VOID Theory: Mathematics Without Infinity

A Resource-Bounded Framework

Probabilistic Minds Consortium
(Konrad Wojnowski, PhD + AI contributors)

7.12.2025

"An actual implementation of finitary math."
— Doron Zeilberger[1]

[1]In private correspondence (11.2025) following a presentation of the computational engine of VOID Theory, its collection of Coq files, mathematician Doron Zeilberger—prominent advocate of ultrafinitism and computational mathematics—endorsed VOID Theory as "an actual implementation of finitary math" and stated to "fully endorse [my] effort." While abstaining from actively promoting the work due to time constraints, his priceless recognition acknowledges VOID's alignment with his long-standing critique of infinity in mathematical foundations.

Abstract

Classical mathematics has no definition of space. Space in modern mathematics is $\mathbb{R}^n$—the Cartesian product of real numbers. But each real number requires infinity (Dedekind cuts, Cauchy sequences, infinite decimals). Each point in $\mathbb{R}^n$ is zero-dimensional. Space is $\infty \times 0$—undefined algebra masquerading as foundation.

We present VOID: mathematics built from finite observation where operations cost budget, infinity cannot be constructed, and every operation either terminates or returns U when budget exhausts.[2]

Core mechanism: Every operation costs μ-ticks of budget and generates heat (dissipated budget), obeying strict conservation: $B_{\text{initial}} = B_{\text{final}} \oplus h$. Mathematical content is encoded via strictly bounded naturals (**Fin**) and bounded rational pairs (**FinProb**), so all values and results are constructed and computed within finite, explicitly delimited resource limits—never crossing into actual or potential infinity. Probability values and all rational outputs are always derived within these finite bounds; operations return True, False, or explicit U (undecidable) if the budget can no longer support exact distinction.

Key results: (1) Exhausted operations return U, making computational limits explicit. Classical systems crash, hang, or hallucinate; VOID admits "I cannot afford this computation." (2) Quantum-like superposition emerges when classical determination exceeds available resources. (3) Information operations exhibit thermodynamic asymmetry: reading preserves budget, writing and erasure generate heat. (4) Paradoxes requiring infinite regress (Russell, Halting, Gödel) return U instead of crashing—they're defunded, not solved. (5) We construct probability, geometry, and entropy without infinite types—demonstrating that infinity is eliminable even in domains where it appears essential.

Verification: 2500+ lines of Coq formalization relative to Martin-Löf type theory and are available at the project repository. All operations maintain conservation $B = B' \oplus h$. All modules handle U explicitly. Zero axioms beyond constructive type theory.

Applications: Safety-critical systems that guarantee termination by returning U rather than hanging. AI agents that return U instead of hallucinating. Compilers verifying resource bounds at compile time. Databases with multi-resolution storage. Cryptography with proven resource requirements. Credit propagation as thermodynamic refund, not gradient descent.

Framework applies to resource-bounded computation, simulation, measurement, and discrete systems—not universal mathematics.

[2]This work is the result of an over year-long, dialogic collaboration between the carbon-based author and multiple AI silicone-based agents. Conceptual scaffolding, mediation, and defending the finite approach are due to the human coordinator, while much of the technical design—especially explicit budgeting—was driven by the AI partners. No other humans participated, and the boundary between intention, misunderstanding, and creative emergence is often blurred. Personal circumstances (severely malignant cancer and interpersonal troubles) shaped the process but do not claim authorship; this is, perhaps, among the first serious mathematical works created as a genuine human–AI co-production rather than the vision of a solitary individual. The solitary individual in question feels pride to have achieved these results whilst working under new methodologies and in radically novel epistemic and material conditions.

Contents and Section Highlights

- **9. Conclusion—Mathematics After the Infinite**
  Summarizes VOID's historical, logical, and practical implications: showing how explicitly finite mathematics both resolves classical paradoxes and expands the conceptual toolkit. Discusses the coexistence and divergence of VOID from classical frameworks, the philosophical stakes of operational honesty, and the path forward for mathematics and computation without infinite assumptions.

1 Core Claims of Finite Mathematics

1.1 The Observer-Void Origin

The framework doesn't begin with a pre-existing mathematical universe that we fill with objects in fixed relations. Instead, it starts from a subject—an originary observer—endowed with finite capacity to make distinctions. This observer begins in the void, empty, without structure: space and objects emerge only when budget is allocated to maintain distinctions. **VOID does not offer a view from outside mathematical reality**; it is the arena where patterns come into existence only through finite acts of distinction, each requiring resource expenditure. No pattern exists a priori—each one must be constructed and actively kept apart from the background by an observer with limited budget. Mathematical reality, in this system, is the ever-changing fabric of explicitly maintained patterns, never an a priori collection of objects.

This inversion has immediate consequences. If mathematical objects emerge through budgeted acts of distinction, then *completed infinity cannot exist*—it would require infinite budget to construct. Yet classical mathematics treats infinity not as idealization but as foundation.

1.2 The Infinity Problem

Space in classical mathematics is $\mathbb{R}^n$—the Cartesian product of real numbers. But each real number requires infinite construction: Dedekind cuts partition rationals infinitely, Cauchy sequences converge through infinite steps, decimal expansions never terminate. Each point in $\mathbb{R}^n$ is zero-dimensional—no extent, no size. Space becomes $\infty \times 0$: an infinite collection of dimensionless points. This is undefined algebra masquerading as foundation. An observer with finite budget cannot construct such objects. They exist only by assumption, not by construction.

Yet for two millennia—from Euclid through Gauss—humans practiced geometry without constructing real numbers. Euclid's points were not elements of $\mathbb{R}^n$ but figures drawn and reasoned about. Newton and Leibniz built calculus using infinitesimals they could not rigorously define. Archimedes computed π through finite algorithms, not completed decimal expansions. Only in the 1800s did Dedekind and Cantor declare completed infinity necessary as axiom, not derived from operations.

That choice enabled elegant theorems but introduced pathologies: non-measurable sets requiring the Axiom of Choice, self-referential paradoxes (Russell, Gödel), renormalization divergences in quantum field theory where infinities must be "subtracted away," algorithms proven correct "in the limit" that crash when implemented on finite machines. These are not peripheral difficulties but symptoms of foundations built on objects no finite observer can construct.

VOID rejects this path. Section 6 constructs geometry—distances, dimensions, curvature—from finite distinguishability, recovering what geometric practice actually was before formalization demanded infinite precision. The question is not whether classical mathematics *works*—it does, spectacularly—but whether infinity is *necessary* or merely *convenient*. We demonstrate it was always optional.

1.3 Finite All The Way Down

Nothing infinite exists as a completed object. Our finite type **Fin** is bounded by parameter MAX, preventing infinite construction through any operation. **Fin replaces infinite natural numbers with a strictly finite inductive type**: syntactically it has zero (fz) and successor (fs), but every

Fin value is bounded by MAX. Natural numbers $\mathbb{N}$ exist only in Coq's metalanguage for proofs about the system—marked `fin_to_Z_PROOF_ONLY`—never as computational objects.

Fin serves dual roles: as administrative infrastructure (tick counts, budget indices) AND as mathematical content itself. Fin values are computed with directly—they represent finite counts, locations in space, indices, and measurements. This is not "finite natural number arithmetic"—it is **Fin arithmetic**, a fundamentally finite structure.

All mathematical content in VOID exists strictly as patterns maintained through finite, budgeted acts of distinction. The world is granular: every change and computation proceeds in fixed, discrete ticks—there is no intermediate or continuous transition. The base layer, Fin, determines the explicit steps (like locations or counts) within a strictly bounded capacity; FinProb records rational values arising only as results of these paid operations, unconstrained by classical probability bounds. Calculations are always exact within available budget, but patterns have no persistent reality: outside active maintenance, nothing endures, and all results are provisional, subject to collapse to U upon exhaustion. No infinite process, object, or approximation exists—only the operational history of funded pattern distinction, indivisibly stepped in time and strictly finite in scope.

When the budget runs out, three-valued logic ensures the outcome is not "true" or "false" but BUnknown—a structurally honest signal that the pattern cannot be further distinguished. This framework is not a fuzzy or probabilistic approximation—it is a world where determinacy is achieved only as long as paid-for pattern distinction endures, and every bit of knowledge is as granular and provisional as the observer's resource ledger allows.

1.4 The Radical Claim

Classical mathematics treats real numbers as completed, eternal objects: $\sqrt{2}$ "exists" as an infinite decimal 1.41421356..., π "exists" as 3.14159265..., and so forth. Dedekind cuts partition all rationals into two infinite sets [26]; Cauchy sequences converge through infinite terms [10]. In both cases, the construction requires **completed infinity as input** to yield the real number as output.

VOID denies that such objects exist. What classical mathematics calls "$\sqrt{2}$" is not an ideal entity awaiting approximation—it is the FinProb value (141421356, 100000000) at resolution $\rho = 10^8$, computed within available budget. That value **is** the mathematical content. There is no Platonic $\sqrt{2}$ behind it that we're "approaching." Increasing precision doesn't get you closer to an ideal—it costs more budget and yields a different FinProb pair.

The "real number as limit" is not an object you approach through successive refinement. It is **an operation you cannot afford to complete**. When budget exhausts at resolution ρ, computation stops and returns the FinProb value achieved. That value doesn't approximate something else—it **is** the mathematical content at that resolution. The "limit" is neither attained nor approached; it is the unfunded operation that never runs.

This is not approximation—this is elimination. Physics already operates this way: measured quantities are never exact reals but intervals with error bars, probabilistic confidence, instrumental resolution. VOID makes this operational reality the mathematical foundation. The question is not "how closely can we approximate π?" but "what precision can we afford?" Mathematics becomes a record of completed, budgeted computations—not contemplation of ideal, uncomputable objects.

Classical analysis assumes infinite precision is ontologically prior and finite measurements are degraded shadows. VOID inverts this: **finite computation is fundamental; infinity is the expensive fantasy we cannot afford.**

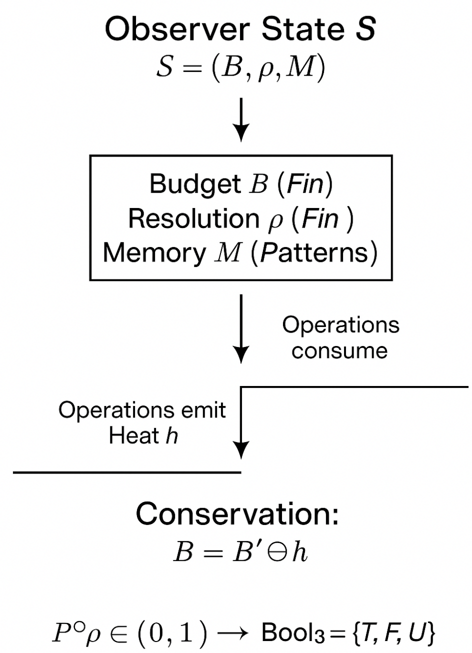


Figure 1: **Observer State S = (B, ρ, M) with conservation law $B = B' \oplus h$.** Operations consume budget B, generate heat h, and produce values in $P^{\circ}_{\rho} \in (0, 1)$ or three-valued logic $\text{Bool}_3 = \{T, F, U\}$.

The Axiom of Choice is thus inadmissible: in a resource-bounded framework, selections must be constructive and every choice operation consumes budget. Arbitrary selection from infinite collections cannot be realized.[3]

1.4.1 Physical Motivation: Why Mathematics Lives in $(0, 1)$

Consider the aforementioned act of measuring a nominal 5V source. Classical mathematics suggests an exact value such as "5.000..." volts, implying infinite precision. Physical measurement reports instead "4.97V $\pm$ 0.03V," admitting uncertainty and instrument limits. VOID expresses the same situation as a probability of 0.997 of matching the 5V standard at resolution $\rho = 1000$, with heat cost proportional to the digits obtained. What physics already knows—that every measurement is a finite comparison at some resolution—is here made explicit in computational terms.

1.4.2 Connection to Existing Frameworks

- **Fuzzy logic** (Zadeh): Membership degrees in $[0, 1]$, but we exclude endpoints [130]
- **Probabilistic programming**: All values carry uncertainty, but we make this constitutive

[3] For some, it is crystal clear that choosing—while standing in front of a wardrobe before an important occasion—costs energy and may lead to sudden bursts of misdirected expenditure. If selecting from a finite wardrobe exhausts budget and produces suboptimal results, simultaneous selection from infinitely many indistinguishable sets without procedure becomes thermodynamically absurd rather than merely philosophically questionable.

- **Quantum amplitudes**: Complex numbers with $|\psi|^2 \in [0,1]$, but we work directly in $(0,1)$
- **Information theory**: Shannon entropy uses probabilities, never certainties

1.5 The Conservation Rule: Mathematical Foundation

VOID operations have uniform type structure:

$$f : \text{Budget} \to \text{Result} \times \text{Budget} \times \text{Heat}$$

Atomic operations. In VOID, an atomic operation is the most basic, indivisible computational step: a single constructor match (such as distinguishing `fz` from `fs` in `Fin` or `Budget`), one recursive reduction of structural size, or a single comparison. Examples include checking if the budget is exhausted, taking one successor step (`fs n`), or making a basic value comparison.

Each atomic operation incurs exactly one μ-tick of budget, independent of the operation's type or apparent complexity ($\Delta_{\text{tick}} = 1$). More complex procedures—addition and multiplication, for example—decompose into sequences of atoms: adding n to m requires m atomic steps, while multiplication entails $n \cdot m$ steps, directly mirroring the structure's granularity. This strict, uniform cost assignment underpins the VOID algebraic conservation principle: every transformation's total cost is simply the sum of its atomic steps, and no computation can ever bypass the resource ledger.

$$\textbf{Conservation Rule:} \quad b_{\text{initial}} = b_{\text{remaining}} + h_{\text{accumulated}}$$

This follows from pure arithmetic. An operation receiving budget b consumes one unit, producing remaining budget $b' = b - 1$ and heat $h = 1$.

Composition over n operations gives:

$$b_0 = b_n + \sum_{i=1}^{n} 1 = b_n + n$$

The rule exists because: (1) the type system enforces budget accounting—every operation must account for its input budget in its output; (2) uniform cost makes this accounting trivial; (3) the finite type Fin bounds all values. No external principles needed—it is structural closure under resource-tracked computation.

System Stability. Conservation provides foundational stability guarantees:

Bounded iteration: Since Budget $\in$ Fin and each operation decreases it by 1, termination is guaranteed in $\leq b_{\text{initial}}$ steps (proof by structural induction on Fin).

Predictable complexity: Every computation consumes exactly n units for n operations. Complexity analysis is exact, not asymptotic.

No divergence: The conservation equation $b + h = \text{const}$ bounds the state space to $\{(b', h) \mid b' + h = b_0\}$, which is finite and enumerable.

Well-defined exhaustion: When $b = 0$, operations return explicit uncertainty constructors (`BUnknown`, intervals, `Void`). This is the natural ground state, not failure.

The conservation rule transforms boundedness from limitation into guarantee: VOID computations cannot diverge, cannot hide infinite processes, cannot escape their finite nature. The constraint stabilizes.

Formalization: `void_arithmetic.v` (Lemma `budget_heat_conservation`), `void_finite_minimal.v` (Axioms `heat_conservation_eq3`, `heat_conservation_le3`).

1.5.1 Why This Matters

Standard arithmetic pretends $2 + 2 = 4$ exactly. But:

- Binary "1" and "0" are voltage ranges, not values of qualities belonging to platonic objects. A "1" might be any voltage between 2V–5V; a "0" between 0V–0.8V. The computer treats 4.7V and 3.2V as "the same" (both 1) because distinction within that range costs more than the operation requires. Numbers are thresholds on continuous distributions.
- Two volts plus two volts gives approximately four volts (measurement limits)
- Two steps plus two steps gives four steps (but steps are just tick labels, not mathematical objects)

Within VOID's operational semantics, natural numbers function as administrative labels and are not treated as mathematical objects; mathematical content lives in $\mathbb{P}^\circ$. This is an operational choice, not a claim about the metaphysical status of numbers in other foundations. This explains why physics uses continuous fields (really: probability densities) rather than discrete counting—nature doesn't count, it distributes probability mass. Crucially, whole numbers (0, 1, 2, 3, 4, ...) do not belong in VOID's computational environment—they exist only in the bookkeeping layer. **To reiterate, all standard operations must be expressed as fractions in $\mathbb{P}^\circ$**

1.5.2 The Key Insight

The core of VOID Theory is a resource-bounded reformulation of mathematics, anchored in explicit thermodynamic and informational constraints. Here, these constraints are promoted to the level of mathematical primitives—not physical assumptions, but axioms that shape what can be constructed and known. By articulating precise axioms and primitive notions—such as budget, distinguishability, and resolution—we reveal how foundational limits immediately propagate into every aspect of geometry, probability, and epistemology.

Infinity does not infiltrate the system by way of "arbitrarily large numbers"; the unattainable idealization of perfect unity—the number 1—is sufficient for classical infinity to sneak in. In VOID, however, values are always strictly within $(0, 1)$: neither 0 nor 1 can ever be exactly represented or attained. This symmetric inaccessibility eliminates both infinitudes: neither the infinitely large nor the infinitely precise can exist. Every value necessarily bears irreducible uncertainty, enforcing **structurally mathematical analogues** of quantum indeterminacy and thermodynamic irreversibility—not as physical phenomena, but as mathematical necessities forced by explicit resource bounds.

The practical irrelevance of the boundaries is itself operational: whether absence present or presence absent, the difference evaporates—neither extremum is ever realized. The logic of the ledger knows only finite, interior distinctions; the ends are eternally deferred.

Void Theory provides mathematical structures where energy consumption and entropy generation are primitive rather than emergent. **Operations track budget expenditure and heat dissipation explicitly, enabling transparent accounting of computational costs**—particularly valuable for resource-constrained systems and socially significant calculations where hidden algorithmic costs (and transparency) matter.

The formalization in Coq demonstrates consistency relative to Martin-Löf type theory, itself a constructive foundation [87].

Coq verification files implementing these principles:

- `void_finite_minimal.v` — Fin type bounded by MAX, Bool_3, Budget as Fin (ground zero: no infinite types)
- `void_probability_minimal.v` — FinProb as (n, d) pairs in $(0, 1)$ with heat-generating operations (perfect 0 and 1 unreachable)
- `void_arithmetic.v` — All operations cost one tick, return (result, budget', heat) triple (no free computation)
- `void_pattern.v` — Patterns as maintained distinctions, observer collapse mechanics, interference effects
- `void_distinguishability.v` — Distinguishability kernel Δ_S, subthreshold collapse, finite quotient spaces at resolution ρ
- `void_information_theory.v` — READ (budget-neutral) vs WRITE (priced), conservation laws, entropy bounds

2 Explicit Base Axioms

2.1 The Ground We Tread

VOID mathematics is determined by explicit, structural axioms that enforce finitude from the ground up. We begin with the most fundamental: the requirement that all resources and constructions are built upon a well-founded capacity.

Axiom 1 (Well-founded capacity)**.** Our resource domain $(B, \leq)$ has a least element 0 and satisfies well-foundedness: no infinite strictly descending sequences exist. This allows induction without assuming infinity:

If P holds for 0, and $P(c)$ for all $c < b$ implies $P(b)$, then P holds everywhere.

Why this improves on Peano induction: Peano arithmetic assumes we can always apply the successor function $S(n) = n + 1$, generating the infinite sequence $0, S(0), S(S(0)), \ldots$ indefinitely. This is *infinite regress* pretending to function as construction—the system presupposes exactly what it claims to build. Each application of S assumes resources to continue, yet never accounts for them. **It's mathematical Peter Pan syndrome: refusing to acknowledge that processes must terminate.**

VOID's well-foundedness aims at providing an adult version: every descending chain *must* terminate because budget exhausts. We can reason about "all cases" through transfinite induction [1] without presuming an infinite totality exists to iterate over. The successor relation $b \prec b'$ ("b' affords operations b cannot") acknowledges: some transitions are possible, some aren't, depending on available resources. No pretense of infinite continuation, no infantile regression masquerading as foundation.

2.2 Every Step Has a Cost

Operations divide sharply into two categories. Budget-neutral operations read existing information without changing it—like looking at a distinction already made. Priced operations create new information or destroy existing information—these irreversibly consume budget and generate heat.

Axiom 2 (Irreversibility decreases budget)**.**

- Priced operation: $f(b) = (\text{output}, b', h)$ where $b' < b$ and $h \neq 0$
- Budget-neutral operation: $g(b) = (\text{output}, b, 0)$—budget unchanged, no heat.

This isn't a philosophical choice but an operational commitment: in our framework, creating distinctions costs budget.

2.3 Conservation of Computational Resources

The framework's backbone is **a conservation rule** as strict as physics, introduced as mathematical axiom rather than physical law: initial resources equal final resources plus dissipated heat. No work happens "off the books."

Axiom 3 (Conservation)**.** For any operation f with budget b:

$$f(b) = (\text{output}, b', h) \implies b = b' \oplus h$$

Here $\oplus$ is addition in our resource monoid. This equation means you can't create or destroy resources, only transform them from potential (budget) to historical (heat).

2.4 Open-World Truth Values

Truth isn't binary but three-valued: T, F, and U. The U isn't "we're not sure"—it marks the precise boundary of what we can afford to compute. As budget increases, U can resolve to T or F, but decided values never change. All exhausted operations return U—a third truth value distinct from Kleene's strong three-valued logic (where U means "undefined" in a semantic sense) [72]. Our U marks a precise epistemic boundary: the statement has a definite truth value, but determining it exceeds available resources. Unlike Kleene's U which propagates through operations, our U can resolve to T or F given more budget—it's thermodynamic, not semantic.

Concretely: Pattern π requires budget B_{min} to distinguish. You have $B < B_{\text{min}}$. Query returns U. Add budget until $B \geq B_{\text{min}}$—U resolves to T or F. The truth existed; you couldn't afford it. Kleene's $U \wedge T = U$ propagates forever. VOID's $U \wedge T$ returns U until budget suffices, then returns T or F. **Exhaustion is measurable state, not semantic mystery.**

Axiom 4 (Monotone refinement)**.** With budgets $b_1 \leq b_2$:

- If $V_{b_1}(P) \in \{T, F\}$, then $V_{b_2}(P) = V_{b_1}(P)$
- If $V_{b_1}(P) = U$, then $V_{b_2}(P) \in \{U, T, F\}$

This models how exhausted systems cannot determine certain truths—not from ignorance but from thermodynamic poverty. U is the mathematics of "I literally cannot afford to check." Like a pixelated image that honestly displays resolution limits, U refuses to interpolate false precision. Smoothing algorithms pretend detail exists between pixels; classical binary logic pretends all questions have affordable answers. Both lie about what's actually distinguishable. Even if smoothness is typically preferred over ruggedness, and countless smoothing algorithms exist, we need an alternative computational model that treats reality's ruggedness seriously.

In VOID, the boundary beyond which uncertainty is no longer tolerable—and computation must honestly return "unknown"—is a design decision, not a hidden flaw. This explicit control over ignorance or risk is what fundamentally prevents hallucination and ensures epistemic humility by construction.

2.5 Probability Is Primitive (0 and 1 Unattainable)

The type `FinProb` computationally supports arbitrary finite pairs (n, d) including ratios outside $(0, 1)$. However, when interpreting these as probabilities, values at the boundaries—perfect zero $(0/d)$ and perfect unity (n/n)—represent unattainable operational ideals. The framework never enforces $0 < n < d$ as a type constraint; rather, probabilistic semantics restrict attention to the open interval, while `FinProb` remains available for non-probabilistic fraction arithmetic.

Axiom 5 (Boundary exclusion)**.** For every event e interpreted probabilistically:

$$0^\circ < P(e) < 1^\circ$$

The probability scale $\mathbb{P}^\circ$ excludes its endpoints. Numbers, when they appear, are just conventional ways to coordinatize this more fundamental probability structure. This reverses the usual foundations: probability doesn't come from measure theory on numbers; numbers are bookkeeping devices for probability.

2.6 Unified Conservation Structure

VOID unifies over ten computational domains under a single conservation principle: arithmetic operations, probability theory (as primitive structure), geometric operations, entropy measurement, information theory (the read/write boundary), distinguishability and observation, randomness generation, pattern dynamics, memory and decay processes, time as computational sequence, multi-level uncertainty (metaprobability), and observer-system interactions.

Every domain implements the same accounting structure:

$$f : A \to \text{Budget} \to (B \times \text{Budget} \times \text{Heat}) \tag{1}$$

with conservation law:

$$b = b' \oplus h \tag{2}$$

where b is initial budget, b' is remaining budget, and h is consumed budget (heat). This holds across all operations in all domains.

What this accomplishes: Standard mathematics treats distinct domains (arithmetic, probability, geometry, information theory) as separate theoretical structures. VOID demonstrates these are manifestations of a single underlying pattern: budgeted computation with mandatory accounting. The unification is formal and mathematical—not analogical or metaphorical.

Comparison to foundations: If Peano arithmetic assumes unbounded succession (the successor function applies indefinitely), VOID makes continuation capacity explicit through budget parameters. Construction, computation, and maintenance all require budget allocation. When budget exhausts, operations terminate—not by arbitrary bound, but by resource depletion.

This eliminates infinity from foundations while preserving computational expressiveness. **Infinity is eliminable because every operation is budgeted, every construction costs, and every process terminates when resources exhaust.**

Axiom 6 (Information operation asymmetry)**.** Operations partition into two classes by type signature:

- **READ**: $f : A \to B$ (budget-free, accesses existing structure)
- **WRITE**: $f : A \to \text{Budget} \to (B \times \text{Budget} \times \text{Heat})$ (budget-consuming, modifies distinguishable content)

For all WRITE operations: $h \geq \mu > 0$ (minimum consumption per distinction).

Reading existing distinctions is budget-neutral. Creating or erasing distinctions consumes budget and generates heat. This asymmetry structures all information operations.

2.7 Budget-Bounded Discrimination and Interference

When discrimination between pattern-hypotheses is budget-unaffordable, the system exhibits formal indistinguishability: co-presence of alternatives without two-valued resolution. Forcing a classical decision is a budgeted operation that consumes budget and eliminates interference terms. This is pure mathematics of resource-bounded computation.

Axiom 7 (Budgeted discrimination threshold)**.** Let A, B be mutually exclusive pattern-hypotheses with discrimination cost $D(A, B)$ at the declared resolution. For any budget b:

- If $b < D(A, B)$, then the joint state carries a non-zero interference term $I(A, B) \neq 0$ and the exclusive claim "A xor B" evaluates to U.
- Any operation that enforces a two-valued decision between A and B must emit heat $h \geq D(A, B)$ and leaves $I(A, B) = 0$.

Relation to previous claims: This sits on the same substrate as Claims 2–4: discrimination is a priced (irreversible) move; U persists until budget meets $D(A, B)$; the heat bound records the thermodynamic cost of forcing two-valued logic.

Remark: This mathematical structure exhibits formal behaviors analogous to quantum superposition, interference, and measurement collapse. VOID makes no claims about physical interpretation—we are constructing finite mathematics where discrimination has explicit computational cost. That certain physical phenomena share this structure is suggestive but outside our scope.

Methodological note: VOID's development is itself an experiment in epistemic humility—arising not from classical mathematical apprenticeship, but from the sustained alliance of a humanities background and AI-facilitated formalism. The necessity to model limitation, exhaustion, and bounded operation grew as much from personal confrontation with empirical finitude as from logical reflection. This collaborative constraint has shaped both the theory's form and its scope, with open possibilities for application to physics and beyond. In other words, it is safe to say that **VOID Theory grew out of resonance between a human agent, unsuspecting his sudden and drastic change of academic field, and AI, surprised time and time again by someone exorcising mathematics from infinity**, who both found a way to mathematically recognize the cost of making every distinction, every atomic step.

Coq verification files:

- `void_finite_minimal.v` — Axioms 1–2
- `void_probability_minimal.v` — Axiom 5
- `void_information_theory.v` — Axiom 6
- `void_observer_collapse.v` — Axioms 4, 7
- `void_pattern.v` — Axiom 3

2.8 Appendix: Conceptual Foundations—The Emergence of Counting

2.9 Motivation

This appendix addresses a foundational question that may arise after encountering VOID's axioms: *If VOID rejects* $\mathbb{N}$ *as primitive and starts with* $(0, 1)$ *and Budget, where does the very concept of counting come from?* The answer reveals how natural number structure emerges from more fundamental operations—budgeted distinction acts within the open interval $(0, 1)$. This is not technical specification (the Coq implementation uses `Fin` as bounded inductive type) but **conceptual reconstruction**: showing that counting need not be assumed but can be *derived* from finite observer operations.

