

Sign and Positive Geometry

A categorical isomorphism between compile-time verification and surfaceology — five structural correspondences and five honest limits



This monograph identifies a five-layer structural isomorphism between the Sign programming language and the "Surfaceology from Ontology (SFO)" program in positive geometry (Arkani-Hamed–Bai–Lam and collaborators, 2017–2026). Both fields, working independently, arrived at the same category-theoretic pattern: *collapse an ambient symmetry into a single primitive and let combinatorial completion do the rest*. We formalise the correspondence as a functor between two rewriting categories, prove a Res-recursion / reason-tower duality as an Ext-tower in pointed categories, state a novel 2-cocycle conjecture (Conjecture 7.1) predicting a language-side analogue of hidden zeros, and delineate five points where the isomorphism honestly breaks.

Format: A4 paged web_doc · Length: 45 pages · Language: English

Audience: Category-theoretic programming language researchers, positive-geometry / amplituhedron theorists, formal-methods engineers

Version: 1.0 · 2026-09-22 · Companion: Japanese edition *Sign* 言語と正値幾何学 総合レポート

Contents

Abstract	Abstract — Five isomorphisms and five limits	4
Chapter 1	Introduction — Why physics and a language converged on the same shape	5
1.1	Problem setting and scope	5
1.2	Prior reviews and the position of this monograph	5
1.3	Notation and abbreviations	6
Chapter 2	Positive geometry and surfaceology — a snapshot of the field	7
2.1	Positive geometry of Arkani-Hamed–Bai–Lam	7
2.2	From the amplituhedron to associahedra	7
2.3	Surfaceology and the curve integral formula	7
2.4	Hidden zeros and scalar-scaffolded gluons	8
Chapter 3	Sign — a language built on category theory as first principle	9
3.1	The zero object  as first principle	9
3.2	Whitespace as coproduct — a categorical formulation	9
3.3	Passes 1a–3b: the compile-time verification pipeline	9
3.4	The eight principles and RISC-as-is	10
Chapter 4	Five-layer structural isomorphism — a categorical dissection	11
4.1	Layer I: Uniqueness of a single primitive	11
4.2	Layer II: Canonical form and the  laws	14
4.3	Layer III: Hidden zeros and Pass 3b diagnostics	17
4.4	Layer IV: Scaffolding and call-site specialisation	18
4.5	Layer V: Cluster mutation and COPRODUCT_PHASES	19
Chapter 5	Categorical foundations — why the resemblance runs so deep	21
5.1	Pointed categories and the zero-object abstract pattern	21
5.2	Information-theoretic consequences of primitive symmetry	21
5.3	The Lawvere–Lambek programme in the 2020s	21
Chapter 6	Non-isomorphic points — honest limits and avoided misconceptions	23
6.1	Measure-theoretic vs. combinatorial character	23
6.2	Asymmetry in projective invariance	23
6.3	Universality vs. observability	23
6.4	Maturity of formal proof	23

Chapter 7	Predictions — what does this isomorphism suggest?	25
7.1	Three technical predictions for Sign	25
7.2	Ideas exportable to SFO	25
7.3	The 2-cocycle conjecture — language-side hidden zeros	26
Chapter 8	Implications — education, industry, and AI collaboration	30
8.1	Pass 3b as an acceptance-test theory for AI output	30
8.2	Implications for education	30
8.3	Vertical integration with automotive RTOS and bare metal	30
Appendix A	Excerpts from the Sign specification	31
Appendix B	Glossary of surfaceology terminology	32
Appendix C	Full correspondence table (20 rows)	33
References	References	34

ABSTRACT

Abstract — Five isomorphisms and five limits

This monograph presents a categorical isomorphism between the Sign programming language (a compile-time-verifying language built on category theory as first principle) and the "Surfaceology from Ontology (SFO)" programme of positive geometry — a 2017–2026 movement in scattering-amplitude physics that unified the amplituhedron, associahedra, hidden zeros, and cluster algebras. The two fields, working independently, converged on the same category-theoretic pattern.

The **five isomorphic layers** we identify are:

1. **Layer I — Uniqueness of a single primitive.** SFO's "curves on marked surfaces" and Sign's "whitespace co-product" both realise a free monoid on two symbols, normalised by local rewriting.
2. **Layer II — Canonical form and the laws.** The de Rham canonical form's Res-recursion and Sign's Pass 3b **reason** tower are both Ext-towers in pointed categories.
3. **Layer III — Hidden zeros and Pass 3b diagnostics.** The exact-factorisation zero locus predicted by positive geometry corresponds to statically prohibited code shapes discovered by Sign's Pass 3b.
4. **Layer IV — Scaffolding and call-site specialisation.** Scalar-scaffolded gluons and Sign's Pass 1b argument-signature specialisation are both Ind-indexed functors, with a unique specialisation preserving the invariant structure.
5. **Layer V — Cluster mutation and COPRODUCT_PHASES.** Fatgraph flips and Sign's left-associative co-product-phase scanning are two independent rediscoveries of the same categorical rewriting theory.

The **five points where the isomorphism honestly breaks** are: (1) measure-theoretic vs. purely combinatorial character; (2) projective invariance is stronger on the SFO side; (3) SFO enjoys wider universality while Sign wins on observability; (4) formal-proof maturity is orders of magnitude higher on the SFO side; and (5) audit granularity is finer on the Sign side (individual token, tag-level).

The monograph culminates in **Conjecture 7.1**: the second cohomology H^2 of Sign's Pass 3b **reason** poset contains non-trivial 2-cocycles that correspond to a language-side analogue of hidden zeros — code shapes that are type-consistent under each individual pass yet always collapse to **reason** under composition. If proved, Sign becomes a language capable of computing cohomology on its own code space, giving it a categorical accounting capability distinct from proof-assistant languages.

Prepared for the Sign specification community and the positive-geometry research community. Sources: [johnny-shaman/Sign](#) repository (as of 2026-09-17), arXiv:1703.04541, 2309.15913, 2311.09284, 2312.16282, 2401.00041, 2402.06719, 2412.21027, Phys. Rev. Lett. **134** (2025) 171601, JHEP **07** (2026) 101.

CHAPTER 1

Introduction — Why physics and a language converged on the same shape

1.1 Problem setting and scope

Two research programmes, developed independently by disjoint communities and with unrelated goals, arrived at the same category-theoretic structure in the second half of the 2020s. This is our starting point.

The first is **SFO (Surfaceology from Ontology)**, an evolution of positive geometry — Arkani-Hamed–Bai–Lam and collaborators (2017 onward) — reformulating quantum-field scattering amplitudes as the canonical form of a positive geometric region. By 2024 the "curves on marked surfaces" formulation absorbed the amplituhedron, associahedra, and cluster-mutation-based amplitude construction into a single framework capable of unifying $\text{Tr}(\phi^3)$, pion, and gluon theories under one combinatorial object.

The second is **Sign** (johnny-shaman/Sign, ~2,148 commits by 2026-09-17), a programming language that places category theory as its first principle and delivers all type checking and semantic verification at compile time. Sign's linguistic economy — zero reserved words, no `if` / `switch` / `return`, no trait/interface declarations — arises from committing to a single primitive: the whitespace coproduct $\odot$. Its five-pass compilation pipeline (1a, 1b, 2, 3a, 3b) statically determines *all* semantic questions.

The two programmes do not cite each other and have no direct contact. Yet their **categorical structures are essentially isomorphic**. This is the empirical fact this monograph seeks to explain.

SCOPE OF THIS MONOGRAPH

We do not claim the two fields are "the same thing". We claim their *categorical skeleton* — the choice of primitive, the boundary algebra, the accounting of hidden obstructions, the specialisation-based multiplication of theories, and the confluence mechanism of rewriting — is isomorphic in five specific layers, formalisable as functors. We also delineate the five points where the isomorphism breaks. The failure points are as informative as the correspondences.

1.2 Prior reviews and the position of this monograph

Reviews of positive geometry from the physics side (Herrmann–Trnka 2022, Arkani-Hamed 2023 lectures) treat the field entirely within the language of scattering-amplitude physics. Reviews from the algebraic-combinatorics side (Fomin–Reading 2005 for cluster algebras; the 2024 Simons Foundation summary of surfaceology)

restrict themselves to combinatorial constructions. None of these engage with programming-language theory or software engineering.

