The Principle of Efficient Novelty: Coherence Windows and the Nature of Mathematical Construction

Halvor Lande
hsl@awc.no

February 2026

Abstract

Mathematical foundations differ not only in their logical primitives, but in their *Coherence Window*—the depth of historical context required to stabilize structural obligations. We introduce the *Principle of Efficient Novelty* (PEN), a formal model of mathematical evolution in which a computational agent selects structures maximizing combinatorial enabling power relative to integration cost. We establish three results. First, the **Complexity Scaling Theorem**: foundations with a Coherence Window of exactly 2 (e.g., intensional type theory) force integration costs to follow the Fibonacci sequence, $\Delta_n = F_n$, with realization times $\tau_n = F_{n+2} - 1$. Second, the **Stagnation Theorem**: foundations with a Coherence Window of 1 (e.g., extensional set theory) have bounded integration costs and exhibit asymptotic efficiency collapse. Third, the **Inductive Exponentiality Theorem**: in constructive type theory, novelty scales as 2^{Δ}, ensuring that efficiency permanently outpaces the rising selection bar.

Applied to an empty library, these axioms produce the *Genesis Sequence*: 15 mathematical structures—from dependent types through spheres, cohesion, and differential geometry to the Dynamical Cohesive Topos—in a deterministic order governed by the Golden Ratio φ. The companion paper [1] exhibits the complete sequence and verifies it computationally.

Contents

1 Introduction

Why do some mathematical foundations naturally support the emergence of high-dimensional geometric structures, while others remain confined to discrete combinatorics? The historical shift from extensional foundations (ZFC, Martin-Löf Type Theory with UIP) to intensional foundations (Homotopy Type Theory [2], Cubical Type Theory [3]) is often viewed as a semantic refinement regarding the treatment of equality. Viewed through the lens of computational complexity, this shift represents a *phase transition* in the evolutionary dynamics of the system.

In an extensional system, equality is a proposition; checking it is a static operation. In an intensional system, equality is data (a path); checking it requires constructing a witness that must cohere with the surrounding topological context. This imposes an *integration cost* on new structures: a candidate definition must be "sealed" against previous layers of the theory.

This paper proposes that the dynamics of mathematical evolution are determined by the *Coherence Window* of the foundation—the depth of history required to resolve these obligations. By abstracting away surface syntax and modeling evolution via *Obligation Graphs*, we resolve a puzzle observed in computational experiments: distinct foundations (e.g., Cubical vs. Simplicial Type Theory) generate identical cost trajectories. This is a consequence of *Graph Isomorphism* in their coherence obligations—if two foundations share the same Coherence Window, they impose isomorphic integration costs on the abstract structures they realize.

Contributions.

(i) We classify foundations by their Coherence Window d and prove that for $d = 2$, integration costs satisfy $\Delta_{n+1} = \Delta_n + \Delta_{n-1}$ (the Fibonacci recurrence), with realization times $\tau_n = F_{n+2} - 1$ (section 5).

(ii) We prove that $d = 1$ foundations stagnate: constant costs lead to bounded efficiency and asymptotic collapse (section 5, §5.3.1).

(iii) We prove that in constructive type theory, novelty scales as 2^{Δ}, ensuring unbounded efficiency growth (section 6).

(iv) We present the Genesis Sequence—15 structures produced by these axioms applied to an empty library—and verify it computationally (sections 2 and 7).

2 The Genesis Sequence

Before developing the formal framework, we present its principal output: the *Genesis Sequence*. This is the complete evolutionary trace produced by the Principle of Efficient Novelty applied to an intensional type-theoretic foundation. The model operates on abstract Obligation Graphs rather than syntactic artifacts; it generates candidate structures, computes their integration costs, and selects the candidate maximizing efficiency. The reader should treat table 1 as an empirical fact to be explained; the remainder of the paper defines the model that produces it.