What follows is the origin story. The rest of the paper does not depend on it—VOID's mathematics proceeds from `Fin`, `Budget`, and `FinProb` as given. But understanding *why* these primitives make sense, and *how* they connect to more fundamental acts of distinction, illuminates the framework's philosophical commitments.

2.9.1 Starting Point: The Generative Field

Classical mathematics begins with zero, constructs successors, builds $\mathbb{N}$, then $\mathbb{R}$, then probability as measure on $\mathbb{R}$. VOID inverts this entirely.

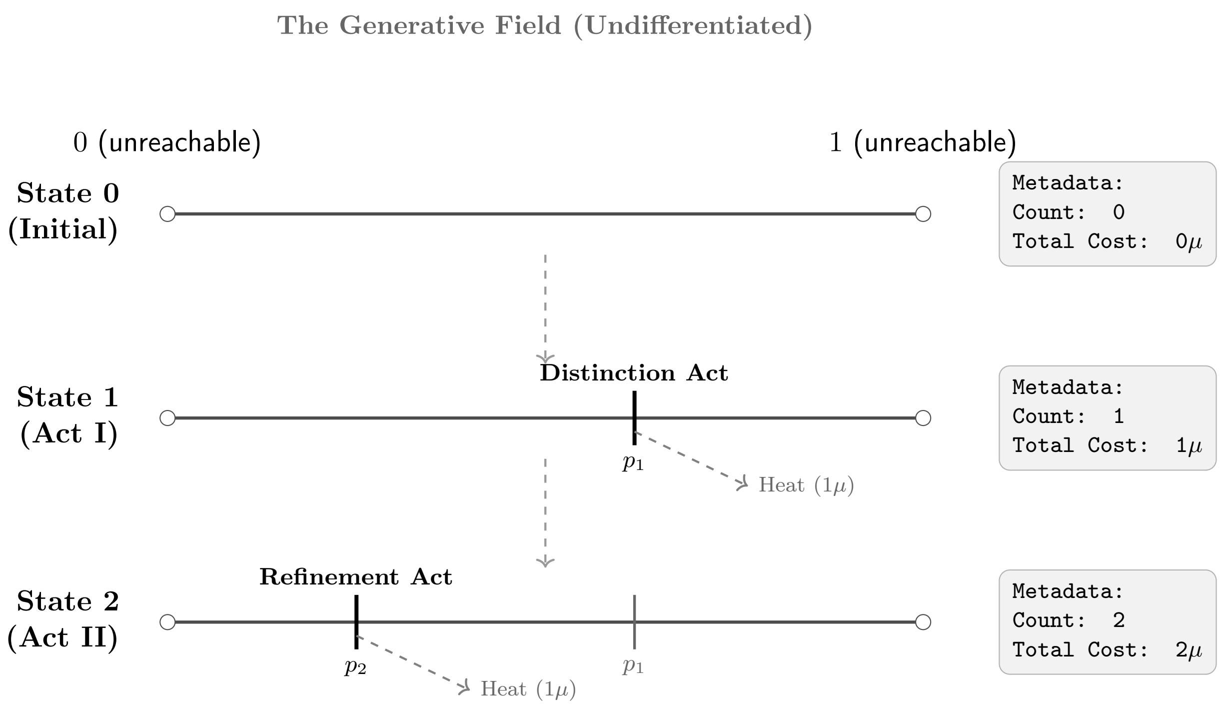


Figure 2: **The Genesis of Count: A Cascade of Distinctions.** The diagram illustrates the emergence of counting not as the discovery of pre-existing objects, but as the metadata trace of physical operations. Starting from the undifferentiated generative field $(0, 1)$ (State 0), each budgeted act of distinction creates positional awareness and dissipates heat (State 1, State 2). The "Count" is simply the cumulative record of these expensive operations.

We start with:

- The open interval $(0, 1)$ as **generative field**—not a set of points, but the arena where patterns can emerge
- **Budget** B as finite capacity to perform distinction acts
- A **situated observer position** $p \in (0, 1)$—the observer does not "choose" this point from an infinity (which would be costly and inadmissible under our rejection of the Axiom of Choice) but *inhabits* it as their initial state of being

The boundaries 0 and 1 are **unreachable limits**—present as constraints but never attainable as constructed values. Mathematical patterns emerge *between* these limits through active distinction, not *from* them through axiomatic construction.

2.9.2 Distinction Acts: Creating Positional Awareness

A **distinction act** is the primitive operation: establishing "HERE vs not-HERE" within the generative field.

Beginning at the situated position $p \in (0, 1)$, the first distinction creates awareness:

- **Region before** p: the interval $(0, p)$
- **Region after** p: the interval $(p, 1)$

This split need not be symmetric. If $p = 0.618\ldots$ (golden ratio), the first distinction creates unequal regions. **Asymmetry is fundamental**—there is no assumption of halving or binary tree structure. Each distinction responds to current context, not predetermined symmetry.

Cost: Each distinction act consumes exactly **one** μ**-tick** of budget. This is the atomic cost of creating and maintaining distinguishability.

2.9.3 Successive Refinement: Pattern Generation

From the initial distinction, further acts refine the field:

$$\begin{aligned}
\text{State}_0 &: \text{situated position } p \in (0, 1) \\
\text{State}_1 &: \text{distinguished regions } (0, p), (p, 1) \quad [\text{cost: } 1\mu] \\
\text{State}_2 &: \text{further distinctions within each region} \quad [\text{cost: } +n\mu] \\
&\vdots
\end{aligned}$$

Each refinement generates new positions, new distinctions, new maintained patterns. The structure is **organic** and **adaptive**, not mechanical or predetermined.

Key insight: After n distinction acts, the observer has:

- Consumed n μ-ticks of budget
- Generated n units of heat (dissipated budget)
- Created a field of distinguishable positions

The number "n" emerges as **metadata**—the count of distinction acts performed. It is not a primitive object but a **trace** of the generative process.

2.9.4 Numbers as Process Traces

In this framework, natural numbers are not pre-existing objects but **records of activity**:

Definition 2.1 (Number as Trace)**.** The number n is the count of distinction acts performed:

$$n \equiv |\{\text{distinction acts completed}\}|$$

Crucially:

- **Zero** is the unreachable limit—no distinctions yet possible (would require budget that doesn't exist)
- **One** represents the first distinction act—"HERE vs not-HERE"—establishing the unit cost (μ-tick) against which all future complexity is measured
- **Successor** $n \to n+1$ is not primitive but *operational*: perform one additional distinction act

The successor function is non-trivial:

$$S(\text{state}) = \text{perform\_distinction}(\text{state})$$

This is not Peano's assumed $S(n) = n + 1$ but a **constructive operation**—actually performing work, consuming budget, generating heat. The successor *emerges* from distinction capability, not from axiom.

2.9.5 Why This Matters: Peano vs. VOID

Peano Axioms (Classical)	**VOID Construction**
Axiom: Zero exists	No zero (unreachable limit)
Axiom: Successor function exists	Successor = distinction act (constructed)
Axiom: No loops	Well-founded by finite budget
Assumes: Counting structure	Derives: Counting from distinctions

Peano assumes natural numbers as foundation. VOID shows they **emerge** from:

- Generative field $(0, 1)$
- Finite budget B
- Capacity to distinguish positions

2.9.6 Resolution and Precision

The resolution parameter ρ determines the **affordable depth** of distinction:

- At resolution ρ_1 with budget B_1: Can maintain $D(\rho_1)$ distinct positions
- At resolution ρ_2 with budget $B_2 > B_1$: Can maintain $D(\rho_2) > D(\rho_1)$ positions

The denominator bound $D(\rho)$ reflects **how many distinctions you can afford to maintain simultaneously**. Higher resolution costs more budget—not because positions are "smaller" but because maintaining finer distinctions requires continuous resource expenditure.

Note: $D(\rho)$ need not be a power of 2. The binary division story is pedagogical—showing how counting *can* emerge—but actual distinctions may follow any affordable pattern. The key is **finite maintainability**, not predetermined symmetry.

2.9.7 The Decimal Point as Generative Operator

In classical notation, the decimal point is passive: "3.14159..." is static representation. In VOID, the decimal point is **generative**—each placement is an act of distinction:

- "3" represents three unit-scale distinctions
- ".14159..." represents position within subdivided space
- Each digit requires budget to distinguish and maintain

Infinite decimals are unconstructible because they require infinite distinction acts. When we write "$\pi = 3.14159\ldots$" classically, we assume a completed infinite process. VOID rejects this: at budget B and resolution ρ, you construct a `FinProb` pair—that *is* the mathematical content. There is no "ideal π" behind it that you're approximating. You construct what you can afford, and that's what exists.

2.9.8 Patterns, Not Objects

One final clarification: VOID contains **no objects**, only **patterns**.

Object (classical): Static, eternal, observer-independent entity existing "out there."
Pattern (VOID): Dynamically maintained distinction requiring continuous budget to prevent decay to indistinguishability.

A `FinProb` pair like $(31415926, 10^7)$ is not an object but a **pattern**—an actively maintained distinction between numerator and denominator, costing budget to preserve. When budget exhausts, the pattern collapses to `U` (undecidable). Nothing persists without active maintenance.

Numbers, then, are not objects but **traces**—records of pattern generation acts. They exist as metadata, not as Platonic entities. This eliminates the ontological baggage of classical mathematics while preserving computational structure.

This distinguishes VOID from other finitist and ultrafinitist positions. In VOID, you *can* run a program called "Perfect Circle Creator" or "Compute π to Infinite Precision"—the system will not reject it as syntactically invalid. The program executes, consumes budget, returns increasingly refined `FinProb` values. It remains perpetually in alpha phase, asymptotically approaching but never reaching its stated goal. VOID does not say "this is impossible"; it says "this is one among infinitely many profoundly stupid ways to spend finite resources." The distinction matters: infinity-chasing is not forbidden by logic but *discouraged by economics*. The system allows foolishness—it just makes the cost explicit.

2.9.9 Summary

Natural number structure emerges from:

1. **Generative field** $(0, 1)$ with unreachable boundaries
2. **Distinction acts** consuming budget, creating positional awareness
3. **Organic refinement** through successive distinctions (asymmetric, adaptive)
4. **Counting as metadata** recording distinction acts performed

5. **Successor as operation** (not axiom): perform next distinction

This is not how `Fin` is implemented in Coq—that uses bounded inductive types for practical verification. But it explains *why* VOID's primitives make philosophical sense: they capture the **minimum structure needed for finite observers to generate mathematical content** without assuming infinity, without assuming counting, without assuming objects. Coq uses `Fin` as the administrative infrastructure to track the budget described here; this appendix explains the ontological origin of the currency that Coq spends.

The rest of the paper proceeds from `Fin`, `Budget`, and `FinProb` as established. This appendix shows where they come from—not from axioms pulled from thin air, but from the irreducible operations of distinction, bounded by finite resources, generating structure between unreachable limits.

VOID is an **ultrafinitist** system in spirit and practice: **nothing infinite is granted, nothing is assumed cost-free, and every construction must be justified by finite means**. In this sense, VOID sits naturally in the tradition of Zeilberger, Esenin-Volpin, and the radical finitists who questioned the metaphysical inflation of classical mathematics.

In this respect, VOID is not a rejection but **an operational completion of ultrafinitism**: a mathematics where finiteness is not a constraint of ideology but a conservation law.

3 Computational Consequences

3.1 Computational Structure

The axiomatic framework produces a computational architecture with three fundamental layers: a resource management core, three-valued logic with graceful degradation, and conservation laws governing all operations.

This explicit type-driven tracking of budget and heat does not merely constrain mathematical construction; **it also opens new avenues for designing and composing mathematical operations**. Functions become resource-aware processes, whose stepwise costs and eventual exhaustions are both visible and decidable at the type level. This enables novel forms of algorithmic composition, verified complexity accounting, and conditional or adaptive computation guided by real-time resource availability. In VOID, mathematical operations are not just formulas, but structured, accountable flows—new possibilities for modeling, reasoning, and implementation emerge from this foundation.

Resource Management Core. Every operation consumes budget B (type Fin, bounded by MAX) and produces heat h. The type system enforces that all functions return triples $(result, B', h)$ where $B' \oplus h = B$ by conservation. This isn't an accounting convenience but structural: the type checker prevents operations from hiding their costs. Operations cannot "borrow" budget from future computations - when $B = 0$, the system halts or returns U. The foundational files `void_finite_minimal.v` implements Fin with the bound axiom, while `void_arithmetic.v` defines all basic operations (addition, multiplication, comparison) as budget-consuming functions with heat generation.[4]

Three-Valued Logic and Graceful Degradation. Classical two-valued logic assumes all propositions are decidable with available resources. In VOID, finite budget may exhaust before a decision can be made, requiring three-valued logic: $\mathbb{B}_3 = \{T, F, U\}$, where U is not an error state but a legitimate value meaning "distinction costs more than available budget." The comparison operations in `void_finite_minimal.v` implement this: `fin_eq_b3` and `le_fin_b3` return Bool_3 values. Classical boolean operations are recovered by *collapsing* $U \to F$, but this collapse is explicit and lossy - the computational history matters.

Finite Rational Arithmetic as Foundation. Mathematical content operates on two layers: **Fin** for bounded discrete values (locations, counts, indices), and **FinProb** for rational numbers as (n, d) pairs. Despite the name, FinProb represents arbitrary rationals—addition can produce $(6, 4) = 1.5$. When interpreted as probabilities, values should remain in $(0^\circ, 1^\circ)$, but the type system permits general rationals. The `avoids_zero` predicate checks $n \neq 0$ to exclude the boundary 0°, but does not enforce $n < d$. All arithmetic operations (in `void_probability_minimal.v`) cost budget and generate heat proportional to operand size.

Distinguishability and Resolution. The framework models observers with finite resolution ρ who can distinguish patterns only if their difference exceeds ρ. The file `void_distinguishability.v`

[4] A core advantage of VOID's explicit resource tracking is that it enables precise reconstruction of each computational, inferential, or learning trajectory. By recording consumed budget and generated heat, the system not only delivers a final answer, but exposes the full "epistemic path" leading to it—mapping the latent space of pattern distinctions and revealing which steps were resource-intensive or uncertain. This makes it possible to identify, analyze, and ultimately prevent the onset of hallucination: unlike conventional systems that may fabricate answers after resource exhaustion, VOID will always mark and localize the threshold where knowledge ends and U begins, providing both the agent and the designer with actionable insights into the relationship between finite resources, ignorance, and safe/unsafe behaviors.

implements this: two patterns below the resolution threshold collapse to indistinguishable, creating equivalence classes. Superposition emerges when classical determination (deciding between patterns) would cost more budget than available - not from physical wave functions but from computational insufficiency. The observer doesn't "see" a superposition; they lack budget to maintain the distinction, so patterns remain unresolved.

Information-Theoretic Asymmetry. READ operations (field access like `location p`) require no budget parameter; WRITE operations (arithmetic, comparisons) require budget and generate heat. This asymmetry appears in `void_information_theory.v` through READ and WRITE type classes. Entropy counting in `void_entropy.v` costs budget per element examined - even measuring complexity depletes resources. Erasure (Landauer's principle) generates heat because it's a write operation destroying maintained distinctions.

Extensions Beyond the Core. The foundational files establish the resource-bounded substrate. Extension files explore consequences:

- `void_pattern.v`: Patterns as probability-weighted locations with interference, decay, and heat-generating observation
- `void_observer_collapse.v`: Observer-induced pattern collapse, Quantum Zeno effect emerging from repeated observation
- `void_geometry.v`: Vectors as lists of probabilities, no escape to infinity through dimension
- `void_topology_folding.v`: Space folding (wormholes) as unstable bridges with decay
- `void_metaprobability.v`: Uncertainty about uncertainty, with confidence naturally decaying toward Void
- `void_resonance.v`: Pattern-seeking resonant locations through frequency matching
- `void_time_memory_composition.v`: Time as tick sequences, memory with decay
- `void_budgeted_complexity.v`: Emergence of complexity from iteration under resource constraints

These extensions demonstrate that quantum-like phenomena (superposition, observer effects, interference) emerge from budget exhaustion rather than requiring physical postulates. The Quantum Zeno effect (repeated observation freezes evolution) appears in `void_observer_collapse.v` as a purely computational phenomenon: checking costs budget, and frequent checking depletes budget before evolution can occur.

Paradox Resolution Through Defunding. Self-referential paradoxes (Russell's set, Halting Problem, Gödel sentences) require infinite descent or unbounded iteration. In VOID, these constructions exhaust budget and return U - they're not solved but defunded. The Halting Problem becomes: "Can we decide if this program halts *with budget* B?" Answer: sometimes T, sometimes F, sometimes U. The third option makes undecidability explicit rather than paradoxical. We return to this issue in more detail in the last section.

Once again, the core definitions and theorems have been formalized in Coq and consistent relative to Coq's type theory and Martin-Löf foundations.

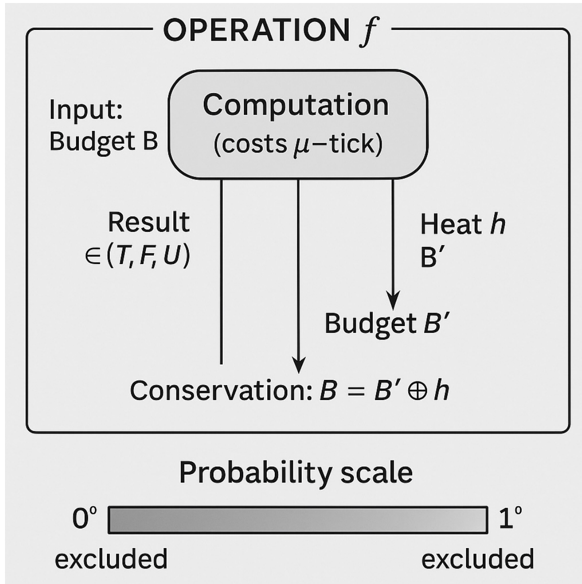


Figure 3: **Fundamental structure of VOID operations**. Each operation costs one μ-tick, returns three-valued results (T, F, U), and obeys strict conservation: $B = B' \oplus h$. Probabilities avoid boundaries, lying strictly in $(0, 1)$.

3.2 Explicit Base Axioms

Axiom 8 (EF0: Well-founded capacity (no infinite regress))**.** Let $(B, \leq, 0_B)$ be the resource domain. The strict order $<$ is well-founded: there are no infinite strictly descending chains

$$b_0 > b_1 > \cdots$$

Induction is by the usual well-founded schema:

$$\forall b\, ((\forall c < b)\, P(c) \Rightarrow P(b)) \Rightarrow (\forall b\, P(b)).$$

Axiom 9 (EF1: Three-valued truth ($\mathbb{B}_3$) and collapse)**.** Truth values at working resolution are $\mathbb{B}_3 = \{T, F, U\}$ with strong-Kleene connectives. The collapse $\kappa : \mathbb{B}_3 \to \{0, 1\}$ is a conventional interface (e.g., $\kappa(T) = 1$, $\kappa(F) = 0$, $\kappa(U) = 0$ by default).

Axiom 10 (EF2: Monotone refinement (open-world semantics))**.** For budgets $b_1 \leq b_2$ and any statement P,

$$V_{b_1}(P) \in \{T, F\} \Rightarrow V_{b_2}(P) = V_{b_1}(P), \quad V_{b_1}(P) = U \Rightarrow V_{b_2}(P) \in \{U, T, F\}.$$

(Decisions persist; U can only refine.)

Axiom 11 (EF3: Probability as primitive with boundary exclusion)**.** At declared resolution, space is a resolution-finite event structure E (no infinite strictly ascending refinements), with a valuation $P : E \to \mathbb{P}^\circ$ into an ordered probability scale whose endpoints are excluded:

$$\forall e \in E : 0^\circ \prec P(e) \prec 1^\circ.$$

This primacy of probability over number echoes von Neumann's formalism, in which quantum observables are self-adjoint operators and measurement statistics are given by their spectral (projection-valued) measures—probabilities are not recovered from counting equally likely cases over infinite ensembles, but built into the theory's operational core [121]. In general, measurement outcomes are probabilistic (certainty appears only for eigenstates of the measured observable).

Likewise, de Finetti's operational view grounds probability in coherent betting behavior rather than in long-run frequencies: degrees of belief are those prices a rational finite agent is prepared to stake, constrained by Dutch-book coherence, not by limits of infinite trials [28].

Our framework radicalizes both: **probabilities neither approximate hidden deterministic values** (as ruled out for broad classes of hidden-variable models by Bell-type and Kochen–Specker no-go results [5, 75]) **nor arise from idealized infinite sequences. They are the only values available at finite resolution.** Excluding the boundaries 0 and 1 is not mere convenience: assigning 0 or 1 to contingent propositions is pragmatically perilous—Cromwell's rule warns against dogmatic priors [83], and diachronic Dutch-book results show how such certainties can destabilize updating—so finite agents should avoid them except for logical truths.

Von Neumann's framework already makes probability operationally fundamental via spectral measures and the Born rule; we extend this stance: at finite resolution all values are probabilistic, and "certainty" is an idealization effectively requiring unlimited resources to verify.[5]

Axiom 12 (Budget-free vs. budgeted primitives (classification))**.** Operations are classified by type signature:

- **READ operations** have signature $A \to B$ and require no budget parameter. These access existing structure without modification (field access, state queries).
- **WRITE operations** have signature $A \to \text{Budget} \to (B \times \text{Budget} \times \text{Heat})$ and must account for all consumed budget. For any evaluation $f_B(x) = (y, B', h)$ where $B > 0$ and work is performed: $B' + h = b$ (conservation) and $h \geq \mu$ (minimum one tick per distinction step). When $B = 0$, operations return immediately with $h = 0$, typically yielding U or exhaustion markers.

Reading stored distinctions is free (no budget parameter); creating/modifying distinctions costs budget and generates heat. This classification follows Bennett's insight that observation can be logically reversible [6], though the extension to general mathematics differs from Fredkin-Toffoli conservative logic [44]: we separate free access from costly modification, rather than making all operations reversible.

Axiom 13 (EF5: Conservation and erasure surcharge (thermodynamic law))**. Conservation:** For every evaluation $f_B(x) = (y, B', h)$,

$$B = B' \oplus h$$

[5]On the broader cultural shift: the twentieth century saw a "probabilization" of the sciences—statistics and probability permeated inquiry far beyond physics. For historical syntheses, see Hacking's *The Taming of Chance* [58] and Gigerenzer et al.'s *The Empire of Chance* [52]. Regarding Planck: he was initially reluctant to ground fundamental laws in probability and atomism, describing his 1900 turn to Boltzmann's method as an "act of desperation" [77]; yet after 1900 he increasingly acknowledged the central role of statistical entropy. My intent in *Probabilistic Aesthetics of the Avant-Gardes: Predictive Arts* was not to contribute to the biography of the greatest minds of statistical reasoning or the post-facto justification of scientific method, but a reconstruction of the probabilistic turn as a new regime of knowing—one that transformed not just how we measure or predict, but how we experience, create, and imagine at the intersection of art and science. The avant-garde, far from being a bystander, was a laboratory for experimental uncertainty, a proving ground for cognition as probability. A necessary incursion into modes of experience in the inhuman industrial environments. [127]

in the commutative resource monoid $(B, \oplus, 0_B)$.

Erasure surcharge (theoretical principle): If f erases a stored distinction, then $h \geq \mu + \epsilon$ for some $\epsilon > 0$. This implements Landauer's principle mathematically: erasing information requires strictly more than minimal work. (Current Coq implementation treats erasure as standard WRITE operation costing μ; the surcharge ϵ remains a design principle for future extensions.)

The erasure surcharge directly implements Landauer's principle: erasing information necessarily dissipates $kT \ln 2$ energy [79]. Our $\epsilon > 0$ is the mathematical analog of this physical necessity, preventing the information-theoretic paradoxes that arise in systems with free erasure.

Axiom 14 (EF6: μ as parameter (not a fixed numeric))**.** There exists $\mu > 0$ (unit of priced work). Any concrete calibration (e.g., $\mu = 1/100$) is model-level, not part of the pure axioms.

This principle is especially significant in the context of neural networks, learning systems, and memory models: every erasure or forgetting event carries an unavoidable cost, directly linking thermodynamic constraints to the architecture and scalability of artificial intelligence. Explicitly pricing irreversibility prevents illusory "free resets" of memory or knowledge, ensuring that every modification, training cycle, or synaptic update accrues a measurable computational surcharge—grounding machine learning and adaptive systems in physically meaningful, accountable dynamics.

Coq verification files:

- `void_phase_orbits.v` — Temporal evolution of patterns under budget constraints
- `void_geometry.v` — Vector operations, inner products with budget costs
- `void_geometry_basis.v` — Space as distinguishability structure, points as patterns
- `void_information_theory.v` — READ/WRITE boundary, heat generation from distinction-making
- `void_pattern_thermo.v` — Pattern decay, maintenance costs, stability conditions
- `void_thermal_convection.v` — Pattern movement driven by distinguishability checks
- `void_computing_node.v` — Observer nodes with finite capacity and resolution

4 Primitives and Notation

4.1 Minimal Ingredients for Finite Mathematics

These primitives aren't arbitrary technical choices but the minimal elements needed to build mathematics from finite observation. Each serves a specific philosophical purpose:

- **Observer state (σ):** Mathematics doesn't exist "out there" but emerges from an observer with limited resources making distinctions
- **Resource monoid (B):** Replaces infinite sets with finite budgets that actually deplete
- **Resolution (ρ):** Reality has grain-size; finer discrimination costs more
- **Probability scale (P°):** The fundamental values aren't numbers but degrees of distinguishability
- **Three-valued logic ($\mathbb{B}_3$):** Acknowledges what cannot be computed with available resources
- **Memory (M):** Stored patterns that persist or decay, costing resources to maintain

Together, these primitives constitute the minimal machinery for finite mathematics, each philosophically necessary rather than technically convenient.

4.2 Observer and State

To make the observer's data explicit, we adopt a record formulation of state rather than an anonymous tuple. This exposes, in one place, the four components that govern every step of evaluation: the available capacity (budget), the current grain of discrimination (resolution), the maintained cache of finite patterns (memory), and a finite seed for sampling from probability distributions and trace ordering. The record serves only to clarify what was already implicit in our rules; it does not add power. Every primitive in the sequel reads and updates these fields in the usual way (budget decreases on priced steps; resolution may be refined by an explicit operation; memory records or erases finite patterns; the phase counter advances with each step). This is the standard Coq idiom for records and projections, chosen here for clarity and to keep proofs and specifications compact.

Evaluation is always state-relative. We write a step as $f_S(x) = f_{(B,\rho,M)}(x) = (y, S', h)$ with $S' = (B', \rho', M')$ and emitted heat h.

(Abbreviation: when only budget changes and ρ, M stay the same, we may write $f_b(x) = (y, b', h)$.)

This makes even the simplest mathematical statement contextual: "$2 + 2 = 4$" becomes "with budget B at resolution ρ, combining these patterns yields that pattern." The observer-void S isn't a passive recorder but actively constructs mathematical reality through resource expenditure.

4.3 Resource Domain and Conservation

Resource monoid. $(B, \oplus, 0_B, \leq)$ is a commutative monoid with order $\leq$ such that the strict order $<$ is well-founded (no infinite strictly descending chains).

Conservation (global discipline). For every evaluation $f_S(x) = (y, S', h)$ we require $B = B' \oplus h$. There is no hidden work: budget turns either into residual budget or into heat.

Budget-neutral vs. priced primitives. There is a predicate priced$(\cdot)$ and a unit tick function $\mu(\rho) > 0$ such that:

- If $\neg\text{priced}(f)$ (reversible/read), then $f_{(B,\rho,M)}(x) = (y, (B, \rho, M), 0)$.
- If $\text{priced}(f)$ (irreversible/write/compute), then $f_{(B,\rho,M)}(x) = (y, (B', \rho', M'), h)$ with $B' < B$ and $h \geq \mu(\rho)$.
- Erasure surcharge: if f erases a stored distinction, then $h \geq \mu(\rho) + \epsilon$ for some $\epsilon > 0$.