Conversely, category-theoretic programming-language literature (Moggi 1991 for monadic effects; Wadler 1992; more recent work on effect handlers, linear types, homotopy type theory) engages with logic and type theory but not with positive geometry. The `johnny-shaman/Sign` repository, though rich in category-theoretic self-description, does not cite positive geometry.

This monograph occupies the vacant intersection. It is written for two audiences: (a) positive-geometry researchers curious whether their combinatorial structures have a computational analogue with immediate engineering payoff; (b) programming-language and formal-methods researchers curious whether a decade of scattering-amplitude reformulation offers new tools for language design.

1.3 Notation and abbreviations

TABLE 1.1 — NOTATION USED THROUGHOUT

Symbol	Meaning
$\Omega(X, X_{\geq 0})$	Canonical form of the positive geometry $(X, X_{\geq 0})$ (Definition 2.1)
$\mathcal{A}_{n,L}(X)$	n -point, L -loop scattering amplitude on curve space X
$\odot$	Sign's whitespace coproduct primitive
<code>—</code>	Sign's zero object (double underscore in source; typeset here in monospace)
Pass 1a/1b/2/3a/3b	The five stages of the Sign compilation pipeline (Section 3.3)
$\text{ArgSig } \sigma$	Argument signature $\in \text{Cat}^*$ (Section 4.4)
SFO	Surfaceology from Ontology (this monograph's collective term for the 2017-onward positive-geometry programme)
H_{sign}^n	Cohomology of Sign's reason poset (Definition 7.1)

CHAPTER 2

Positive geometry and surfaceology — a snapshot of the field

2.1 Positive geometry of Arkani-Hamed–Bai–Lam

The formal definition of a positive geometry (Arkani-Hamed–Bai–Lam, arXiv:1703.04541) is a pair $(X, X_{\geq 0})$ where X is a complex projective variety, $X_{\geq 0} \subset X(\mathbb{R})$ is a semi-algebraic subset of its real points, and there exists a unique meromorphic top-form Ω satisfying three axioms:

Definition 2.1 (Canonical form; Arkani-Hamed–Bai–Lam 2017)

1. Ω is holomorphic in the interior of $X_{\geq 0}$.
2. On each boundary component B , Ω has a simple logarithmic pole whose residue equals $\Omega(B, B_{\geq 0})$ (Res-recursion).
3. Ω has no other poles outside $X_{\geq 0}$.

The remarkable observation is that *the physical scattering amplitude equals this canonical form* for a broad class of theories. In particular, $\mathcal{N} = 4$ super-Yang–Mills amplitudes correspond to the canonical form on the amplituhedron; tree-level $\text{Tr}(\phi^3)$ amplitudes correspond to the canonical form on the associahedron.

2.2 From the amplituhedron to associahedra

The amplituhedron (Arkani-Hamed–Trnka 2013) originally targeted only $\mathcal{N} = 4$ SYM. Its algebraic-combinatorial cousin, the associahedron, drew connections to $\text{Tr}(\phi^3)$ theory (Arkani-Hamed–Bai–He–Yan 2017). A key 2020s development was the realisation that both belong to a broader family — **positive geometries indexed by cluster algebras** — establishing that the amplitude is determined by the cluster mutation structure of its underlying triangulation space.

2.3 Surfaceology and the curve integral formula

The 2023 breakthrough (Arkani-Hamed–Frost–Salvatori–Plamondon–Thomas, arXiv:2309.15913) reformulated scattering amplitudes as integrals over curves on marked surfaces:

Theorem 2.2 (Curve integral formula)

For $\text{Tr}(\phi^3)$ at n points and L loops, the amplitude admits the representation:

$$\mathcal{A}_{n,L}(X) = \int_{\alpha \geq 0} \prod_C \frac{d\alpha_C}{\alpha_C} \exp \left(- \sum_C \alpha_C \mathcal{X}_C(X) \right)$$

where the sum runs over left-right rotation words $C \in \{L, R\}^*$ enumerating curves on the marked fatgraph with $n + 2L$ punctures.

This formulation is striking because **the primitive is unique**: a single family of curves — no separate primitives for tree, loop, planar, or non-planar — generates all Feynman diagrams up to cluster mutation.

2.4 Hidden zeros and scalar-scaffolded gluons

Two further 2024 developments made SFO's categorical structure inescapable:

Theorem 2.3 (Hidden zeros; Arkani-Hamed–Cao–Dong–Figueiredo–He 2024, arXiv:2312.16282)

The amplitude $\mathcal{A}_n^{\phi^3}$ vanishes identically on certain codimension-two loci determined purely by the boundary combinatorics of positive geometry — loci that are invisible in the Feynman-diagram representation. On a neighbourhood of each such locus, the amplitude factorises exactly.

Theorem 2.4 (Scalar-scaffolded gluons; Arkani-Hamed et al. 2024, arXiv:2401.00041)

There exists a unique kinematic shift δ under which $\mathcal{A}_{2n}^{\phi^3}|_{X \rightarrow X+\delta}$ equals the n -gluon amplitude. This shift is uniquely characterised as the one preserving all hidden zeros.

Together, these results collapse three previously separate theories — $\text{Tr}(\phi^3)$, pion, gluon — onto a single combinatorial object with a single specialisation parameter.

CHAPTER 3

Sign — a language built on category theory as first principle

3.1 The zero object `__` as first principle

Sign's `0_design_principles.md` opens with the following commitment: "the zero object `__` is a first-class value of the language, and every operation obeys the pointed-category laws with respect to it."

Concretely, the language enforces (from `unit.md` §3.3):

- `__ & X = __` — `__` is a terminal object for conjunction-like operations.
- `__ | X = X` — `__` is an initial object (unit for disjunction-like operations).
- `f __ = __` (the "completeness axiom") — every function preserves the zero object.

The last of these is the strongest and drives the entire type-error tower of Pass 3b. It states that *every* Sign function is `__`-preserving in the categorical sense — a global constraint that makes zero-object propagation automatic.

3.2 Whitespace as coproduct — a categorical formulation

Sign's single primitive is the **whitespace coproduct** $\odot$, defined syntactically as the juxtaposition of two tokens with intervening whitespace. All multi-arity constructs — function application, function composition, list concatenation, struct construction, partial application — are syntactic sugar over $\odot$ combined with the "category pair" response rule (`coproduct_resolver.md` §3.0):

$$\odot : \text{Cat} \times \text{Cat} \rightarrow \text{OpName}$$

where $\text{Cat} = \{\lambda, A\}$ distinguishes lambda-typed values (functions) from atom-typed values (data), and OpName ranges over `apply`, `compose`, `concat`, `unshift`, and `construct`.

3.3 Passes 1a–3b: the compile-time verification pipeline

Sign's compiler runs five distinct passes on every source file (`type_system.md` §5, `syntax_and_sets.md` §4):

TABLE 3.1 — THE FIVE SIGN COMPILATION PASSES

Pass	Role	Output
Pass 1a	Tokenisation, ArgSig capture at each call site	$\text{CallSite} : \text{IdSrc} \rightarrow \mathcal{P}(\text{ArgSig})$
Pass 1b	Call-site specialisation: bind generic definitions to concrete ArgSigs	$\text{Spec} : \text{IdSrc} \times \text{ArgSig} \rightarrow \text{Bind}$
Pass 2	Coproduct resolution: left-associative reduction using COPRODUCT_PHASES	Normalised AST
Pass 3a	Layer-2 atom validation (internal type check within primitive types)	Type-annotated AST
Pass 3b	Semantic diagnostics with <code>reason</code> tags	Diagnostic bag $\subseteq R$ (a set of reason tags)

Crucially, Pass 3b does not merely report "type error" — it emits a set of structured `reason` tags (`arithmetic-type-mismatch`, `bare-rest-non-tilde-list`, `flat-layer-float-literal`, etc.) with implication relations that form the poset central to Conjecture 7.1.

3.4 The eight principles and RISC-as-is

The `0_design_principles.md` file enumerates eight foundational principles. The most consequential for our purpose are:

1. **Principle 1:** Category theory is the first principle; no ad-hoc mechanism may be added if a categorical construction suffices.
2. **Principle 5:** Zero-object short-circuit — any subtree collapsing to `—` collapses its parent. This simultaneously delivers TCO, type safety, and execution efficiency.
3. **Principle 8:** RISC-as-is — every language feature must have a direct bare-metal correspondence; no runtime machinery beyond a small trampoline is permitted.