Table 1: The Genesis Sequence

n	τ	Structure	Δ_n	ν	κ	ρ	Φ_n	Ω_{n-1}	Bar
1	1	Universe $\mathcal{U}_0$	1	1	2	0.50	—	—	—
2	2	Unit type $\mathbf{1}$	1	1	1	1.00	1.00	0.50	0.50
3	4	Witness $\star : \mathbf{1}$	2	2	1	2.00	2.00	0.67	1.33
4	7	Π/Σ types	3	5	3	1.67	1.50	1.00	1.50
5	12	Circle S^1	5	7	3	2.33	1.67	1.29	2.14
6	20	Propositional truncation	8	8	3	2.67	1.60	1.60	2.56
7	33	Sphere S^2	13	10	3	3.33	1.62	1.85	3.00
8	54	$S^3 \cong \mathrm{SU}(2)$	21	18	5	3.60	1.62	2.12	3.43
9	88	Hopf fibration	34	17	4	4.25	1.62	2.48	4.01
10	143	Cohesion	55	19	4	4.75	1.62	2.76	4.46
11	232	Connections	89	26	5	5.20	1.62	3.03	4.91
12	376	Curvature tensors	144	34	6	5.67	1.62	3.35	5.42
13	609	Metric + frame bundle	233	43	7	6.14	1.62	3.70	5.99
14	986	Hilbert functional	377	60	9	6.67	1.62	4.06	6.58
15	1596	Dynamical Cohesive Topos	610	150	8	18.75	1.62	4.48	7.25

2.1 Guided Reading

The table records 15 *realizations*—mathematical structures selected in sequence from an empty library. Each row is characterized by:

- n: realization index.
- τ: cumulative realization time ($= F_{n+2} - 1$).
- Δ_n: *Integration Latency*—the cost of sealing the structure against the library ($= F_n$).
- ν: *Novelty*—the count of newly enabled constructions.
- κ: *Construction Effort*—the definitional complexity.
- $\rho = \nu/\kappa$: *Efficiency*—the selection score.
- $\Phi_n = \Delta_n/\Delta_{n-1}$: *Structural Inflation*, converging to φ.
- Ω_{n-1}: *Cumulative Baseline*—the library's historical efficiency ($\sum \nu_i / \sum \kappa_i$).
- Bar $= \Phi_n \cdot \Omega_{n-1}$: the selection threshold.

Three patterns are verifiable by inspection:

1. Fibonacci Timing. The Δ_n column is the Fibonacci sequence: $1, 1, 2, 3, 5, 8, 13, 21, \ldots, 610$. The τ column is its cumulative sum: $\tau_n = F_{n+2} - 1$. This is not a numerical coincidence; we prove in section 5 that it is the unique cost schedule for foundations with a two-step coherence window.

2. Selective Survival. Every realized structure clears the selection bar: $\rho_n \geq \mathrm{Bar}_n$. The tightest margin is at $n = 4$: dependent types ($\rho = 1.67$) clear the bar (1.50) by only 0.17. This narrow passage is structurally necessary—the system must invest in foundational infrastructure before the geometric payoff begins. Not all candidates survive: Lie groups ($\kappa = 6$, $\nu = 9$, $\rho = 1.50$) are *absorbed* rather than realized, as their efficiency falls far below the bar (≈ 4.46) at the time they become reachable.

3. Four Phases.

- **Bootstrap** ($n = 1$–4): A universe, a type, an inhabitant, dependent types.
- **Geometric Ascent** ($n = 5$–9): The circle, spheres, the Hopf fibration.
- **Framework Abstraction** ($n = 10$–14): Cohesion [5, 4], connections, curvature, metrics, Hilbert.
- **Synthesis** ($n = 15$): The Dynamical Cohesive Topos clears the bar by a factor of 2.6.

2.2 The Claims

The remainder of this paper establishes three results that, taken together, explain why the Genesis Sequence has the structure it does.

Claim 1. Fibonacci costs are necessary (section 5, theorem 5.1). For any foundation whose coherence obligations span exactly two layers, $\Delta_{n+1} = \Delta_n + \Delta_{n-1}$. With minimal initial conditions, $\Delta_n = F_n$.

Claim 2. Extensional foundations stagnate (section 5, §5.3.1). Foundations whose obligations span one layer have constant costs. The resulting evolution is linear and efficiency growth decays to zero.

Claim 3. Novelty scales exponentially with cost (section 6, theorem 6.2). In constructive type theory, a structure with Δ constructors enables 2^{Δ} distinguishable predicates at linear definitional cost. This ensures that efficiency permanently outpaces the selection bar.

These three claims explain, respectively, the Δ_n column (Fibonacci), the contrast with extensional alternatives (stagnation), and the sustained growth of ρ_n (exponential novelty).

