	theorem	`PaperXI.TwistedActionEquivalence.finrank_sq`
66	theorem	`PaperXI.TwistedActionEquivalence.finrank_sqrt`
71	theorem	`PaperXI.TwistedActionEquivalence.action_injective`
77	theorem	`PaperXI.TwistedActionEquivalence.action_surjective`
92	theorem	`PaperXI.TwistedActionEquivalence.action_bijective`
96	def	`PaperXI.TwistedActionEquivalence.actionEquiv`
100	theorem	`PaperXI.TwistedActionEquivalence.actionEquiv_apply`
103	theorem	`PaperXI.TwistedActionEquivalence.actionEquiv_basis`

A.398 `PaperXI.TwistedAlgebra`

Selected layer: v0.44 parent. **Source lines:** 109.

SHA-256: `578da8f15a3a094d50dbc2d501fc66ea1c8ad354c2a29fb8acb5fd3f4c7b2020`.

```
A concrete realization of the complex twisted group algebra, including its
actual basis, multiplication, units, and dimension. The ambient endomorphism
algebra supplies the ring laws; all closure and basis assertions are proved.
```

Direct project imports: `PaperXI.TwistedRegular`.

Line	Kind	Name
16	def	`PaperXI.TwistedAlgebra.operatorSpan`
19	theorem	`PaperXI.TwistedAlgebra.regular_mem`
22	theorem	`PaperXI.TwistedAlgebra.one_mem`
27	theorem	`PaperXI.TwistedAlgebra.mul_mem`
45	def	`PaperXI.TwistedAlgebra.subalgebra`
57	abbrev	`PaperXI.TwistedAlgebra.Algebra`
62	def	`PaperXI.TwistedAlgebra.basis`
65	theorem	`PaperXI.TwistedAlgebra.coe_basis`
71	theorem	`PaperXI.TwistedAlgebra.basis_mul`
78	theorem	`PaperXI.TwistedAlgebra.basis_one`
84	def	`PaperXI.TwistedAlgebra.basisUnit`

Line	Kind	Name
96	theorem	`PaperXI.TwistedAlgebra.coe_basisUnit`
99	theorem	`PaperXI.TwistedAlgebra.basis_isUnit`
102	instance	`PaperXI.TwistedAlgebra.finiteDimensional`
105	theorem	`PaperXI.TwistedAlgebra.finrank_eq_card`

A.399 `PaperXI.TwistedMatrix`

Selected layer: v0.44 parent. **Source lines:** 54.

SHA-256: `99153ad911c9dab5fd807658f17f2d4d1583b71edd10538aad5c08bfb601e959`.

```
The matrix-algebra part of Paper XI, Proposition 8.3. Simplicity is proved
for the constructed twisted algebra before invoking Wedderburn–Artin. No
finite-image condition is imposed on the cocycle.
```

Direct project imports: `PaperXI.TwistedSimple`.

Line	Kind	Name
18	theorem	`PaperXI.TwistedMatrix.exists_matrix`
27	theorem	`PaperXI.TwistedMatrix.matrix_sqrt`
39	theorem	`PaperXI.TwistedMatrix.order_square`
46	theorem	`PaperXI.TwistedMatrix.classification`

A.400 `PaperXI.TwistedModules`

Selected layer: v0.44 parent. **Source lines:** 54.

SHA-256: `75c7bf4253db1b19641a24e2fbeaa7317eac3362bc5c6427038f89481a072edb`.

```
The simple-module conclusions of Paper XI, Proposition 8.3, for the actual
constructed twisted algebra. Existence, uniqueness, finite dimensionality,
and the square-root dimension are proved under the source hypotheses.
```

Direct project imports: `PaperXI.TwistedMatrix`, `PaperXI.SimpleModuleClassification`.

Line	Kind	Name
18	theorem	`PaperXI.TwistedModules.exists_simple_ideal`
26	theorem	`PaperXI.TwistedModules.equivalent`
40	theorem	`PaperXI.TwistedModules.finiteDimensional`
45	theorem	`PaperXI.TwistedModules.finrank_sq`
51	theorem	`PaperXI.TwistedModules.finrank_sqrt`

A.401 `PaperXI.TwistedRegular`

Selected layer: v0.44 parent. **Source lines:** 81.

SHA-256: `542e57f3ada86dab99247efc614f2e360c4ed62aa6278ac5d318f7cad41242b0`.

```
The faithful twisted left regular operators for an arbitrary normalized
complex unit-valued cocycle. No finite-image hypothesis is used.
```

Direct project imports: `PaperXI.CocycleCommutator`.

Line	Kind	Name
18	def	`PaperXI.TwistedRegular.regular`
23	theorem	`PaperXI.TwistedRegular.regular_apply`
26	theorem	`PaperXI.TwistedRegular.regular_one`
31	theorem	`PaperXI.TwistedRegular.regular_mul`
46	def	`PaperXI.TwistedRegular.deltaOne`
50	theorem	`PaperXI.TwistedRegular.regular_deltaOne`
66	def	`PaperXI.TwistedRegular.evaluateOne`
71	theorem	`PaperXI.TwistedRegular.regular_linearIndependent`

A.402 `PaperXI.TwistedRescaling`

Selected layer: v0.44 parent. **Source lines:** 158.

SHA-256: `1bb12f819e8232e254d6147b5fac209667d032836c4aa02fe72c472015d40e0d`.

```
Actual algebra equivalences under normalized scalar changes of basis.
The equivalence is constructed from the canonical bases of the two concrete
twisted algebras; its multiplicativity follows from their proved tables.
```

Direct project imports: `PaperXI.TwistedAction`, `PaperXI.CocycleBasis`.