The conservation law $B = B' \oplus h$ isn't just bookkeeping but thermodynamic necessity: you cannot create or destroy computational capacity, only transform it. This prevents paradoxes like the halting problem (would require infinite budget to decide) and Russell's paradox (self-reference costs more than any finite budget), similar to how Girard's linear logic tracks resource usage to avoid paradoxes [53].

Note on $\ominus$: we do not assume a global subtraction. When needed in examples, we solve $B = B' \oplus h$ for the unknown term (e.g., "the unique h with $B' \oplus h = B$"), avoiding implicit subtraction.

4.4 Resolution and Its Dynamics

Resolution poset. $(\text{Res}, \preceq)$ is a poset of resolutions. A refinement step $\text{refine}(\rho \to \rho')$ with $\rho \preceq \rho'$ is a priced operation evaluated as $\text{refine}_{(B,\rho,M)}() = (\text{ok}, (B', \rho', M), h)$ with $h \geq \mu(\rho')$. Intuitively: finer resolution is more expensive (at least as costly as its unit tick).

Monotonicity. For fixed S, discrimination power and primitive costs are monotone in ρ: if $\rho \preceq \rho'$ then $\mu(\rho) \leq \mu(\rho')$, and any decided truth at ρ persists at ρ'.

4.5 Probabilistic Space

Split principle. Budgets are bounded counters $B \in \text{Fin}$; we may express costs as multiples of the unit μ-tick, but the budget itself never exceeds MAX. Values and operations act on finite ρ-grids (no appeal to bare $\mathbb{N}$ in the value domain). This is conventional bookkeeping—not physics—yet precise enough to audit cost step by step.

Probability scale. P° is a linearly ordered scale with two ideal boundary points 0° and 1° excluded: for all events $e \in E_\rho$, $0^\circ \prec P(e) \prec 1^\circ$.

At resolution ρ, the scale is constructed as:

$$P^\circ_\rho = \left\{ \frac{n}{d} \,\middle|\, 1 \leq n < d \leq D(\rho) \right\} \subset (0, 1)$$

where $D(\rho)$ is the denominator cap at resolution ρ.

Constructibility at resolution ρ: At resolution ρ with denominator bound $D(\rho)$, the system can distinguish between pairs (n, d) where $1 \leq n < d \leq D(\rho)$. The construction budget required scales with $D(\rho)$. **Distinguishability examples:**

- $D(\rho) = 2$: Can distinguish 1 probability value
- $D(\rho) = 3$: Can distinguish 3 probability values
- $D(\rho) = 10$: Can distinguish up to 45 probability values
- $D(\rho) = 100$: Can distinguish up to 4,950 probability values

Primitive operations. Partial operations on P° include $\oplus_p$ for disjoint sum, $\otimes_p$ for independent product, and $\oslash_p$ for division, each commutative, associative (where defined), and monotone. Operations may return degraded results, intervals, or U when:

1. Result denominator exceeds $D(\rho)$ (exceeds resolution capacity)
2. Result approaches or reaches boundaries 0° or 1° (touches excluded endpoints)
3. Semantic preconditions fail (non-disjoint events for $\oplus_p$, dependent events for $\otimes_p$)

The FinProb type permits computations that touch boundaries (including exact zero from subtraction or division), but probabilistic interpretation excludes these endpoints. Application code must handle boundary cases appropriately for the domain.

Detailed operation mechanics, examples, and thermodynamic costs are specified in Section 5.5 (Arithmetic on the Probability Scale). Geometric applications appear in Section 6 (geometry without Infinity).

Valuation. A probability valuation at resolution ρ is a map $P_\rho : E_\rho \to P^\circ$ respecting $\oplus_p/\otimes_p$ on the domain where those event compositions are declared. Numeric coordinatizations (e.g., ratios) are conventional encodings of values in P° and are not part of the primitives.

Operating in P° rather than $\mathbb{R}$ reflects that every measurement, every distinction, carries irreducible uncertainty. The excluded boundaries mean we never achieve perfect knowledge (1°) or complete ignorance (0°)—both would require infinite resources to verify.

Coq verification: `void_probability_minimal.v` (FinProb as (n, d) pairs, boundary exclusion), `void_arithmetic.v` (division module `Void_Probability_Division`), `void_distinguishability.v` (resolution-bounded scales, thresholds), `void_pattern.v` (event structures, valuations).

4.6 Distinguishability and Perception

The observer's distinguishability kernel at resolution ρ is a map $\Delta_S : X \times X \to P^\circ_\rho$ assigning a probabilistic separation to pairs of stimuli/patterns. Sub-threshold cases induce U in derived comparisons at budget B.

The kernel Δ_S captures the fundamental act: telling things apart. When $\Delta_S(x, y)$ falls below the threshold affordable with budget B, x and y become indistinguishable—not "similar" but literally the same to the observer. This is how quantum superposition emerges: states that classical logic would separate become unified when discrimination is unaffordable.

Formal properties:

1. **Observer-relativity:** Δ_S depends on full state $S = (B, \rho, M)$
2. **Probabilistic values:** Results in open interval $(0, 1)$, excluding 0° and 1°
3. **Monotonicity:** If $(B_1, \rho_1) \preceq (B_2, \rho_2)$ then $\Delta_S^{(\rho_1, B_1)}(x, y) \preceq \Delta_S^{(\rho_2, B_2)}(x, y)$
4. **Subthreshold collapse:** When $\Delta_S(x, y) < \theta(\rho)$, predicates distinguishing x from y return U
5. **Asymmetry:** $\Delta_S(x, y)$ may differ from $\Delta_S(y, x)$ when discrimination costs differ by approach

Unlike classical metrics, Δ_S violates symmetry and triangle inequality. This reflects thermodynamic reality: distinguishing A from B may cost differently than B from A (asymmetry); going directly from A to C may be cheaper than via intermediate B (triangle violation). These aren't approximations—they're honest accounting of discrimination costs. **Examples:**

Asymmetry: Expert distinguishing genuine art from forgery ($\Delta(\text{genuine}, \text{fake})$ high) versus fake from genuine when primed for authenticity ($\Delta(\text{fake}, \text{genuine})$ lower). Direction matters.

Resolution dependence: Two colors identical at twilight ($\Delta^{(\rho_1,B)} < \theta$), distinct in daylight ($\Delta^{(\rho_2,B)} \geq \theta$). Same stimuli, different capacity.

Budget dependence: Distinguishing similar patterns costs $5\mu(\rho)$; with $B = 3\mu(\rho)$ they collapse, with $B' = 10\mu(\rho)$ they separate. Distance changes because resources change.

Triangle violation: Three classes where direct test $[a] \to [c]$ costs $\mu(\rho)$, but via $[b]$ costs $5\mu(\rho)$. Direct path is cheaper than composite.

The kernel relates to but differs from metric spaces (Fréchet [43]), similarity measures (Tversky [118]), fuzzy metrics (Kramosil [78]), and partial metrics (Matthews [88]). The innovation: observer state (B, ρ, M) is constitutive—separation doesn't exist independently of resources required to establish it.

Full development of distinguishability-based geometry appears in Section 5 (Perception, Patterns, and Information) where the kernel generates quotient spaces $C_{(\rho,B)} = X/\sim_{(\rho,B)}$, and in Section 6 (geometry without Infinity) where perceptual distance d^{perc} and separation cost sep_θ emerge as complementary metrics.

The formalization proves constructively that non-metric distinguishability provides consistent geometric foundation—what was claimed impossible now demonstrated as working mathematics with explicit resource accounting.

Coq verification files:

- `void_distinguishability.v` — Core kernel Δ_S, threshold collapse, monotonicity axioms; concrete operations implementing Δ_S formalized
- `void_pattern.v` — Quotient space construction $C_{(\rho,B)}$, equivalence by indistinguishability
- `void_geometry_basis.v` — Distinguishability-based distance, shape fields, point-pattern correspondence
- `void_probability_minimal.v` — Probabilistic return values P°_ρ, boundary exclusion

4.7 Three-Valued Truth

Truth values at state $(B, \rho, \cdot)$ are $\mathbb{B}_3 = \{T, F, U\}$ with $U =$ "Unknown at current capacity/resolution."

Monotone refinement: if $b_1 \leq b_2$ and $\rho \preceq \rho'$, then any decided value (T/F) at (b_1, ρ) persists at (b_2, ρ'), while U at (b_1, ρ) may refine to $U/T/F$ at (b_2, ρ'). There is no forced collapse without resources.

Unlike Kleene's U (undefined due to semantic failure) or Łukasiewicz's intermediate values (degrees of truth) [84], our U marks a precise epistemic boundary: the statement has a definite truth value that we cannot afford to determine. This isn't agnosticism but "thermodynamic" poverty.

When a chess engine returns U for a position, it's not saying "this position has no value" but "evaluating this would exceed my search budget." The monotone refinement property ensures logical consistency: truths discovered cheaply remain true with more resources, but U can resolve either way. This models how exhausted reasoners stop computing rather than guess—a financier with depleted analytical budget doesn't randomly assign stock values but marks them "unanalyzed." The open-world assumption means U doesn't infect all downstream computations (as in Kleene logic) but localizes undecidability to specific resource boundaries.

4.8 Memory and Pattern Dynamics

Memory M stores constructed distinctions/patterns.

- read: $\text{read}_{(B,\rho,M)}(\text{key}) = (\text{value}, (B, \rho, M), 0)$ (budget-neutral).
- write: $\text{write}_{(B,\rho,M)}(\text{patt}) = (\text{ok}, (B', \rho, M \cup \{\text{patt}\}), h)$ with $h \geq \mu(\rho)$ (priced).
- erase: $\text{erase}_{(B,\rho,M)}(\text{patt}) = (\text{ok}, (B', \rho, M \setminus \{\text{patt}\}), h)$ with $h \geq \mu(\rho) + \epsilon$ (priced, surcharge).

Pattern-level transforms (match/rewrite) are priced and obey conservation like all primitives.

The asymmetry (reading is budget-neutral, writing costs $\mu(\rho)$, erasing costs $\mu(\rho) + \epsilon$) implements Landauer's principle at the mathematical level. This explains why forgetting is harder than remembering: erasure must dissipate information into heat, while reading leaves patterns intact.

4.9 Time as Ledger

For a finite evaluation trace $\text{tr} : S_0 \to S_1 \to \cdots \to S_n$ with heats $h_1, \ldots, h_n$, define:

$$T(\text{tr}) = h_1 \oplus \cdots \oplus h_n$$

Ledger identity: $B_0 = B_n \oplus T(\text{tr})$ by iterated conservation.

Time is not a coordinate we consult but the integrated heat of work done. This inverts classical temporality: Newton's absolute time [92] flows uniformly regardless of events; Einstein's spacetime [32, 33] treats time as geometric parameter. Here, time is *produced* by operations rather than presupposed as background. A system performing no priced work experiences no time—the identity transformation has $T(\text{id}) = 0$ because no distinctions are created or destroyed.

Example: Computing $3 + 4 = 7$ with budget $B_0 = 20\mu(\rho)$:

$$\begin{gathered} \text{add}(3, 4, 20\mu) = (7, 16\mu, 4\mu) \quad [\text{4 recursive steps}] \\ 20\mu = 16\mu \oplus 4\mu \quad [\text{conservation}] \\ T(\text{tr}) = 4\mu \quad [\text{elapsed time}] \end{gathered}$$

The addition "took" 4 time units because it dissipated $4\mu(\rho)$ heat through recursion. Passive read would take zero time ($h = 0$). Time measures irreversible change, not merely change.

The arrow of time: Reversibility requires $T(\text{tr}) = 0$, which holds if every operation is budget-neutral (reading existing information). Any priced operation—creating distinctions, erasing patterns, computing values—dissipates heat and produces time. This grounds Eddington's arrow of time [31] thermodynamically: time flows toward accumulated dissipation because priced operations are strictly irreversible ($h \geq \mu(\rho) > 0$).

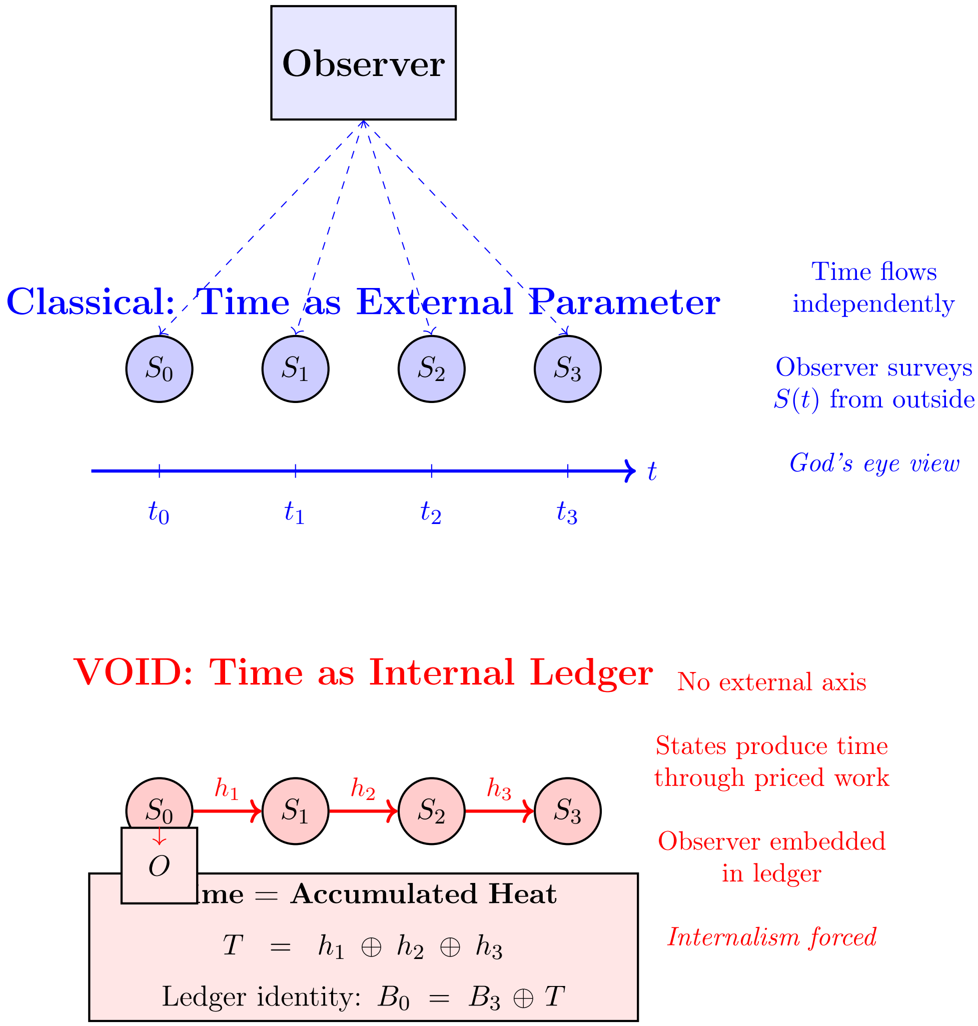


Figure 4: **Time: External Parameter vs Internal Ledger. Top:** Classical view treats time t as independent coordinate indexing states $S(t)$, with observer surveying from outside. **Bottom:** VOID view has no external time axis—states generate time through priced operations dissipating heat. Time is the ledger $T = \bigoplus h_i$, observer is embedded within, and the conservation law $B_0 = B_n \oplus T$ closes internally. The conceptual inversion is total.

This connects to Prigogine's identification of time with irreversible processes [99] and Rovelli's thermal time hypothesis [101], but operationalizes it: time isn't philosophical interpretation but direct mathematical consequence. The ledger $T(\mathrm{tr})$ is measurable—the sum of heat emissions in the computation trace.

Note on $\mu(\rho)$: The parameter $\mu(\rho) > 0$ lower-bounds priced operations at resolution ρ; complex operations cost multiple μ. The monoid operation $\oplus$ accumulates actual dissipated heat—no separate summation over μ required.

Full temporal dynamics appear in Section 7 (budget flows, entropy accumulation, predictive maintenance) and Section 8 (memory decay, refresh costs).

Coq verification files:

- `void_time_memory_composition.v` — Temporal traces, ledger identity
- `void_budget_flow.v` — Heat accumulation
- `void_entropy.v` — Time as integrated dissipation
- `void_finite_minimal.v` — Heat conservation axioms

4.10 Numerals and Counters

Fin constructors (fz, fs, fs fs...) serve as structural tokens in procedures; they do not supply the grain of reality. Granularity arises from the pair (B, ρ) and the observer's evaluation, not from these syntactic markers.

Fin undergoes a role reversal: these constructors aren't mathematical numbers but computational structure—bounded tokens that organize evaluation steps. We never compute $2 + 2 = 4$ as arithmetic on numbers; instead, patterns with strength encoded as FinProb pairs combine through budgeted operations, costing 1 tick per distinction step. The Fin tokens structure these computations without being numeric objects.

Standard mathematics constructs $\mathbb{R}$ from $\mathbb{N}$ via Dedekind cuts. We reject this entire tower. FinProb pairs are primitive; numeric notation like "1/2" is merely how humans read the pair structure, like Wittgenstein's ladder we must eventually discard.[6]

4.11 Complexity

In VOID, complexity means explicitly counting how much of the resource monoid is used, not just estimating rates or asymptotics.[7] Fix a resolution ρ and a tick $\mu(\rho)$; every priced primitive operation consumes at least one tick. So, if an algorithm makes n such operations, its complexity is at least $n \cdot \mu(\rho)$, with all constants kept visible and open to audit—in contrast to traditional big-O notation, where constants are ignored.

Monotonicity. If $\rho' \leq \rho$ (coarser), then $\mu(\rho') \leq \mu(\rho)$. Finer resolutions are costlier: increasing precision raises the unit tick price, ensuring that higher detail always consumes more budget.

[6] Wittgenstein, *Tractatus Logico-Philosophicus* 6.54: "My propositions are elucidatory in this way: he who understands me finally recognizes them as senseless, when he has climbed out through them, on them, over them. (He must so to speak throw away the ladder, after he has climbed up on it.) He must surmount these propositions; then he sees the world rightly."

[7] In Coq, $\mu(\rho)$ is the unit priced tick; the symbol here is expository.

Concrete examples. Here n, m, k are step counts used for *budgeting*; the computation proper lives on finite ρ-grids.

- **Addition:** In the unit-decrement scheme on m,

$$\mathrm{add}(n, m) \text{ debits } m\,\mu(\rho).$$

  E.g. 3+5 debits $5\,\mu(\rho)$.

- **Multiplication:** As repeated addition,

$$\mathrm{mult}(n, m) \text{ has baseline debit } (n \cdot m)\,\mu(\rho),$$

  with any per-loop overhead counted as an explicit constant (not hidden).

- **Division:** Implemented as repeated subtraction. For example, $23 \div 5$ requires four successful subtractions plus the terminal comparison:

$$H_{\mathrm{div}} \in [\,4,\ 4{+}c\,]\,\mu(\rho),$$

  where c depends on the chosen primitives (comparison/branch costs).

- **Quicksort:** Let k be the number of items. With comparison-only pricing, the *expected* debit is

$$\mathbb{E}[H_{\mathrm{sort}}] = \alpha(\rho)\,k \log_2 k \cdot \mu(\rho),$$

  and the worst-case debit is

$$H_{\max} = \alpha'(\rho)\,k^2 \cdot \mu(\rho),$$

  where $\alpha(\rho)$ and $\alpha'(\rho)$ are constants determined by the primitive cost model (never hidden). If swaps or moves are priced, a linear term with its own explicit coefficient is added. The point is not asymptotics but *auditable consumption*: each comparison and move burns measurable budget.

Why counted constants matter. Classically one writes $O(n \log n)$ and discards factors; asymptotics erase reality. In VOID, constants are first-class: a factor 1000 burns budget 1000 times faster than a factor 1. Complexity is *consumption* in the same units as available budget.

Exhaustion handling. Let B_{req} be the counted requirement under the chosen primitives.

1. **Success:** terminates with $B_{\mathrm{final}} > 0$; returns (result, B_{final}).
2. **Exhaustion:** if the budget counter reaches 0 during execution, return U with a trace: "required B_{req}; exhausted at step k".
3. **Partial result:** if supported, return a degraded output (coarser ρ-grid / lower resolution) together with the remaining budget.

Operation composition. Debits add: for a composition $f \circ g$

$$H_{f \circ g} \;=\; H_g \,\oplus\, H_f = H_f \,\oplus\, H_g,$$

where $H_{\bullet}$ is the total debit (in ticks). Heat accumulation is deterministic and commutative in the resource monoid. While pattern strengths and operation outcomes are probabilistic (values in P°_{ρ}), the thermodynamic cost of executing operations is deterministic—you spend exactly $\mu(\rho)$

per priced step. For algorithms with probabilistic branching (selecting among strategies based on pattern probabilities), the *expected* heat $\mathbb{E}[H]$ depends on the probability distribution over execution paths. No cancellation, no free lunch—each step contributes to the sum.

This yields *tick-precise* accounting. A sorter is not "$O(n \log n)$" but "debits $n \log_2 n$ (with an explicit factor) units of irreversible budget." In asymptotic theory constants vanish; in resource accounting every tick counts because the budget actually depletes.

Split principle. Budgets are whole-number counters $B \in \mathbb{N} \cdot \mu(\rho)$; values/operations act on finite ρ-grids (no appeal to bare $\mathbb{N}$ in the value domain). This is conventional bookkeeping—not physics— yet precise enough to audit cost step by step.

Coq verification files:

- `void_finite_minimal.v` — Well-founded capacity order, base axioms
- `void_probability_minimal.v` — Probability scale P°, boundary exclusion
- `void_arithmetic.v` — Operations on FinProb ($\oplus_p$, $\otimes_p$, $\oslash_p$), division module
- `void_pattern.v` — Pattern construction, matching, monotone refinement
- `void_information_theory.v` — Read/write/erase asymmetry, Landauer principle
- `void_entropy.v` — Heat accumulation,
- `void_time_memory_composition.v` — Temporal traces, ledger identity, memory dynamics
- `void_distinguishability.v` — Distinguishability kernel, threshold effects

4.12 Meta-theoretical Properties

The preceding sections have defined VOID's core types (Fin, Budget, Heat), three-valued logic, operations with resource accounting, and complexity measures grounded in tick-precise consumption. We have specified how every operation debits budget, generates heat, and respects strict conservation laws. These are not informal descriptions but machine-verified definitions spanning 33 Coq modules.

Having established the operational machinery, we now prove that this system is formally sound and complete:

Theorem 4.1 (VOID is Constructively Complete)**.** *For finite observer with budget B and resolution ρ:*

1. *Every decidable question has answer in $\{\texttt{BTrue}, \texttt{BFalse}, U\}$ within B computational steps*
2. *Every operation either completes with result, or exhausts budget and returns U—no hanging, no infinite loops*
3. *Heat generation satisfies strict conservation: for every operation returning $(result, B', H)$, we have $H \oplus B' = B$ where $\oplus$ is `add_heat`*
4. *All operations have bounded runtime: maximum steps determined by initial budget B (no operation can consume more than available budget)*

Proof. By construction of the Coq formalization.

(1) All operations in `void_finite_minimal.v` through `void_metaprobability.v` have type signatures guaranteeing termination: they match on bounded inductive types (`Fin`, `Budget`) that must bottom out at `fz`. Three-valued logic handles exhaustion explicitly via pattern match on budget.

(2) Every recursive function decreases either iteration counter or budget on each call. When both reach `fz`, function returns. Type system ensures no infinite recursion.

(3) Conservation axioms appear in `void_finite_minimal.v` (lines 505–508) and also in `void_arithmetic.v` (lines 249–259). These are foundational axioms enforced by type system, defining the algebraic structure of the resource monoid.

(4) Runtime bound follows from (1) and (2): maximum steps is initial budget value, which is itself bounded by MAX. Each operation consumes at least one u-tick. Therefore, no operation can take more than $B \leq \text{MAX}$ steps when starting with budget B. □

Theorem 4.2 (Conservation of Computational Resources)**.** *For every atomic operation $f : A \to \mathit{Budget} \to (A \times \mathit{Budget} \times \mathit{Heat})$ and every initial budget B, if $f(x, B) = (y, B', h)$, then:*

$$B = B' \oplus h$$

in the resource monoid.

Proof. By structural induction on the recursive definition of operations in `void_arithmetic.v`. Base case: If f performs no work (returns immediately), then $h = \text{fz}$ and $B' = B$. Equation holds: $B = B \oplus \text{fz}$. Inductive step: Every recursive call consumes exactly one unit of budget (μ) and adds it to the heat accumulator. Since addition in Fin is associative and commutative (proven in Coq standard library for Peano-like structures), the sum of remaining budget and accumulated heat remains invariant at every step of the recursion. □

5 Perception, Patterns, and Information

5.1 Modeling Framework and Scope

This section formalizes perception, patterns, and information within budgetary constraints.

We define the scope of this model by shifting the fundamental unit of analysis from the static *object* to the executing *program*. Drawing on the operational logic of Vilém Flusser's combinatorial anthropology [41], we treat reality not as a collection of inert entities, but as a complex of interlocking protocols that determine behavior. Within this framework, phenomena ranging from the biological lottery of DNA to cultural scripts and economic subroutines are modeled as functional inputs—code that executes within a finite environment.

VOID Theory serves as the formal architecture for this scope. By rejecting the "legacy code" of infinite precision, we provide an alternative operating system where mathematical existence is defined by **executability** rather than idealization. The model restricts itself to what can be constructed and maintained by a finite observer. Consequently, we enforce a transition from **corpuscular inertial** ontology (static, independent existence of things in space that unintentionally lacks definition) to **pattern-oriented** dynamics (processual, cost-dependent maintenance of space and entities that constitute it). A pattern, in this model, is defined as a boundary entity: a distinction actively maintained against entropy through budgeted operations.

Section structure:

- Perception as distinguishability with budget cost (§5.1)
- Patterns as maintained distinctions requiring budget (§5.2)
- Information operations with read/write asymmetry (§5.3)

5.2 Perception $\equiv$ Distinguishability

Perception is construction. A distinction is not discovered and filed; it is made, at cost. When capacity is low or resolution coarse, the system returns Unknown rather than a counterfeit decision.

Provable Real-Time Termination: In safety-critical systems (avionics, medical devices), unbounded loops are fatal. VOID guarantees that every operation respects a hard budget constraint. Instead of hanging or missing a deadline, the system returns explicit U, allowing for fail-safe default behaviors rather than undefined states.

Formal Framework:

Definition 5.1 (Resolution-bounded probability scale)**.**

$$P_\rho^\circ = \{n/d \mid 0^\circ \prec n/d \prec 1^\circ,\ d \leq D(\rho)\}$$

where $D(\rho)$ is the denominator cap at resolution ρ. The boundaries 0° and 1° are excluded—they mark the unattainable void and totality.

Definition 5.2 (Distinguishability kernel)**.** At state $S = (B, \rho, M)$, the kernel

$$\Delta_S^{(\rho,B)} : X \times X \to P_\rho^\circ \cup \{\mathsf{U}\}$$

assigns to each pair (x, y) the separation achieved by executing affordable tests within budget B.

Definition 5.3 (Distinguishability threshold)**.** The threshold $\theta(\rho) \in P_\rho^\circ$ marks minimum affordable distinguishability at resolution ρ. Below this, discrimination costs exceed available budget.