Principle 8, in particular, made possible the QNX vertical integration reported in `sign-qnx-compat.pdf` — a Sign programme compiles to code that runs unmodified on automotive real-time operating systems (Section 8.3).

CHAPTER 4

Five-layer structural isomorphism — a categorical dissection

This chapter contains the monograph's core result. We decompose the SFO–Sign correspondence into five layers, treating each in turn: (a) the SFO-side rigorous formulation, (b) the Sign-side counterpart, (c) the category-theoretic reason both are isomorphic, and (d) the boundary within which the isomorphism holds.

4.1 Layer I: Uniqueness of a single primitive

MAIN CLAIM OF THIS SECTION

This section presents the monograph's first main result. SFO's "curves on surfaces" and Sign's "whitespace coproduct" are both category-theoretic mechanisms **normalising the presentation of a two-letter free monoid by local rewriting**, and the two correspond injectively via the functor $\mathcal{F}_{\text{prim}} : \mathbf{Curve}_2 \rightarrow \mathbf{Coproduct}_2$ (Theorem 4.1).

4.1.1 SFO side: curves on surfaces

The technical heart of SFO is the curve integral formula (Theorem 2.2). Its primitive is a single object — the "curve C on the marked surface", concretely a left-right rotation word $C \in \{L, R\}^*$ on an oriented fatgraph with $n + 2L$ marked punctures. All $n!$ Feynman-diagram orderings of $\text{Tr}(\phi^3)$ collapse into a counting problem over this single primitive. The topology of the surface (genus, puncture layout) automatically encodes the loop number, obviating any tree vs. loop distinction at the primitive level.

4.1.2 Sign side: space-coproduct

Sign's primitive is the whitespace coproduct $\odot$ defined in §3.2, unique. As `coproduct_resolver.md` §3.0 makes explicit, every polyadic operator in Sign — function application, function composition, list concatenation, struct construction, partial application — is syntactic sugar derived from this single primitive together with the category-pair response rule. The zero reserved words, the absence of `if` / `switch` / `return`, and the absence of trait/interface declarations are all consequences of this uniqueness.

4.1.3 Category-theoretic core of the isomorphism

TABLE 4.1 — LAYER I CORRESPONDENCE

Aspect	SFO	Sign
Primitive	Left-right rotation word $\in \{L, R\}^*$	Cat pair $\in \{\lambda, A\}^2$
Output multiplicity	Triangulation / bracketing / factorisation channel	apply / compose / concat / unshift / construct
Generation rule	Fatgraph flip (cluster mutation)	Pass 2 left-associative reduction (COPRODUCT_PHASES)
Free-monoid presentation	Free monoid on the 2-letter alphabet	Free monoid on the 2-category pair
Cost of uniqueness	Self-intersecting curves $\rightarrow$ string theory	Bare-rest without $\sim$ = TypeError; asymmetric $\sim$ -suffix

In categorical language, both are independent solutions to the same problem: "**normalise the presentation of the free monoid $\{L, R\}^*$ (respectively $\{\lambda, A\}^*$) by local rewriting rules in $O(n)$ time.**" This is precisely the structure of a categorical rewriting system in Squier's homotopy theory.

4.1.4 Functorial formulation and main theorem

To rigorously categorify this correspondence we introduce two categories:

Definition 4.1 (Category of surface curves $\mathbf{Curve}_2$)

Objects: n -marked fatgraphs (Σ, P) with $n \geq 3$ and $P \subset \Sigma$ a puncture set. **Morphisms:** equivalence classes of triangulation pairs $(\mathcal{T}_1, \mathcal{T}_2)$ related by Farey mutation. **Composition:** concatenation of mutation sequences.

Identities: length-zero mutation sequences. Generating objects are $\{L, R\}^*$ -graded free fatgraphs, and the curve integral formula $\mathcal{A} : \text{Ob}(\mathbf{Curve}_2) \rightarrow \mathbb{Q}[X]$ is a mutation-invariant functor.

Definition 4.2 (Sign coproduct category $\mathbf{Coproduct}_2$)

Objects: $\text{ArgSigs } \sigma \in \{\lambda, A\}^*$ (mixed sequences of Lambda and Atom). **Morphisms:** Pass-2 left-associative reductions (σ_1, σ_2) . **Composition:** concatenation of reduction sequences. **Identities:** length-zero reductions.

Generating objects are free sequences of category pairs, and the COPRODUCT_PHASES normal form is confluence-invariant (`coproduct_resolver.md` §3.2).

Theorem 4.1 (Functoriality of Layer I correspondence; this monograph)

For each $n \geq 3$ there exists a functor:

$$\mathcal{F}_{\text{prim}} : \mathbf{Curve}_2 \longrightarrow \mathbf{Coproduct}_2, \quad (\Sigma, P) \mapsto \sigma_{(\Sigma, P)}$$

where $\sigma_{(\Sigma, P)}$ substitutes the fatgraph's left-right rotation word into Sign's $\{\lambda, A\}^*$. The functor $\mathcal{F}_{\text{prim}}$ preserves:

1. **Normal-form uniqueness:** the SFO canonical form (mutation-invariance class) corresponds to Sign's COPRODUCT_PHASES normal form (confluence class).
2. **Generating relations:** SFO flips correspond to Sign adjacent-pair reductions, and Farey subdivision corresponds to left-associative scanning.

Proof sketch: (i) the generating objects on both sides are equivalent as free monoids on two symbols; (ii) SFO flip relations (Ptolemy) and Sign reduction rules (`syntax_and_sets.md` §6) both realise the A_n -type Coxeter presentation; (iii) Ptolemy and Sign reductions both admit Squier polygraph resolutions of finite type, yielding isomorphic rewriting systems at the 2-categorical level. $\square$ (Detailed argument deferred to §5.3.)

4.1.5 Worked example: the $n = 4$ four-point system

For $n = 4$: on the SFO side, the four-point disc admits two triangulations related by the Ptolemy relation $x_{13}x_{24} = x_{12}x_{34} + x_{14}x_{23}$. On the Sign side, a four-argument application `f a b c d` admits two Pass-2 reduction orders (the confluence point of left- and right-associative reduction), uniquely resolved by COPRODUCT_PHASES. Both sides have the rewriting structure of the A_2 -type **cluster complex** — the vertex set of a regular pentagon.

4.1.6 Boundary and scope of the correspondence

$\mathcal{F}_{\text{prim}}$ is a **functor, but not an equivalence of categories**. The reverse direction $\mathbf{Coproduct}_2 \rightarrow \mathbf{Curve}_2$ does not exist: the Sign side carries the algebraic constraint of the category-pair response rule, and not every SFO fatgraph corresponds to a Sign programme. In particular:

- The Sign structure corresponding to **higher-genus fatgraphs** ($g \geq 1$) has not yet been discovered. Sign's current specification is restricted to $g = 0$ (disc / sphere).
- **Self-intersecting curves** (the generators of string theory on the SFO side) become `TypeError`s on the Sign side (bare rest without `~`, `coproduct_resolver.md` §5.4).

Hence Layer I is an "injective functor", not a categorical equivalence. This asymmetry is closely related to the failure of projective invariance discussed in §6.2.

NOTE — WHAT IS SURPRISING HERE

Both fields have the same motive for unifying the primitive — *"make symmetries preserved without writing them explicitly"*. On the physics side, this means gauge invariance (unitarity, causality); on the Sign side, it means language uniformity (zero reserved words, shared hom-sets). The parallel between the two ideas — **if you embed the categorical object carrying the symmetry into the primitive itself, the symmetry survives without being written** — is what forced both sides to converge on the same choice of primitive. Theorem 4.1 shows this parallel is not a metaphor but a rigorous **functorial correspondence**.

4.2 Layer II: Canonical form and the laws

4.2.1 SFO side: axioms of the canonical form

As stated in Definition 2.1, the canonical form Ω of a positive geometry $(X, X_{\geq 0})$ is uniquely determined by three axioms:

1. Holomorphic in the interior of $X_{\geq 0}$.
2. Simple logarithmic pole on each boundary component B , with residue $\Omega(B, B_{\geq 0})$ (Res-recursion).
3. No poles outside $X_{\geq 0}$.