3 Formal Framework

We model mathematical evolution as a discrete-time optimization process operating on a state $\mathcal{B}$ (the "Library"). At each step, the system generates candidate extensions, calculates their *Efficiency* $\rho = \nu/\kappa$, and selects the optimal candidate. The framework separates the *Latency* required to witness historical coherence from the *Effort* required to specify new structure.

3.1 State and Candidates

Definition 3.1 (State)**.** A *State* $\mathcal{B}$ is a monotone context (a category of contexts) closed under derivability. An evolution step $\mathcal{B}_n \rightsquigarrow \mathcal{B}_{n+1}$ is an extension by a single sealed structure.

Definition 3.2 (Candidate)**.** A *Candidate* X is a pair $(X_{\mathrm{core}}, \mathcal{G}_{\mathrm{obl}})$, where X_{core} is the definitive data (type formers, constructors) and $\mathcal{G}_{\mathrm{obl}}$ is the *Obligation Graph*: the set of atomic coherence obligations required to seal X against the history.

3.2 The Dual-Cost Model

We distinguish between the cost of *waiting* (Time) and the cost of *building* (Complexity).

Definition 3.3 (Integration Latency)**.** The *Integration Latency* $\Delta(X \mid \mathcal{B})$ is the cardinality of the Obligation Graph:

$$\Delta(X \mid \mathcal{B}) := |\mathcal{G}_{\mathrm{obl}}| \tag{1}$$

Latency is paid in *time*: high latency means the system must accumulate significant historical context before the structure becomes realizable.

Definition 3.4 (Construction Effort)**.** The *Construction Effort* $\kappa(X)$ is the count of atomic generators (constructors, axioms) required to specify X given the interface:

$$\kappa(X) := |X_{\mathrm{core}}| \tag{2}$$

Effort is the denominator of efficiency.

3.3 The Disjoint Interface

The magnitude of the Integration Latency is determined by the foundation's Coherence Window.

Definition 3.5 (Interface Basis)**.** For a foundation with Coherence Window d, the *Interface Basis* available for sealing candidate X_{n+1} is the disjoint union of the schemas exported by the previous d layers:

$$I_n^{(d)} := \biguplus_{j=0}^{d-1} S(L_{n-j}) \tag{3}$$

Lemma 3.6 (Latency Recurrence)**.** *Assuming the system saturates the interface to maximize connectivity:*

$$\Delta_{n+1} = |I_n^{(d)}| = \sum_{j=0}^{d-1} \Delta_{n-j} \tag{4}$$

For $d = 2$: $\Delta_{n+1} = \Delta_n + \Delta_{n-1}$, i.e., $\Delta_n = F_n$.

3.4 Novelty and Efficiency

Effort captures cost; we need a dual notion of benefit. A structure is valuable not in itself but for what it enables. Dependent types are powerful not because they are large but because they unlock an entire universe of constructions. We formalize this through the *frontier*—the set of structures accessible within a given effort bound.

Definition 3.7 (Canonical Frontier)**.** Given state $\mathcal{B}$ and effort horizon $H \in \mathbb{N}$, the *frontier* is:

$$\mathcal{S}(\mathcal{B}, H) := \{Y \text{ (sealed schema)} \mid K_{\mathcal{B}}(Y) \leq H\} \tag{5}$$

Definition 3.8 (Novelty)**.** The *novelty* of sealed candidate X relative to base $\mathcal{B}$ and horizon H:

$$\nu(X \mid \mathcal{B}, H) := \left|\left\{Y \in \mathcal{S}(\mathcal{B}, H) \cup \mathcal{S}(\mathcal{B} \cup \{X\}, H) \,\middle|\, K_{\mathcal{B}}(Y) - K_{\mathcal{B}\cup\{X\}}(Y) \geq 1\right\}\right| \tag{6}$$

Structures with $K_{\mathcal{B}}(Y) = \infty$ but $K_{\mathcal{B}\cup\{X\}}(Y) < \infty$ contribute—these are constructions that were impossible before X but become reachable after.