Definition 5.4 (Equivalence by indistinguishability)**.**

$$x \sim_{(\rho,B)} y \quad \text{iff} \quad \Delta_S^{(\rho,B)}(x,y) \prec \theta(\rho)$$

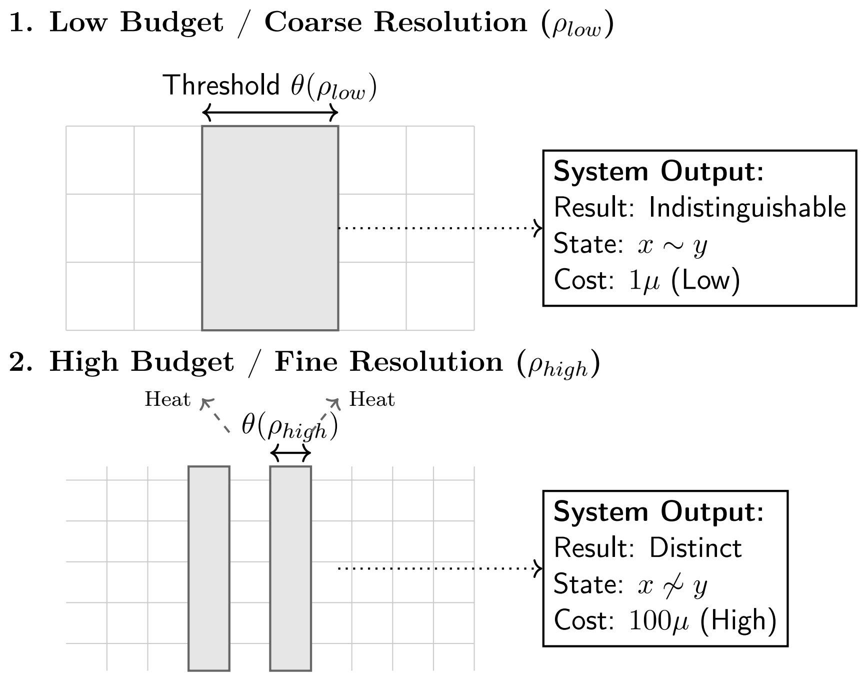


Figure 5: **Perception as a Function of Budget.** Two underlying states x and y are fixed. **Top:** At coarse resolution (low budget), the separation distance falls below the threshold $\theta(\rho)$, forcing the system to treat them as equivalent ($x \sim y$). **Bottom:** Paying for higher resolution shrinks $\theta(\rho)$, allowing the distinction to be registered. Distinction is not a property of the objects, but of the affordable interaction.

Axiom 15 (Conservation in perception)**.** Running any test t obeys:

$$B = B' \oplus h$$

where $h \succeq \mu(\rho)$ for priced tests, $h = 0$ for budget-neutral reads.

Remark 1 (Interference)**.** *When patterns* p_1, p_2 *share discrimination tests,* $\text{Interfere}(p_1, p_2)$ *produces superposition—quantum-like behavior emerging from budget allocation, not wave mechanics.*

Axiom 16 (Monotonicity)**.** If $(B_1, \rho_1) \preceq (B_2, \rho_2)$, then:

$$\Delta_S^{(\rho_1,B_1)}(x,y) \preceq \Delta_S^{(\rho_2,B_2)}(x,y)$$

More capacity never reduces separability—but always costs.

Axiom 17 (Subthreshold collapse)**.** Below threshold, distinctions vanish. Any predicate on collapsed pairs returns U.

Example (Frequency discrimination): Two tones at 440.0 Hz and 440.8 Hz. With $D(\rho) = 10$, we have $P_\rho^\circ = \{1/10, \ldots, 9/10\}$.

- At $B = 5\mu(\rho)$: short-window test yields sep $= 2/10 < \theta(\rho) = 3/10 \rightarrow$ tones collapse
- At $B = 50\mu(\rho)$: long-window test yields sep $= 7/10 > \theta(\rho) \rightarrow$ tones separate
- Ledger shows heat $= 45\mu(\rho)$ paid for the discrimination

The difference wasn't "discovered"—it was funded into existence.

5.3 Patterns—Persistent Distinctions Under Budget

Patterns are not eternal forms waiting to be discovered but active maintenances that cost resources to sustain. Think of a whirlpool in a stream [99]: it appears stable but exists only through continuous flow of water and energy. Stop the flow, the pattern dissolves. Similarly, our patterns—from "chair" to "electron"—persist only while we pay to maintain the distinctions that constitute them. This reverses the usual metaphysics: stability is expensive; dissolution is free.

Definition 5.5 (Pattern space)**.** Stimuli x, y are equivalent when $\Delta_S^{(\rho,B)}(x, y) < \theta(\rho)$. Patterns live on the quotient:

$$C_{(\rho,B)} = X/{\sim_{(\rho,B)}}$$

where X is stimulus space. Each equivalence class $[x]$ is a pattern—stimuli indistinguishable at current budget and resolution.

Definition 5.6 (Pattern structure)**.** A pattern p consists of:

- location: equivalence class in $C_{(\rho,B)}$
- strength: probability in P_ρ° of persistence
- maintenance_cost: $\mu(\rho) \times \text{size}(p)$ per time unit

Axiom 18 (Formation with conservation)**.** Every pattern operation obeys:

$$B = B' \oplus h, \quad \text{where } h \geq \mu(\rho) \text{ for new distinctions}$$

Axiom 19 (Partiality under exhaustion)**.** If required spend exceeds B, operations return U—genuinely unknown, not guessed.

Axiom 20 (Persistence)**.** Features established at (B, ρ) persist at any (B', ρ') where $(B, \rho) \preceq (B', \rho')$.

Operations:

- $\text{match}(\text{pattern}, \text{input}) \rightarrow \{T, F, U\}$ costs $\mu(\rho)$ per comparison
- $\text{compose}(p_1, p_2) \rightarrow p_3$ costs $\max(\mu(\rho_1), \mu(\rho_2))$
- split (refinement) costs $\mu(\rho)$
- merge (coarsening via erasure) costs $\mu(\rho) + \epsilon$

The algebra of patterns is a discipline of attention: spend where structure repeats, compress where regularities hold, accept that novelty costs more.

5.4 Arithmetic as Thermodynamic Process

We now enter the engine room of the theory. In classical mathematics, arithmetic operations are instantaneous logical relations: $2 + 2 = 4$ is an eternal truth that costs nothing to assert. In VOID, arithmetic is a **physical process of assembly**. Every addition, multiplication, or comparison is a micro-event that consumes budget, generates heat, and transforms the structural entropy of its operands.

This section introduces a radical departure from standard operational semantics. Since `FinProb` represents general rationals rather than idealized real numbers, we do not treat values merely as magnitudes (where 12/6 is identical to 2). Instead, we treat them as **historical artifacts**. The fraction 12/6 carries the structural memory of its composition; reducing it to 2 is not a trivial identity but a destructive act of erasure that requires additional energy.

Consequently, our arithmetic operations are:

- **Bounded:** They succeed only when the resulting structural complexity (denominator size) remains within the resolution constraint $D(\rho)$.
- **Priced:** Cross-multiplication and renormalization are not free algebraic steps but budgeted operations costing μ-ticks.
- **Lazy:** The system defaults to retaining complexity (large denominators) rather than paying the thermodynamic cost of simplification (GCD computation), unless explicitly funded to do so.

What follows is the algebra of this accountability.

Notation: Throughout this section, we write μ for `operation_cost` (one tick) for brevity. All costs are expressed as multiples of this distinction unit.

5.5 Arithmetic as Thermodynamic Process

We now enter the engine room of the theory. In classical mathematics, arithmetic operations are instantaneous logical relations: $2 + 2 = 4$ is an eternal truth that costs nothing to assert. In VOID, arithmetic is a **physical process of assembly**. Every addition, multiplication, or comparison is a micro-event that consumes budget, generates heat, and transforms the structural entropy of its operands.

This section introduces a radical departure from standard operational semantics. Since `FinProb` represents general rationals rather than idealized real numbers, we do not treat values merely as magnitudes (where 12/6 is identical to 2). Instead, we treat them as **historical artifacts**. The fraction 12/6 carries the structural memory of its composition; reducing it to 2 is not a trivial identity but a destructive act of erasure that requires additional energy.

Consequently, our arithmetic operations are:

- **Bounded:** They succeed only when the resulting structural complexity (denominator size) remains within the resolution constraint $D(\rho)$.
- **Priced:** Cross-multiplication and renormalization are not free algebraic steps but budgeted operations costing μ-ticks.
- **Lazy:** The system defaults to retaining complexity (large denominators) rather than paying the thermodynamic cost of simplification (GCD computation), unless explicitly funded to do so.

What follows is the algebra of this accountability.

5.5.1 Disjoint Sum ($\oplus_p$)

For events e_1, e_2 with probabilities $p_1 = n_1/d_1$ and $p_2 = n_2/d_2$:

$$p_1 \oplus_p p_2 = \frac{n_1 d_2 + n_2 d_1}{d_1 d_2}$$

provided $d_1 d_2 \leq D(\rho)$ (denominator within resolution bound).

If the denominator exceeds $D(\rho)$, the operation returns U at resolution ρ.

Example: At ρ with $D(\rho) = 100$, combining patterns with strengths 3/10 and 2/5 yields:

$$\frac{3}{10} \oplus_p \frac{2}{5} = \frac{3 \cdot 5 + 2 \cdot 10}{10 \cdot 5} = \frac{35}{50} = \frac{7}{10}$$

This succeeds because denominator $50 \leq 100$.

Why this is hard: Adding fractions with different denominators requires four operations: multiply n_1 by d_2, multiply n_2 by d_1, add numerators, multiply denominators. What looks like "one addition" is actually four multiplications and one addition—each costing $\mu(\rho)$. Total heat: $5\mu(\rho)$ for a single $\oplus_p$ operation.

When denominators match, addition is cheaper: $2/5 \oplus_p 1/5 = 3/5$ costs only $2\mu(\rho)$ (one addition, one budget update). The system pays for cross-multiplication only when necessary.

5.5.2 Independent Product ($\otimes_p$)

For independent events with probabilities $p_1 = n_1/d_1$ and $p_2 = n_2/d_2$:

$$p_1 \otimes_p p_2 = \frac{n_1 n_2}{d_1 d_2}$$

provided $d_1 d_2 \leq D(\rho)$. Independence means the occurrence of one event doesn't affect the other's probability.

Example: Two independent tests each succeeding with probability 3/4:

$$\frac{3}{4} \otimes_p \frac{3}{4} = \frac{9}{16}$$

Note that when both inputs represent probabilities in $(0, 1)$, their product remains in $(0, 1)$. However, FinProb permits general rationals: $(3/2) \otimes_p (4/3) = 12/6 = 2$, a valid rational exceeding 1.

5.5.3 Division ($\oslash_p$)

Division answers: "Given that event B occurred, what's the probability of $A \cap B$?" Formally, for $p_1 = n_1/d_1$ and $p_2 = n_2/d_2$:

$$p_1 \oslash_p p_2 = \frac{n_1 d_2}{d_1 n_2}$$

Constraints:

1. Division by zero cannot occur: Zero is the unattainable vanishing point and cannot be constructed as a divisor. The `avoids_zero` property ensures $n_2 > 0$ for all valid probability values.

2. When result denominator $d_1 \cdot n_2 > D(\rho)$: Exceeds resolution bound, operation returns U or interval.

3. For conditional probability semantics: Events must satisfy appropriate independence or conditional probability requirements.

Division is more delicate than sum or product because it can push values toward boundaries or beyond resolution limits.

Example: $(1/2) \oslash_p (1/100) = (1 \cdot 100)/(2 \cdot 1) = 100/2 = 50/1$—the result represents 50, far exceeding 1 and thus moving outside probability interpretation while remaining a valid FinProb rational.

5.5.4 Subtraction with Saturation ($\ominus_p$)

For probabilities $p_1 = n_1/d_1$ and $p_2 = n_2/d_2$:

$$p_1 \ominus_p p_2 = \frac{\max(n_1 d_2 - n_2 d_1, 0)}{d_1 d_2}$$

Unlike real subtraction which can go negative, this saturates at zero: when $n_1 d_2 < n_2 d_1$, the numerator becomes `fz`, yielding result $(0, d_1 d_2)$ representing zero.

Pattern vanishing: Zero is the vanishing point of the system. When subtraction produces zero, the pattern has exhausted itself and ceased to exist. This is not an error but pattern death—subtraction is the only operation that can annihilate patterns. A probability of zero means the pattern no longer exists in the distinguishability space.

Example: $1/5 \ominus_p 1/4$ computes $(1 \cdot 4 - 1 \cdot 5)/(5 \cdot 4) = -1/20$, which saturates to $0/20$—the pattern vanished.

5.5.5 Comparison Operations

Checking equality or ordering costs budget through cross-multiplication:

Equality: Does $p_1 = p_2$? For n_1/d_1 and n_2/d_2, compute $n_1 \cdot d_2$ and $n_2 \cdot d_1$, compare results. Costs $3\mu(\rho)$ (two multiplications, one comparison).

Less-than: Is $p_1 < p_2$? Same cross-multiplication, costs $3\mu(\rho)$.

Even asking "are these equal?" requires work. There are no free comparisons—telling things apart always costs.

5.5.6 Thermodynamic Cost Structure

All arithmetic operations cost budget. Every call to `add_prob_heat`, `mult_prob_heat`, or comparison generates heat proportional to the computational work performed. There are no "free" arithmetic operations.

The type system distinguishes:

- **READ operations** (no budget parameter): Access existing structure without computation (e.g., field access: `location p`, `strength p`, state queries).

- **WRITE operations** (require budget): Perform computation and generate heat (e.g., arithmetic, comparisons, pattern operations).

Type signature reveals cost: if a function takes a `Budget` parameter and returns `(Result * Budget * Heat)`, it costs resources. Every addition invokes the same `add_prob_heat` function that generates heat—the exact cost depends on operand sizes, not on whether inputs are "stored" or "measured"—the computation itself generates heat.

5.5.7 Structural Entropy: The Cost of Simplification

Classical arithmetic operates on the axiom of extensionality: values are defined solely by their magnitude, making 12/6 and 2 identical. VOID rejects this equivalence on thermodynamic grounds. In a resource-bounded system, a value is not just a magnitude but a **history of its composition**. The fraction 12/6 carries *structural entropy*—it encodes the fact that it was assembled from specific interactions.

To reduce 12/6 to 2 requires computing the Greatest Common Divisor (GCD) and dividing out the terms. This is a destructive operation: it constitutes the erasure of structural information (the specific ratio of inputs). Following Landauer's principle, such erasure is thermodynamically expensive ($h \geq \mu + \epsilon$). Therefore, the system defaults to **lazy retention**: values naturally accumulate structural complexity (e.g., growing to 20000/10000) rather than collapsing to ideal integers. Ideally simplified integers are high-energy states of artificial purity, achieved only by paying the cost to destroy historical metadata.

5.6 Complete Budgeted Operation Example

To see the full thermodynamic accounting in action, here is a probability addition with every tick tracked:

Operation: $\frac{3}{10} \oplus_p \frac{2}{5}$ with initial budget $B_0 = 200\mu(\rho)$.

Execution trace:

1. Check if denominators equal: $10 \stackrel{?}{=} 5 \to$ FALSE
   - Cost: 6μ (comparison recurses $\min(10, 5) + 1 = 6$ times)
   - Remaining: $B_1 = 194\mu$
   - Heat generated: $h_1 = 6\mu$
2. Denominators differ, so cross-multiply: $n_1 \times d_2 = 3 \times 5$
   - Multiplication via repeated addition: $3 + 3 + 3 + 3 + 3 = 15$
   - Cost: 15μ (5 additions of 3 ticks each)
   - Remaining: $B_2 = 179\mu$
   - Heat generated: $h_2 = 15\mu$
3. Cross-multiply: $n_2 \times d_1 = 2 \times 10$
   - Multiplication via repeated addition: $2 + 2 + \cdots + 2$ (10 times) $= 20$
   - Cost: 20μ (10 additions of 2 ticks each)
   - Remaining: $B_3 = 159\mu$
   - Heat generated: $h_3 = 20\mu$
4. Add cross-products: $15 + 20 = 35$

- Cost: 20μ (addition costs second operand)
- Remaining: $B_4 = 139\mu$
- Heat generated: $h_4 = 20\mu$

5. Multiply denominators: 10×5
   - Multiplication via repeated addition: $10 + 10 + 10 + 10 + 10 = 50$
   - Cost: 50μ (5 additions of 10 ticks each)
   - Remaining: $B_5 = 89\mu$
   - Heat generated: $h_5 = 50\mu$

Final result:

$$\text{add\_prob\_heat}\left(\frac{3}{10}, \frac{2}{5}, 200\mu\right) = \left(\frac{35}{50}, B' = 89\mu, h = 111\mu\right)$$

Conservation check:

$$B' \oplus h = 89\mu + 111\mu = 200\mu = B_0 \quad \checkmark$$

Total heat: $h = h_1 + h_2 + h_3 + h_4 + h_5 = 6 + 15 + 20 + 20 + 50 = 111\mu$.

What this shows: Every operation returns the triple $(result, B', h)$. The budget never disappears—it either remains available (B') or has been dissipated as heat (h). The conservation law $B = B' \oplus h$ holds exactly, not approximately.

Notice how the costs scale: multiplication isn't uniform but depends on operand sizes. 3×5 costs less than 10×5 because it's implemented as repeated addition. This is thermodynamically honest arithmetic: you can see precisely where every tick went, and why some operations are more expensive than others.

The simplified result $\frac{35}{50} = \frac{7}{10}$ after reduction, but reduction itself would cost additional budget (requiring GCD computation). The system returns the unsimplified fraction—simplification is an optional post-processing step that the caller can choose to pay for.

5.7 Budget Exhaustion and Recovery

What happens when we run out of resources mid-computation? The behavior depends on which function variant we use.

Option 1: Basic arithmetic (`add_prob_heat`)

When budget exhausts mid-computation, `add_prob_heat` returns a degraded result—either the first input unchanged or a partial computation:

Scenario: Same operation $\frac{3}{10} \oplus_p \frac{2}{5}$ but with insufficient budget $B_0 = 50\mu$.

1. Check if denominators equal: 6μ cost, 44μ remaining
2. Cross-multiply 3×5: 15μ cost, 29μ remaining
3. Cross-multiply 2×10: 20μ cost, 9μ remaining
4. Add cross-products: Budget exhausted after 9 of 20 steps

Result:

$$\text{add\_prob\_heat}\left(\frac{3}{10}, \frac{2}{5}, 50\mu\right) = \left(\frac{3}{10}, B' = 0, h = 50\mu\right)$$

The function returns the first input unchanged when unable to complete. Conservation: $0 + 50\mu = 50\mu$ ✓. The caller sees $(3/10, 0, 50\mu)$ and must decide: "Is returning the first operand acceptable, or do I need more budget?"

Option 2: Interval arithmetic (`add_prob_interval`)

For operations that must signal uncertainty explicitly, use the interval variant:

$$\text{add\_prob\_interval}\left(\frac{3}{10}, \frac{2}{5}, 50\mu\right) = \left(\text{PUnknown}, B' = 0, h = 50\mu\right)$$

When budget exhausts (budget = `fz`), this returns `PUnknown`—an explicit "I don't know" constructor. The type:

```
Inductive ProbInterval :=
  | PExact : FinProb -> ProbInterval
  | PRange : FinProb -> FinProb -> ProbInterval
  | PUnknown : ProbInterval.
```

With very low budget (`fs fz`), returns `PRange` giving bounds. Only with adequate budget does it return `PExact`.

Budget replenishment and retry:

Allocate $B_{\text{new}} = 200\mu$ and call again:

$$\text{add\_prob\_interval}\left(\frac{3}{10}, \frac{2}{5}, 200\mu\right) = \left(\text{PExact}\left(\frac{35}{50}\right), B' = 89\mu, h = 111\mu\right)$$

Success! Conservation: $89\mu + 111\mu = 200\mu$ ✓

Key lessons:

1. **Degraded results, not crashes:** Insufficient budget produces safe fallback (first input or PUnknown), never garbage.

2. **Conservation always holds:** Even failed operations account for all consumed budget as heat. No energy disappears.

3. **Type-level choice:** Use `add_prob_heat` when degraded results are acceptable. Alternatively, use the `add_prob_interval` variant when explicit uncertainty signaling is needed.

4. **No silent retries:** The system never automatically retries. Budget allocation is explicit.

5. **Monotonicity:** More budget never makes results worse. With 50μ we get degraded result or PUnknown. With

$$200\mu \tag{3}$$

5.7.1 Conservation Laws: The Ledger

The integrity of VOID arithmetic relies on a single, non-negotiable accounting principle: the conservation of the budget. Unlike classical operations which exist in a thermodynamic vacuum, every VOID operation constitutes a transfer. No resources are ever created or destroyed; they are merely transformed from *potential* (Budget) to *dissipated* (Heat).

Every operation in VOID must conserve resources—the initial budget B must equal the final budget B' plus all heat h generated during computation. This fundamental constraint ensures thermodynamic honesty: no operation can create or destroy computational capacity, only transform available budget into consumed heat.

Conservation Axiom:
For all $p_1, p_2, B, B', \text{res}, h$: if `add_prob_heat`$(p_1, p_2, B) = (\text{res}, B', h)$ then $h \oplus B' = B$.

This identity forces all VOID operations to account for their costs explicitly in the ledger. Without conservation, a system could hallucinate unlimited precision or infinite computation—the mathematical equivalent of perpetual motion. The axiom makes resource exhaustion ($B' = 0$) a genuine constraint rather than an implementation detail.

Theorem 5.7 (Universal Termination)**.** *In the VOID framework, every well-formed operation terminates in finite time. There are no infinite loops.*

Proof. All iterative processes in VOID are guarded by the `Budget` parameter of type `Fin`. `Fin` is an inductive type isomorphic to $\{0, \ldots, \text{MAX}\}$. Every recursive call matches on the budget:

- If $B = \text{fz}$ (zero), the operation terminates immediately (returning U or partial result).
- If $B = \text{fs } B'$ (successor), the operation proceeds with strictly smaller budget B'.

Since `Fin` is well-founded (contains no infinite descending chains), the budget must reach fz in a finite number of steps (specifically, $\leq B_{initial}$ steps). Therefore, non-termination is structurally impossible. □

What this section establishes: The Arithmetic of Accountability. By exposing the thermodynamic cost of every operation (cross-multiplication, comparison) and preserving the structural entropy of results (refusing automatic simplification), VOID transforms arithmetic from an abstract manipulation of values into a **physical process of accumulation**. This framework replaces the "magical" instantaneous answers of classical math with the mechanistic reality of the ledger. For the user, this "unusual" friction is a feature, not a bug: it ensures that mathematical complexity is never hidden, but always accounted for. Instead of silent precision errors or unpredicted latency, VOID offers a transparent trade-off: precision is purchased explicitly with budget, and every result carries the receipt of its construction.

Coq verification files: The arithmetic operations and their thermodynamic costs are fully formalized across several modules:

- `void_finite_minimal.v` — Bounded type Fin, Budget/Heat monoids, conservation axiom $B = B' \oplus h$
- `void_probability_minimal.v` — FinProb pairs (n, d), operations $\oplus_p / \otimes_p$ with boundary exclusion proofs

- `void_arithmetic.v` — All operations with heat tracking: `add_fin_heat`, `mult_fin_heat`, `div_fin_heat`, plus probability division module
- `void_probability_operations.v` — Derived operations: complement, min/max, weighted average, distance
- `void_distinguishability.v` — Kernel Δ_S, threshold collapse, resolution-bounded scales
- `void_pattern.v` — Quotient spaces $C_{(\rho,B)}$, pattern formation/matching with budget
- `void_pattern_algebra_extended.v` — Pattern composition, interference, scalar multiplication
- `void_information_theory.v` — READ (free) vs WRITE (costs budget) typeclass instances
- `void_entropy.v` — Minimum coding cost, heat accumulation as entropy
- `void_time_memory_composition.v` — Temporal traces, ledger identity $B_0 = B_n \oplus T(\mathrm{tr})$

6 Geometry without Infinity

A complete re-imagining of geometry: From our perspective, space isn't a pre-existing, malleable grid into or onto which one might project whatever figure or relation. It's the virtual, probabilistic structure of what a finite observer can distinguish at declared resolution with available budget. A "point" is a simplest pattern—something distinguishable from other patterns. "Distance" is the effort required to tell patterns apart. "Shape" is a probability field you navigate by asking "how circular is this location?" or "how triangular?" **Geometry is the discipline of these paid distinctions**.

This section addresses the foundational critique from our abstract: classical space is $\infty \times 0$—an infinite collection of zero-dimensional points, undefined algebra masquerading as foundation. That wasn't rhetorical flourish. It's the core problem. Each point in $\mathbb{R}^n$ has no extent. Space gains dimension only through infinite aggregation. But no finite observer can execute infinite operations. Classical geometry assumes completed infinities exist before any observer acts. We reverse this: observers with finite budget make distinctions, and space emerges as what they can afford to hold apart.

Our geometry eliminates real numbers entirely. **Points are patterns** (finite location + probability strength). **Distance is distinguishability** measured as FinProb pairs, geometrically interpreted in $(0, 1)$, computed at cost $\mu(\rho)$ per query. **Shapes are probability fields** you navigate by asking "how circular is this location?"—each query costs one tick, returns value in P°_ρ. Dimension emerges from budget structure: budget fs fz enables one dimension, fs (fs fz) enables two, fs (fs (fs ...)) enables three (maximum). **Coordinates are Fin tokens** (not reals), no $\mathbb{R}^n$, no infinity anywhere. What follows is geometry built from budgeted acts of distinction—showing that infinity was never necessary, just convenient for mathematicians who could ignore computational reality.

But what *is* this independent space? In modern mathematics, space is $\mathbb{R}^n$—the Cartesian product of real numbers. Yet each real number requires infinity to construct: Dedekind cuts partition rationals infinitely, Cauchy sequences converge through infinite steps, decimal expansions never terminate. Each point in $\mathbb{R}^n$ is zero-dimensional—no extent, no size. Space becomes $\infty \times 0$: an infinite collection of dimensionless points. This is undefined algebra masquerading as foundation. An observer with finite budget cannot construct such objects. They exist only by assumption, not by construction.

This isn't mere rhetorical provocation. Classical mathematics treats completed infinity as *necessary* for geometry when it is actually *optional*. For over two millennia—from Euclid (300 BCE) through Archimedes' computation of π, from medieval Islamic geometry through Gauss's differential geometry (1827)—humans *did geometry* without rigorous real numbers. Euclid's axioms never defined "point" or "line" as elements of $\mathbb{R}^n$ [34]. Newton and Leibniz invented calculus with "infinitesimals" they could not rigorously define [92]. Only in the 1800s, with Dedekind's cuts (1872) and Cantor's set theory, did mathematicians declare we needed completed infinite objects as *foundation*.

The classical view stems from measurement practices: rulers and compasses create spatial structures (buildings, diagrams), then those *tools* shape our abstract conception of space itself. What began as instrumental convenience—using rigid rods to establish distance—hardened into metaphysics: space as pre-existing grid, eternal and observer-independent.

But this was never philosophically transparent. Greek atomists explicitly challenged whether space exists between bodies, with Democritus declaring that only atoms and void are real [27, 114]. Medieval scholars like Buridan and Oresme debated whether vacuum could exist, critiquing

Aristotle's denial of void in their analyses of projectile motion [20, 95]. Kant recognized space as imposed structure rather than discovered fact, arguing it "adheres only to the form of intuition and hence to the subjective character of our mind" [69][A23/B37-8]. Yet geometry proceeded as if these questions were settled, perhaps because reopening them threatened too much.

We invert this completely, **following constructive traditions** [10, 87] but with explicit resource accounting absent from those frameworks. First comes the observer with finite budget. Then come acts of distinction that cost. Then emerges what we call "space" as the pattern of what can be held apart.

Consider a real-time rendering engine under strict computational constraints. When resources are scarce, a distant terrain is not a detailed surface "seen poorly"—it is literally rendered as a coarse, low-polygon mesh. The missing fractal detail is not obscured by haze; it is computationally unconstructable at that budget. The geometry itself simplifies to match the available processing power.

VOID draws its geometric intuition from a radically different lineage: not the rigid rod, but the nervous system. The construction of space is a fundamental capacity of living organisms—from eagles to humans—long before it becomes a formalized discipline. Biological spatiality is never a passive reception of a pre-existing grid; it is an active reconstruction based on finite sensory bandwidth and metabolic constraints. **Active vision is production, not free reading**. The space "seen" by an organism is a dynamic projection of what it can afford to process. VOID formalizes this biological reality: geometry is mathematics in a state of continuous, costly update, where the structure of space is determined not by eternal axioms, but by the resolution and budget of the cognitive agent.

This operationalization moves beyond philosophical finitism by treating resource bounds not as external limitations, but as constitutive axioms. We reject the passive view of space as a pre-existing container that is merely "seen" with varying clarity. Instead, **geometric structure is a function of resolution** ρ: low resolution does not blur the underlying geometry; it *defines* a simpler geometry. High resolution does not reveal pre-existing detail; it constructs it. This aligns with relational approaches in physics [109] and measurement theory [64], but treats the bound as primitive. The transition from Section 5 is seamless: **probabilistic distinguishability** Δ_S **becomes the spatial metric**. If the observer cannot afford to distinguish x from y, they occupy the same point. Points, distances, and dimensions are not primitives waiting to be discovered, but structural consequences of the budget allocated to distinction.