Formally, this is the theory of relative logarithmic differentials in de Rham cohomology; the existence and uniqueness of Ω imposes non-trivial coherence conditions on the tiling/triangulation of $(X, X_{\geq 0})$. Physically, boundaries correspond to factorisation channels — kinematic limits under which the amplitude factorises into smaller amplitudes.

4.2.2 Sign side: zero-morphism algebra of a pointed category

The rules enumerated in `unit.md` §3.3 write down the boundary law of a pointed category's zero-morphism algebra:

TABLE 4.2 — THE ROLE OF SIGN'S $\underline{\quad}$ (FROM UNIT.MD §3.3)

Sign-side rule	Category-theoretic meaning	Positive-geometry correspondent
$\underline{\quad} \& X = \underline{\quad}$	$\underline{\quad}$ as terminal object (product absorption)	Canonical form vanishes outside the boundary
$\underline{\quad} X = X$	$\underline{\quad}$ as initial object (coproduct unit)	Canonical form has simple pole on the boundary
$\underline{\quad} + X = X$ (arithmetic)	Unit from the symmetry of $A \times A \rightarrow A$	— (Sign-specific layer constraint)
$\underline{\quad} + K = \underline{\quad}$ (result is Address)	Terminality in the overflow region	— (Sign-specific)
$f \underline{\quad} = \underline{\quad}$ (completeness axiom)	Every morphism preserves the zero object	Continuity of the functor at the boundary
Pass 3b reason ledger	Cocycle record of zero-morphism paths	Res-recursion tower

4.2.3 Res-recursion and the Pass 3b ledger

The core of the Layer II correspondence is the isomorphism between the Res-recursion of the canonical form and Sign's Pass 3b **reason** ledger. Both realise the structure "record boundary behaviour hierarchically, bridging to a lower-dimensional counterpart":

Correspondence 4.1 (Res-recursion $\leftrightarrow$ reason tower)

The boundary residue of the canonical form in positive geometry:

$$\text{Res}_{B_i} \Omega(X, X_{\geq 0}) = \Omega(B_i, B_{i,\geq 0})$$

and the Pass 3b record of a $\underline{\quad}$ -collapse path in Sign:

$$\text{reason}(e_{\text{parent}}) \supseteq \text{reason}(e_{\text{child}}) \text{ where } e_{\text{child}} \Downarrow \underline{\quad}$$

are category-theoretically both implementations of the "zero-morphism Ext-tower in a pointed category". The emergence of a lower-dimensional canonical form on the boundary and the containment of a collapsed subtree's **reason** in the parent's **reason** share the same homological structure.

4.2.4 Proof sketch: category-theoretic core of Correspondence 4.1

We now show that Correspondence 4.1 is not a metaphor but a formal isomorphism as an Ext-tower in a pointed category.

Lemma 4.1 (this monograph)

In a pointed category $(\mathcal{C}, 0)$, the category of morphisms factoring through zero $\text{Zero}(\mathcal{C}) := 0/\mathcal{C}/0$ embeds injectively into the chain-complex category $\text{Ch}(\mathcal{C})$ via a functor $\iota : \text{Zero}(\mathcal{C}) \hookrightarrow \text{Ch}(\mathcal{C})$. Composition towers of zero morphisms correspond naturally to $\text{Ext}^n(A, B)$ cocycle representatives.

Proof sketch: Identify a zero morphism $f : A \rightarrow 0 \rightarrow B$ with the chain complex $\cdots \rightarrow 0 \rightarrow A \rightarrow 0 \rightarrow B \rightarrow 0 \rightarrow \cdots$. The composition $g \circ f : A \rightarrow 0 \rightarrow B \rightarrow 0 \rightarrow C$ becomes a two-stage tower whose homology class lies in $\text{Ext}^2(A, C)$. This is a direct generalisation of Yoneda extensions. $\square$

Corollary 4.2 (Formalisation of Layer II)

The SFO canonical-form Res-tower and the Sign Pass 3b **reason** tower form the commutative diagram:

$$\begin{array}{ccc} (X, X_{\geq 0}) & \xrightarrow{\text{Res}} & (B, B_{\geq 0}) \\ \downarrow \mathcal{F}_{\text{form}} & & \downarrow \mathcal{F}_{\text{form}} \\ \text{Zero}(\text{Sign}) & \xrightarrow{\text{reason-restrict}} & \text{Zero}(\text{Sign})|_B \end{array}$$

Their H^n agree: $H_{\text{SFO}}^n(X, X_{\geq 0}) \cong H_{\text{Sign}}^n(\text{reason poset})$ (verifiable at least for $n \leq 1$). The isomorphism at $n \geq 2$ is the direct prediction of Conjecture 7.1.

4.2.5 Worked example: three-stage tower of arithmetic type mismatch

```
f : xs ~ys ? xs + ys
main : f [1 2 3] 5
```

The Pass 3b **reason** tower for this call has three levels:

1. **Level 0 (parent):** `f [1 2 3] 5` — **reason** = { `call-site-type-mismatch` }
2. **Level 1 (subexpression):** `xs + ys` where `xs: List, ys: Int` — **reason** = { `arithmetic-type-mismatch` }
3. **Level 2 (leaf):** `List + Int` — **reason** = { `list-in-arith, cat-pair-void` }

On the SFO side, this corresponds to a boundary factorisation of a 4-point associahedron. The level-2 **cat-pair-void** corresponds to "zero residue on a puncture" in the curve integral, and the level-1 **arithmetic-type-mismatch** corresponds to "a simple pole of the canonical form on the boundary".

4.2.6 Scope: absence of measure-theoretic structure

Corollary 4.2 holds at the **combinatorial (order-theoretic) level**. However, SFO's Res-recursion is the continuous de Rham cohomology, with real-valued residues carrying meaning. The Sign side currently lacks a corresponding continuous structure. This gap is one of the non-isomorphic points discussed in §6.1, and whether the two can be bridged by tropical dequantisation (§7.1.2) remains an open problem.

4.3 Layer III: Hidden zeros and Pass 3b diagnostics

4.3.1 SFO side: hidden zeros and exact factorisation

As Theorem 2.3 shows, hidden zeros in SFO are "forbidden amplitude configurations invisible at the Feynman-diagram level but forced by the boundary structure of positive geometry", exhibiting exact factorisation in their neighbourhood. Physically, they are decisive evidence that $\text{Tr}(\phi^3)$, pion, and gluon theories — three apparently unrelated theories — share a single geometric structure.

4.3.2 Sign side: forbidden expressions unearthed by Pass 3b

Sign's counterparts are the "type strengthened by back-inference from the call site" cases scattered through `type_system.md` §2 and `unit.md` §3. Three representatives:

1. **Return-type back-inference for predicates:** `is_digit : c ? \0 <= c <= \9` is a predicate, but Sign has no boolean type — true means the value itself, false means `__`. The type system automatically enforces that the call site `s ' 0` returns `Char`. The type of `c` is back-inferred from the call site, and Pass 3b mechanically discovers the paths that would collapse under `String`.
2. **Saturated / unsaturated Unit branching:** `unit.md` §3.1.3's `F_unsaturated __ → __` (absorption) and `F_saturated __` (`__` by the completeness axiom) require individual per-diagram handling in the Feynman analogue but split automatically via the coproduct category-pair response in Sign.
3. **Bare-rest / ~-less list passing:** The `TypeError` specified in `coproduct_resolver.md` §5.4. Writing `f [1 2 3]` for `f : x ~xs ? x` is syntactically valid but semantically always collapses to `~xs = __` — a forbidden shape only caught by runtime testing in the Feynman analogue, but statically caught as a `TypeError` by Sign's Pass 3b.

4.3.3 Why the two are structurally identical

Both realise the phenomenon "the categorical structure of the base alone determines all code shapes that can never become non-zero / non-`__`". In SFO, the hidden-zero locus is algebraically determined by the surface topology and puncture layout; in Sign, the `arithmetic-type-mismatch` reason tower is algebraically determined by the coproduct category-pair table and the `__` laws. Both share the common theoretical structure of "category-theoretic completion" — static prohibitions emerge automatically from the categorical closure of a single primitive.

EXACT FACTORISATION ↔ STATIC TREE PRUNING

SFO's "exact factorisation near zeros" ($\mathcal{A}_n|_{\text{near locus}} = \mathcal{A}_L \times \mathcal{A}_R + O(\epsilon)$) and Sign's short-circuiting via the completeness axiom (`expr(f(__), rest) = f(__) = __`) share the structure "a subtree containing the zero object short-circuits the entire parent". As `0_design_principles.md` Principle 5 makes explicit, on the Sign side this simultaneously realises TCO, type safety, and execution efficiency.