Line	Kind	Name
15	def	`PaperXI.TwistedRescaling.congrEquiv`
20	theorem	`PaperXI.TwistedRescaling.congrEquiv_basis`
29	def	`PaperXI.TwistedRescaling.linearEquiv`
36	theorem	`PaperXI.TwistedRescaling.linearEquiv_basis`
46	theorem	`PaperXI.TwistedRescaling.scaled_basis_mul`
56	theorem	`PaperXI.TwistedRescaling.linearEquiv_one`
61	theorem	`PaperXI.TwistedRescaling.linearEquiv_mul`
88	def	`PaperXI.TwistedRescaling.rescaleEquiv`
95	theorem	`PaperXI.TwistedRescaling.rescaleEquiv_basis`
103	theorem	`PaperXI.TwistedRescaling.rescaleEquiv_symm_basis`
116	def	`PaperXI.TwistedRescaling.transport`
122	theorem	`PaperXI.TwistedRescaling.transport_basis`
131	theorem	`PaperXI.TwistedRescaling.transport_unique`
145	def	`PaperXI.TwistedRescaling.actionEquiv`

A.403 `PaperXI.TwistedSimple`

Selected layer: v0.44 parent. **Source lines:** 205.

SHA-256: `f420d05bd41765009035ee02338ae4721a2914e0946fb52c572fcee8ca333c60`.

```
Nondegenerate scalar commutators force simplicity of the actual twisted
algebra. Coordinate elimination is applied to two-sided ideals, so no
semisimplicity or matrix-algebra hypothesis enters the argument.
```

Direct project imports: `PaperXI.CocycleBasis`, `PaperXI.TwistedAlgebra`, `PaperXI.SchrodingerFourier`.

Line	Kind	Name
19	theorem	`PaperXI.TwistedSimple.commutator_separates`
34	theorem	`PaperXI.TwistedSimple.basis_commute`
44	theorem	`PaperXI.TwistedSimple.ideal_smul_mem`
52	theorem	`PaperXI.TwistedSimple.simple_of_basis`
117	theorem	`PaperXI.TwistedSimple.isSimpleRing`
129	theorem	`PaperXI.TwistedSimple.radical_eq_bot_iff`
135	theorem	`PaperXI.TwistedSimple.isSimpleRing_of_nondegenerate`
144	theorem	`PaperXI.TwistedSimple.diagonal_mul_basis`
157	theorem	`PaperXI.TwistedSimple.center_scalar_of_basis`
189	theorem	`PaperXI.TwistedSimple.center_eq_scalars`

A.404 `PaperXI.TypeNormalizerAction`

Selected layer: v0.44 parent. **Source lines:** 108.

SHA-256: `623d5fefcb3b3c856476bbd194fd5a1101347efa5511dbdd0c6a68ffc6e7c2f2`.

```
The actual ambient conjugation action on N/H and its descent to J/N.
The quotient action is constructed from groups, not from choices of operators.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.TypeNormalizerAction.conjugation`
25	theorem	`PaperXI.TypeNormalizerAction.conjugation_coe`
28	theorem	`PaperXI.TypeNormalizerAction.conjugation_inner`
34	def	`PaperXI.TypeNormalizerAction.Stable`
37	theorem	`PaperXI.TypeNormalizerAction.stable_map`
48	def	`PaperXI.TypeNormalizerAction.action`
65	theorem	`PaperXI.TypeNormalizerAction.action_mk`
72	theorem	`PaperXI.TypeNormalizerAction.subgroup_le_kernel`
85	def	`PaperXI.TypeNormalizerAction.quotientAction`
90	theorem	`PaperXI.TypeNormalizerAction.quotientAction_mk`
96	theorem	`PaperXI.TypeNormalizerAction.quotientAction_unique`

A.405 `PaperXI.TypeNormalizerForm`

Selected layer: v0.44 parent. **Source lines:** 85.

SHA-256: `1f05cc7cd75bdb3f15a7b10c3d2153b865baf51da1f73e0a4fe87998c5d6969c`.

```
The source-facing field-valued form: the ambient quotient action is
transported to the declared vector-space coordinates, then preservation is
proved from the actual character commutator. Residue-field linearity is explicit.
```

Direct project imports: `PaperXI.TypeNormalizerPairing`.

Line	Kind	Name
15	def	`PaperXI.TypeNormalizerForm.coordinateAction`
26	theorem	`PaperXI.TypeNormalizerForm.coordinateAction_apply`

Line	Kind	Name
33	def	`PaperXI.TypeNormalizerForm.linearAut`
44	def	`PaperXI.TypeNormalizerForm.vectorAction`
66	theorem	`PaperXI.TypeNormalizerForm.source_form_preserved`

A.406 `PaperXI.TypeNormalizerPairing`

Selected layer: v0.44 parent. **Source lines:** 154.

SHA-256: `47cff0132a4d65fdb6193671065f95af0c45479300e676253cef05a50957f105`.

```
Character invariance yields preservation of the actual quotient pairing.
No form-preservation hypothesis is inserted into the normalizer theorem.
```

Direct project imports: `PaperXI.TypeNormalizerAction`, `PaperXI.TypeSideExtension`.

Line	Kind	Name
17	def	`PaperXI.TypeNormalizerPairing.subgroupAut`
26	def	`PaperXI.TypeNormalizerPairing.CharacterInvariant`
32	theorem	`PaperXI.TypeNormalizerPairing.inertiaInvariant`
43	theorem	`PaperXI.TypeNormalizerPairing.characterCommutator_preserved`
61	def	`PaperXI.TypeNormalizerPairing.pairing`
69	theorem	`PaperXI.TypeNormalizerPairing.pairing_on_representatives`
77	theorem	`PaperXI.TypeNormalizerPairing.quotientAction_preserves_pairing`
93	theorem	`PaperXI.TypeNormalizerPairing.pairing_alternating`
98	theorem	`PaperXI.TypeNormalizerPairing.pairing_mul_left`
106	theorem	`PaperXI.TypeNormalizerPairing.pairing_mul_right`
119	theorem	`PaperXI.TypeNormalizerPairing.bilinear_character_separates`
142	theorem	`PaperXI.TypeNormalizerPairing.fieldForm_preserved`

A.407 `PaperXI.TypeProjectiveImplementation`

Selected layer: v0.44 parent. **Source lines:** 157.

SHA-256: `3621d11e4ad48c3a7d4986dc55e21778d7bf926a4a008b6de91a38ccc21db617`.