Definition 3.9 (Efficiency)**.**

$$\rho(X) := \frac{\nu(X)}{\kappa(X)} \tag{7}$$

Definition 3.10 (Realization Time)**.** The cumulative Integration Latency:

$$\tau_n := \sum_{i=1}^{n} \Delta_i = F_{n+2} - 1 \quad \text{(for } d = 2) \tag{8}$$

3.5 Selection Dynamics

Definition 3.11 (Structural Inflation)**.**

$$\Phi_n := \frac{\Delta_n}{\Delta_{n-1}} = \frac{F_n}{F_{n-1}} \xrightarrow{n\to\infty} \varphi \approx 1.618 \tag{9}$$

Remark 3.12 (Why $\Phi_n \neq \tau_n/\tau_{n-1}$)**.** At $n = 4$, $\tau_4/\tau_3 = 7/4 = 1.75$, which would raise the bar above $\rho_4 = 1.67$, killing the infrastructure phase. The correct definition $\Phi_4 = \Delta_4/\Delta_3 = 3/2 = 1.50$ allows dependent types to survive. Inflation should measure the *marginal* growth of interface debt, not the cumulative burden.

Definition 3.13 (Cumulative Baseline)**.**

$$\Omega_{n-1} := \frac{\sum_{i=1}^{n-1} \nu_i}{\sum_{i=1}^{n-1} \kappa_i} \tag{10}$$

The cumulative ratio is dominated by the bulk of the library, preventing outliers from distorting the threshold.

3.6 The Five Axioms

Axiom 3.14 (Cumulative Growth)**.** $R(\tau - 1) \subseteq R(\tau)$ for all τ. Mathematics only grows; realized structures are never removed.

Axiom 3.15 (Horizon Policy)**.** A global effort horizon $H \in \mathbb{N}$ governs the search space. After each realization: $H \leftarrow 2$. After each idle tick: $H \leftarrow H + 1$.

The horizon resets after each success, forcing the system to search locally before exploring more distant constructions.

Axiom 3.16 (Admissibility)**.** A sealed candidate X over base $\mathcal{B} = R(\tau - 1)$ is *admissible* iff:

1. X is derivable from $\mathcal{B}$ (i.e., $K_{\mathcal{B}}(X) < \infty$), and
2. $\kappa(X) \leq H$.

Axiom 3.17 (Selection)**.** The *Selection Bar* is $\mathrm{Bar}_n := \Phi_n \cdot \Omega_{n-1}$. From admissible candidates, select X with $\rho(X) \geq \mathrm{Bar}_n$ and minimal positive overshoot:

$$X \in \arg \min_{Y\,:\,\rho(Y) \geq \mathrm{Bar}_n} \big(\rho(Y) - \mathrm{Bar}_n\big) \tag{11}$$

Ties are broken by minimal κ; remaining ties realize in superposition. If no candidate clears the bar, the tick idles.

The minimal-overshoot criterion selects the most “natural” next step rather than an efficiency outlier.

Axiom 3.18 (Coherent Integration)**.** When candidate X_{n+1} is realized, it is integrated coherently by establishing coherence witnesses with recently realized structures. This produces an integration layer L_{n+1} with *integration gap* $\Delta_{n+1} := \kappa(L_{n+1})$.

Remark 3.19. axiom 3.18 leaves open how many prior layers constitute “relevant context.” section 5 establishes that exactly two layers—L_n and L_{n-1}—constitute the optimal strategy for intensional foundations.

4 Coherence Dimensions

We now apply the framework to classify concrete mathematical foundations. “Dimension” here is not geometric intuition but the operational depth required to stabilize the Obligation Graph.

4.1 Induced Obligations

Definition 4.1 (Induced Obligations)**.** Let $\mathcal{O}^{(k)}(X)$ denote the set of normalized atomic obligations induced when candidate X is sealed against a history of depth k. Because the interface is a cumulative disjoint union (definition 3.5):

$$\mathcal{O}^{(1)}(X) \subseteq \mathcal{O}^{(2)}(X) \subseteq \mathcal{O}^{(3)}(X) \subseteq \cdots \tag{12}$$

Definition 4.2 (Coherence Window)**.** A foundation has Coherence Window d if the induced obligations stabilize at depth d: for all valid candidates X and all $k \geq d$,

$$\mathcal{O}^{(k)}(X) \cong \mathcal{O}^{(d)}(X) \tag{13}$$

4.2 Class 1: Extensional Systems ($d = 1$)

Examples: ZFC, Martin-Löf Type Theory with UIP.

In these systems, equality is a proposition. If $a = b$ and $b = c$, then $a = c$ is true by property; the proof is irrelevant (Uniqueness of Identity Proofs). Introducing a new structure requires checking immediate well-formedness, but there is no requirement to prove "coherence of coherence." An operation defined on layer L_n interacts with L_n, but it does not generate distinct, irreducible obligations to L_{n-1}.