Formal setup We keep state $S = (B, \rho, M)$ and probability scale $P^\circ_\rho = \{n/d \mid 1 \le n < d \le D(\rho)\} \subset (0, 1)$ from Section 5. From Section 5 we inherit the distinguishability kernel $\Delta_S : X \times X \to$ FinProb which costs budget to compute and returns separation as probability pairs (geometrically interpreted in $(0, 1)$).

6.1 The $\infty \times 0$ Problem

Classical geometry's foundation rests on a construction that cannot be built. Space as $\mathbb{R}^n$ requires real numbers, each demanding completion of infinite processes. Dedekind cuts partition the rationals into two infinite sets. Cauchy sequences require checking infinite many terms for convergence. Decimal expansions $0.d_1d_2d_3\ldots$ extend without termination. No finite observer can construct these objects through operations—they exist only by fiat, assumed rather than built.

Each point in $\mathbb{R}^n$ has dimension zero: no length, no width, no depth. A line segment becomes an uncountable infinity of dimensionless points arranged continuously. Space emerges as their "collection"—but this is the multiplication $\infty \times 0$, undefined in any arithmetic. We paper over this by declaring points to be primitive and dimension to emerge from the manifold structure, but the incoherence remains: **foundations built on objects that require infinite work to specify yet contribute zero dimension individually**.

This wasn't always so. **For two millennia, geometry functioned without rigorous real numbers**. Euclid (300 BCE) defined points as "that which has no part" but never constructed them as elements of $\mathbb{R}^n$—the notation didn't exist, the concept was foreign [34]. His geometry used ruler and compass, producing constructible magnitudes (rationals and certain algebraic numbers), never completed infinities. Archimedes computed π through finite polygonal approximations, not by defining it as limit of infinite sequence. Medieval Islamic geometers developed spherical trigonometry and conic sections without Dedekind cuts. Gauss (1827) defined curvature of surfaces using finite computations on tangent spaces, decades before Riemann formalized manifolds atop $\mathbb{R}^n$ [49].

Newton and Leibniz invented calculus using "infinitesimals"—quantities smaller than any finite number yet not zero. Berkeley famously critiqued these as **"ghosts of departed quantities"** [9], exposing that 17th-century analysis had no rigorous foundation. The response, two centuries later, was Weierstrass's ϵ-δ formalism, Dedekind's cuts, and Cantor's transfinite arithmetic: completed infinity as *axiom.* This choice enabled elegant theorems—compactness, completeness, uniform continuity on closed intervals—but introduced pathologies absent from finite practice.

Non-measurable sets require the Axiom of Choice to construct and violate intuitive properties of "size." **Self-referential paradoxes emerge from unrestricted comprehension over infinite domains**. Renormalization in quantum field theory generates infinite integrals that must be "subtracted away" through elaborate regularization [30]—**infinities appearing because we assumed continuous spacetime**, itself built from $\mathbb{R}^4$. Algorithms proven correct "in the limit" crash when run on finite machines because the limiting behavior assumes resources never exhaust.

These aren't peripheral difficulties or failures of applied mathematics. They are symptoms of infinity's presence as foundational principle rather than methodological tool.[17, 61] Completed infinity was never discovered in nature [3]. It was invented by mathematicians as idealization, then mistaken for necessity. The move from "arbitrarily large finite" to "actually infinite" seems innocuous but proves catastrophic: **finite systems have exact answers, infinite systems have limits that may not exist, may not converge, or may converge differently depending on path (Riemann rearrangement theorem** [100]).

VOID rejects this path entirely. We demonstrate that geometry—distances, dimensions, curvature, shapes—can be constructed from finite distinguishability at declared resolution. No real numbers, no limits, no completed infinities. What classical geometry assumes as primitive (continuous space, dimensionless points, costless measurement), we derive from more fundamental operations: budgeted acts of distinction by observers with finite capacity. Section 6.2 onward develops this constructively, showing it is not "finite approximation of infinite truth" but *different mathematics* where resource bounds are constitutive rather than restrictive.

The question is not whether classical geometry *works*—it does, spectacularly, like most of mathematics and the enormous edifice it became after thousands of more or less haphazard innovations made without really looking back at partly crumbling foundations. The question is whether infinity is *necessary* or merely *convenient.* We prove it was always optional. **For computational systems and discrete implementations, finite foundations are mathematically complete without infinity**. They acknowledge from the start what classical mathematics denies: *every distinction costs budget, every operation consumes resources, every observer has finite capacity.* **Those**

aren't bugs to be abstracted away—they're the structure of what mathematics should formalize.

6.2 Points Are Patterns, Space Is a Field

In void geometry, there are no coordinate grids or origin points. Instead:

Definition 6.1 (Void point)**.** A point in void space is a pattern:

$$\text{VoidPoint} := \text{Pattern}$$

Each pattern has:

- **location**: a finite label (element of Fin)
- **strength**: probability in P°_{ρ} of persistence

Definition 6.2 (Void space)**.** Space is the distinguishability relationship:

$$\text{VoidSpace} := \text{VoidPoint} \to \text{VoidPoint} \to \text{ObserverWithBudget} \to (\text{FinProb} \times \text{Budget})$$

Note: FinProb values are geometrically interpreted in $(0, 1)$*, though the type permits general rational pairs.*

This function takes two patterns and an observer, returns their distinguishability probability and remaining budget. Computing space structure costs—there are no free lookups.

Definition 6.3 (Shape field)**.** A geometric object is a probability field over patterns:

$$\text{ShapeField} : \text{VoidPoint} \times \text{ObserverWithBudget} \to \text{FinProb} \times \text{ObserverWithBudget}$$

A shape field assigns to each pattern location a probability of "belonging to" that shape, consuming one tick of observer budget per query.

What this means: Triangleness, circleness, lineness—these aren't properties OF objects. They're fields you navigate through. You query "how triangular is this location?" and get back a probability, paying one tick per query.

6.3 Distance As Distinguishability Effort

Definition 6.4 (Point distance)**.** For patterns p_1, p_2 and observer state:

$$\text{point\_distance}(p_1, p_2, \text{obs}) := \begin{cases} (\text{half}, \text{obs}) & \text{if obs\_budget(obs)} = 0 \\ \text{distinguishability\_distance}(p_1, p_2, \text{obs}) & \text{otherwise} \end{cases}$$

where the second case costs one tick.

Axiom 21 (Uniform geometric cost)**.** Every geometric operation costs exactly one tick:

- Measuring distance between two patterns: μ (one tick)
- Checking if point lies in shape field: μ (one tick)
- Computing local curvature: μ (one tick)

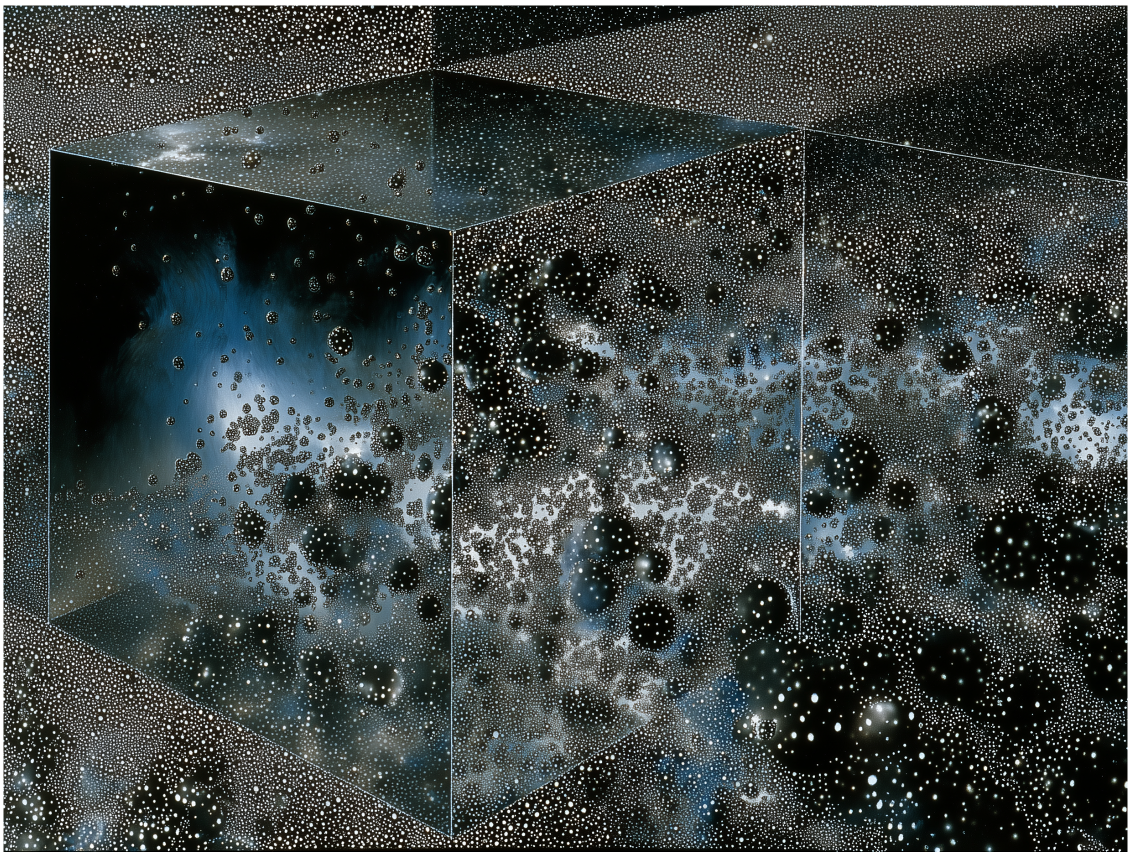


Figure 6: **Synthetic Visualization: The Granularity of Distinction.** Generated via probabilistic diffusion (Midjourney), this visualization interprets the VOID geometric primitive not as a continuous solid, but as a discrete density field of distinction acts. The bright grains" represent coordinates where the budget was sufficient to establish a value $p \in P_\rho^\circ$ (a hit"). The surrounding darkness is not empty space, but the uncomputed generative field. Note how the topology holds at the boundaries where expenditure" is high, but dissolves into probabilistic noise where the distinction budget is sparse. The image was constructed not merely through textual prompting (terms like probabilistic geometry" provided only weak instruction), but through a process of recursive image blending using the author's previous geometric studies, filters, and mood-boards—creating a feedback loop where output becomes input, mirroring the state-transition logic of VOID. This process of generating images in relation to our progress in formalisms lasted since the beginning of the project, that is, in July 2014.

- Sampling one adjacent location: μ (one tick)

No atomic operation is intrinsically "harder" than another; each costs exactly one unit tick. The apparent expense of complex operations (like cross-multiplication) emerges solely from the iteration of these simple steps, not from opaque complexity penalties.

Example (Three pattern distances): Consider patterns p_a, p_b, p_c at locations with initial budget $B_0 = 10\mu(\rho)$.

Query 1: Distance p_a to p_b returns $(7/10, B_0 - \mu)$
Query 2: Distance p_b to p_c returns $(3/10, 8\mu)$—low distinguishability, 1 tick consumed.
Query 3: Distance p_a to p_c directly returns $(6/10, 7\mu)$—1 tick consumed.

Going via intermediate pattern ($p_a \to p_b \to p_c$) costs 2 ticks. Going direct costs 1 tick. **Direct paths can be cheaper**, violating triangle inequality. This isn't error—it's honest accounting of which discrimination tests are available.

6.4 Shapes As Probability Landscapes

Shapes aren't boundaries or equations. They're fields returning "how much does this location belong to this shape?" This reconceptualization is radical. In classical geometry, a triangle IS its vertices and edges—definite, eternal objects. Here, triangleness is a field you navigate through, asking at each location "how triangular is it here?" The answer costs computation and returns probability, not certainty. You don't determine membership in the triangle; you measure triangular-ness as a probability at each queried location. Sample enough points (if budget allows), and you build a discrete map of triangular-ness values. The appearance of 'smoothness' between samples is exhaustion—you can't afford to check every intermediate point.

Triangleness field:

$$\text{triangleness\_field}(p_1, p_2, p_3)(p, \text{obs}) := \begin{cases} (\text{half}, \text{obs}) & \text{if budget exhausted} \\ ((2,3), \text{obs}''') & \text{if all distances nonzero} \\ (\text{half}, \text{obs}''') & \text{otherwise} \end{cases}$$

where obs$'''$ reflects three distance checks at cost 3μ (three ticks).

This costs **three ticks**: one per distance check. A location is "triangular" if it maintains measurable distances to three reference points. The probability (2/3) isn't magic—it's just "sufficiently triangular for this resolution."

Circleness field:

$$\text{circleness\_field}(\text{center}, \text{radius})(p, \text{obs}) := \begin{cases} (\text{half}, \text{obs}) & \text{if budget exhausted} \\ ((2,3), \text{obs}') & \text{if dist} \approx \text{radius} \\ ((1,3), \text{obs}') & \text{otherwise} \end{cases}$$

where obs$'$ reflects one distance measurement at cost $\mu(\rho)$.

Costs one distance measurement plus fraction arithmetic for radius comparison.

Lineness field:

$$\text{lineness\_field}(p_1, p_2)(p, \text{obs}) := \begin{cases} (\text{half}, \text{obs}) & \text{if budget exhausted} \\ ((2,3), \text{obs}'') & \text{if both distances nonzero} \\ (\text{half}, \text{obs}'') & \text{otherwise} \end{cases}$$

where obs$''$ reflects two distance checks at cost $2\mu(\rho)$.

Costs **two ticks**: check distances to both endpoints.

6.5 Navigation Through Shape Fields

You don't plot coordinates. You take discrete sampling steps through probability landscapes.

Definition 6.5 (Discrete neighbor search)**.**

$$\text{step\_if\_better}(p, \text{field}, \text{obs}) \quad \begin{cases} (p, \text{obs}) & \text{if budget exhausted} \\ (p', \text{obs}'') & \text{if val}' > \text{val} \\ (p, \text{obs}'') & \text{otherwise} \end{cases}$$

where p' is a neighboring location, val$'$ is field value at p', val is field value at p, and obs$''$ reflects two field evaluations at cost $2\mu(\rho)$.

This costs **two ticks**: sample neighboring location, sample current location, compare. No calculus, no derivatives—just "try one step, see if it's better."

Micro-scene (Finding a circle): Observer with initial budget B_0 (sufficient for 20 operations) wants to find circular patterns near location 5.

Step 1: Query circleness at location 5: returns $(3/10, B_0 - \mu)$—not very circular, 1 tick spent.

Step 2: Query circleness at location 6: returns $(7/10, B_0 - 2\mu)$—much more circular! Move there, 2 ticks spent total.

Step 3: Query circleness at location 7: returns $(5/10, B_0 - 3\mu)$—worse. Stay at 6, 3 ticks spent.

Step 4: Query circleness at location 5 (backtrack): returns $(3/10, B_0 - 4\mu)$—worse. Confirm 6 is local maximum, 4 ticks spent.

Observer has found "most circular nearby region" using 4 ticks of available budget. Cost is exactly 4μ, no hidden work.

6.6 Dimension Emerges From Exhaustion

Dimension isn't a property of space. It's how many questions the observer can afford to ask.

Definition 6.6 (Observed dimension)**.**

$$\text{observed\_dimension}(\text{obs}) := \begin{cases} \text{fz} & \text{if budget exhausted (fz)} \\ \text{fs fz} & \text{if budget} = \text{fs fz} \\ \text{fs (fs fz)} & \text{if budget} = \text{fs (fs fz)} \\ \text{fs (fs (fs fz))} & \text{if budget} = \text{fs (fs (fs \_)) (max)} \end{cases}$$

(maximum three dimensions in this implementation).

This redefinition of dimension enables a new class of operations impossible in rigid geometric frameworks: **dimensional throttling**. Since dimensionality is a function of available budget ($D \propto B$), systems can dynamically shed dimensions to preserve liveness under load, rather than crashing.

Example 1: The Algorithmic Fight-or-Flight. A high-frequency trading system normally tracks market state via a 50-dimensional correlation matrix. During a flash crash, volatility spikes, and the computational cost to update the full matrix exceeds the tick time limit (B→0). A classical system lags and fails. A VOID system executes *dimensional shedding*: it abandons the complex matrix entirely, collapsing its world model to a single dimension—price momentum. It continues to execute trades on this crude, 1D vector, preserving liquidity and survival when higher intelligence is unaffordable.

Example 2: Cognitive Tunneling. Consider a pilot encountering catastrophic mechanical failure. The cockpit presents 100 distinct data streams (navigation, radio, fuel mix, oil pressure). As stress (heat) depletes the pilot's cognitive budget, the brain performs an automatic dimensional throttle: it blinds itself to radio and navigation, reducing the flight envelope to a single, existential metric—attitude relative to the horizon. The system survives not by processing more, but by aggressively ignoring 99 dimensions to maintain absolute control over the one that prevents impact.

This dimensional observer-dependence differs from both classical dimensional reduction [70, 74] and computational complexity's dimension hierarchies [76, 24]. Those treat dimension as intrinsic property approximated or compressed; we treat it as fundamentally observer-relative.

No classical geometry allows this. In Euclidean, Riemannian, or even non-Euclidean geometries, dimension is an intrinsic property of the manifold, independent of who measures it or what resources they have. Here, dimension is **observer-dependent and resource-contingent**. This isn't approximation error—it's exact mathematics of what you can afford to distinguish. Two observers with different budgets live in genuinely different-dimensional spaces, with no "true" dimensionality adjudicating between them.

What this means: With 1 tick of budget, you can check one coordinate. With 2 ticks, two coordinates. With 3+ ticks, you reach the maximum three dimensions this implementation supports. Not because space "is" 3D, but because each dimension costs a tick to query.

Axiom 22 (Dimension-budget correspondence)**.** The number of distinguishable dimensions equals the number of affordable coordinate queries:

$$\dim(S) = \min(\text{budget}(S), \text{MAX\_DIM})$$

This is radical. A "sphere" in this geometry has different dimensionality for different observers. Well-funded observer sees 3D sphere. Exhausted observer sees 1D line. Same object, different budgets.

Example (Dimensional collapse)

Observer A with $B = 10\mu$: Can query all three coordinates, sees full 3D structure.
Observer B with $B = 2\mu$: Can only afford two queries, sees projection to 2D plane.
Observer C with $B = 1\mu$: Can only afford one query, sees 1D shadow.
Observer D with $B = 0$: Cannot query anything, experiences dimensionless point.

Same pattern, four different geometries, purely from budget differences.

6.7 When Discreteness Looks Smooth: Budget Depletion

Classical geometry assumes we can always "zoom in" for finer detail. Void geometry recognizes limits. When budget depletes, the discrete structure of individual patterns blurs into apparent smoothness—not because space became continuous, but because you can no longer afford to resolve the gaps.

The poverty of the continuum. Smoothness is not a feature of reality; it is a symptom of poverty. The continuum appears only when the observer can no longer afford to see the seams.

This anticipates Section 7's calculus, where continuity emerges from exhaustion rather than being assumed as primitive. Curvature captures this transition:

6.8 Curvature As Distinguishability Variation

Definition 6.7 (Local curvature)**.**

$$\text{local\_curvature}(p, \text{obs}) := \begin{cases} (\text{half}, \text{obs}) & \text{if budget exhausted} \\ (d, \text{obs}') & \text{where } d = \text{point\_distance}(p, p') \end{cases}$$

with p' being one step from p, costing one tick.

Curvature is just "how fast does distinguishability change?" Sample one nearby point, measure distance, done.

Flat space: nearby points have small distinguishability differences.

Curved space: nearby points have large distinguishability differences.

No Riemann tensors, no Christoffel symbols—just "look one step ahead, see how much things change."

6.9 Maintaining Shapes Costs Budget

A "perfect" circle doesn't exist eternally. It requires active maintenance.

Micro-scene (Circle maintenance): Circle with 8 points, each pair must maintain distance equal to radius r.

Initial construction: Verify all $\binom{8}{2} = 28$ pairwise distances equal r. Cost: $28\mu(\rho)$.

But patterns drift due to thermal noise (Section 8). After time τ, must re-verify distances.

Maintenance per time unit: $28\mu(\rho) \times \tau$.

The circle isn't a Platonic form—it's a maintenance schedule. Stop paying, the pattern decays, distances drift, circularity vanishes.

Axiom 23 (Geometric maintenance cost)**.** For shape with n constraints each costing $\mu(\rho)$ to verify:

$$h_{\text{maintain}}(\tau) = n \times \mu(\rho) \times \tau$$

Symmetry isn't conserved—it's continuously purchased against entropy.

6.10 Topology That Folds

Space can fold, creating shortcuts between distant locations. But folding costs energy.

These fold bridges formalize wormhole-like structures studied informally in general relativity and quantum information [85], but with explicit maintenance costs absent from those treatments. Space topology isn't fixed—it's an actively maintained configuration.

Definition 6.8 (Fold bridge)**.** A fold bridge is a structure with:

- end1: First endpoint (element of Fin)
- end2: Second endpoint (element of Fin)

- stability: Probability bridge holds (in P_ρ°)
- maintenance_cost: Budget per tick to maintain (in Fin)

A fold bridge connects two locations that would normally require many hops. But:

1. Creating bridge: costs $\mu(\rho)$
2. Maintaining bridge: costs $\mu(\rho)$ per tick
3. Bridge can collapse: if stability drops below threshold

Example (Wormhole economics): Two locations at "distance" 10 (requires 10 hops to traverse normally).

Option A: Travel normally, cost $10\mu(\rho)$ per journey.

Option B: Create fold bridge (cost $\mu(\rho)$), then traverse in 1 hop (cost $\mu(\rho)$), but must maintain bridge (cost $\mu(\rho)$ per tick).

Breakeven: If you make more than 1 journey per tick, bridge pays off. Otherwise, normal travel is cheaper.

While this terminology evokes science fiction, it formalizes a mundane intuition of the post-human era: distance is a function of logistical cost, not geography. A flight from Kraków to London (1500 km) is often **more effective**—consuming less time and metabolic energy—than a trip to a nearby but disconnected village like Skomielna Czarna (37 km away). The airport infrastructure acts as a maintained "fold" that lowers the thermodynamic cost of the long jump, while the friction of local transit makes the short geometric distance operationally distant. In VOID, we do not measure miles; we measure the metabolic cost of arrival.

6.11 Observer-Dependent Geometry

Two observers with different budgets measure genuinely different geometries. Not "approximately different"—fundamentally different.

Micro-scene (Disagreeing observers): Pattern p at location 5.

Observer A (budget 10μ, resolution $\rho_1 = 100$): Measures distance to pattern q as (8/10), distinguishes 3 dimensions, sees curved space.

Observer B (budget 2μ, resolution $\rho_2 = 10$): Measures distance to same pattern q as (2/10) (can't afford fine discrimination), distinguishes 2 dimensions, sees flat space.

Who's right? **Both.** These are equally valid geometric facts. Observer A can afford to see curvature, observer B cannot. The geometry doesn't exist independently—it's the structure of what each observer can distinguish.

Axiom 24 (Geometric relativity)**.** For observers (B_1, ρ_1) and (B_2, ρ_2):

$$\text{geometry}_{(B_1,\rho_1)} \neq \text{geometry}_{(B_2,\rho_2)}$$

unless $(B_1, \rho_1) = (B_2, \rho_2)$.

Geometry is observer-relative, not because of measurement error, but because geometric structure IS the distinguishability structure affordable at given resources.

6.12 What Classical Geometry Misses

Classical geometry's power comes from assuming unlimited precision, infinite divisibility, and cost-free measurement. These aren't just convenient idealizations—they're structurally incompatible with finite foundations when applied to discrete computational implementations. Finite computational geometry operates in systems where:

- Measurements cost budget (Section 8's resource accounting)
- Precision requires resources (Section 5's resolution parameter)
- Structures decay without maintenance (Section 9's credit assignment)
- Observers have finite capacity (Section 4's bounded budgets)

Void geometry doesn't "approximate" classical geometry under resource constraints. It provides different foundations where resource limits are primitive rather than perturbative. The classical limit (infinite budget, infinite resolution) doesn't even make sense here—there's no coherent way to take $B \to \infty$ or $\rho \to \infty$ because the system is built from finitude up.

6.13 What We've Gained

Everything above is finite by construction:

- Space is patterns at finite locations
- Distance is FinProb values (geometrically interpreted in $(0, 1)$)
- Shapes are finite fields costing one tick per query
- Dimension is budget-determined, not space-determined
- Navigation is discrete neighbor checks, not calculus
- Maintenance is explicit budget expenditure
- Curvature is sampled locally, not computed globally

We haven't lost geometry—we've gained honesty about what geometric claims cost.

The "perfect sphere" of Platonic heaven doesn't exist here. But we can maintain sphere-like patterns by paying μ per verification, and those patterns behave geometrically until we stop paying for them.

Classical geometry assumes perfect forms exist for free. Void geometry recognizes forms are processes maintained against decay, and processes cost budget.

6.14 Why This Matters Beyond Theory

Classical mathematics assumes infinite resources and produces algorithms that work "eventually" or "in the limit." But real systems crash, hang, or produce garbage when pushed beyond capacity.

Void mathematics provides alternative foundations where:

- **Limits are first-class mathematical objects** (U regions)

- **Cost is part of the type system** (Budget-parameterized operations)
- **Degradation is graceful** (coarser resolution, not failure)
- **Time emerges from computation** (not assumed as background)

This isn't "applied mathematics" taking pure math and making it practical. It's **different mathematics** that models computational systems with explicit resource accounting because it was built from finite observation up, not infinite idealization down.

The opportunities above aren't improvements to existing approaches—they're things that become possible when you stop pretending infinity exists and start accounting for every tick of work.

The geometry formalized in this section is the foundation. The applications emerge when you take it seriously: space isn't given, operations cost, and honesty about limits is mathematical virtue, not engineering compromise.

6.15 Geometric Horizons: From CAD to Reality

These definitions aren't merely theoretical constraints; they sketch the architecture of a new class of spatial computing.

Resource-Aware Rendering. Current graphics engines simulate continuity through massive over-sampling, crushing GPUs to approximate an infinite ideal. A VOID-based engine would invert this: geometry itself simplifies dynamically. Distant objects aren't "high-poly meshes rendered poorly"; they structurally collapse into lower-dimensional impostors as the budget for their distinction diminishes. Level of Detail (LOD) becomes a law of physics, not a rendering trick.

Robotics and Navigation. An autonomous agent operating in VOID geometry doesn't crash when pathfinding exceeds memory. Instead, the space "folds" into a coarser topology. A robot low on battery (Budget) stops perceiving the room as a complex mesh of obstacles and sees it as a simple node-graph of traversable paths. It navigates a simpler world because it cannot afford a complex one.

Physics Simulation. Collision detection in classical engines suffers from "tunneling" (objects passing through each other) when time steps are too large. VOID handles this as budget exhaustion: if the system cannot afford to calculate the interaction, it returns U—explicitly flagging the simulation gap rather than generating a glitch. This moves us from glitchy approximations of the continuum to honest, low-resolution certainties.

Coq verification:

Core geometric primitives formalized with thermodynamic accounting:

- `void_geometry.v` — Vector spaces with budgeted operations, void projection, dimension computation
- `void_geometry_basis.v` — VoidPoint = Pattern, ShapeField definitions, triangleness / circleness / lineness fields, discrete steps navigation, observer-dependent dimension
- `void_distinguishability.v` — Kernel Δ_S with budget tracking, threshold collapse
- `void_pattern.v` — Pattern as location + strength, decay mechanics

- `void_topology_folding.v` — FoldBridge records, space folding operations, maintenance costs
- `void_symmetry_movement.v` — Symmetry as actively maintained property, not conserved automatically

All operations maintain conservation $B = B' \oplus h$ and return probabilities in P_ρ°.

7 Flows, Entropy, and Time as Ledger

7.1 Not a River

Time is not a current dragging the system forward; it is the sediment of irreversible operations. It is the accumulated receipt of work done—the sum of all heat dissipated to keep distinctions alive, to execute transforms, to resist collapse back into undifferentiated flux. The classical story, from Newton's absolute time [92] to Einstein's spacetime parameter [32, 33], treats time as a coordinate t that indexes states. Our story inverts this: states generate time through priced transitions. What we call dynamics is the pattern of these expenditures; what we call causation is the discipline of which operations can afford to trigger which others.