```
Implementation by an actual extension representation. Scalars compare
intertwiners for the same automorphism, not arbitrary quotient representatives.
The section defect is retained as an operator until scalarity is proved.
```

Direct project imports: `PaperXI.TypeNormalizerPairing`, `PaperXI.SchurScalar`, `PaperXI.CliffordDatum`.

Line	Kind	Name
17	def	`PaperXI.TypeProjectiveImplementation.operator`
31	theorem	`PaperXI.TypeProjectiveImplementation.operator_apply`
36	theorem	`PaperXI.TypeProjectiveImplementation.implements_conjugation`
50	theorem	`PaperXI.TypeProjectiveImplementation.implementer_unique`
60	theorem	`PaperXI.TypeProjectiveImplementation.two_implementers`
73	theorem	`PaperXI.TypeProjectiveImplementation.section_operators_mul`
81	theorem	`PaperXI.TypeProjectiveImplementation.representative_change`
86	def	`PaperXI.TypeProjectiveImplementation.ProjectiveDescends`

Line	Kind	Name
91	theorem	`PaperXI.TypeProjectiveImplementation.projectiveDescends_iff`
113	theorem	`PaperXI.TypeProjectiveImplementation.scalar_element_commutator`
127	theorem	`PaperXI.TypeProjectiveImplementation.source_operator_commutator`
138	theorem	`PaperXI.TypeProjectiveImplementation.nontrivial_pairing_obstructs_descent`

A.408 `PaperXI.TypeSideDescent`

Selected layer: v0.44 parent. **Source lines:** 293.

SHA-256: `e066d77666e2671d02c454f4bd06e3650750cfcb143a6c4635339d8b6a0adb2f`.

```
The actual type-side scalar restriction and quotient action. The source
input that an irreducible representation contains the invariant character
is used through its nonzero character eigenspace; scalar restriction on
the whole representation is a conclusion, not an additional assumption.
```

Direct project imports: `PaperXI.QuotientRepresentation`, `PaperXI.GeneratedIrreducibility`.

Line	Kind	Name
18	def	`PaperXI.TypeSideDescent.conjugate`
22	def	`PaperXI.TypeSideDescent.Invariant`
26	def	`PaperXI.TypeSideDescent.characterSpace`
39	def	`PaperXI.TypeSideDescent.Contains`
43	def	`PaperXI.TypeSideDescent.characterRepresentation`
53	theorem	`PaperXI.TypeSideDescent.contains_iff_nonzero_intertwiner`
86	theorem	`PaperXI.TypeSideDescent.characterSpace_stable`
99	def	`PaperXI.TypeSideDescent.characterSubrepresentation`
105	theorem	`PaperXI.TypeSideDescent.scalar_restriction`
121	def	`PaperXI.TypeSideDescent.characterKernel`
124	theorem	`PaperXI.TypeSideDescent.characterKernel_le`
129	theorem	`PaperXI.TypeSideDescent.mem_characterKernel`
139	instance	`PaperXI.TypeSideDescent.characterKernel_normal`
147	theorem	`PaperXI.TypeSideDescent.characterKernel_le_ker`
157	def	`PaperXI.TypeSideDescent.representation`
163	theorem	`PaperXI.TypeSideDescent.representation_apply`
169	theorem	`PaperXI.TypeSideDescent.representation_irreducible`
177	theorem	`PaperXI.TypeSideDescent.finite_quotient`
188	def	`PaperXI.TypeSideDescent.quotientProjection`
193	theorem	`PaperXI.TypeSideDescent.quotientProjection_apply`
198	theorem	`PaperXI.TypeSideDescent.quotientProjection_surjective`
203	theorem	`PaperXI.TypeSideDescent.quotientProjection_ker`
210	theorem	`PaperXI.TypeSideDescent.central_image`
229	def	`PaperXI.TypeSideDescent.centralCharacter`
232	theorem	`PaperXI.TypeSideDescent.centralCharacter_apply`
235	theorem	`PaperXI.TypeSideDescent.centralCharacter_injective`
240	def	`PaperXI.TypeSideDescent.centerInclusion`
248	theorem	`PaperXI.TypeSideDescent.centerInclusion_apply`
253	theorem	`PaperXI.TypeSideDescent.centerInclusion_injective`
260	theorem	`PaperXI.TypeSideDescent.centerInclusion_range`
272	theorem	`PaperXI.TypeSideDescent.exact_center_quotient`
277	theorem	`PaperXI.TypeSideDescent.centerInclusion_central`

Line	Kind	Name
285	theorem	`PaperXI.TypeSideDescent.central_action`

A.409 `PaperXI.TypeSideExtension`

Selected layer: v0.44 parent. **Source lines:** 170.

SHA-256: `f00297a9c260398048b04cca886e47dbd0ac912e029104cee4892063e5aab11c`.

```
The actual type-side central extension, its source commutator formula,
and scalar nondegeneracy transferred from a supplied source pairing. These
interfaces retain the simple-character facts as explicit inputs.
```

Direct project imports: `PaperXI.TypeSideDescent`, `PaperXI.CentralExtensionCocycle`.

Line	Kind	Name
20	instance	`PaperXI.TypeSideExtension.centralQuotient_isMulCommutative`
24	theorem	`PaperXI.TypeSideExtension.finite_center`
27	theorem	`PaperXI.TypeSideExtension.centralCharacter_ne_one`
36	def	`PaperXI.TypeSideExtension.datum`
48	theorem	`PaperXI.TypeSideExtension.commutator_mem`
54	def	`PaperXI.TypeSideExtension.characterCommutator`
60	theorem	`PaperXI.TypeSideExtension.scalar_commutator_source`
81	theorem	`PaperXI.TypeSideExtension.scalar_commutator_eq_pairing`
94	theorem	`PaperXI.TypeSideExtension.nondegenerate_of_pairing`
117	theorem	`PaperXI.TypeSideExtension.bilinear_character_nondegenerate`
138	theorem	`PaperXI.TypeSideExtension.nondegenerate_of_bilinear_pairing`
162	theorem	`PaperXI.TypeSideExtension.representation_central_action`

A.410 `PaperXI.TypeSideUniqueness`

Selected layer: v0.44 parent. **Source lines:** 96.

SHA-256: `3a4c2a7589f7cccf49eec7161e463a3c6d21f217693a596505efa5d339330a63`.