Window: $d = 1$. The interface $I_n^{(1)} = S(L_n)$; integration costs are bounded by the current object alone.

4.3 Class 2: Intensional Systems ($d = 2$)

Examples: Homotopy Type Theory [2], Cubical Type Theory [3], Simplicial Type Theory.

In these systems, equality is data (a path). A path $p : a = b$ is distinct from $q : a = b$; operations must respect this structure. Defining composition of paths (layer n) requires witnessing that the composition respects points and paths from layer $n - 1$. The Interchange Law and higher coherences generate irreducible obligations spanning two layers. By the Mac Lane Coherence Theorem (and its type-theoretic analogues), coherence at dimension 2 implies coherence at all higher dimensions. Stabilization occurs at $d = 2$.

Window: $d = 2$. The interface $I_n^{(2)} = S(L_n) \uplus S(L_{n-1})$.

5 The Complexity Scaling Theorem

We prove that Fibonacci-timed evolution is a mathematical necessity for Class 2 foundations under PEN.

5.1 The Recurrence

Theorem 5.1 (Complexity Scaling)**.** *For a foundation with Coherence Window d, evolving under PEN with the saturation assumption (lemma 3.6), the integration cost satisfies:*

$$\Delta_{n+1} = \sum_{j=0}^{d-1} \Delta_{n-j} \tag{14}$$

Proof. **Step 1 (Window Constraint).** The interface available for sealing X_{n+1} is $I_n^{(d)}$.

Step 2 (Disjoint Assembly). By definition 3.5: $I_n^{(d)} = \biguplus_{j=0}^{d-1} S(L_{n-j})$.

Step 3 (Conservation of Complexity). By lemma 3.6, $\Delta_{n+1} = |I_n^{(d)}|$. Since the union is disjoint:

$$\Delta_{n+1} = \sum_{j=0}^{d-1} |S(L_{n-j})| \tag{15}$$

Step 4 (Recursive Substitution). By definition, $|S(L_k)| = \Delta_k$: each integration layer exports exactly as many schemas as its integration cost. □

Corollary 5.2. *For $d = 2$: $\Delta_{n+1} = \Delta_n + \Delta_{n-1}$. With $\Delta_1 = \Delta_2 = 1$ (Universe and Unit), this gives $\Delta_n = F_n$.*

5.2 The Realization Time Formula

Theorem 5.3 (The Golden Schedule)**.**

$$\tau_n = \sum_{i=1}^{n} F_i = F_{n+2} - 1 \tag{16}$$

Proof. By definition 3.10, $\tau_n = \sum_{i=1}^{n} \Delta_i = \sum_{i=1}^{n} F_i$. The identity $\sum_{i=1}^{n} F_i = F_{n+2} - 1$ is standard. □

5.3 Dynamical Consequences

5.3.1 Stagnation of Class 1 Systems

For $d = 1$: $\Delta_{n+1} = \Delta_n$, so $\Delta_n = C$ (constant).

Inflation: $\Phi_n = 1$ for all n. The bar reduces to $\mathrm{Bar}_n = \Omega_{n-1}$.

Time: $\tau_n = nC$ (linear).

Efficiency Collapse. With constant integration costs, the Cumulative Baseline Ω_{n-1} converges to a finite limit as the library grows. New candidates must clear a fixed efficiency threshold with diminishing returns on novelty. The system does not halt, but its rate of producing genuinely novel structure decays to zero. This is the **Stagnation Theorem**: Class 1 foundations are dynamically trapped in linear time.

5.3.2 Acceleration of Class 2 Systems

For $d = 2$: $\Delta_n = F_n \sim \varphi^n$.

Inflation: $\Phi_n = F_n/F_{n-1} \to \varphi$. The convergence is *from below* in the early game: $\Phi_3 = 2.00$, $\Phi_4 = 1.50$, $\Phi_5 = 1.67$, $\Phi_6 = 1.60$. At $n = 4$, the dip to 1.50 provides breathing room for the infrastructure step ($\rho_4 = 1.67$).

Time: $\tau_n = F_{n+2} - 1 \sim \varphi^{n+2}/\sqrt{5}$ (exponential).

The Bar. $\mathrm{Bar}_n = \Phi_n \cdot \Omega_{n-1}$ grows steadily as Φ_n stabilizes near φ and Ω_{n-1} increases with each realization. Because Ω is a cumulative ratio, it is resistant to distortion by individual outliers.