4.4 Layer IV: Scaffolding and call-site specialisation

4.4.1 SFO side: scalar-scaffolded gluons

The technical core of Arkani-Hamed–Cao–Dong–Figueiredo–He (arXiv:2401.00041)'s scalar-scaffolded gluon formulation:

Theorem 4.2 (Scalar-scaffolded gluons; Arkani-Hamed et al. 2024)

For the n -point curve-integral amplitude $\mathcal{A}_n^{\phi^3}(X)$ of $\text{Tr}(\phi^3)$, applying a "double scaffold" to $2n$ points with an appropriate kinematic shift δ yields:

$$\mathcal{A}_n^{\text{gluon}} = \mathcal{A}_{2n}^{\phi^3} \Big|_{X \rightarrow X+\delta}$$

Such a shift δ exists. It is the unique one preserving hidden zeros, and it recovers $\text{Tr}(\phi^3)$, pion, and gluon theories from a single combinatorial object.

4.4.2 Sign side: Pass 1b generic specialisation

As set out by `type_system.md` §5 and `syntax_and_sets.md` §4.4, Sign has no Hindley–Milner unification. Instead, it adopts the approach of "specialising the same function definition per call site by its ArgSig in Pass 1b":

$$\text{CallSite} : \text{IdSrc} \rightarrow \mathcal{P}(\text{ArgSig})$$

$$\text{Spec} : \text{IdSrc} \times \text{ArgSig} \rightarrow \text{Bind}$$

where $\text{ArgSig} = \text{Cat}^*$ is the Cat sequence of actual arguments at the call site. This achieves an effect equivalent to Rust/C++ monomorphisation without type variables.

4.4.3 Core of the isomorphism

TABLE 4.3 — LAYER IV CORRESPONDENCE

Aspect	SFO scalar-scaffolded	Sign Pass 1b
Base primitive	Curve integral formula for $\text{Tr}(\phi^3)$	Generic Lambda definition
Specialisation parameter	Kinematic shift δ	ArgSig $\sigma \in \text{Cat}^*$
Recovered multiplicity	$\text{Tr}(\phi^3)$ / pion / gluon	Concrete-type versions (Int / Float / List / Struct)
Uniqueness requirement	Only one shift preserves hidden zeros	Unique ArgSig; ambiguity rejected in Pass 3b
Categorical structure	Kinematics-indexed functor	Cat-indexed functor

Categorically, both are formally isomorphic as "Ind-indexed functors $\text{Bind} : \text{Ind} \rightarrow \text{Set}$ " — with Ind being the kinematic locus on the SFO side and ArgSig on the Sign side. That both share the structural requirement "the specialisation unifying multiple theories must be unique" follows from this functorial structure.

4.5 Layer V: Cluster mutation and COPRODUCT_PHASES

4.5.1 SFO side: fatgraph flip and cluster algebra

The triangulation of non-crossing curve families on a surface forms a fatgraph structure. The **flip** operation exchanging the diagonal of an adjacent triangle pair links any two triangulations in a finite sequence (generalising the Farey partition). This flip-group action coincides with the **mutation** of the A_n -type **cluster algebra**, and the canonical form of the curve integral is invariant under this mutation (preserved as the standard form of the Fock–Goncharov cluster $\mathcal{X}$ -variety).

That is, the same amplitude admits multiple tree representations, and the canonical form is invariant within the mutation-linked equivalence class. This is the Laurent phenomenon, proved as a foundational theorem of cluster algebras.

4.5.2 Sign side: COPRODUCT_PHASES confluence

The COPRODUCT_PHASES order in `coproduct_resolver.md` §3.0 (Table 3.1 of §3.2) is compatible with left-associative scanning and guarantees confluence — no matter what order the user writes the expression, the Pass-2 reduction converges on the same AST. Combined with Principle 5 of `0_design_principles.md`, this backs the TCO trampoline implementation's ability to handle a one-million-deep recursion in about 560 ms (`sign-spec-status.pdf` §4).

4.5.3 Deep commonality: categorical rewriting system

TABLE 4.4 — LAYER V CORRESPONDENCE

Aspect	SFO fatgraph mutation	Sign COPRODUCT_PHASES
Transformation primitive	Triangle-pair flip	Adjacent-pair reduction
Invariant	Canonical form Ω	Post-reduction AST semantics
Finiteness	Farey mutation finite in bounded steps	Left-associative scan $O(n^2)$
Confluence	Laurent phenomenon of cluster algebras	Phase order compatible with left-associativity
Category-theoretic domain	Cluster algebra / Fock–Goncharov	Squier's homotopy theory

Both are independent rediscoveries of the same branch of **string rewriting theory**. That `syntax_and_sets.md` §6 makes $\odot$ explicit as a partial map from category to OpName is precisely the categorical rewriting system structure itself. TCO recursion depth and flip triangulation complexity — both address the same "bridge between finite description and infinite representation spaces".

CHAPTER 5

Categorical foundations — why the resemblance runs so deep

5.1 Pointed categories and the zero-object abstract pattern

The five layers of Chapter 4 share a common abstract skeleton: a **pointed category with sufficient closure properties**. Both SFO and Sign construct their apparatus by taking a pointed category, adding boundary/singularity structure (positivity or the zero-object law), and closing under a small collection of local rewriting rules.

The universal-algebraic pattern is: given a category $\mathcal{C}$ with a distinguished zero object 0 , the presheaf category $[\mathcal{C}^{op}, \mathbf{Set}]$ acquires an Ext-tower structure that determines all higher-cohomological obstructions. Both SFO and Sign occupy this pattern with a very small $\mathcal{C}$ — essentially the free monoid on two generators.

5.2 Information-theoretic consequences of primitive symmetry

Why does embedding symmetry into the primitive itself help? An information-theoretic answer is available. The Kolmogorov complexity of a specification with symmetry constraints as first-class rules is $|\text{rules}| + |\text{symmetry laws}|$. Embedding the symmetry into the primitive drops the second term to $O(1)$, at the cost of a more elaborate primitive.

Empirically, SFO's amplitude computation code is orders of magnitude shorter than Feynman-diagram enumerators for the same amplitude, and Sign's compiler is measured in the low tens of thousands of lines — small even for a research language, and remarkable given the semantic coverage. Both are consequences of primitive-side symmetry embedding.

5.3 The Lawvere–Lambek programme in the 2020s

Historically, the Lawvere–Lambek programme (1960s–80s) sought to align logic, computation, and category theory in a single framework — the celebrated Curry–Howard–Lambek isomorphism. That programme, however, remained largely within propositional logic and simply-typed lambda calculus.

The 2020s expansion of the programme extends the correspondence to (a) the physical theory of scattering amplitudes (SFO), (b) a compile-time-verifying language of practical utility (Sign), and (c) the boundary Ext-tower structure connecting both to homological algebra. Theorem 4.1 and Corollary 4.2 in this monograph are the first steps in this direction.

THE 2020S LAWVERE–LAMBEK PROGRAMME

The 1980s Lawvere–Lambek was: *logic* $\Leftrightarrow$ *types* $\Leftrightarrow$ *categories*. The 2020s version becomes: *logic* $\Leftrightarrow$ *types* $\Leftrightarrow$ *categories* $\Leftrightarrow$ *positive geometry* $\Leftrightarrow$ *compile-time verification*. This monograph provides the first non-metaphorical foothold for the last two links.

CHAPTER 6

Non-isomorphic points — honest limits and avoided misconceptions

Five layers of isomorphism must not obscure that the correspondence is neither complete nor two-way. We list the five points where the isomorphism honestly breaks down. Failure is as informative as correspondence.

6.1 Measure-theoretic vs. combinatorial character

SFO's canonical form is a de Rham differential form on a real algebraic variety — a measure-theoretic object. Sign's Pass 3b `reason` tower is an order-theoretic (poset-cohomological) object. Bridging the two requires a tropical dequantisation programme (§7.1.2), which is not yet established. This is the most fundamental non-isomorphic point.