```
Uniqueness and dimension for the original actual type-side actions.
Character containment is converted to scalar restriction and quotient
descent by proved constructions. The source simple-character structure and
nondegenerate pairing remain explicit inputs. The first intertwiner and
the representation dimensions are conclusions.
```

Direct project imports: `PaperXI.TypeSideExtension`, `PaperXI.CentralExtensionUniqueness`.

Line	Kind	Name
21	theorem	`PaperXI.TypeSideUniqueness.finiteDimensional`
34	theorem	`PaperXI.TypeSideUniqueness.finrank_sqrt`
50	theorem	`PaperXI.TypeSideUniqueness.exists_equiv`
71	theorem	`PaperXI.TypeSideUniqueness.exists_unique_intertwiner`

A.411 `PaperXI.UnramifiedBaseChangeFactor`

Selected layer: v0.44 parent. **Source lines:** 159.

SHA-256: `0f6e188bb24bf02711cdd8d676182d365a8d0fac73456675899fa0f4564a7340`.

```
# Integer-degree base change for the finite unramified Weil model

The finite unramified model is canonically equivalent to `Multiplicative ℤ`.
Multiplication of the degree by `n` therefore gives the restriction map that
models unramified base change of residue degree `n`. This module proves the
resulting Frobenius-power and Euler-factor formulas.

No actual tower of local fields is constructed here; the theorem is scoped to
this integer-degree Weil model.
```

Direct project imports: `PaperXI.RankOneEulerFactor`.

Line	Kind	Name
30	def	`PaperXI.FiniteUnramifiedWeil.integerDegreeDilation`
38	def	`PaperXI.FiniteUnramifiedWeil.degreeDilation`
43	theorem	`PaperXI.FiniteUnramifiedWeil.degreeEquiv_degreeDilation`
49	theorem	`PaperXI.FiniteUnramifiedWeil.degree_degreeDilation`
61	theorem	`PaperXI.FiniteUnramifiedWeil.degreeDilation_geometricLift`
71	theorem	`PaperXI.FiniteUnramifiedWeil.degreeDilation_comp`
84	def	`PaperXI.FiniteUnramifiedWeil.baseChangeRepresentation`
91	theorem	`PaperXI.FiniteUnramifiedWeil.baseChange_geometricFrobenius`
100	theorem	`PaperXI.FiniteUnramifiedWeil.baseChangeRepresentation_comp`
113	theorem	`PaperXI.FiniteUnramifiedWeil.baseChange_frobeniusPolynomial`
121	theorem	`PaperXI.FiniteUnramifiedWeil.baseChange_localFactor`
131	theorem	`PaperXI.FiniteUnramifiedWeil.irreducible_baseChange_frobeniusPolynomial`
151	theorem	`PaperXI.FiniteUnramifiedWeil.irreducible_baseChange_localFactor`

A.412 `PaperXI.UnramifiedConductor`

Selected layer: v0.44 parent. **Source lines:** 131.

SHA-256: `1a940bc977e7fb0565338370fce87b395fc254c3cbbc39c220d61eef0a6de682`.

```
Restriction across an inertia-containing subgroup, with a power change of
residue Frobenius. The congruence groups, fixed spaces and EVERY lower conductor
summand are preserved, for wild as well as tame representations. This is not the
ramified-quotient/Herbrand change-of-extension theorem.
```

Direct project imports: `PaperXI.ResidueDegreeBaseChange`, `PaperXI.ValuationConductor`.

Line	Kind	Name
18	theorem	`PaperXI.UnramifiedConductor.lower_eq_comap`
21	def	`PaperXI.UnramifiedConductor.lowerEquiv`
28	theorem	`PaperXI.UnramifiedConductor.lower_card`
33	def	`PaperXI.UnramifiedConductor.embedding`
47	theorem	`PaperXI.UnramifiedConductor.embedding_degree`
52	theorem	`PaperXI.UnramifiedConductor.embedding_injective`
65	theorem	`PaperXI.UnramifiedConductor.embedding_lift`
77	theorem	`PaperXI.UnramifiedConductor.fixed_eq`
95	theorem	`PaperXI.UnramifiedConductor.summand_eq`
103	theorem	`PaperXI.UnramifiedConductor.swan_eq`
112	theorem	`PaperXI.UnramifiedConductor.artin_eq`
120	theorem	`PaperXI.UnramifiedConductor.summable_restriction`

A.413 PaperXI.UnramifiedConductorZero

Selected layer: v0.44 parent. **Source lines:** 109.

SHA-256: 35a3f791e30467fdb145456cea233a2a85f3fc669b8cc9a5216db7b5a705f19d.

```
# Conductor zero for the finite unramified Weil model

This module does two separate things.

* It proves a generic theorem in the existing `ValuationConductor` interface:
  a representation has Artin conductor zero when the actual degree-kernel inertia
  subgroup is trivial.
* It packages the corresponding codimension/Swan/Artin exponents for the finite
  unramified Weil model constructed in Block 39.

The model-specific exponents are not asserted to be the conductor of an absolute
Weil representation. They record the exact unramified specialization of the
existing fixed-space formula.
```

Direct project imports: `PaperXI.UnramifiedLocalFactor`, `PaperXI.ValuationConductor`.

Line	Kind	Name
34	theorem	`PaperXI.ValuationConductor.artin_zero_of_degree_kernel_bot`
73	def	`PaperXI.FiniteUnramifiedWeil.inertiaCodimension`
80	def	`PaperXI.FiniteUnramifiedWeil.swanConductorExponent`
84	def	`PaperXI.FiniteUnramifiedWeil.artinConductorExponent`
89	theorem	`PaperXI.FiniteUnramifiedWeil.inertiaCodimension_eq_zero`
97	theorem	`PaperXI.FiniteUnramifiedWeil.swanConductorExponent_eq_zero`
103	theorem	`PaperXI.FiniteUnramifiedWeil.artinConductorExponent_eq_zero`

A.414 PaperXI.UnramifiedFieldTower

Selected layer: v0.44 parent. **Source lines:** 270.

SHA-256: 79699a103ad55a1254dbaa1ea632d59bd77ce2f4ac03367b50d40d1b35f287ee.