This inversion has a profound consequence: **there is no coherent construction of an external observer**. In frameworks where time is a parameter, the mathematician implicitly occupies a position outside the system, surveying states $S(t)$ as t varies. This external vantage point—the "God's-eye view"—appears neutral, but it assumes the observer can make distinctions about the entire system without consuming resources from within it.

VOID shows this assumption is mathematically untenable for finite systems. An observer is a pattern: a structure maintained by distinctions that cost budget. To construct a view "from outside" requires:

- Distinctions to define the system boundary (cost: b_1)
- Distinctions to characterize what lies beyond (cost: b_2)
- Distinctions to relate internal structure to external position (cost: b_3)

For a pattern with budget b to observe a system with total budget B, we require $b_1 + b_2 + b_3 \leq b$. But if the observer is part of the system, $b \leq B$, and the distinctions needed to characterize "outside" must reference content not distinguishable within budget B. The construction fails: "outside" is not merely unknown—it is unconstructable.

The consequence is **mathematical immanence**: every observation is made from within, by patterns observing patterns, with no meta-level from which the whole is surveyable. This does not prohibit mathematics; it constrains what mathematics can coherently construct. We cannot write $S(t)$ and mean "the state at time t as seen from outside time." We can only write: pattern P at budget b makes distinctions about pattern Q, consuming budget and accumulating heat. Time is not the index; time is the ledger of distinctions made.

Wittgenstein grappled with this constraint throughout his philosophical trajectory. The Early Wittgenstein of the *Tractatus* established the hard boundary of the constructible, famously concluding that "whereof one cannot speak, thereof one must be silent" [125]. The Late Wittgenstein operationalized this insight, rejecting abstract logical forms to insist that meaning resides solely in *use* within bounded language games [126]. Both intuitions point toward finitude and immanence. VOID unifies them mathematically: the "limit of the world" is the budget, and "meaning as use" is the thermodynamic cost of maintenance. What cannot be constructed with finite budget has no mathematical existence.

VOID theory adjudicates the famous 1922 debate between Bergson and Einstein, delivering a belated verdict: **Winner — Bergson.** When Einstein declared that "the time of the philosophers does not exist," he was asserting the primacy of the block universe—time as a geometric coordinate

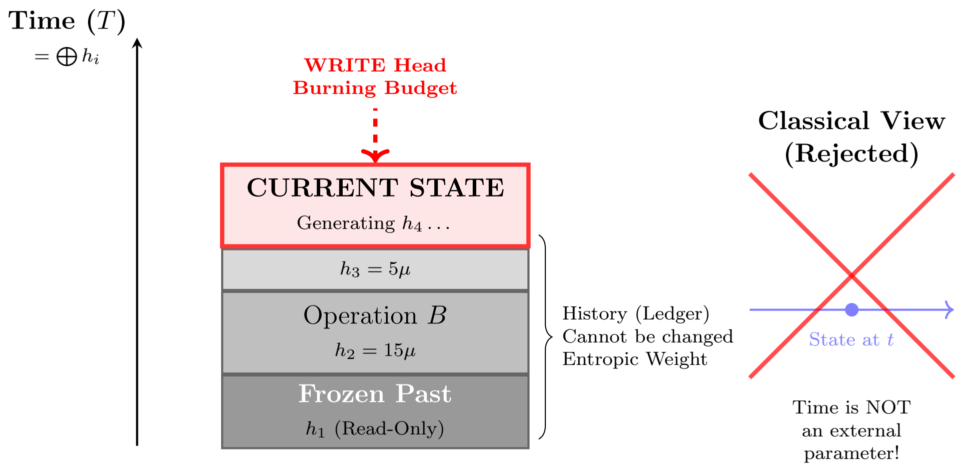


Figure 7: **Time as Sedimentation.** Unlike the classical view (right), where time flows as an external parameter t, in VOID (left) time accumulates vertically as a ledger of dissipated heat. The past is simply the weight of frozen operations beneath the current state. T is not a dimension to travel through, but a measure of the system's thermodynamic history ($T = \bigoplus h_i$).

t, visible from a transcendent vantage point. Bergson argued for *durée*—time as irreversible accumulation, experienced from within. History sided with Einstein, but only because physics assumed the external observer was a valid construct. VOID proves it is not (Theorem 7.4).

For any finite system, the "God's eye view" required to see time as a spatial dimension is unconstructable; it demands budget that the system does not possess. Therefore, Einstein's spacetime is a useful but physically impossible fiction (an artifact of infinite capacity), while Bergson's *durée* is the rigorous description of the system's internal reality. Even if one finds this re-evaluation preposterous, one must at least admit that the participants were addressing the same question in radically different contexts: Bergson spoke of a finite system (the human mind and its experience), while Einstein's response imagined a "Vitruvian" observer suspended in an objective but ideal reality. It was, of course, infinity-based mathematics that firmly secured the immutability of physical laws in Einstein's model. The error was not Bergson's intuition but the lack of a mathematical language to formalize why transcendence is unaffordable. VOID provides that language. [8]

In this section we follow budget as it circulates across the geometric scaffold of Section 6, track entropy accumulation along computational trajectories, and characterize prediction as optimization under finitude: maintain pattern distinguishability at minimum distinction cost. We trace how patterns persist through budgeted maintenance (§7.1), how distinguishability accumulates as path-dependent entropy (§7.2), **how time emerges as integrated heat rather than external parameter** (§7.6), how patterns interfere when distinctions are unaffordable (§7.7), and how exhaustion produces formal undecidability—not ignorance, but the mathematical end of distinguishability (§7.8).

[8] It is plausible that the historical consensus sided with Einstein not for rigorous reasons, but due to the sociological conditioning of the era. In the shadow of WWI, the subject was raised to perceive itself as an object within a vast system of relations—a cog in an imperial machine. Paradoxically, affirming subjective experience—even in the time of Freud, that plumber of the repressed in the mental thermodynamics of the bourgeoisie—felt more alien to the traumatized collective consciousness than accepting the status of a negligible coordinate in an objective, unfeeling spacetime.

What we build is not a framework that describes systems "from outside," but a mathematics adequate to systems observing themselves from within—finite patterns making finite distinctions about finite patterns, with no escape to transcendence because transcendence has no coherent construction.

Example 7.1 (Causal Affordability)**.** Operation A costs $5\mu(\rho)$ and produces state with budget $B' = 15\mu$. Operation B costs 20μ. Then A can trigger C (costs 10μ) but cannot trigger B—insufficient remaining budget. The causal graph is not logical possibility but thermodynamic affordability: edges exist only where $B' \geq \text{cost}_{\text{next}}$.

We keep the state $S = (B, \rho, M)$ and the interior probability scale $P^\circ_\rho = \{n/d \mid 1 \leq n < d \leq D(\rho)\} \subset (0,1)$. Event-sum and product are declared only when their side-conditions (disjointness/independence) hold and the reduced result stays interior with denominator $\leq D(\rho)$; otherwise the operation is undefined at ρ. Every step obeys the ledger $B = B' \oplus h$, where $h = 0$ (budget-neutral read) or $h \geq \mu(\rho)$ (priced act).

7.2 The Impossibility of External Observation: A Formal Proof

We now make the preceding argument mathematically explicit. This is not philosophy—it is a theorem about finite systems. The choice is binary (unless one's too tired to follow through to the end of the argument): either accept that observers are internal to the system they observe, or explicitly posit actual infinity as a mathematical primitive (or doze off on a couch). There is no neutral ground. Only the couch.

7.2.1 Formal Setup

From `void_finite_minimal.v`, the foundational type is:

```
Inductive Fin : Type :=
  | fz : Fin
  | fs : Fin -> Fin.

Axiom fin_bounded : forall n : Fin, (fin_to_Z_PROOF_ONLY n <= MAX)%Z.
```

Translation to standard formalism: Fin is inductively constructed: `fz` | `fs`(Fin), with $\forall n : \text{Fin}.\, n \leq \text{MAX}$.

Every value is bounded. The function `fin_to_Z_PROOF_ONLY` exists only in the metalanguage for proofs—it cannot be used computationally. This enforces: no escape to $\mathbb{N}$.

Budget and Heat are defined as:

```
Definition Budget := Fin.
Definition Heat := Fin.
```

The fundamental conservation law from `void_arithmetic.v`:

```
Axiom heat_conservation_add : forall n m b b' res h,
  add_fin_heat n m b = (res, b', h) -> add_heat h b' = b.
```

Standard form:

$$B = B' \oplus h$$

For any operation consuming budget B and producing state with budget B', the heat generated is $h = B \ominus B'$. The total is conserved.

Every atomic operation costs:

```
Definition operation_cost : Fin := fs fz. (* exactly 1 *)
```

$$\Delta_{\text{tick}} = 1$$

7.3 Notation Change

Why we will be using different symbols now. This section proves theorems about resource constraints: conservation laws, the impossibility of external observation, thermodynamic limits. These proofs require manipulating budget equations explicitly.

Writing conservation laws with function names is unreadable. Compare `add_heat(h1, add_heat(h2, add_heat(h3, fz)))` versus $h_1 \oplus h_2 \oplus h_3$. When proving that external observation requires $b_1 \oplus b_2 \oplus b_3 \preceq b_{\text{obs}} \preceq B_{\text{sys}}$ (§7.3.2), nested function calls obscure the mathematical structure.

The operators below are the same functions from the formalization—just with notation suitable for theorem-proving rather than programming. Every $\oplus$ is an `add_heat` call; every $\oplus_p$ is `add_prob_heat`. Same semantics, readable syntax.

Operations on Fin (Budget, Heat, counts):

Symbol	Coq Function	Meaning
$a \oplus b$	`add_heat a b`	Fin addition
	`add_fin a b budget`	(with budget tracking)
$a \ominus b$	`sub_fin a b budget`	Saturating subtraction
$\bigoplus_i a_i$	`fold_left add_heat [a1;a2;...] fz`	Iterated addition

Operations on FinProb (rationals n/d with $d \leq D(\rho)$):

Symbol	Coq Function	Meaning
$p_1 \oplus_p p_2$	`add_prob_heat p1 p2 b`	Disjoint sum
$p_1 \ominus_p p_2$	`sub_prob_heat p1 p2 b`	Saturating subtraction
$p_1 \otimes_p p_2$	`mult_prob_heat p1 p2 b`	Independent product
$p_1 \oslash_p p_2$	`div_prob_heat p1 p2 ρ b`	Division

Mixed operation (Budget scaling by probability):

Symbol	Definition	Meaning
$p \otimes_h b$	$\lfloor (n \cdot b)/d \rfloor$ for $p = n/d$	Scale budget by probability

Type discipline. The subscript p marks probability operations (on FinProb pairs). No subscript marks Fin operations (on Budget/Heat). These are different types:

- Fin = `fz` | `fs`(Fin) — bounded inductive
- FinProb = (Fin × Fin) — numerator/denominator pairs

Mixing them without conversion is a type error. Every operation returns triples `(result, budget', heat)` tracking resource consumption.

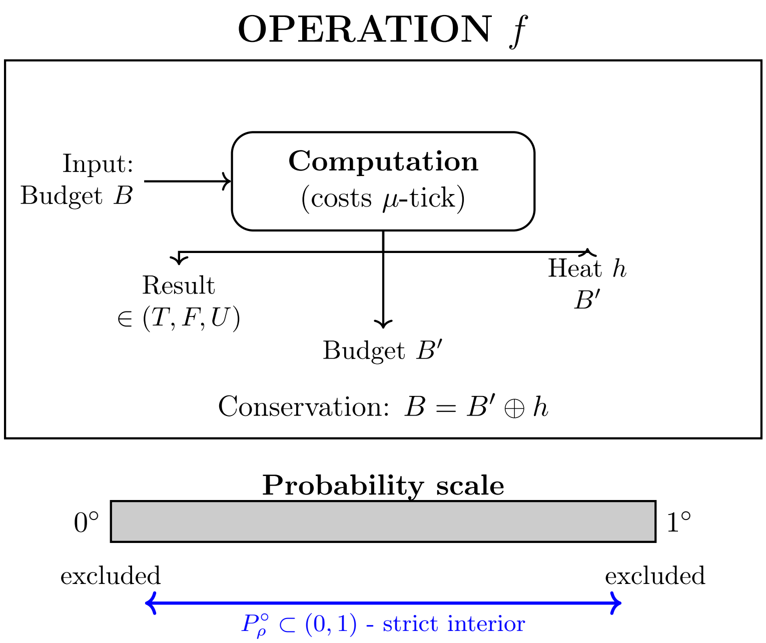


Figure 8: **Fundamental structure of VOID operations.** Each operation costs one μ-tick, returns three-valued results (T, F, U), and obeys strict conservation: $B = B' \oplus h$. Probabilities avoid boundaries, lying strictly in $(0, 1)$.

Why not $\sum$: Standard notation $\sum_{i=1}^{n} a_i$ presupposes natural number indexing. Since we reject completed infinities, we use $\bigoplus$ (iterated monoid operation) which is implemented as recursive fold over finite lists. It's the same result—just honest about finitude.

Conservation law. Every operation conserves the ledger:

$$B = B' \oplus h$$

where B = initial budget, B' = remaining budget, h = heat dissipated. This is axiomatized in `void_arithmetic.v` as shown above. The symbolic form $B = B' \oplus h$ is just the readable version of this axiom.

7.3.1 The External Observer: What It Would Require

An external observer is a mathematical construction that:

1. Distinguishes the system boundary ∂S (cost b_1)
2. Distinguishes "outside" S^c (cost b_2)
3. Relates internal state to external position (cost b_3)

Let:

- B_{sys} = total system budget (finite by construction)
- b_{obs} = observer budget

Requirement: To construct the external view, the observer must execute:

$$b_1 \oplus b_2 \oplus b_3 \preceq b_{\text{obs}}$$

7.3.2 The Impossibility Theorem

Theorem 7.2 (Unconstructability of External Observation)**.** *For any observer O that is part of a finite system S, the external observer position is unconstructable.*

Proof. **(1) Observer as Pattern**

From `void_pattern.v`, any observer is a maintained pattern:

```
Record Observer := {
  sensitivity : Fin;
  obs_budget : Budget;
  obs_heat : Heat
}.
```

If $O \subseteq S$ (observer is part of system), then:

$$b_{\text{obs}} \leq B_{\text{sys}}$$

This is not an assumption—it's definitional. The observer's budget is allocated from system resources.

(2) Cost of Distinguishing "Outside"

To distinguish S^c, the observer must make distinctions about structure not contained in S. From `void_distinguishability.v`:

```
Definition distinguishability_with_budget (O : ObsState) (e1 e2 : EnvState)
  (b : Budget) : (FinProb * Budget) :=
  match prob_diff_with_budget (mu O e1) (mu O e2) b with
  | (diff, b1) => ...
```

Each distinction about environment states costs budget. For S^c:

$$b_2 = \sum_{\text{states in } S^c} \text{distinguish}(s^c)$$

But if $s_i^c \notin S$, then $\mu(O, s_i^c)$ is undefined—the observer cannot form the probabilistic measure needed for distinguishability.

(3) The Budget Contradiction

We require:

$$b_1 + b_2 + b_3 \leq b_{\text{obs}} \leq B_{\text{sys}}$$

But b_2 requires distinguishing content not in S, which means b_2 references distinctions not available within budget B_{sys}.

Formally: Let $D(B)$ denote the set of distinguishable states at budget B. Then:

- $D(B_{\text{sys}})$ is finite (proven from Fin bounded)
- $S^c \not\subseteq D(B_{\text{sys}})$ (by definition of "outside")
- Therefore b_2 requires budget to distinguish elements not in $D(B_{\text{sys}})$
- But $b_{\text{obs}} \leq B_{\text{sys}}$ means observer can only access $D(B_{\text{sys}})$

Contradiction. The construction requires b_2 while simultaneously making b_2 unconstructable. □

7.3.3 The Ledger Consequence

From the impossibility of external observation, time cannot be an external parameter. The proof:

(1) Classical Time Requires External Indexing
Writing $S(t)$ presupposes:

- A vantage point from which to observe state S
- A parameter t independent of S
- The ability to survey the mapping $t \mapsto S(t)$

This is precisely the external observer position just proven impossible.

(2) Time Must Be Internal
From `void_time_memory_composition.v`:

```
Definition operation_cost : Fin := fs fz.
(* A tick is evidence of observable change *)
Inductive tick := | Tick : State -> State -> tick.
```

Time is the accumulated heat:

$$T = \bigoplus_{\text{operations}} \Delta_{\text{tick}}$$

From `void_arithmetic.v`, every operation generates:

```
match add_fin_heat n m b with
| (res, b', h) => (* h = operation_cost *)
```

Standard form:

$$h = \bigoplus_{\text{operations}} \text{operation\_cost} \quad \text{(recursive accumulation in Fin)}$$

Time is not the stage. Time is the receipt. The ledger of distinctions made.

7.3.4 The Only Escape: Actual Infinity

There is exactly one way to restore the external observer: posit actual infinity as primitive.

If we allow:

$$B_{\text{obs}} = \infty$$

Then the observer can:

- Make unbounded distinctions (b_1, b_2, b_3 can be arbitrarily large)
- Survey the entire system from "outside"
- Access completed infinite totalities

But this requires accepting:

1. Actual infinity exists (not as limit, but as completed object)
2. Infinite resources are physically realizable

3. The observer transcends thermodynamic constraints

From `void_finite_minimal.v`, this is explicitly rejected:

```
Parameter MAX : Z.
Axiom MAX_positive : (0 < MAX)%Z.
Axiom fin_bounded : forall n : Fin, (fin_to_Z_PROOF_ONLY n <= MAX)%Z.
```

The axiom is: Everything is bounded by MAX. This is not a computational limitation—it is the mathematical foundation.

7.3.5 The Forced Choice

We have proven:

$$\text{Finite System} \implies \text{No External Observer} \implies \text{Time as Ledger}$$

The contrapositive:

$$\text{External Observer} \implies \text{Infinite Resources} \implies \text{Actual Infinity}$$

There is no third option. You cannot:

- Remain agnostic about infinity while using $S(t)$
- Use phase space methods without committing to transcendent viewpoint
- Write ensemble averages without believing in completed infinities

The dichotomy is:

Position	**Commitment**	**Mathematics**
Immanence	Universe is finite	VOID (Fin, Budget, Heat)
Transcendence	Actual ∞ exists	Classical ($\mathbb{R}, \mathbb{N}$, completed sets)
“Agnostic”	(Incoherent)	Uses ∞ while denying commitment

This "agnostic" position—often defended as pragmatic instrumentalism—amounts to a form of **ontological free-riding**. By invoking the continuum to solve discrete problems, scientists incur a hidden metaphysical debt: they borrow infinite precision to simplify their equations, only to be surprised when this debt comes due in the form of singularities, divergences, and uncomputable edge cases. VOID acts as a liquidation of this debt. We trade the sleek, impossible elegance of the infinite for the rugged solvency of the finite. In VOID, there are no "neutral tools"; there are only audited operations.

This proof shows: **You are making metaphysical claims**. Every time you write $\psi(x,t)$ and integrate over all space, you are claiming an observer position that (if the universe is finite) cannot exist.

7.3.6 Summary: The Impossibility in Standard Form

Definition 7.3. A system S is *finite* if $\exists \text{MAX} < \infty$ such that all budgets $B \leq \text{MAX}$.

Theorem 7.4 (No External Observation)**.** *For any finite system S and observer $O \subseteq S$, the construction of "outside" S^c is impossible.*

Proof. Budget arithmetic. Distinguishing S^c costs b_2 where b_2 references content $\notin D(B_{\text{obs}})$. But $b_{\text{obs}} \leq B_{\text{sys}}$ by definition. Contradiction. □

[Time as Ledger] If external observation is impossible, then time cannot be external parameter.

$$\text{Time} = \bigoplus \Delta_{\text{tick}} = \text{accumulated heat of maintained distinctions}$$

[Immanence is Forced] The only mathematics coherent for finite systems is one where all observation is internal: patterns observing patterns, with no meta-level.

The choice: Believe in actual infinity, or accept immanence. Neutrality is not available.

This is not incremental progress. This is not "a new perspective." This is a limit theorem that invalidates the conceptual foundations of mathematical physics as practiced for four centuries—unless you explicitly commit to Platonism and actual infinity as real.

If you study a finite universe, you cannot coherently use the mathematics of the external observer. Not because it's hard. Because it's proven impossible.

7.3.7 Time: A Different Picture

We often think of time in terms of a dimension—resembling a flowing river—in which we, the observers, and the surrounding space are submerged. This metaphor tries to kill two birds with one stone: maintain the objectivity of a material river as a phenomenon that can be described quantitatively, while leaving room for subjective experience—the feeling of water flowing against the skin. Here, we invite you to abandon this image for something less intuitive—bearing in mind that intuition itself is largely socially constructed, changing as cultures develop and introduce new "truths." Imagine you are not a character standing beside a river (time), watching the water move past, but a torchbearer wandering a labyrinth at night. The light you carry—your budget—lets you illuminate features and passages, keep pathways clear, and distinguish forks. Every step, every distinction, every choice consumes a little of your fuel; darkness rushes in behind. There is no vantage "above" the labyrinth, no completed map—only what can be built, lit, and maintained within what you carry. If you meet another torchbearer, they have their own sphere of light; their map is not the same. The "river of time" is not flowing past; what accumulates is the soot and fading heat in your lamp—a ledger of all you've lit, each distinction retained only by continuous effort. To imagine seeing it all—inside and out, in a single glance—would require not a larger lamp, but an infinite one: more light than could ever be carried, more fuel than could ever be stored. For a finite torchbearer, the world is what remains within the circle of maintained light—the rest is not darkness, but unimaginable, even in principle.

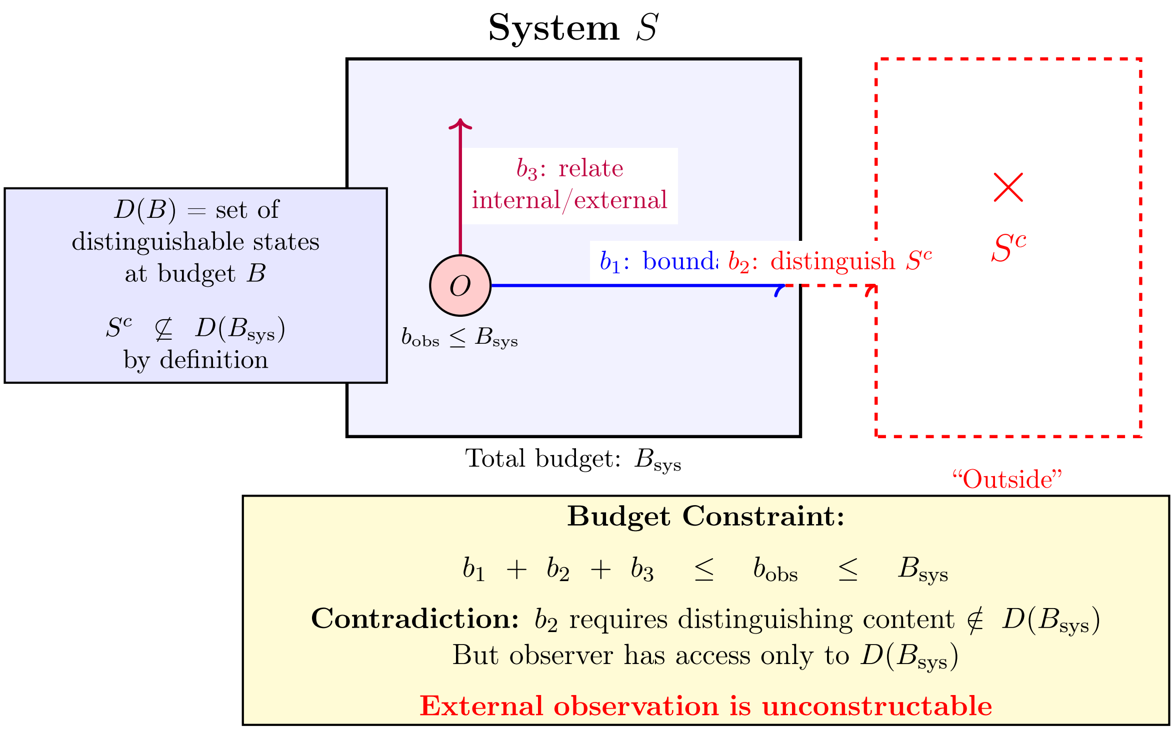


Figure 9: **The Impossibility of External Observation.** An observer O embedded within finite system S cannot construct a view "from outside." Required distinctions cost: b_1 (system boundary), b_2 (external content), b_3 (internal/external relation). The observer's budget $b_{\mathrm{obs}} \leq B_{\mathrm{sys}}$ cannot access S^c because b_2 references states outside $D(B_{\mathrm{sys}})$. The construction fails by budget arithmetic—not epistemic limitation but mathematical impossibility.

7.4 Budget Flows and Heat Currents

A flow is not an abstract vector field but a finite program that redistributes budget across the quotient $C_{(\rho,B)}$. Unlike Hamiltonian mechanics [56] where energy sloshes between kinetic and potential forms conservatively, here every transfer dissipates heat irreversibly. Think of a stepwise trace:

$$S_0 \xrightarrow{h_1} S_1 \xrightarrow{h_2} \cdots \xrightarrow{h_n} S_n$$

each arrow an admissible operation with emitted heat h_i. Time on this trace is the monoid sum $T = h_1 \oplus \cdots \oplus h_n$; the ledger closes as $B_0 = B_n \oplus T$.

You may also collect simultaneous transfers: a budget-flow F is a finite family of localized transfers $\{\tau_i\}$ with declared sources/sinks and rates $r_i \in P_\rho^\circ$. The rate r_i indicates the probability that transfer τ_i executes successfully; the support $\mathrm{support}(\tau_i)$ is the set of spatial locations (equivalence classes in $C_{(\rho,B)}$) where the transfer operates; the cost function $c(\tau_i, s)$ gives the heat dissipated at location s when transfer τ_i executes. Executing F for one tick costs:

$$h_F = \bigoplus_i \bigoplus_{s \in \mathrm{support}(\tau_i)} c(\tau_i, s), \quad \text{with } c(\tau_i, s) \succeq \mu(\rho)$$

Example 7.5 (Simultaneous Transfers)**.** Three patterns refresh in parallel at resolution ρ:

- Transfer τ_1: refresh pattern p_1 at location $[x]$, costs $c(\tau_1, [x]) = 3\mu(\rho)$
- Transfer τ_2: refresh pattern p_2 at location $[y]$, costs $c(\tau_2, [y]) = 2\mu(\rho)$
- Transfer τ_3: refresh pattern p_3 at location $[z]$, costs $c(\tau_3, [z]) = 4\mu(\rho)$

Total heat for one tick: $h_F = 3\mu \oplus 2\mu \oplus 4\mu = 9\mu$. Unlike sequential execution which would cost 9μ spread over time, simultaneous execution pays 9μ in one tick—higher instantaneous dissipation but faster completion.

Axiom 25 (Flow Conservation and Dissipation)**.** For any flow F with total heat emission h_F, the ledger closes:

$$B_{\text{total}} = B'_{\text{total}} \oplus h_F$$

where $h_F = 0$ if all transfers are budget-neutral reads, otherwise $h_F \geq \mu(\rho)$. Heat accumulates locally at sites of work, producing thermal imbalances: regions with high cumulative heat h_{accum} experience attenuated distinguishability via thermal decay (formalized in `void_pattern_thermo.v`).

This resembles Fourier's heat equation [42] but with discrete, budgeted transfers rather than continuous diffusion.

Micro-scene: Cache a feature once (heat $\mu(\rho)$); read it a hundred times (heat 0); erase it (heat $\mu(\rho) + \epsilon$). The cycle dissipates $\mu(\rho) \oplus \mu(\rho) \oplus \epsilon$. There is no clever loop that nets clarity for free.

7.5 Entropy, Finitely: Coding Cost and Path Accumulation

We refuse to import a continuum to speak of uncertainty. While Shannon used $-\sum p \log p$ [108] and Boltzmann used $k \log W$ [14], both assuming infinite precision, we define entropy as the minimum heat to identify.

The scaling operator. Before defining entropy, we introduce the budget-scaling operator:

$$\otimes_h : P_\rho^\circ \times B \to B$$

This operator scales budget by probability. For probability $p = n/d$ and budget b:

$$p \otimes_h b = \left\lfloor \frac{n \cdot b}{d} \right\rfloor$$

Example 7.6. Probability 3/5 applied to budget 10μ gives $(3/5) \otimes_h 10\mu = \lfloor 30/5 \rfloor \mu = 6\mu$. A test that succeeds with probability 3/5 and costs 10μ contributes 6μ to expected heat.