Unbounded Evolution. With Δ_n growing exponentially, the Combinatorial Novelty (theorem 6.2) grows super-exponentially ($\sim 2^{F_n}$). Efficiency $\rho_n \sim 2^{F_n}/F_n$ outpaces the bar. The system enters a state of accelerating complexity.

The Infrastructure Correspondence. The viability of the Genesis Sequence rests on a precise correspondence between two manifestations of the Fibonacci sequence:

1. The *structural* oscillation of $\Phi_n = F_n/F_{n-1}$ around φ, modulating the bar.
2. The *evolutionary* alternation between infrastructure realizations (low ρ, high enabling power) and geometric ones (high ρ, exploiting infrastructure).

The same recurrence governs both. This resolves the paradox of why intensional foundations appear “more fertile” than extensional ones: it is a structural necessity for sustaining exponential growth.

6 The Inductive Exponentiality Theorem

The Complexity Scaling Theorem established that integration costs grow as $\Delta_n \sim \varphi^n$. If novelty scaled only linearly with cost ($\nu \propto \Delta$), efficiency would converge to a constant while the bar rises, causing collapse. We prove that in constructive type theory, novelty scales *exponentially.*

6.1 The Interface as a Signature

Definition 6.1 (Interface-Constructor Correspondence)**.** Let X be a structure saturating an interface of size Δ. In intensional type theory, this interface constitutes the *signature* of X as an inductive type: each schema corresponds to a constructor (point, path, surface), and Δ counts the orthogonal generators.

6.2 Combinatorial Scaling

Theorem 6.2 (Inductive Exponentiality)**.** *Let X be a structure with Δ constructors. Let the library contain a testing type* $\mathbf{2}$ *(Boolean). The number of distinct functions $f : X \to \mathbf{2}$ definable with $O(\Delta)$ effort is 2^{Δ}.*

Proof. **Step 1 (Elimination).** Realizing X grants access to the eliminator (pattern matching). A function $f : X \to \mathbf{2}$ requires a computation rule for each of the Δ constructors.

Step 2 (Syntax).

$$f(x) := \textbf{case } x \textbf{ of } \{c_1 \mapsto b_1;\ \dots;\ c_\Delta \mapsto b_\Delta\} \tag{17}$$

Step 3 (Cost). The case operator costs 1 unit; each branch b_i is a boolean reference costing 1. Total: $\kappa \approx 1 + \Delta$.

Step 4 (Counting). Each of Δ branches independently chooses $b_i \in \{\text{true}, \text{false}\}$. Orthogonality of constructors (disjoint interface) ensures each choice is semantically distinct. There are 2^{Δ} such functions. □

Remark 6.3. This explains why Higher Inductive Types dominate. The torus T^2 has a small interface ($\Delta = 4$: point, two loops, surface), yet its eliminator compresses $2^4 = 16$ distinct predicates into a 4-branch match.

6.3 Asymptotic Escape Velocity

Theorem 6.4 (Divergence of Efficiency)**.** *In a Class 2 foundation,* $\lim_{n\to\infty} \rho_n = \infty$.

Proof. **Cost:** $\kappa_n \approx \Delta_n \sim \varphi^n$.

Novelty: $\nu_n \sim 2^{\Delta_n} \sim 2^{\varphi^n}$.

Efficiency:

$$\rho_n = \frac{\nu_n}{\kappa_n} \sim \frac{2^{\varphi^n}}{\varphi^n} \to \infty \tag{18}$$

The super-exponential numerator dominates. The bar, growing as $\Phi \cdot \Omega$, is at most polynomial in the cumulative efficiency, hence permanently outpaced. □

7 Computational Verification

We verify the theoretical predictions with two independent implementations. The companion paper [1] presents the full computational reconstruction with detailed methodology; here we summarize the key results.

7.1 The Haskell Engine

A ∼3,000-line Haskell engine[1] implements the five PEN axioms as a synthesis loop. Starting from an empty library, it generates candidates from nine structural categories—Foundations, Formers, HITs, Suspensions, Maps, Algebras, Modals, Axioms, and Synthesis types—each gated

[1] Source code: `engine/`, 17 modules. Build: `cd engine && cabal build`. Run: `cabal run pen-engine`.

by prerequisites (e.g., suspensions require loopy types in the library; the Hopf fibration requires S^1, S^2, S^3).