```
# Recursive unramified local-field towers

This module constructs the upper field over the already constructed lower field,
using the checked residue-field equivalence to transport a finite residue tower.
The resulting map on the integer-degree unramified Weil models is defined by the
actual relative residue degree.  No absolute Weil group is asserted.
```

Direct project imports: `PaperXI.UnramifiedBaseChangeFactor`.

Line	Kind	Name
32	abbrev	`PaperXI.UnramifiedFieldTower.LowerField`
36	def	`PaperXI.UnramifiedFieldTower.residueTowerMap`
53	abbrev	`PaperXI.UnramifiedFieldTower.UpperField`
69	def	`PaperXI.UnramifiedFieldTower.lowerToUpper`
75	theorem	`PaperXI.UnramifiedFieldTower.residue_finrank`
83	theorem	`PaperXI.UnramifiedFieldTower.relative_degree`
90	theorem	`PaperXI.UnramifiedFieldTower.total_degree`
98	def	`PaperXI.UnramifiedFieldTower.upperResidueAlgEquiv`
104	theorem	`PaperXI.UnramifiedFieldTower.relative_ramificationIdx_one`
111	def	`PaperXI.UnramifiedFieldTower.restriction`

Line	Kind	Name
120	theorem	`PaperXI.UnramifiedFieldTower.restriction_degree_square`
127	theorem	`PaperXI.UnramifiedFieldTower.restriction_degree`
136	theorem	`PaperXI.UnramifiedFieldTower.restriction_geometricLift`
159	theorem	`PaperXI.UnramifiedFieldTower.restriction_injective`
187	theorem	`PaperXI.UnramifiedFieldTower.restriction_range`
217	def	`PaperXI.UnramifiedFieldTower.restrictionEquivPeriod`
228	theorem	`PaperXI.UnramifiedFieldTower.restriction_unique`
246	def	`PaperXI.UnramifiedFieldTower.restrictRepresentation`
253	theorem	`PaperXI.UnramifiedFieldTower.restrictRepresentation_geometricLift`
261	theorem	`PaperXI.UnramifiedFieldTower.restriction_frobeniusPolynomial`

A.415 `PaperXI.UnramifiedLocalFactor`

Selected layer: v0.44 parent. **Source lines:** 168.

SHA-256: `ca0d3e6fba2b3caa2549ac65cd50e4fe66b15dc4cf96c9398531f826b265ebbd`.

```
# Unramified local factors for the finite Weil model
```

Direct project imports: `PaperXI.FiniteUnramifiedWeil`.

Line	Kind	Name
28	abbrev	`PaperXI.FiniteUnramifiedWeil.Inertia`
33	theorem	`PaperXI.FiniteUnramifiedWeil.inertia_le_kernel`
39	theorem	`PaperXI.FiniteUnramifiedWeil.inertia_invariants_eq_top`
47	def	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial`
52	def	`PaperXI.FiniteUnramifiedWeil.unramifiedLocalFactor`
59	def	`PaperXI.FiniteUnramifiedWeil.inertiaFixedEquiv`
64	def	`PaperXI.FiniteUnramifiedWeil.fixedGeometricFrobenius`
70	theorem	`PaperXI.FiniteUnramifiedWeil.fixedGeometricFrobenius_charpoly`
77	def	`PaperXI.FiniteUnramifiedWeil.inertiaFrobeniusPolynomial`
83	theorem	`PaperXI.FiniteUnramifiedWeil.inertiaFrobeniusPolynomial_eq`
90	def	`PaperXI.FiniteUnramifiedWeil.inertiaLocalFactor`
95	theorem	`PaperXI.FiniteUnramifiedWeil.inertiaLocalFactor_eq`
103	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_eq_det`
114	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_coeff_zero`
121	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_eval_zero`
127	theorem	`PaperXI.FiniteUnramifiedWeil.frobeniusPolynomial_ne_zero`
136	theorem	`PaperXI.FiniteUnramifiedWeil.mappedFrobeniusPolynomial_ne_zero`
143	theorem	`PaperXI.FiniteUnramifiedWeil.unramifiedLocalFactor_mul_polynomial`
151	theorem	`PaperXI.FiniteUnramifiedWeil.polynomial_mul_unramifiedLocalFactor`
159	theorem	`PaperXI.FiniteUnramifiedWeil.unramifiedLocalFactor_ne_zero`

A.416 `PaperXI.UnramifiedPolynomial`

Selected layer: v0.44 parent. **Source lines:** 90.

SHA-256: `2dfea600d7050f62c8adde4b0242ca02748ec7427f610c1876bc40b1d7a2f64b`.

```
An irreducible residue polynomial constructs a finite etale DVR and its
actual fraction field, without any supplied ambient Galois extension.
```

Direct project imports: `PaperXI.MonicResidueAlgebra`.

Line	Kind	Name
21	theorem	`PaperXI.UnramifiedPolynomial.irreducible`
24	instance	`PaperXI.UnramifiedPolynomial.domain`
27	instance	`PaperXI.UnramifiedPolynomial.faithful`
30	theorem	`PaperXI.UnramifiedPolynomial.not_field`
37	instance	`PaperXI.UnramifiedPolynomial.dvr`
44	theorem	`PaperXI.UnramifiedPolynomial.uniformizer`
50	abbrev	`PaperXI.UnramifiedPolynomial.field`
52	instance	`PaperXI.UnramifiedPolynomial.baseFieldAlgebra`
55	instance	`PaperXI.UnramifiedPolynomial.baseFieldTower`
58	theorem	`PaperXI.UnramifiedPolynomial.natDegree_pos_of_residue_irreducible`
62	theorem	`PaperXI.UnramifiedPolynomial.field_degree`
66	instance	`PaperXI.UnramifiedPolynomial.finiteFieldExtension`
71	theorem	`PaperXI.UnramifiedPolynomial.integralClosure`
78	theorem	`PaperXI.UnramifiedPolynomial.etale`
85	theorem	`PaperXI.UnramifiedPolynomial.ramificationIdx_one`

A.417 `PaperXI.UnramifiedValuation`

Selected layer: v0.44 parent. **Source lines:** 145.

SHA-256: `001406e3abfac1991c15f4006373c9a8673cd0e8e16df3d80a9ef6c276a50054`.