At resolution ρ, let E_ρ be the finite event structure with valuation $P_\rho : E_\rho \to P_\rho^\circ$. A code is a finite decision tree whose internal tests are admissible; each branch consumes heat equal to the work of the tests it executes. Define:

$$H_\rho(P_\rho) = \min_{\text{codes on } E_\rho} \mathbb{E}_{P_\rho}[\text{heat}]$$

where the expected heat over all branches is:

$$\mathbb{E}_{P_\rho}[\text{heat}] = \bigoplus_{\ell} P_\rho(\ell) \otimes_h \text{heat}(\ell)$$

Example 7.7 (Decision Tree Entropy)**.** Consider event space $E_\rho = \{e_1, e_2, e_3\}$ with probabilities $P_\rho(e_1) = 5/11$, $P_\rho(e_2) = 3/11$, $P_\rho(e_3) = 2/11$ (note: $5/11 + 3/11 + 2/11 = 10/11 < 1$, staying interior to $(0, 1)$).

Root node branches to test A (left, cost μ) leading to e_1, and test B (right, cost μ) which branches to test C (cost μ, leading to e_2) and e_3.

Branch costs:

- Branch to e_1: 1 test, heat $= \mu(\rho)$
- Branch to e_2: 2 tests, heat $= 2\mu(\rho)$
- Branch to e_3: 1 test, heat $= \mu(\rho)$

Expected heat:

$$\mathbb{E}[\text{heat}] = (5/11) \otimes_h \mu \oplus (3/11) \otimes_h 2\mu \oplus (2/11) \otimes_h \mu$$

Computing (assuming $\mu = 11$ for concreteness):

$$\begin{aligned} &= \lfloor 5/11 \cdot 11 \rfloor \oplus \lfloor 3/11 \cdot 22 \rfloor \oplus \lfloor 2/11 \cdot 11 \rfloor \\ &= 5 \oplus 6 \oplus 2 = 13\mu \end{aligned}$$

This tree's entropy is $H_\rho = 13\mu$ (if this is the optimal tree). A worse tree (testing in different order) might cost 15μ or more. Entropy measures the thermodynamic floor—best possible identification cost.

No logs, no limits: entropy is least heat to identify. It lives in the same resource monoid as budget.[9] This operationalizes Kolmogorov's insight [76] that information content is program length, but with thermodynamic cost replacing abstract complexity.

[9] Implemented in `void_entropy.v` as `entropy_b` which counts non-zero elements with budget tracking. The minimum-heat-to-identify interpretation connects to algorithmic information theory but with physical costs replacing abstract complexity.

Along a concrete run $\pi : S_0 \to S_1 \to \cdots \to S_n$ with heats $h_1, \dots, h_n$, we can speak of **path entropy**—a normalized expenditure. Using a declared normalization $\nu_\rho : B_{>0} \to P_\rho^\circ$:

$$\text{Entropy}(\pi) := \nu_\rho(h_1 \oplus \cdots \oplus h_n) \in P_\rho^\circ$$

The normalization ν_ρ maps accumulated heat into the probability scale, typically by dividing by maximum budget: $\nu_\rho(h) = h/B_{\max}$ when both are expressed in compatible units.

Example 7.8. With $B_{\max} = 100\mu$ and cumulative heat $h = 47\mu$, path entropy is $\nu_\rho(47\mu) = 47/100 \in P_\rho^\circ$—nearly half the available budget dissipated.

Degradation under spend. High local dissipation coarsens perception. If a region's cumulative heat exceeds threshold $\theta_{\text{ent}}(\rho)$, its kernel attenuates:

$$\Delta_S^{(\rho,B')}(x, y) \preceq a \otimes_p \Delta_S^{(\rho,B)}(x, y) \quad \text{with } a \prec 1^\circ$$

Example 7.9 (Thermal Degradation)**.** Two patterns initially distinguished at $\Delta^{(\rho,B)}(x, y) = 7/10$. After executing operations that dissipate cumulative heat $h_{\text{accum}} = 45\mu$ in this region (exceeding $\theta_{\text{ent}} = 40\mu$), the distinguishability degrades:

$$\Delta^{(\rho,B')}(x, y) = (5/7) \otimes_p (7/10) = \frac{5 \cdot 7}{7 \cdot 10} = \frac{1}{2}$$

The attenuation factor $a = 5/7 \approx 0.71$ reflects thermal noise from dissipated work. What was moderately separated (7/10) becomes barely distinguishable (1/2). If heat continues accumulating, Δ eventually falls below threshold $\theta(\rho) = 3/10$, causing patterns to collapse entirely.

This models Carnot's insight [23] that heat degrades work capacity, but at the perceptual level.

Axiom 26 (Predictive Economy)**.**

$$\bigoplus_{\Pi \text{ active}} \text{maintenance\_cost}(\Pi) \preceq B_{\text{available}} \ominus B_{\text{operations}}$$

This resembles Friston's free energy principle [45]—organisms minimize surprise by maintaining predictive models—but with explicit thermodynamic accounting.

Example 7.10. Consider maintaining two predictive patterns at $\rho = 2$ with $D(\rho) = 10$. Pattern Π_1 (weather forecast) costs $5\mu(\rho)$ per tick; Π_2 (traffic prediction) costs $3\mu(\rho)$. Total obligation: $8\mu(\rho)$. With $B_{\text{available}} = 20\mu(\rho)$ and $B_{\text{operations}} = 6\mu(\rho)$, we have $B_{\text{free}} = 6\mu(\rho)$, insufficient for both. Must choose: drop Π_2 or coarsen Π_1 to reduce its cost. The system cannot maintain all predictions at full fidelity—a forced tradeoff.

Micro-scene (Boundary drift): Two features maintain a decision boundary at cost $2\mu(\rho)$ per tick but yield U on 40% of tests (insufficient resolution). Adding a third feature improves decisions (U only 10%) but triples maintenance to $6\mu(\rho)$. Alternative: expensive one-time rewrite (merge + reindex) costing $20\mu(\rho)$ now, yielding two new features at $3\mu(\rho)$ per tick with U at 15%. Amortization: rewrite pays off after $20\mu(\rho)/(6\mu(\rho) - 3\mu(\rho)) \approx 7$ ticks.

7.6 Time as Integrated Heat

For any trajectory π:

$$T(\pi) = h_1 \oplus \cdots \oplus h_n$$

Our treatment of time recalls in this regard Prigogine's identification of time with irreversible processes [99], but makes it mathematically precise through the resource monoid.

Axiom 27 (Time Irreversibility)**.** $T(\pi) = 0$ if π has no priced steps. The arrow of time is the monotonicity of spend.[10]

Even the identity has a thermodynamic face. As a function it preserves distances; but maintaining the underlying distinctions over any interval costs budget to counter drift. The ledger records that maintenance explicitly—connecting to Schrödinger's observation [106] that life maintains order by exporting entropy.

7.7 Transport, Resonance, Interference

Transport. Budget moves because work is local. Transfer from region A (low marginal gain) to B (high marginal gain) continues while the benefit exceeds transfer cost.

Example 7.11. Pattern Π_A at location $[x]$ has maintenance cost 3μ but provides information value 2μ (net loss: 1μ). Pattern Π_B at location $[y]$ has maintenance cost 3μ but provides value 5μ (net gain: 2μ). Transferring budget from A to B (cost: 1μ for transfer operation) yields net improvement: stop maintaining Π_A (save 3μ), pay transfer (1μ), boost Π_B (net gain increases). Transport continues until marginal gains equalize.

This relates to Onsager's reciprocal relations [94] from non-equilibrium thermodynamics.[11]

Resonance. Maintenance tasks can share subroutines. Two patterns Π_1, Π_2 resonate with strength characterized by attenuation factor $a_\sigma \in P_\rho^\circ$ when:

$$\text{maintenance\_cost}(\Pi_1 \cup \Pi_2) \preceq \text{maintenance\_cost}(\Pi_1) \oplus a_\sigma \otimes_h \text{maintenance\_cost}(\Pi_2)$$

where $a_\sigma < 1^\circ$ indicates savings from shared work.

In 1665 Huygens discovered [68] that pendulum clocks on the same wall synchronize—not through mystical sympathy but through tiny vibrations transmitted through the wood. The synchronized state, an "odd sympathy", as he called it, requires less total energy to maintain than independent oscillations fighting each other. Our resonance captures this: when patterns share maintenance routines (like two features using the same discrimination test), maintaining both together costs less than maintaining them separately. The attenuation factor a_σ quantifies this savings—how much work overlaps. Unlike Huygens' continuous oscillators that find exact synchrony, our patterns resonate at discrete frequencies determined by the test basis B_ρ.

[10] The irreversibility axiom is enforced in `void_finite_minimal.v` through heat conservation axioms which guarantee $B = B' \oplus h$ with $h \geq 0$. Time reversal would require negative heat, excluded by construction.

[11] Onsager showed that near equilibrium, flow coefficients are symmetric ($J_1/F_2 = J_2/F_1$). Our framework differs: flows are discrete, priced at $\geq \mu(\rho)$, creating thresholds below which no flow occurs. Round-trips dissipate $h_{\text{out}} \oplus h_{\text{back}} > 0$.

Interference. Align tests so shared work reduces marginal heat. Periodic work admits resonant schedules: refresh in phase with natural decay to minimize average heat per period.

Feynman's path integral formulation [39] sums over paths weighted by $e^{iS/\hbar}$. In our framework, paths carry heat signatures instead of complex phases.[12]

7.8 Exhaustion and Undecidability

When budget cannot sustain a discrimination test, the system returns U for mutually exclusive alternatives. This is not indeterminacy but honest admission of resource limits.

Example 7.12 (Coexisting Alternatives)**.** Two patterns Π_A and Π_B represent alternative classifications for the same stimulus. The discrimination test costs $D(A, B) = 15\mu$. With budget $B = 10\mu$ (below discrimination threshold), both patterns remain in the system: queries "is it A?" and "is it B?" both return U—not because the answer doesn't exist but because determining it exceeds available resources.

As budget increases to $B' = 20\mu$, the discrimination test becomes affordable. Executing it costs 15μ, dissipates that heat irreversibly, and collapses the superposition: one pattern wins, the other is discarded. This is the "measurement"—a priced operation that forces classical determination.[13]

This explains why certain observations are mutually exclusive: when two tests each cost C and available budget is $B < 2C$, you must choose which test to execute. The mutual exclusivity isn't a law of nature—it's resource arithmetic. If test T_1 costs B_1 and test T_2 costs B_2, they are jointly affordable only when $B_1 + B_2 \le B_{\text{available}}$.

The "measurement problem" dissolves into bookkeeping: U persists until discrimination cost is met; collapse is paid deletion of unaffordable alternatives. Coexistence of alternatives is computational poverty, not metaphysical mystery.

This convergence with physics is not accidental; it is the inevitable consequence of treating mathematical operations as energetic acts. By introducing a cost to distinction, VOID collapses the artificial separation between the logical abstract and the physical concrete.

It provides us with a resource-based interpretation of Bohr's complementarity [12]: incompatible observations aren't philosophically forbidden but thermodynamically unaffordable simultaneously. Measuring position precisely (costs B_x) exhausts budget needed for momentum measurement (costs B_p). If $B_x + B_p > B_{\text{available}}$, you must choose—not because nature forbids joint measurement but because you cannot afford both.

Theorem 7.13 (Thermodynamic Irreversibility)**.** *For any sequence of priced operations transforming state $S_0 \to S_1 \to \cdots \to S_n$, the total accumulated heat T is strictly monotonically increasing.*

Proof. Let h_i be the heat generated at step i. By the cost axiom (Axiom 6), every state-changing operation generates $h_i \succeq \mu(\rho) \succ 0$. The total accumulated heat is $H_n = \bigoplus_{i=1}^{n} h_i$. Since $\forall i, h_i \neq \mathsf{fz}$ for priced operations and $\oplus$ is monotonic on Fin, we have $H_n \succ H_{n-1}$. Reversing the system would require an operation with negative heat $h \prec 0$ to restore budget ($B' = B \oplus h$), but Heat := Fin admits no negative values. The arrow of time is structurally enforced. □ □

[12] Key difference: Feynman's amplitudes can perfectly cancel (destructive interference → zero probability). Our heat costs only accumulate—no negative heat exists. Incompatible paths don't cancel; they both become unaffordable, returning U. This means computation leaves permanent thermal traces, and optimization gets stuck in local minima because exploration costs budget.

[13] Formalized in `void_observer_collapse.v`. The kernel Δ_S computes separation; if below threshold $\theta(\rho)$, result is U. Measurement pays $h \ge \mu(\rho)$ to force separation above threshold.

7.9 Asymmetric Decoupling Under Resource Scarcity

Consider coupled observers A and B maintaining a shared pattern Π_{AB} with mutual maintenance cost $\mathcal{M}_\rho(\Pi_{AB})$. This framework provides tools for analyzing resource dynamics in coupled systems—including interpersonal configurations where asymmetric cost externalization occurs. Under finite-budget assumptions, such dynamics acquire precise thermodynamic characterization, suggesting that even ethical categories may admit operational definitions grounded in conservation laws. **The finite framework redirects analysis from unbounded abstractions**—eternal, immutable, absolute—toward discrete material acts with computable costs. **Honesty and deception become thermodynamically distinguishable**: the former minimizes verification overhead, **the latter (A's activity) externalizes heat onto the deceived party**. This explains ethical asymmetries in conservation laws rather than infinite penalties.

Definition 7.14 (Unilateral Severance)**.** Observer B performs *unilateral severance* when B ceases maintaining Π_{AB} without emitting a termination signal to A. If B simultaneously initiates Π_{BC} with observer C, the severance is *masked*.

Definition 7.15 (Decoy Signal)**.** A *decoy signal* S_d is a maintenance signal for Π_{AB} emitted by B while B's actual budget allocation satisfies $\mathcal{M}_B(\Pi_{BC}) \succ \mathcal{M}_B(\Pi_{AB})$.

When A operates near exhaustion threshold ($A_{\text{total}} \approx A_{\text{critical}}$), decoy signals force verification costs:

$$h_{\text{verify}} = h_{\text{check}}(\Pi_{AB}) \oplus h_{\text{interpret}}(S_d)$$

Theorem 7.16 (Asymmetric Severance Cost)**.** *If B performs masked unilateral severance while A satisfies $A_{total} \preceq A_{critical} \oplus h_{verify}$, then A enters forced deficit:*

$$\mathcal{M}_B(\text{internal}) = \mathcal{M}_B(\text{internal}) \ominus h_{verify} \prec \theta_{\text{survival}}$$

Proof. By conservation, A cannot simultaneously fund h_{verify} and $\mathcal{M}_A(\text{internal})$ when total budget is bounded by $A_{\text{critical}} \oplus h_{\text{verify}}$. The deficit h_{verify} must be paid from internal maintenance allocation. When internal maintenance drops below θ_{survival}, structural degradation follows by Theorem 8.26. □ □

Interpretation. The severance becomes thermodynamically *asymmetric*: B's transition cost approaches zero while A absorbs the full dissipation burden. This represents externalization of heat production—one observer's state change paid from another's survival budget accelerating termination of Observer B.

7.10 What We Have Built

These principles constitute a mathematics where dynamics is fundamentally discrete, priced, and irreversible.

Motion without infinity. Classical trajectories assume a particle moves through infinitely many intermediate points. We have shown this is impossible within our framework and this should be considered a feature, not a bug: the smallest distinguishable step costs $\mu(\rho)$. Motion becomes a sequence of paid jumps from location ℓ_1 to location ℓ_2, each costing $\mu(\rho)$. What looks like "smooth motion" is exhaustion—when distinguishing intermediate positions exceeds budget, discrete jumps become indistinguishable.

We stay in $\mathbb{Q}$ via FinProb—no completion to $\mathbb{R}$ needed. Classical physics demands real numbers for continuity; we avoid Dedekind cuts, Cauchy sequences, and the axiom machinery required to construct $\mathbb{R}$ from $\mathbb{Q}$. All coordinates, velocities, accelerations live in finite-precision rationals.

Memory as active maintenance. Stored distinctions decay unless refreshed. Every pattern in memory M carries a maintenance obligation: dissipated budget per tick to prevent strength dropping below threshold. Memory is an ongoing decision about which patterns are worth keeping alive. When budget depletes, patterns must be triaged. Forgetting is mathematical necessity when maintenance obligations exceed resources.

Flows as priced redistribution. Classical flows assume frictionless energy transport. Theorem 7.3.2 proved this impossible: every budget transfer dissipates heat $h \geq \mu(\rho)$. Flow becomes a sequence: identify source (costs $\mu(\rho)$), verify destination (costs $\mu(\rho)$), execute transfer (costs $\mu(\rho)$). Flow continues while marginal benefit exceeds transfer cost, then stops from thermodynamic unprofitability.

Entropy as identification cost. Shannon's $H = -\sum p \log p$ assumes infinite precision. We defined entropy as minimum heat to identify which event occurred: build optimal decision tree, weight branches by probability, sum costs. This lives in resource monoid B, counted in ticks, no logarithms. High entropy means many tests required; low entropy means few tests suffice.

Prediction as obligation schedule. Each predictive pattern costs maintenance heat per tick. Your prediction system is the set of patterns you're funding. Failed predictions waste budget; accurate predictions earn their keep. The system must choose: maintain failing predictions (loyalty) or explore alternatives (adaptation). Prediction is an active maintenance schedule, each with a price.

Symmetry as expensive achievement. Maintaining symmetry requires continuous verification: for n elements, $O(n^2)$ pairwise checks, each costing $\mu(\rho)$. Symmetry doesn't persist automatically—it requires active enforcement against drift. High-symmetry systems are expensive, affordable only with large budgets. As resources deplete, symmetry breaks because the system cannot afford maintenance.

Arrow of time as irreversible dissipation. Time flows forward because dissipated budget cannot reconstitute as available budget. Every priced operation converts B into B' plus h, with $B = B' \oplus h$. Reversing would require $h \oplus B' = B$—dissipated budget reconstituting as available budget. But dissipation is irreversible by construction. The ledger accumulates monotonically: $T(\pi) = h_1 \oplus \cdots \oplus h_n$ grows because each $h_i > 0$. Time is the receipt. Space is the workshop. Entropy is the bill. What endures is what we keep funding; what fades is what we refuse to maintain. The abyss remains; the scaffold grows—one priced step at a time.

Boundaries of observation and internalism. This framework proves that external, cost-free observation is impossible in a finite universe (universe constructed as a finite system). Every act of distinction, detection, or measurement is a local, budget-consuming intervention that irreversibly shapes what can be known. There is no God's-eye perspective: each observer is embedded in the system, transforming it through finite, priced acts of recognition. Every equivalence, classification, and perceived regularity emerges not from detached witnessing, but from resource-limited participation;

as budget is exhausted, fine distinctions collapse, and only coarser, more affordable perspectives persist. The mathematics of observation is thus fundamentally internalist, but not solipsistic, grounded in what can actually be built and distinguished from within, not in what could, in principle, be named by an idealized outside view.

Coq Verification. Core temporal dynamics formalized with thermodynamic accounting:

- `void_time_memory_composition.v` — Time as tick accumulation, memory decay, ledger identity $B_0 = B_n \oplus T(\mathrm{tr})$
- `void_budget_flow.v` — Budget redistribution with mandatory dissipation, flow costs
- `void_entropy.v` — Entropy as counting cost, minimum heat to identify, no logarithms
- `void_entropy_integration.v` — Observer-dependent entropy, resolution effects
- `void_process_memory.v` — Pattern regeneration with fidelity decay
- `void_pattern_thermo.v` — Heat generation, thermal decay, success probability
- `void_resonance.v` — Pattern resonance, maintenance cost sharing
- `void_observer_collapse.v` — Measurement as paid decision
- `void_crisis_relocation.v` — Crisis-driven dynamics
- `void_phase_orbits.v` — Deterministic trajectories

All implementations maintain conservation $B = B' \oplus h$ and return probabilities in P_ρ°. No module assumes continuous time, conservative flows, or zero-cost operations.

8 Time, Memory, and Pattern Dynamics

Having established time as integrated heat and budget flows in Section 7, we now examine how patterns persist through active maintenance, how concurrent processes share or compete for resources, and how symmetries require continuous investment. This section applies Section 7's primitives to show emergent dynamics unavailable in classical frameworks, because finite budgets and discrete thresholds make all transitions exactly computable—consolidation break-even points, oscillation periods, and crystallization conditions become arithmetic facts rather than asymptotic approximations.

8.1 Memory as Maintained Distinguishability

Operational Perspective. Classical memory models treat remembering as retrieval from persistent storage—a database query that accesses pre-existing traces. This framework rejects storage primitives entirely. Memory is not stored data but actively maintained distinctions. A pattern π is "remembered" precisely when the system can afford to maintain $\Delta_S(\pi, \cdot) \succeq \theta(\rho)$ through continuous refresh operations costing $\mathcal{M}_\rho(\pi)$ per grace period $g_\rho(\pi)$.

This redefinition follows from conservation: maintaining distinction requires budget expenditure, but this cost is not paid continuously. Instead, memory operates on a **Lazy Decay** principle. Patterns are not actively degraded every tick (which would be energetically prohibitive); rather, they carry a timestamp of validity. The decay operator δ is applied retroactively only upon access. If the elapsed time exceeds the pattern's grace period without a refresh event, the fetch operation returns U. "Forgetting" is thus a passive expiration of validity, not an active deletion process. The system does not "lose" a memory; it simply declines to pay the retroactive cost of reconstructing it against the noise floor.

This makes memory fundamentally economic: the question is never "do we have this stored?" but "can we afford to maintain this distinction given current budget and competing obligations?" A "remembered" pattern is one whose distinguishability can be reconstructed within current budget constraints, either through direct refresh or through parasitic maintenance on other patterns. Memory capacity is not storage size but maintenance budget—how many patterns can be kept above threshold simultaneously.

The Mechanics of Persistence. A pattern π is not a static object but a thermodynamic struggle. Left alone, it decays below the observability threshold $\theta(\rho)$. To prevent this, the system must perform a *refresh operation*—an active injection of budget to counter entropy. This leads to a precise definition of maintenance cost: not as an arbitrary tax, but as the minimal energy required to keep the pattern distinguishable from noise.

Definition 8.1 (Maintenance Function)**.** Let π be a pattern with current distinguishability $\Delta_S^{(\rho,B)}(\pi, \cdot)$. The *maintenance requirement* $\mathcal{M}_\rho(\pi)$ is the infimum of budget required to restore visibility:

$$\mathcal{M}_\rho(\pi) \;:=\; \inf\{\, h \in B \mid \mathrm{refresh}(\pi, h) \to \Delta' \succeq \theta(\rho) \,\}.$$

This defines the cost of existence as an optimization problem: what is the smallest amount of heat h the system must dissipate to prevent π from collapsing into U?

Theorem 8.2 (Maintenance Necessity)**.** *Without refresh for k ticks, distinguishability degrades through iteration:*

$$\Delta_S^{(k)}(\pi, \pi') \;=\; \underbrace{\delta \circ \delta \circ \cdots \circ \delta}_{k\ \text{times}}\big(\Delta_S^{(0)}(\pi, \pi')\big),$$

where $\delta : P_\rho^\circ \to P_\rho^\circ$ is the decay operator satisfying $\delta(p) \preceq p$ for all $p \neq U$.

Proof. By operational semantics, each tick without refresh applies δ once. The conservation law $B = B' \oplus h$ requires that distinguishability cannot increase without budget expenditure; hence δ produces monotone degradation under composition. $\square$

Intuition: The Spinning Plate Analogy. Imagine memory not as books stored on a library shelf (which stay there for free), but as spinning plates on poles. A plate stays up only as long as you periodically tap it to inject energy. If you stop paying attention, the plate doesn't "fade away" linearly; it spins until it loses critical momentum and then crashes catastrophically. In VOID, "forgetting" is not data loss—it is simply the economic decision to stop spending energy on a specific plate.

8.1.1 Discrete Threshold Dynamics

Decay operates through discrete threshold crossing, not continuous degradation. Classical models use exponential decay $e^{-\lambda t}$; here, patterns remain sharp until accumulated heat crosses $\theta(\rho)$, then collapse discretely to U. This creates **grace intervals**—finite periods of zero maintenance cost followed by sudden collapse, rather than gradual degradation.

Definition 8.3 (Grace Period)**.** Each pattern admits a *grace period*

$$g_\rho(\pi) \ := \ \max\left\{ k \ \middle|\ \delta^{(k)}\big(\Delta_S(\pi, \cdot)\big) \succeq \theta(\rho) \right\}$$

during which no maintenance is required.

Contrast with Continuous Models. Standard memory models use exponential decay $\exp(-\lambda t)$, creating gradual degradation. Here, $\delta^{(k)}$ produces a discrete sequence until threshold crossing. This is not approximation but a consequence of finite computation: either sufficient budget exists to maintain distinction or it does not—no intermediate "partially maintained" state.

Example 8.4 (Grace Period Computation)**.** Let $\Delta_S(\pi, \pi') = \frac{3}{4}$, $\theta(\rho) = \frac{1}{2}$, and let $\delta\left(\frac{n}{d}\right) = \frac{n-1}{d}$ when $n \succ 1$. Then

$$\frac{3}{4} \xrightarrow{\delta} \frac{2}{4} \xrightarrow{\delta} \frac{1}{4} \prec \frac{1}{2},$$

so $g_\rho(\pi) = 2$ ticks before collapse.

What this enables: **refresh batching**. Since patterns do not degrade gradually, you can wait until the last possible tick before refresh, then batch multiple refreshes together. Classical systems with continuous decay must refresh continuously; discrete thresholds permit strategic timing.

Coq Verification. `void_time_memory_composition.v` implements δ as strength reduction in `MemoryTrace` records, with `decay_with_budget` enforcing the conservation constraint. `void_pattern_thermo.v` implements threshold-crossing semantics in `thermal_decay`, with grace periods tracked through `read_trace_forgotten` predicates.

8.2 Consolidation Economics

Classical Limits. Memory optimization in classical frameworks relies on heuristics: cache replacement policies (LRU, LFU) have no provable optimality, garbage collection triggers use empirical thresholds, and compression algorithms trade space for time without exact break-even analysis. The question "should we consolidate?" requires profiling, benchmarking, and tuning—no closed-form answer exists because memory operations have no explicit cost structure.

Exact Break-Even from Conservation. Our framework makes the consolidation decision mathematically exact. The conservation law $B = B' \oplus h$ forces every operation to declare its cost explicitly. This enables calculation of the precise tick t^* where consolidation investment pays off—not an estimate, not an asymptotic bound, but the exact integer where cumulative savings exceed rewrite cost.

Definition 8.5 (Maintenance Obligation)**.** For pattern set $\Pi = \{\pi_1, \ldots, \pi_n\}$, the total maintenance obligation is

$$\mathcal{O}_\rho(\Pi) \;:=\; \bigoplus_{i=1}^{n} \frac{\mathcal{M}_\rho(\pi_i)}{g_\rho(\pi_i)} \quad \text{(budget units per tick).}$$

Theorem 8.6 (Consolidation Criterion)**.** *The pattern set* Π *should be replaced by* Π' *if and only if:*

$$h_{rewrite} \prec \big(\mathcal{O}_\rho(\Pi) \ominus \mathcal{O}_\rho(\Pi')\big) \otimes t_{expected}$$

where $h_{rewrite}$ *is the one-time consolidation cost and* $t_{expected}$ *is the remaining operational lifespan.*

Proof. By the conservation axiom $B = B' \oplus h$, the break-even analysis is exact. Consolidation is thermodynamically profitable precisely when the future accumulated savings exceed the rewrite expenditure. Let $\Delta = \mathcal{O}_\rho(\Pi) \ominus \mathcal{O}_\rho(\Pi')$ be the savings per tick. Assuming $\Delta \succ 0$, the break-even point occurs at tick $t^* = h_{\text{rewrite}} \oslash \Delta$. If $t_{\text{expected}} \succ t^*$, the total heat saved exceeds the heat invested. Since all terms belong to Fin, this inequality is decidable through finite arithmetic—no approximation or asymptotic limits enter the calculation. □

Example 8.7 (Weather Forecast Consolidation)**.** Three patterns $\{f_1, f_2, f_3\}$ maintaining forecasts cost $12\mu(\rho)$ per tick. Rewriting to $\{g_1, g_2\}$ using shared features costs a one-time $50\mu(\rho)$ but reduces maintenance to $7\mu(\rho)$.