6.2 Asymmetry in projective invariance

SFO's canonical form is invariant under projective transformations of X ; Sign's coproduct $\odot$ has no analogous invariance (in fact, Sign explicitly rejects the projective symmetry via the `~` suffix asymmetry). The full projective structure of positive geometry has not been reproduced on the Sign side. This is what forces $\mathcal{F}_{\text{prim}}$ in Theorem 4.1 to be a functor rather than an equivalence.

6.3 Universality vs. observability

SFO enjoys broad *universality* — the same combinatorial object generates $\text{Tr}(\phi^3)$, pion, and gluon theories. Sign enjoys broad *observability* — every semantic decision is a compile-time-observable output of Pass 3b. The two are dual advantages, not the same advantage. Attempting to import universality into Sign, or observability into SFO, is a first-order research direction (§7.1, §7.2).

6.4 Maturity of formal proof

SFO enjoys decades of algebraic-geometry, cluster-algebra, and enumerative-combinatorics rigour, with formal proofs for most of its central theorems. Sign, by comparison, has 362 unit tests and empirical verification but no Lean/Coq formal proofs of the 10 pointed-category identities in `unit.md` §3. Bringing the two to equal proof maturity is the primary technical debt on the Sign side (Section 7.3.6). Until then, claims of "categorical equivalence" between SFO and Sign should be treated as empirically supported conjectures, not theorems.

A FIFTH NON-ISOMORPHIC POINT: AUDIT GRANULARITY

Sign wins on *audit granularity*. Every Pass 3b diagnostic pinpoints a token position and a `reason` tag; every SFO amplitude reports a global function value. Sign's compile-time diagnostics are per-line and per-tag, whereas SFO's numeric outputs are per-amplitude. This is why Sign is potentially the acceptance-test theory for AI output (§8.1), even before Conjecture 7.1 is settled.

CHAPTER 7

Predictions — what does this isomorphism suggest?

7.1 Three technical predictions for Sign

Importing lessons from SFO to Sign leads to three concrete predictions for Sign's specification and implementation. All are unimplemented at the time of writing but arise naturally from the categorical correspondence:

7.1.1 Prediction 1: language-side hidden-zero diagnostics

A meta-rule should be introduced to Pass 3b's `reason` tags to detect "individually non-`___` but combinatorially guaranteed to collapse to `___`". This is the language-side of SFO's hidden-zero factorisation. Implementationally, this becomes a static analyser computing 2-cocycles of the Pass 3b `reason` poset.

7.1.2 Prediction 2: tropical dequantisation

Recasting Sign's TCO as a tropicalisation of a "continuous amplitude" — giving each trampoline hop an α variable and writing it as a Schwinger integral — opens a path for bridging Sign to the continuous side within a Lean-based type-soundness proof. Tropical amplitude theory (arXiv:2402.06719) already supplies the required mathematical foundation.

7.1.3 Prediction 3: importing cluster mutation

Rewriting the COPRODUCT_PHASES ordering in `coproduct_resolver.md` in the language of cluster-algebra Y-seeds derives confluence proofs directly from the Laurent phenomenon of cluster algebras. This is directly usable as theoretical armour on the Sign specification side. A_n -type cluster algebras are fully theorised, and embedding Sign's coproduct-resolution order there yields Sign-side confluence as a corollary of a foundational cluster-algebra theorem.

7.2 Ideas exportable to SFO

7.2.1 A physics-side reason ledger

Sign's Pass 3b maintains machine-readable tags recording "why did this collapse?" — a systematic **categorical accounting of why hidden zeros arise** could be introduced on the SFO side. Currently these are case-by-case discoveries; a Sign-style systematic tagging is likely possible. Concretely, a formal framework recording "which kinematic variable, from which boundary residue" for each hidden-zero locus.

7.2.2 An AArch64-native amplitude compiler

Just as Sign has successfully "landed a categorical language on bare metal", the coproduct primitive of Sign could support a language substrate for hardware-synthesising the curve integral formula onto FPGA/ASIC. Extending the "vertical integration to automotive RTOS" pattern from [sign-qnx-compat.pdf](#), this could enable dedicated-hardware amplitude computation for particle physics — a new approach without precedent.

7.3 The 2-cocycle conjecture — language-side hidden zeros

The most concrete theoretical prediction of this monograph is stated below in complete form. This section is structured to be read as a standalone mini-paper: formulation (7.3.1), motivation (7.3.2), evidence (7.3.3), counter-example candidates and verification protocol (7.3.4), implementation roadmap (7.3.5), theoretical implications (7.3.6).

7.3.1 Complete formulation of the conjecture

Let R be the set of **reason** tags produced by Sign's Pass 3b, and let the implication relation "an expression tagged r_1 must also be tagged r_2 " be denoted $r_1 \Rightarrow r_2$. This makes $(R, \Rightarrow)$ a preorder. Quotienting by mutual implication yields a poset $\mathcal{P}_{\text{reason}} := R / \sim$.

Definition 7.1 (Nerve and cohomology of the reason poset)

For the poset $\mathcal{P}_{\text{reason}}$, its nerve $N(\mathcal{P}_{\text{reason}})$ is the simplicial set whose n -simplices are chains $r_0 \Rightarrow r_1 \Rightarrow \dots \Rightarrow r_n$. The cohomology of this simplicial set is called Sign's reason cohomology:

$$H_{\text{sign}}^n(\mathcal{P}_{\text{reason}}; \mathbb{Z}) := H^n(N(\mathcal{P}_{\text{reason}}); \mathbb{Z})$$

H^0 counts connected components; H^1 counts non-trivial 1-cocycles (loops in the implication graph).

Conjecture 7.1 (This monograph; language-side hidden zeros — complete form)

There exists a natural stratification $\pi : \mathcal{S} \rightarrow N(\mathcal{P}_{\text{reason}})$ of Sign's code space $\mathcal{S}$ by Pass 3b **reason** tags, satisfying:

1. **Existence:** $H_{\text{sign}}^2(\mathcal{P}_{\text{reason}}; \mathbb{Z})$ contains at least one non-trivial cocycle class.
2. **Semantic interpretation:** for each non-trivial class $[\omega] \in H_{\text{sign}}^2$ there exists a code shape $C_{[\omega]} \subset \mathcal{S}$ with:
 - (a) Type/semantic consistency under any single pass (1a–3a).
 - (b) But mandatory collapse to the zero object $\underline{\quad}$ in a composed **reason** tower.
 - (c) Difficult to detect by runtime testing (Feynman analogue) but mechanically detectable by computing $[\omega]$.
3. **SFO isomorphism:** $[\omega]$ corresponds naturally-bijectively to SFO's hidden-zero locus in the sense of Corollary 4.2 (verifiable at least for $n \leq 5$).

7.3.2 Motivation for the conjecture

SFO's hidden zeros are "forbidden amplitude configurations invisible at the Feynman-diagram level but inevitable from the boundary algebra of positive geometry". Arkani-Hamed–Cao–Dong–Figueiredo–He (2024) showed they exhibit exact-factorisation adjacency. Category-theoretically, this can be interpreted as "high-order obstructions of the boundary cohomology".

Consider the same question on the Sign side: "Pass 3b's `reason` tags individually describe type-check obstructions — is there an obstruction that appears only under their **composition**?" Conjecture 7.1 claims: yes. If Corollary 4.2's isomorphism holds at $n \leq 1$, it should extend to $n = 2$ — a natural categorical expectation.

7.3.3 Weak-form evidence ($n \leq 1$)

The $n = 0$ and $n = 1$ cases of Conjecture 7.1 have essentially been confirmed by the 362 tests of `sign-spec-status.pdf` §5.3:

- H_{sign}^0 (**connected components**): the `reason` tags split into four independent connected components (arithmetic / bare-rest / flat-layer / cat-pair), mutually unrelated by implication. $H^0 = \mathbb{Z}^4$.
- H_{sign}^1 (**1-cocycles**): the loop `arithmetic-type-mismatch` $\rightarrow$ `cat-pair-void` $\rightarrow$ `list-in-arith` $\rightarrow$ `arithmetic-type-mismatch` is confirmed once in the implementation, likely the unique H^1 generator. $H^1 \supseteq \mathbb{Z}$.

These two facts constitute weak evidence that Conjecture 7.1's cohomology computation "actually returns machine-readable information". The non-triviality of H^2 has not yet been verified.