```
Source-facing unramified residue-base attachment to an actual valuation
subring and decomposition-group action. The subgroup fixes the actual embedded
finite residue extension, rather than being assigned an index by a certificate.
This still does not construct its lift to a local-field unramified extension.
```

Direct project imports: `PaperXI.FiniteResidueSubgroup`, `PaperXI.UnramifiedConductor`, `PaperXI.ValuationTameEndpoint`.

Line	Kind	Name
22	abbrev	`PaperXI.UnramifiedValuation.fixingGroup`
27	theorem	`PaperXI.UnramifiedValuation.frobenius_agrees`
32	theorem	`PaperXI.UnramifiedValuation.contains_inertia`
37	def	`PaperXI.UnramifiedValuation.embedding`
45	theorem	`PaperXI.UnramifiedValuation.degree_formula`
54	def	`PaperXI.UnramifiedValuation.degreeSubgroupEquiv`
61	theorem	`PaperXI.UnramifiedValuation.embedding_injective`
66	theorem	`PaperXI.UnramifiedValuation.embedding_range`
93	theorem	`PaperXI.UnramifiedValuation.degree_index`
103	theorem	`PaperXI.UnramifiedValuation.norm_compatibility`
118	theorem	`PaperXI.UnramifiedValuation.source`

A.418 `PaperXI.ValuationConductor`

Selected layer: v0.44 parent. **Source lines:** 167.

SHA-256: `3040c48c4558f4796813146cac1147f5d7bf87a20ff05cabecf53a37fce73cb0`.

```
Conductor sums from actual maximal-ideal congruence groups. For a finite
Galois extension with discrete valuation these are the lower-numbering expressions.
No change-of-extension theorem, absolute Weil comparison, or Hasse--Arf integrality
is inferred from this definition. Finite support is proved under the stated
Noetherian/finite faithful action conditions.
```

Direct project imports: `PaperXI.ResidueWeil`, `PaperXI.RamificationSeparation`.

Line	Kind	Name
19	abbrev	`PaperXI.ValuationConductor.residue`
23	def	`PaperXI.ValuationConductor.lift`
32	theorem	`PaperXI.ValuationConductor.lift_degree`
38	def	`PaperXI.ValuationConductor.fixed`
43	theorem	`PaperXI.ValuationConductor.fixed_zero`
61	def	`PaperXI.ValuationConductor.codimension`
66	def	`PaperXI.ValuationConductor.summand`
70	def	`PaperXI.ValuationConductor.swan`
74	def	`PaperXI.ValuationConductor.artin`
79	theorem	`PaperXI.ValuationConductor.positive_fixed_top`
88	theorem	`PaperXI.ValuationConductor.tame_summand_zero`
96	theorem	`PaperXI.ValuationConductor.tame_swan_zero`
102	theorem	`PaperXI.ValuationConductor.tame_irreducible_dimension`
116	theorem	`PaperXI.ValuationConductor.tame_artin_dichotomy`
127	theorem	`PaperXI.ValuationConductor.fixed_top_of_lower_bot`
138	theorem	`PaperXI.ValuationConductor.summable_summands`
151	theorem	`PaperXI.ValuationConductor.zeroth_summand`
160	theorem	`PaperXI.ValuationConductor.artin_eq_full_sum`

A.419 `PaperXI.ValuationFixingField`

Selected layer: v0.44 parent. **Source lines:** 41.

SHA-256: `fdaeae5757fb111b416bf264d6b76e43cead77d0d338c4bc16110191bedaeb06`.

```
Restricting the base field in the literal decomposition subgroup.
```

Direct project imports: `PaperXI.HenselianFieldLift`.

Line	Kind	Name
12	def	`PaperXI.ValuationFixingField.decompositionEquiv`
27	theorem	`PaperXI.ValuationFixingField.decompositionEquiv_apply`
31	theorem	`PaperXI.ValuationFixingField.ring_action`
35	theorem	`PaperXI.ValuationFixingField.residue_action`

A.420 `PaperXI.ValuationIntegralModel`

Selected layer: v0.44 parent. **Source lines:** 72.

SHA-256: `c126174463d0f9039e8425c76709f0ab71b7babdb8e5ed6b19aa18a6169ffd3c`.

```
Integral closure and finiteness of an invariant valuation ring.
```

Direct project imports: `PaperXI.HenselianGaloisResidue`.

Line	Kind	Name
13	abbrev	`PaperXI.ValuationIntegralModel.base`
15	instance	`PaperXI.ValuationIntegralModel.baseAlgebra`
17	instance	`PaperXI.ValuationIntegralModel.baseTower`
19	instance	`PaperXI.ValuationIntegralModel.baseFieldTower`
21	instance	`PaperXI.ValuationIntegralModel.baseLocal`
25	def	`PaperXI.ValuationIntegralModel.ringAction`
31	theorem	`PaperXI.ValuationIntegralModel.invariant`
50	theorem	`PaperXI.ValuationIntegralModel.integral`
56	theorem	`PaperXI.ValuationIntegralModel.integralClosure`
62	theorem	`PaperXI.ValuationIntegralModel.finite`
67	theorem	`PaperXI.ValuationIntegralModel.rank`

A.421 `PaperXI.ValuationRamification`

Selected layer: v0.44 parent. **Source lines:** 96.

SHA-256: `92ddb509dc327389c2168f44ddfd5f7e16f4315af16835a54398077fe81146ca`.

```
Lower congruence subgroups are defined by their action on an actual local ring.
For a discrete valuation ring in a finite Galois extension these are the usual
nonnegative lower ramification groups. The generic local-ring definition is NOT
asserted to be an absolute Weil ramification filtration.
```

Direct project imports: none.

Line	Kind	Name
15	def	`PaperXI.ValuationRamification.lower`
30	theorem	`PaperXI.ValuationRamification.mem_lower`
34	theorem	`PaperXI.ValuationRamification.smul_mem_power`
48	instance	`PaperXI.ValuationRamification.lower_normal`
55	theorem	`PaperXI.ValuationRamification.antitone`
60	theorem	`PaperXI.ValuationRamification.lower_zero_eq_residue_kernel`
80	theorem	`PaperXI.ValuationRamification.lower_le_inertia`
85	theorem	`PaperXI.ValuationRamification.positive_le_one`
91	theorem	`PaperXI.ValuationRamification.valuation_lower_zero`

A.422 `PaperXI.ValuationResidueEmbedding`

Selected layer: v0.44 parent. **Source lines:** 48.

SHA-256: `ee803ae38b207f0084f4fe3d41d3b6348f38ade5e33a9ca430804b2e35a4c98f`.