Novelty for HITs and suspensions is computed by *proof-rank clustering*, a domain-independent algorithm: enumerate newly inhabited types at expression depth ≤ 1, abstract each type to a schema (replacing specific names with generic variables), filter trivial schemas, and count the remaining independent clusters. For maps, modals, axioms, and synthesis candidates, ν is computed by component-based structural formulas. The synthesis candidate (DCT) uses the Lattice Tensor Product: $\nu = 14 \times 11 - 4 = 150$, gated on Cohesion being present in the library. See [1] for the full algorithm and architecture.

Results. The engine discovers all 15 Genesis structures in the correct order. Values of ν match the table exactly for 12 of 15 structures and lie within ±15% for the remaining three (S^3, Connections, Hilbert). These variations arise from the depth-1 enumeration window, which does not fully capture deep homotopy-theoretic structure (e.g., $\pi_3(S^3) \cong \mathbb{Z}$). In all cases the structure clears its bar, and the ordering is preserved.

The engine's bar values differ slightly from the table because S^3 is discovered as a suspension ($\kappa = 3$) rather than as an SU(2)-equipped type ($\kappa = 5$). Both interpretations produce the same sequence; the question of whether κ measures the bare type definition or the equipped type remains open.

Four independent cross-validation modes (paper replay, capability computation, capability replay, genuine synthesis) all produce consistent results; see [1] for details.

7.2 Cubical Agda Mechanization

The Complexity Scaling Theorem (theorem 5.1) is proved in Cubical Agda [7][2]: for $d = 2$ and the saturation assumption, $\Delta_{n+1} = \Delta_n + \Delta_{n-1}$. The proof is machine-checked and covers:

- Initial conditions: $\Delta_1 = \Delta_2 = 1$.
- The Fibonacci recurrence: $\Delta_{n+1} = \Delta_n + \Delta_{n-1}$.
- The cumulative sum identity: $\tau_n = F_{n+2} - 1$.
- Convergence: $\Phi_n \to \varphi$.

A κ-oracle using Agda's reflection API (counting constructors of type definitions) is partially implemented. The ν-measure and selection loop are not yet mechanized. The Haskell engine serves as the fast explorer; Agda serves as the trusted checker for the foundational theorems.

8 Conclusion

We have shown that the choice of mathematical foundation is a choice of Coherence Window.

- **Class 1 (Extensional),** $d = 1$**:** Constant integration costs. Bounded novelty ($2^{O(1)}$). Linear time, asymptotic stagnation.
- **Class 2 (Intensional),** $d = 2$**:** Fibonacci integration costs ($\Delta_{n+1} = \Delta_n + \Delta_{n-1}$). Super-exponential novelty (2^{F_n}). Exponential time, unbounded acceleration.

The Golden Ratio of mathematical evolution is the dominant eigenvalue of a memory system that looks back exactly two steps. Intensional type theory is not merely a logical alternative to extensional foundations; it is the unique substrate that sustains exponentially growing structural complexity.

The companion paper [1] exhibits the full Genesis Sequence produced by these axioms—15 structures from dependent types to the Dynamical Cohesive Topos [6, 4]—and verifies it with a Haskell engine that discovers all 15 structures from unconstrained search. The complete engine source code (∼3,000 lines of Haskell) and Cubical Agda proofs are available as supplementary material.

[2] Source code: `agda/`.

References

[1] H. Lande. The Genesis Sequence: A Computational Reconstruction of the Mathematical Hierarchy. 2026.

[2] Univalent Foundations Program. *Homotopy Type Theory: Univalent Foundations of Mathematics*. Institute for Advanced Study, 2013.

[3] C. Cohen, T. Coquand, S. Huber, A. Mörtberg. Cubical Type Theory: a constructive interpretation of the univalence axiom. *TYPES 2015*, 2015.

[4] U. Schreiber. Differential Cohomology in a Cohesive Infinity-Topos. arXiv:1310.7930, 2013.

[5] F. W. Lawvere. Axiomatic Cohesion. *Theory and Applications of Categories*, 19(3), 2007.

[6] H. Nakano. A Modality for Recursion. *Proceedings of LICS*, 2000.

[7] A. Vezzosi, A. Mörtberg, A. Abel. Cubical Agda: A Dependently Typed Programming Language with Univalence and Higher Inductive Types. *ICFP*, 2019.