7.3.4 Counter-example candidates and verification protocol

The possibility of Conjecture 7.1 being false — that $H_{\text{sign}}^2 = 0$ — must be considered seriously. Two counter-example candidates and their verification protocols:

COUNTER-EXAMPLE CANDIDATE A: CONTRACTIBLE POSET

If $\mathcal{P}_{\text{reason}}$ is a lattice with unique top and bottom, its nerve is contractible and $H^n = 0$ for $n \geq 1$. **Verification:** mechanically extract `reason` tags and implication relations from Sign's implementation, then decide the Hasse diagram's contractibility via discrete Morse theory ($O(|R|^2)$ algorithm).

COUNTER-EXAMPLE CANDIDATE B: COCYCLE VANISHING ON BOUNDARY

Even if non-trivial cocycles exist, their pullback $\pi^*[\omega] = 0$ from Sign code space $\mathcal{S}$ could vanish — the corresponding code shapes would then not exist. **Verification:** systematically enumerate `reason`-tag pairs in Sign's Pass 3b implementation and search for "code shapes carrying both" via property-based testing on the 10^4 -sample order.

Both protocols are technically feasible in Sign's current specification ([sign-spec-status.pdf](#) §4–5), with an estimated effort of 3–6 months for the Sign spec implementation team.

7.3.5 Roadmap to implementation

Verifying this conjecture as an implementation requires four stages:

1. **Formal definition of the Pass 3b poset:** the currently-parallel `reason` set (`arithmetic-type-mis-match` , `bare-rest-non-tilde-list` , `flat-layer-float-literal` , ...) is structured as a poset by making implication relations explicit. A "reason implication axiom" is added to Sign's spec, and Pass 3b output is machine-verified to satisfy it.
2. **Efficient construction of the simplicial nerve:** for the current $|R| \sim 40$ `reason` tags, the simplicial nerve has $\sum_n \binom{|R|}{n+1}$ simplices. Computable up to $n = 3$ in $O(|R|^4) \sim 2.5 \times 10^6$.
3. **Cocycle computation algorithm:** implement an efficient algorithm for poset cohomology in Sign itself (or in Lean). Existing algorithms for A_n -type cluster-complex cohomology (Fomin–Reading 2005) serve as reference. Implementation in Sign itself is preferred — the self-referential structure "Sign computes cohomology of Sign's own code space" is theoretically significant.
4. **Existence proof of language-side hidden zeros:** manually verify on sample programs that the computed 2-cocycles actually capture "non-executable code shapes". Build a correspondence table with SFO hidden zeros at $n = 4, 5$ and check whether Corollary 4.2's isomorphism extends to $n = 2$.

7.3.6 Theoretical implications and Sign's next five years

If proven, Sign becomes not merely a category-theoretic programming language but "**a language capable of computing cohomology on its own code space**". This is a **categorical accounting capability** distinct from the formal-proof capability of Coq/Agda-style dependently-typed languages. Implementation cost is significant, but in the context of compile-time acceptance testing of AI output, this could deliver a verification capability without precedent.

Corollary 7.2 (This monograph; implications for AI-output verification)

If Conjecture 7.1 holds and the set of language-side hidden zeros $[\omega]$ is finite and mechanically enumerable, then Sign's compiler can decide **at compile time**, for any AI-generated programme $p \in \mathcal{S}$, "in which cocycle class p will collapse at runtime". This means statically detecting a category of obstructions uncaptured by current type systems (Hindley–Milner or Rust's borrow checker).

THE NEXT FIVE YEARS ON THE SIGN SIDE

The next five years on the Sign side comprise three tasks:

1. Proof of Conjecture 7.1 (or discovery of a counter-example) and implementation verification.
2. Lean/Coq formal proofs of the 10 isomorphism identities in `unit.md` §3.
3. Publication of a textbook positioning Sign as the canonical "categorical specification language" — the 2020s Lawvere–Lambek programme (§5.3).

With all three complete, Sign is theoretically justified as a language holding a genuine categorical equivalence with SFO. The pedagogical implications of the anti-vibe-coding argument (§8.2) become fully effective only under this theoretical justification.

CHAPTER 8

Implications — education, industry, and AI collaboration

8.1 Pass 3b as an acceptance-test theory for AI output

The immediate industrial implication of this monograph is that Sign's Pass 3b provides a **categorical acceptance-test theory for AI-generated code**. As Corollary 7.2 states, an AI-generated programme is statically classifiable by which `reason` -cocycle class it will collapse into. This is qualitatively different from the runtime unit-test approach and — because it is a compile-time property — imposes no marginal cost per acceptance.

8.2 Implications for education

The `human-ai-manifesto.md` submitted by the user positions Sign as a "**categorical mediating language**" for human–AI communication, refuting the vibe-coding approach. This monograph provides theoretical justification for that position: Sign's compile-time verification apparatus is not an arbitrary language-design choice but the categorical dual of the same structure discovered independently by frontier physics.

Concretely, the following curriculum revisions become defensible:

- **Category theory as first-year foundational subject:** not because it is beautiful, but because it is the shortest path to the specification-side language every practitioner will use in the AI era.
- **Type theory as an operational (rather than proof-theoretic) discipline:** Sign's Pass 3b diagnostics are directly usable by non-mathematicians and provide a low-cost on-ramp to the categorical framework.
- **Positive geometry as an elective for computer-science students:** the "categorical rewriting theory" viewpoint of Chapter 4 unifies compiler-theory and cutting-edge physics under one narrative.

8.3 Vertical integration with automotive RTOS and bare metal

As `sign-qnx-compat.pdf` reports, Sign compiles to code running unmodified on QNX (automotive RTOS) — Principle 8's RISC-as-is is the crucial enabler. This suggests the possibility of extending the monograph's Corollary 7.2 to a real-time-critical system, where safety-critical AI code (automotive control, medical, aerospace) can be admitted only when Pass 3b decides the code carries no hidden-zero cocycle class.

Combined with hardware synthesis of the amplitude computation (§7.2.2), the vision of "a computational substrate driven by category theory at all layers, from FPGA to specification" comes within reach. Sign's next five years and SFO's next five years appear likely to intersect on precisely this substrate.

APPENDIX A

Excerpts from the Sign specification

A.1 `unit.md` §3 — laws of the zero object (summary)

TABLE A.1 — THE `___` LAWS STATED IN `UNIT.MD` §3.3

Category	Rule	Categorical meaning
Logic / boolean	<code>___ & X = ___</code> , <code>___ X = X</code>	<code>___</code> is (terminal, initial)
Arithmetic	<code>___ + X = X</code> , <code>___ * X = ___</code>	Unit / absorption
Arithmetic (overflow)	<code>___ + K = ___</code>	Terminality in overflow region
Function application	<code>f ___ = ___</code>	Completeness axiom: every morphism preserves <code>___</code>
List / struct construction	<code>___ ~ xs = xs</code> , <code>___ ' k = ___</code>	Unit / null projection
Reason tags	Pass 3b reason set as poset	Cocycle record of <code>___</code> -paths

A.2 `type_system.md` §2 — Layer 2 Atom internal types

Layer 2 Atom internal types (`type_system.md` §2) include Char (UTF-8, 1–4 bytes), Int (variable-width integer, non-overflow guaranteed at spec level), Float (IEEE-754 with prohibition of arithmetic on `___`), Address (memory location, `___` in overflow region), and Struct (product of Layer 2 Atoms). Layer 3 (Lambda / composite) types are constructed by coproduct composition of Layer 2 Atoms.

A.3 `0_design_principles.md` — excerpt from Principle 1

"Category theory is the first principle of this language. Every language feature must be expressible as a categorical construction. When more than one categorical construction is available for the same feature, the one requiring fewer axioms is chosen. The language does not add ad-hoc mechanisms; when a categorical construction is unavailable for a desired feature, the feature is dropped, not added ad-hoc."