```
Residue embedding of an actual restricted valuation ring.
```

Direct project imports: `PaperXI.HenselianFieldLift`.

Line	Kind	Name
10	def	`PaperXI.ValuationResidueEmbedding.inclusion`

Line	Kind	Name
16	theorem	`PaperXI.ValuationResidueEmbedding.inclusion_val`
18	instance	`PaperXI.ValuationResidueEmbedding.inclusion_local`
33	def	`PaperXI.ValuationResidueEmbedding.residueMap`
35	theorem	`PaperXI.ValuationResidueEmbedding.residueMap_residue`
37	theorem	`PaperXI.ValuationResidueEmbedding.residueMap_injective`
39	theorem	`PaperXI.ValuationResidueEmbedding.mem_range_iff`

A.423 `PaperXI.ValuationTameEndpoint`

Selected layer: v0.44 parent. **Source lines:** 52.

SHA-256: `613a021b32c397cfc2425e8ee5c9cb796856bc8643069f24fde8b8f634df23d9`.

```
Actual valuation-ring specialization with a constructed geometric residue
Frobenius. The finite residue-base embedding remains declared. There is no
claimed identification with an absolute Weil group or extension-independent
arithmetic conductor.
```

Direct project imports: `PaperXI.ValuationConductor`.

Line	Kind	Name
20	def	`PaperXI.ValuationTameEndpoint.geometricFrobenius`
24	theorem	`PaperXI.ValuationTameEndpoint.geometric_power`
28	abbrev	`PaperXI.ValuationTameEndpoint.WeilPullback`
30	def	`PaperXI.ValuationTameEndpoint.inertiaIdentification`
38	theorem	`PaperXI.ValuationTameEndpoint.conductor`

APPENDIX B

Navigation across the original 66 obligations

This navigation table extracts the original 66 proof-obligation records from the historical register and gives an entry point into each corresponding source family. Entry modules lead to the definitions and related results for the whole section. They are not assertions that one module alone proves every displayed claim. The full source hypotheses and relevant ordinary mathematical attachments remain decisive.

The historical register contains 102 numbered records, including definitions, imported facts, examples and remarks. Original obligation 66 is printed Theorem 16.2, all-records row 95. Its historical "proved internally" flag is superseded in this reading edition by the conjectural-realization and conditional-comparison split. The other historical record types are excluded from this 66-row table. The printed numbers in this historical column now link to the corresponding current monograph destinations. Historical PDF page numbers remain only in the unchanged JSON crosswalk. In the last row, historical 16.2 links to Conjecture 66. The separate conditional comparison is the comparison of one realized candidate. These are reading links, not new theorem-equivalence or compiler certificates.

Obl.	Row	Printed	Original title	Source family entry
1	3	2.3	Exact block reconstruction	`PaperXI.AtomicReconstruction`
2	4	2.4	Scalar covariance model	`PaperXI.AtomicReconstruction`
3	5	2.5	Formal atomic reconstruction	`PaperXI.AtomicReconstruction`
4	7	3.1	Tame character projectors	`PaperXI.TameProjectors`
5	8	3.2	Tame orbit induction	`PaperXI.TameProjectors`
6	9	3.3	Dimension and regularity	`PaperXI.TameProjectors`
7	10	3.4	Dual and self dual tame packets	`PaperXI.TameProjectors`
8	11	3.5	Unramified twist	`PaperXI.TameProjectors`
9	12	3.6	Möbius orbit count	`PaperXI.TameProjectors`
10	15	4.2	Primitive cuspidal idempotent	`PaperXI.CharacterProjector`
11	16	4.3	Parabolic annihilation	`PaperXI.CharacterProjector`
12	17	4.4	Frobenius cycle polynomial	`PaperXI.CharacterProjector`
13	18	4.5	Unramified orientation character	`PaperXI.CharacterProjector`
14	19	5.1	Greatest common divisor and least common multiple packets	`PaperXI.TameTensorSource`
15	20	5.2	Packet count and depth	`PaperXI.TameTensorSource`
16	21	5.3	Refined exponent formula	`PaperXI.TameTensorSource`
17	22	5.4	Local polynomial of one tame packet	`PaperXI.TameTensorSource`
18	23	5.5	Tame tensor local factor	`PaperXI.TameTensorSource`
19	24	5.6	Trivial tensor packet criterion	`PaperXI.TameTensorSource`
20	25	5.7	Tame Hom test	`PaperXI.TameTensorSource`
21	26	5.8	Conductor of a nontrivial tame atom	`PaperXI.TameTensorSource`
22	27	6.1	Gauss magnitude	`PaperXI.GaussCharacters`
23	29	6.3	Duality of Gauss phases	`PaperXI.GaussCharacters`
24	33	7.2	Finite Schrödinger representation	`PaperXI.HeisenbergRepresentation`
25	34	7.3	Character support	`PaperXI.HeisenbergRepresentation`
26	36	7.5	Isotropic averaging projector	`PaperXI.HeisenbergRepresentation`
27	37	7.6	Primitive line selector	`PaperXI.HeisenbergRepresentation`

Obl.	Row	Printed	Original title	Source family entry
28	38	8.1	Scalar choice invariance	`PaperXI.TwistedMatrix`
29	39	8.2	Alternating commutator form	`PaperXI.TwistedMatrix`
30	40	8.3	Nondegenerate twisted group algebra	`PaperXI.TwistedMatrix`
31	41	8.4	Radical free envelope	`PaperXI.TwistedMatrix`
32	44	9.2	Clifford multiplicity action	`PaperXI.CliffordSource`
33	46	9.4	Intrinsic commutator pairing	`PaperXI.CliffordSource`