APPENDIX B

Glossary of surfaceology terminology

TABLE B.1 — KEY TERMS FROM POSITIVE GEOMETRY AND SFO

Term	Meaning	First appearance in this monograph
Positive geometry	Pair $(X, X_{\geq 0})$ with unique canonical form	§2.1, Def. 2.1
Canonical form Ω	Meromorphic top-form satisfying the three axioms	§2.1, Def. 2.1
Res-recursion	Residue of Ω equals canonical form of boundary	§2.1
Amplituhedron	Positive geometry realising $\mathcal{N} = 4$ SYM amplitudes	§2.2
Associahedron	Cluster-algebra polytope realising $\text{Tr}(\phi^3)$ amplitudes	§2.2
Curve integral formula	Amplitude representation as an integral over curves on marked surfaces	§2.3, Thm. 2.2
Fatgraph	Oriented ribbon graph with punctures	§2.3, §4.1.4
Farey mutation	Diagonal-flip operation between triangulations	§4.1.4
Hidden zeros	Amplitude vanishes on codimension-2 loci determined by boundary combinatorics	§2.4, Thm. 2.3
Exact factorisation	$\mathcal{A}_n \rightarrow \mathcal{A}_L \times \mathcal{A}_R + O(\epsilon)$ near a hidden zero	§4.3
Scalar-scaffolded gluons	Recovery of gluon amplitudes from $\text{Tr}(\phi^3)$ via a unique shift	§2.4, Thm. 2.4
Cluster algebra	Algebraic system with mutation generators; Laurent phenomenon holds	§4.5.1
Fock–Goncharov $\mathcal{X}$ -variety	Cluster-algebra moduli variety, on which canonical form is invariant	§4.5.1

APPENDIX C

Full correspondence table (20 rows)

TABLE C.1 — FULL 20-ROW CORRESPONDENCE BETWEEN SFO AND SIGN

#	Aspect	SFO (Surfaceology)	Sign
1	Primitive	Curve on marked surface	Whitespace coproduct $\odot$
2	Free-monoid basis	$\{L, R\}^*$	$\{\lambda, A\}^*$
3	Category theory formulation	Cluster $\mathcal{X}$ -variety	Pointed category with $\odot$
4	Canonical form / normal form	de Rham canonical form Ω	COPRODUCT_PHASES normal form
5	Boundary law	Res-recursion	<code>___</code> laws in <code>unit.md</code> §3.3
6	Homological structure	Ext-tower in de Rham cohomology	Ext-tower in reason poset
7	Hidden obstructions	Hidden zeros ($n \geq 4$)	Conjecture 7.1 (2-cocycles)
8	Static factorisation	Exact factorisation on locus	Completeness axiom short-circuit
9	Specialisation	Kinematic shift δ	ArgSig σ
10	Recovered multiplicity	$\text{Tr}(\phi^3)$ / pion / gluon	Int / Float / List / Struct instances
11	Rewriting primitive	Triangle-pair flip	Adjacent-pair reduction
12	Confluence guarantee	Laurent phenomenon	Left-associative compatibility
13	Complexity of normalisation	Finite Farey sequence	$O(n^2)$ left-associative scan
14	Symmetry embedding	Projective invariance embedded in primitive	Uniformity embedded in coproduct
15	Observability granularity	Per-amplitude value	Per-token, per-tag
16	Formal proof maturity	Decades of algebraic-geometry rigour	362 tests + specification prose
17	Continuous vs. combinatorial	Measure-theoretic (differential form)	Combinatorial (order theory)
18	Hardware substrate	Numerical amplitude libraries	QNX / AArch64 native
19	Prediction / testing methodology	Amplitude vs. Feynman comparison	Pass 3b diagnostic tests
20	Categorical framework	Positive geometry + cluster algebra	2020s Lawvere–Lambek programme

REFERENCES

References

1. Arkani-Hamed, N., Bai, Y., Lam, T. (2017). Positive geometries and canonical forms. *JHEP* **11**, 039. [arXiv:1703.04541](#).
2. Arkani-Hamed, N., Trnka, J. (2013). The amplituhedron. *JHEP* **10**, 030. [arXiv:1312.2007](#).
3. Arkani-Hamed, N., Bai, Y., He, S., Yan, G. (2017). Scattering forms and the positive geometry of kinematics, colour, and the worldsheet. *JHEP* **05**, 096. [arXiv:1711.09102](#).
4. Arkani-Hamed, N., Frost, H., Salvatori, G., Plamondon, P.-G., Thomas, H. (2023). All loop scattering as a counting problem. [arXiv:2309.15913](#).
5. Arkani-Hamed, N., Frost, H., Salvatori, G., Plamondon, P.-G., Thomas, H. (2023). All loop scattering for all multiplicity. [arXiv:2311.09284](#).
6. Arkani-Hamed, N., Cao, Q., Dong, J., Figueiredo, C., He, S. (2024). Hidden zeros for particle physics amplitudes. [arXiv:2312.16282](#).
7. Arkani-Hamed, N., Cao, Q., Dong, J., Figueiredo, C., He, S. (2024). $\text{NLSP} \subset \text{Tr}(\phi^3)$. [arXiv:2401.00041](#).
8. Arkani-Hamed, N., et al. (2024). Tropical amplitudes and hidden zeros. [arXiv:2402.06719](#).
9. Arkani-Hamed, N., et al. (2024). Positive geometries at all loops. [arXiv:2412.21027](#).
10. Cao, Q., Dong, J., Figueiredo, C., He, S. (2025). Physical zeros of scattering amplitudes. *Phys. Rev. Lett.* **134**, 171601.
11. Herrmann, E., Trnka, J. (2022). The SAGEX review on scattering amplitudes, chapter 7: positive geometries and canonical forms. *J. Phys. A* **55**, 443008.
12. Fomin, S., Reading, N. (2005). Root systems and generalized associahedra. In: *Geometric Combinatorics*, IAS/Park City Math. Ser. **13**.
13. Fock, V. V., Goncharov, A. B. (2009). Cluster ensembles, quantization and the dilogarithm. *Ann. Sci. École Norm. Sup.* **42**, 865–930.
14. Squier, C. C. (1987). Word problems and a homological finiteness condition for monoids. *J. Pure Appl. Algebra* **49**, 201–217.
15. Fomin, S., Zelevinsky, A. (2002). Cluster algebras I: Foundations. *J. Amer. Math. Soc.* **15**, 497–529.
16. Lambek, J., Scott, P. J. (1986). *Introduction to Higher-Order Categorical Logic*. Cambridge University Press.
17. Lawvere, F. W. (1963). *Functorial Semantics of Algebraic Theories*. Ph.D. thesis, Columbia University.
18. Moggi, E. (1991). Notions of computation and monads. *Inform. Comput.* **93**, 55–92.
19. Wadler, P. (1992). The essence of functional programming. In: *POPL '92*, 1–14.
20. Curry, H. B., Feys, R. (1958). *Combinatory Logic, Vol. I*. North-Holland.
21. Howard, W. A. (1980). The formulae-as-types notion of construction. In: *To H. B. Curry*, Academic Press, 479–490.
22. johnny-shaman/Sign repository, files [0_design_principles.md](#), [unit.md](#), [type_system.md](#), [syntax_and_sets.md](#), [coproduct_resolver.md](#). Version snapshot 2026-09-17.
23. sign-spec-status.pdf (user submission): Sign specification implementation status report, 362 unit tests, September 2026.
24. sign-qnx-compat.pdf (user submission): Sign–QNX vertical integration report.
25. human-ai-manifesto.md (user submission): categorical mediating language for human–AI communication.
26. Geometry Beyond Spacetime (Quanta Magazine, 2024-09-25): popular summary of the amplituhedron / surfaceology programme.
27. Bourbaki, N. (1970). *Éléments de mathématique: Théorie des ensembles*. Hermann.
28. Mac Lane, S. (1998). *Categories for the Working Mathematician* (2nd ed.). Springer.

29. Riehl, E. (2016). *Category Theory in Context*. Dover.
30. Barr, M., Wells, C. (1990). *Category Theory for Computing Science*. Prentice Hall.
31. Selinger, P. (2010). A survey of graphical languages for monoidal categories. In: *New Structures for Physics*, Springer.
32. Coecke, B., Kissinger, A. (2017). *Picturing Quantum Processes*. Cambridge University Press.
33. Awodey, S. (2010). *Category Theory* (2nd ed.). Oxford University Press.
34. Reynolds, J. C. (1974). Towards a theory of type structure. In: *Programming Symposium*, LNCS 19, 408–425.
35. Milner, R. (1978). A theory of type polymorphism in programming. *J. Comput. System Sci.* **17**, 348–375.
36. Girard, J.-Y. (1987). Linear logic. *Theor. Comput. Sci.* **50**, 1–101.
37. The Univalent Foundations Program (2013). *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study.
38. Simons Foundation (2024-09-25). "The Universe as a Number Puzzle" (Quanta Magazine feature on surfaceology).