34	48	9.6	Nondegenerate one break packet	`PaperXI.CliffordSource`
35	49	9.7	Isomorphism invariance	`PaperXI.CliffordSource`
36	50	9.8	Tame stabilizer action	`PaperXI.CliffordSource`
37	51	9.9	Heisenberg representation attached to one break	`PaperXI.CliffordSource`
38	55	10.3	Type side projective commutator	`PaperXI.TypeNormalizerAction`
39	57	11.2	Restricted Heisenberg bridge	`PaperXI.CyclicBridge`
40	58	11.3	Cyclic discrepancy character	`PaperXI.CyclicBridge`
41	59	11.4	Linearization over a cyclic stabilizer	`PaperXI.CyclicBridge`
42	60	11.5	One phase closes the cyclic bridge	`PaperXI.CyclicBridge`
43	61	11.6	Bridge uniqueness	`PaperXI.CyclicBridge`
44	65	12.3	Symplectic invariance	`PaperXI.OddPhaseCertificate`
45	66	12.4	Orthogonal multiplicativity	`PaperXI.OddPhaseCertificate`
46	67	12.5	Characteristic polynomial certificate	`PaperXI.OddPhaseCertificate`
47	68	12.6	Fixed and moving decomposition	`PaperXI.OddPhaseCertificate`
48	70	12.8	Hyperbolic phase formula	`PaperXI.OddPhaseCertificate`
49	71	12.9	Phase only bridge test	`PaperXI.OddPhaseCertificate`
50	74	13.2	Affine space of refinements	`PaperXI.DyadicAffine`
51	77	13.5	Dyadic Gauss sign	`PaperXI.DyadicAffine`
52	78	13.6	Orthogonal dyadic multiplicativity	`PaperXI.DyadicAffine`
53	79	13.7	The form alone does not determine the phase	`PaperXI.DyadicAffine`
54	80	13.8	Translation of quadratic refinements	`PaperXI.DyadicAffine`
55	81	13.9	Number of dyadic refinements	`PaperXI.DyadicAffine`
56	83	14.2	Matrix holonomy criterion	`PaperXI.GraphGauge`
57	84	14.3	Unbased gauge classification	`PaperXI.GraphGauge`
58	85	14.4	Scalar and projective gluing	`PaperXI.GraphGauge`
59	86	14.5	Projective cocycle form	`PaperXI.GraphGauge`
60	87	14.6	Exact edge residual from cycle defect	`PaperXI.GraphGauge`
61	88	14.7	Quantitative finite gluing	`PaperXI.GraphGauge`
62	91	15.2	Heisenberg tensor factorization	`PaperXI.FiniteBreakAssembly`
63	92	15.3	Breakwise phase product	`PaperXI.FiniteBreakAssembly`
64	93	15.4	Breakwise restricted bridge	`PaperXI.FiniteBreakAssembly`
65	94	16.1	Finite tame comparison package	`PaperXI.FiniteTameComparison`
66	95	16.2	Conditional atomic comparison	`PaperXI.ConditionalAtomicComparison`

APPENDIX C

Excerpt and claim crosswalk

The following index maps teaching excerpts to the preserved formal source. Original proof obligation 66 corresponds to printed Theorem 16.2 and all-records row 95 in the historical 102-record register. Historical register row 66 is a different record, Proposition 12.4. The historical row 95 status "proved internally" is preserved as historical metadata and is superseded by this edition's conjectural-realization and conditional-comparison split. The `ConditionalAtomicComparison.source` interface is a proved conditional implication, while its general arithmetic-realization premise is the conjectural publication endpoint. Proposition 16.1's finite package appears in `FiniteTameComparison`; an arithmetic local-factor interpretation is a separate attachment.

ID	Module	Source lines	Page
E01	`PaperXI.FourierProjectors`	13–15	8
E02	`PaperXI.FourierProjectors`	68–77	8
E03	`PaperXI.PacketTransport`	62–66	8
E04	`PaperXI.TameProjectors`	45–52	9
E05	`PaperXI.InducedCycleInvariants`	33–33	10
E06	`PaperXI.InducedCycleInvariants`	75–77	10
E07	`PaperXI.TameTensorSource`	66–69	10
E08	`PaperXI.TameConductor`	56–58	12
E09	`PaperXI.FiniteTameComparison`	93–107	12
E10	`PaperXI.HeisenbergRepresentation`	14–16	13
E11	`PaperXI.HeisenbergRepresentation`	58–59	13
E12	`PaperXI.CentralProduct`	39–41	13
E13	`PaperXI.FiniteBreakAssembly`	22–37	14
E14	`PaperXI.FiniteBreakAssembly`	111–115	14
E15	`PaperXI.OddPhaseCertificate`	46–50	15
E16	`PaperXI.DyadicAffine`	74–75	15
E17	`PaperXI.DyadicGauss`	51–58	15
E18	`PaperXI.DyadicRefinementCounts`	42–44	16
E19	`PaperXI.GraphGauge`	155–159	17
E20	`PaperXI.GluedPacketSections`	22–26	17
E21	`PaperXI.GluedPacketSections`	138–144	17
E22	`PaperXI.MonodromyAtom`	64–66	19
E23	`PaperXI.AtomicReconstruction`	17–21	19
E24	`PaperXI.AtomicReconstruction`	143–144	19
E25	`PaperXI.ValuationRamification`	15–28	21
E26	`PaperXI.RamificationPolyphase`	311–313	21
E27	`PaperXI.RamifiedUnramifiedQuotient`	42–43	22
E28	`PaperXI.NormalizedFrobeniusQuotient`	79–79	22
E29	`PaperXI.ActualProfileCandidate110`	34–88	23
E30	`PaperXI.RankOneSourceCandidate110`	29–60	24
E31	`PaperXI.MoritaActualCartan110`	137–152	26
E32	`PaperXI.OnePlusIntersection110`	179–189	26
E33	`PaperXI.ActualSourceCandidate111`	40–84	27
E34	`PaperXI.ScalarCocycle112`	171–174	28

ID	Module	Source lines	Page
E35	`PaperXI.EquivariantRepresentation113`	67–91	28
E36	`PaperXI.CyclicNormalization113`	167–173	29
E37	`PaperXI.LocalFactorIdentification`	14–18	30
E38	`PaperXI.LocalFactorIdentification`	25–28	30
E39	`PaperXI.LocalFactorIdentification`	49–55	30
E40	`PaperXI.ConditionalAtomicComparison`	127–143	31