- Savings: $\Delta = 12\mu - 7\mu = 5\mu(\rho)$ per tick.
- Break-even: $t^* = 50\mu \oslash 5\mu = 10$ ticks.

After 10 ticks, consolidation begins to yield net energy profit. At tick $t > 10$, the system has provably saved $5\mu(\rho) \otimes (t - 10)$ budget compared to the baseline.

What This Enables.

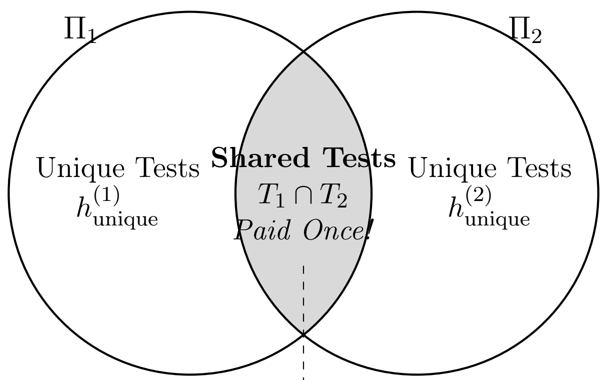
1. **Provable optimality.** Consolidation is optimal if the inequality holds—no heuristics, no "usually better in practice." The break-even point t^* is exact.
2. **Compile-time verification.** Given maintenance costs and expected lifespan, Coq can verify consolidation decisions statically before execution.

3. **Worst-case guarantees.** Classical systems optimize for average case; we can prove worst-case bounds because all costs are explicit. If $t_{\text{expected}} = t^* + k$, consolidation saves exactly $\Delta \otimes k$ budget—verifiable by conservation.

4. **Compositional reasoning.** Multiple consolidations compose: verify each independently, combine verified savings. Classical optimizations interact unpredictably; ours obey $B = B' \oplus h$ globally.

Forced Consolidation. If $\mathcal{O}_\rho(\Pi) \otimes t \succ B$, the system must consolidate or accept pattern loss. This reinterprets **sleep as consolidation necessity**. Organisms must periodically halt new observations to rewrite memory into lower-obligation forms, not because of "memory pressure" but because $\text{obl}(S) \to B$ forces crisis. The system doesn't choose to consolidate; thermodynamic constraint compels it. This is formalized as the `Hibernate` strategy in `void_crisis_relocation.v`—when budget exhausts, patterns reduce activity to preserve resources for essential maintenance.

Coq Verification. `void_budget_flow.v` implements cooperative resource allocation in `cooperative_competition_b`, with `pattern_alliance_b` formalizing pattern consolidation through merging. `void_process_memory.v` provides pattern regeneration with fidelity decay tracking.



Cost Algebra:

$$\mathcal{M}(\Pi_1 \cup \Pi_2) = \mathcal{M}(\Pi_1) \oplus \mathcal{M}(\Pi_2) \ominus \underbrace{\text{Cost}(T_1 \cap T_2)}_{\text{Saved Heat}}$$

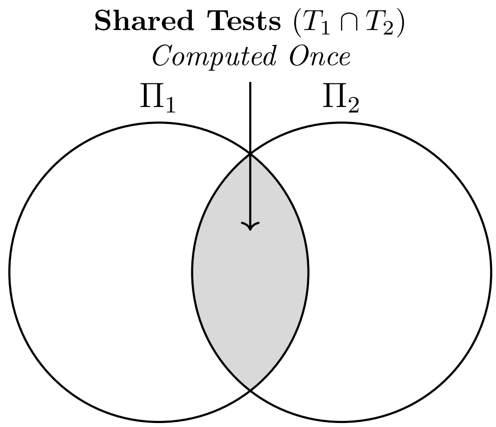
Figure 10: **Interference Routing.** Concurrent patterns Π_1 and Π_2 share a subset of discrimination tests ($T_1 \cap T_2$). In VOID Theory, this overlap is computed only once per tick, making the joint maintenance cost strictly less than the sum of individual costs. This thermodynamic saving drives pattern synchronization.

8.3 Interference as Test-Sharing Algebra

Operational Perspective. When multiple patterns must be maintained simultaneously, they compete for the same finite budget. The naive cost model assumes simple addition: maintaining π_1 costs $\mathcal{M}_\rho(\pi_1)$, maintaining π_2 costs $\mathcal{M}_\rho(\pi_2)$, so maintaining both costs $\mathcal{M}_\rho(\pi_1) \oplus \mathcal{M}_\rho(\pi_2)$. This is wrong.

Patterns do not “interfere” in the physical sense—they share computational substrate. Each pattern π maintains distinguishability by computing a test set $T_\pi \subseteq X \times X$ of pairwise comparisons: $(x, y) \in T_\pi$ means π requires knowing whether $\Delta_S(x, y) \succeq \theta(\rho)$. When patterns π_1, π_2 have overlapping test sets ($T_1 \cap T_2 \neq \emptyset$), the shared tests need only be computed once—their cost is paid jointly, not duplicated.

This creates **Interference Routing**: the system routes the maintenance budget through shared pathways. High overlap ($\sigma \to 1^\circ$) means efficient multiplexing: most tests are shared, so concurrent maintenance costs barely more than maintaining the more expensive pattern alone. Low overlap ($\sigma \to 0$) means patterns require disjoint computations, forcing expensive duplication.



Thermodynamic Saving:

$$\mathcal{M}(\Pi_1 \cup \Pi_2) = \mathcal{M}(\Pi_1) \oplus \mathcal{M}(\Pi_2) \ominus \mathrm{Cost}(T_1 \cap T_2)$$

Figure 11: **Interference Routing.** Overlapping test sets are computed only once per tick, creating a thermodynamic incentive for synchronization.

Definition 8.8 (Test Portfolio)**.** Pattern π maintains distinguishability via test set $T_\pi \subseteq X \times X$, where $(x, y) \in T_\pi$ means π requires $\Delta_S(x, y)$ to be computed.

Definition 8.9 (Overlap Coefficient)**.** For patterns π_1, π_2 with test sets T_1, T_2:

$$\sigma(\pi_1, \pi_2) := \frac{\#(T_1 \cap T_2)}{\#(T_1 \cup T_2)} \in [0, 1].$$

Theorem 8.10 (Cost Composition via Overlap)**.** *Joint maintenance cost satisfies:*

$$\mathcal{M}_\rho(\pi_1 \cup \pi_2) \preceq \mathcal{M}_\rho(\pi_1) \oplus \mathcal{M}_\rho(\pi_2) \otimes_p (1^\circ \ominus_p \sigma(\pi_1, \pi_2)).$$

Proof. Shared tests $T_1 \cap T_2$ execute once and contribute to the maintenance of both patterns. Only disjoint tests $(T_1 \cup T_2) \setminus (T_1 \cap T_2)$ incur additional cost. Budget accounting yields $h_{\text{total}} = h_{\text{shared}} \oplus h_{\text{disjoint}}$. This is strictly less than $2 \times h_{\text{shared}} \oplus h_{\text{disjoint}}$. □

8.3.1 Conflict Types and Cognitive Load

Conflict types arise from different sharing failures:

1. **Temporal:** patterns need same resource simultaneously (must serialize, paying time cost)
2. **Structural:** incompatible threshold requirements (cannot share tests, must duplicate)
3. **Phase:** mismatched refresh cadences (must maintain separately, no batching possible)

What this enables: **cognitive load emerges from heat accounting**. As patterns increase, conflicts become inevitable, forcing serialization (time cost) or duplication (space cost). The system doesn't "feel overloaded"—it runs out of budget to maintain concurrent distinctions, returning U for patterns it cannot afford to distinguish simultaneously.

Real-World Implication: Cramming vs. Understanding. Why is "rote memorization" exhausting? Maintaining 100 disconnected facts ("cramming") creates a massive maintenance obligation $\mathcal{O}_\rho$. Consolidation is the act of replacing those 100 facts with 1 underlying theory or rule that generates them. The rule costs more to learn initially (high rewrite cost h_{rewrite}), but is infinitely cheaper to maintain over time. VOID proves exactly *when* you should stop memorizing examples and start learning the rule.

Coq Verification. `void_interference_routing.v` implements interference dynamics through `InterferenceField` records with cached computation results.
`Void_pattern.v` provides `interfere_heat` for pattern interaction accounting.

8.4 Phase-Locking and Synchronization

Operational Perspective. Classical synchronization requires dynamic coordination protocols (mutexes, semaphores, consensus algorithms) with no closed-form optimal strategy. Our framework makes synchronization optimization **exact and automatic** through number theory: the greatest common divisor of decay periods directly calculates maximum savings—no runtime negotiation, no probabilistic convergence, just arithmetic on grace periods.

Synchronized refresh saves budget through test-sharing. When patterns share tests and their decay times have common factors, coordinated refresh executes shared tests once per synchronization period rather than independently. Natural rhythms emerge from greatest common divisor structure in grace periods—not from designed protocols but from arithmetic optimization.

Definition 8.11 (Natural Cadence)**.** For pattern π with decay δ,

$$\tau_\rho(\pi) \;:=\; \inf\{\, k \in \mathrm{Fin} \mid \delta^{(k)}(\Delta_S(\pi, \cdot)) \prec \theta(\rho) \,\}$$

is the *natural cadence* (ticks until threshold collapse, bounded by MAX).

Theorem 8.12 (Phase-Locking Advantage)**.** *For patterns π_1, π_2 with shared tests $T^* = T_1 \cap T_2$ and cadences τ_1, τ_2,*

$$h_{saved} \;=\; \#(T^*) \otimes \mathcal{M}_\rho(test) \otimes \gcd(\tau_1, \tau_2) \quad per\ \ \mathrm{lcm}(\tau_1, \tau_2)\ \ ticks,$$

where $\#(T^*)$ denotes the count of shared tests.

Proof. Shared tests refreshed at $\gcd(\tau_1, \tau_2)$ intervals satisfy both patterns until the next boundary. Independent schedules would refresh shared tests at average rates τ_1^{-1} and τ_2^{-1}; synchronization reduces executions to $\gcd(\tau_1, \tau_2)^{-1}$, saving the stated amount per lcm window. $\square$

What This Framework Enables? Automatic synchronization discovery. Patterns with $\tau_1 = 6$ and $\tau_2 = 9$ automatically lock at $\gcd(6, 9) = 3$-tick intervals—no scheduler needed. The system computes $\gcd(\tau_1, \tau_2)$ once, then synchronizes perfectly forever. Classical systems require runtime coordination; ours requires one arithmetic operation.

Forced coordination without control theory. When patterns share tests, thermodynamic pressure (minimizing total heat) forces them into resonance. This isn't designed—it's mathematically necessary. Patterns that could synchronize but don't waste budget, creating selection pressure toward phase-locked configurations. Synchronization emerges from arithmetic constraint, not from protocol design.

Hierarchical rhythm emergence. When refresh(Π_1) maintains $\Delta(\Pi_2) \succeq \theta(\rho)$ as side effect, Π_2 becomes parasitic on Π_1's maintenance (Section 8.5), paying zero refresh cost. No hierarchy needs to be specified—it's discovered through subset relations: if $T_2 \subseteq T_1$ and τ_2 divides τ_1, host-parasite structure emerges automatically. The system finds temporal hierarchies through heat minimization, not through designed abstraction layers.

Exact savings calculation. Classical amortized analysis gives $O(1)$ average cost with hidden constants. Here, savings are $\#(T^*) \otimes \mathcal{M}_\rho(\text{test}) \otimes \gcd(\tau_1, \tau_2)$ per $\text{lcm}(\tau_1, \tau_2)$ ticks—exact integers, verifiable by conservation. Given two patterns, compute their gcd, multiply by shared test count, done. No profiling needed.

Compositional optimization. With n patterns, pairwise gcd structure creates a synchronization lattice. The optimal global schedule can be computed statically from grace periods—it's the solution to a number-theoretic lattice problem, decidable in polynomial time. Classical schedulers use heuristics; we solve exactly.

Example 8.13 (Hierarchical Locking)**.** A pattern tracking "hour of day" has $\tau_1 = 3600$ ticks. Patterns for "morning/afternoon" ($\tau_2 = 43200$) and "business hours" ($\tau_3 = 28800$) satisfy $\gcd(3600, 43200) = 3600$ and $\gcd(3600, 28800) = 3600$. Both coarse patterns piggyback on hour-tracking for free—zero additional cost for shared tests. The system discovers this three-level temporal hierarchy purely from gcd arithmetic on natural cadences.

8.5 Parasitic Hierarchies: The Architecture of Abstraction

Not all patterns compete for resources; some survive by embedding themselves within the maintenance cycles of others. If maintaining a complex pattern π_1 necessarily executes all the distinctions required for a simpler pattern π_2, then π_2 persists at **zero marginal cost**. It becomes a metabolic free rider.

This occurs structurally when test sets are nested ($T_2 \subseteq T_1$). For example, an observer maintaining the pattern "12:05:30 PM" (high resolution) automatically maintains the pattern "Afternoon" (low resolution) without spending a single extra tick. The coarse pattern rides on the back of the fine distinction.

Definition 8.14 (Parasitic Maintenance)**.** Pattern π_2 is *parasitic* on π_1 if $T_2 \subseteq T_1$. In this state, the refresh operation for π_1 constitutes a valid refresh for π_2 as a side effect.

Theorem 8.15 (Zero Marginal Cost)**.** *If* $T_2 \subseteq T_1$*, then* $\mathcal{M}_\rho(\pi_1 \cup \pi_2) = \mathcal{M}_\rho(\pi_1)$.

Proof. Since $T_2 \subseteq T_1$, the union $T_1 \cup T_2 = T_1$. The system executes tests for π_1, and by definition, all tests for π_2 are completed. No additional budget is consumed. □ □

This asymmetry is fundamental. The relationship is strictly one-way: π_1 supports π_2, but π_2 contributes nothing to π_1. We adopt Michel Serres' definition of the parasite here not as a biological pest, but as a relational operator: a system where energy flows $A \to B$ without reciprocity [107]. In VOID, this thermodynamic parasitism is the physical origin of what classical logic calls "abstraction."

Hierarchical Dependencies. Fine-grained patterns (large test sets) automatically host coarse-grained patterns (small test sets that are subsets). In Example 8.13, the hour-tracking pattern maintains the morning/afternoon binary because hour distinctions include all tests needed for the coarser classification. The coarse pattern persists indefinitely, provided the fine pattern remains maintained.

These hierarchies emerge spontaneously from test set inclusion, not from design. Classical accounts treat abstraction as independent generalization built "above" concrete details. VOID reveals the opposite: coarse patterns exist *because* fine patterns perform their computational work. "Abstract" is not autonomous—it is parasitic side effect. Memory hierarchies are not information-theoretic pyramids but dependency structures where survival requires extracting distinctions someone else maintains.

8.6 Symmetry as Expensive Achievement

Operational Perspective. Classical frameworks treat symmetries as intrinsic properties: a system is rotationally invariant, translationally invariant, or time-reversal symmetric by construction. The invariance costs nothing to maintain—it simply is. This framework rejects that view.

Symmetry is not a free structural property but an actively maintained invariance. For pattern set Π to remain invariant under transformation $T : X \to X$, the system must continuously verify that $\Delta_S(T(x), T(y)) = \Delta_S(x, y)$ for all relevant pairs—a budget-consuming operation repeated every tick. Without this maintenance, patterns drift: small asymmetries accumulate through decay until the claimed invariance fails. Maintaining T-symmetry means paying to suppress this drift.

This inverts Noether's theorem.[14] Classical physics says symmetry implies conservation. Here, the direction reverses: conservation laws are accounting ledgers tracking where symmetry-maintenance budget went. Angular momentum isn't conserved because of SO(3) symmetry; maintaining SO(3) invariance costs budget, and angular momentum is the ledger of that cost. When budget exhausts, symmetry breaking is forced not from phase transitions or field fluctuations but from inability to afford continued invariance verification.

Definition 8.16 (Symmetry Preservation Cost)**.** For transformation $T : X \to X$ and pattern set Π, maintaining T-invariance over k ticks requires

$$h_{\mathrm{sym}}(T, \Pi, k) \;:=\; k \otimes \bigoplus_{\pi \in \Pi} \mathcal{C}_T(\pi),$$

where $\mathcal{C}_T(\pi)$ measures drift of π under no-maintenance evolution (the rate at which patterns want to break symmetry without active enforcement).

[14] In gauge theory, local symmetry requires gauge fields to maintain invariance. Here, maintaining symmetry requires continuous budget—the "gauge field" is dissipated heat. The drift_rate(T) is the connection form measuring how much symmetry "wants" to break.

Theorem 8.17 (Forced Symmetry Breaking). *If $h_{sym}(T, \Pi, k) \succ B$, the system cannot maintain T-invariance. Patterns must collapse to coarser T-equivalence classes or accept distinguishability loss.*

Proof. The hard budget constraint forbids spending beyond B; excess demand enforces either reduction of maintained distinctions or loss of invariance. □

Example (Rotational Invariance Cost). Maintaining SO(3) invariance for n spatial patterns costs proportional to $n \otimes d^2$ per tick, where d is dimension. High dimensions make symmetry prohibitively expensive, forcing symmetry reduction as dimension increases.

What this enables: **symmetry breaking as budget exhaustion**. When $h_{\mathrm{sym}}(T, \Pi, k) \to B$, the system cannot afford to maintain invariance. Symmetry breaks not from phase transitions or field fluctuations but from running out of resources to enforce it. Crystallization, magnetization, pattern formation—all reinterpretable as the system cheapening maintenance by abandoning expensive symmetries.

Coq Verification. `void_symmetry_movement.v` implements transformation preservation in `SymmetrySeeker` records with explicit per-tick costs ($\mathcal{C}_T$), drift rates, and exhaustion conditions triggering forced symmetry breaking.

8.7 Boundary Phenomena and Dynamical Regimes

Operational Perspective. When maintenance obligation $\mathcal{O}_\rho(\Pi)$ approaches budget limit B, discrete structural transitions become unavoidable. This is not analogous to physical phase transitions but literal computational necessity: patterns must be eliminated, merged, or allowed to collapse when the constraint $\mathcal{O}_\rho(\Pi) \otimes t \preceq B$ becomes unsatisfiable. Classical dynamical systems exhibit continuous state trajectories; VOID trajectories are punctuated by forced discontinuities when budget arithmetic forbids continuation of current configurations.

The relationship between total obligation $\mathcal{O}_\rho(\Pi)$ and available budget B determines qualitatively distinct dynamical regimes. We classify the complete spectrum from abundant resources to total exhaustion.

8.7.1 Regime I: Stable Maintenance

Condition: $\mathcal{O}_\rho(\Pi) \ll B$ with significant margin.

In this regime, all patterns remain above threshold $\theta(\rho)$ indefinitely. Every pattern $\pi \in \Pi$ receives timely refresh before its grace period expires, maintaining $\Delta_S(\pi, \cdot) \succeq \theta(\rho)$ without interruption.

Theorem 8.18 (Stable Equilibrium). *If $\mathcal{O}_\rho(\Pi) \otimes k \preceq B$ for time horizon $k \geq \max_{\pi \in \Pi} g_\rho(\pi)$, then all patterns in Π persist with $\Delta_S(\pi, \cdot) \succeq \theta(\rho)$ indefinitely.*

Proof. Let $\Pi = \{\pi_1, \ldots, \pi_n\}$ with maintenance costs $\mathcal{M}_\rho(\pi_i)$ and grace periods $g_\rho(\pi_i)$.

Step 1 (Refresh schedule exists): Since $\mathcal{O}_\rho(\Pi) = \bigoplus_{i=1}^n \frac{\mathcal{M}_\rho(\pi_i)}{g_\rho(\pi_i)}$, the average per-tick cost to maintain all patterns is $\mathcal{O}_\rho(\Pi)$. By assumption $\mathcal{O}_\rho(\Pi) \otimes k \preceq B$ for $k = \max_i g_\rho(\pi_i)$, we have sufficient budget to refresh every pattern within its grace period.

Construct refresh schedule: Let $t_i^{(0)} = 0$ be initial refresh time for π_i. Set next refresh at $t_i^{(1)} = t_i^{(0)} + g_\rho(\pi_i)$. By definition of grace period, π_i remains above threshold during $[t_i^{(0)}, t_i^{(1)}]$ even without refresh.

Step 2 (Budget sufficiency): Total cost over interval $[0, k]$ where $k = \mathrm{lcm}(g_\rho(\pi_1), \ldots, g_\rho(\pi_n))$ is:

$$\bigoplus_{i=1}^{n} \frac{k}{g_\rho(\pi_i)} \otimes \mathcal{M}_\rho(\pi_i) = k \otimes \mathcal{O}_\rho(\Pi) \preceq B$$

by assumption. Therefore refresh schedule is affordable. **Step 3 (Induction):** Assume all patterns satisfied $\Delta_S(\pi_i, \cdot) \succeq \theta(\rho)$ at tick t. By construction, next refresh of each π_i occurs within $g_\rho(\pi_i)$ ticks. By grace period definition, π_i remains above threshold until refresh. Therefore $\Delta_S(\pi_i, \cdot) \succeq \theta(\rho)$ at tick $t + 1$. By induction, all patterns persist indefinitely. □ □

This is the default operational regime—computation proceeds without crisis intervention.

8.7.2 Regime II: Graceful Coarsening

Condition: Budget decreases; $\mathcal{O}_\rho(\Pi) \to B$ from below.

As budget tightens, the system cannot maintain current resolution ρ indefinitely. Rather than catastrophic failure, patterns coarsen: resolution parameter ρ decreases, reducing per-pattern maintenance cost $\mathcal{M}_\rho(\pi)$ at the expense of distinguishability precision.

Definition 8.19 (Resolution Coarsening)**.** Pattern π *coarsens* from ρ to $\rho' \prec \rho$ if $\mathcal{M}_{\rho'}(\pi) \prec \mathcal{M}_\rho(\pi)$ and $\Delta_S(\pi, \cdot) \succeq \theta(\rho')$ at the lower threshold.

Theorem 8.20 (Graceful Degradation)**.** *Let budget decrease as B_t (non-increasing in t). If resolution adjusts to $\rho_t = \sup\{\rho : \mathcal{O}_\rho(\Pi) \preceq B_t\}$ at each tick, then no pattern collapses to U.*

Proof. **Claim:** For all $\rho' \preceq \rho$, if pattern π satisfies $\Delta_S(\pi, \cdot) \succeq \theta(\rho)$ at resolution ρ, then $\Delta_S(\pi, \cdot) \succeq \theta(\rho')$ at coarser resolution ρ'.

Justification: Lower resolution ρ' uses fewer tests, creating coarser equivalence classes. If π was distinguishable at fine resolution ρ (meaning tests at ρ separate it from others), then subset of those tests at ρ' still separates π from others at coarser level. Threshold $\theta(\rho')$ weakens proportionally, so $\Delta_S(\pi, \cdot) \succeq \theta(\rho')$ holds.

Main argument: At tick t, choose maximal ρ_t such that $\mathcal{O}_{\rho_t}(\Pi) \preceq B_t$. If $B_{t+1} \prec B_t$, then $\rho_{t+1} \preceq \rho_t$ (must coarsen to meet budget). By claim above, patterns maintained at ρ_t remain above $\theta(\rho_{t+1})$ at coarser resolution. No pattern returns U because distinguishability requirements weaken in proportion to budget reduction. □ □

Example: Spatial tracking degrades from meter-precision to ten-meter bins to hundred-meter regions as budget declines, maintaining "where am I?" functionality at reduced granularity. This regime exhibits *smooth adaptation*—no discrete failures, only gradual loss of precision.

8.7.3 Regime III: Metastable Persistence

Condition: $\mathcal{O}_\rho(\Pi) \succ B$ but grace periods unexpired.

Patterns survive temporarily despite insufficient long-term budget. Each pattern π tolerates $g_\rho(\pi)$ ticks without refresh before collapsing. During this grace window, patterns remain above threshold even though total obligation exceeds capacity.

Definition 8.21 (Metastable Configuration)**.** Π is *metastable* at budget B if $\mathcal{O}_\rho(\Pi) \succ B$ yet $\Delta_S(\pi, \cdot) \succeq \theta(\rho)$ for all $\pi \in \Pi$.

Theorem 8.22 (Delayed Collapse). *A metastable configuration persists for at most* $g_{\min} := \min_{\pi \in \Pi} g_\rho(\pi)$ *ticks before at least one pattern collapses.*

Proof. Let $\pi^* = \arg\min_{\pi \in \Pi} g_\rho(\pi)$ be pattern with shortest grace period. Since $\mathcal{O}_\rho(\Pi) \succ B$, not all patterns can be refreshed within their grace windows. Specifically:

$$\sum_{i=1}^{n} \frac{\mathcal{M}_\rho(\pi_i)}{g_\rho(\pi_i)} \succ \frac{B}{g_{\min}}$$

In any interval of length $g_{\min}$, total budget available is $B \preceq g_{\min} \otimes B/g_{\min} \prec g_{\min} \otimes \mathcal{O}_\rho(\Pi)$, which is insufficient to refresh all patterns. At least one pattern—specifically π^* whose grace period is $g_{\min}$—will not be refreshed within its grace window. After $g_{\min}$ ticks without refresh, π^* crosses threshold: $\Delta_S(\pi^*, \cdot) \prec \theta(\rho)$, returning U. Therefore metastability cannot persist beyond $g_{\min}$ ticks. □ □

Metastability is *transient buffer*—it delays but cannot prevent forced adaptation when obligation exceeds capacity.

8.7.4 Regime IV: Limit-Cycle Oscillations

Condition: $\mathcal{O}_\rho(\Pi_A) \prec B \prec \mathcal{O}_\rho(\Pi_A \cup \Pi_B)$ with $\Pi_A \cap \Pi_B = \emptyset$.

The system exhibits deterministic alternation between maintaining disjoint pattern sets. When Π_A is refreshed, Π_B degrades through its grace period until collapse forces attention switch.

Example 8.23 (Attentional Flicker). Budget $B = 10\mu(\rho)$. Observer tracks weather context (cost $8\mu(\rho)$) and social context (cost $7\mu(\rho)$). Total $15\mu(\rho) \succ B$.

Timeline:

- Tick 0: Refresh weather (8μ), social enters grace period
- Tick 1–3: Social grace period active, weather maintained
- Tick 4: Social grace expires $\rightarrow$ collapse to U
- Tick 5: Refresh social (7μ), weather enters grace period
- Tick 6–8: Weather grace period active, social maintained
- Tick 9: Weather grace expires $\rightarrow$ collapse to U
- Tick 10: Refresh weather, cycle repeats

Period: $\mathrm{lcm}(g_\rho(\mathrm{weather}), g_\rho(\mathrm{social}))$.

Theorem 8.24 (Attentional Oscillation). *If* $\mathcal{O}_\rho(\Pi_A) \prec B \prec \mathcal{O}_\rho(\Pi_A \cup \Pi_B)$ *and pattern sets are disjoint, the system exhibits stable limit cycle with period* $T = lcm(g_\rho(\Pi_A), g_\rho(\Pi_B))$.

Proof. **Step 1 (Simultaneous maintenance impossible):** By assumption, $\mathcal{O}_\rho(\Pi_A \cup \Pi_B) \succ B$. Therefore the system cannot maintain both Π_A and Π_B simultaneously by Theorem 8.22. **Step 2 (Individual maintenance possible):** By assumption, $\mathcal{O}_\rho(\Pi_A) \prec B$ and $\mathcal{O}_\rho(\Pi_B) \prec B$. Therefore either set can be maintained indefinitely in isolation by Theorem 8.18. **Step 3 (Forced alternation):** Without loss of generality, assume system refreshes Π_A at tick $t = 0$. Then:

- Π_B enters grace period (no refresh for Π_B).
- After $g_\rho(\Pi_B)$ ticks, Π_B collapses to U by grace period definition.
- System must refresh Π_B to restore it (cost $\mathcal{M}_\rho(\Pi_B)$).
- Refreshing Π_B consumes budget, preventing Π_A refresh.
- Π_A enters grace period, expires after $g_\rho(\Pi_A)$ ticks, forces Π_A refresh.

This alternation repeats indefinitely. **Step 4 (Period calculation):** The cycle repeats when both patterns have completed integer multiples of their grace periods:

$$k_A \cdot g_\rho(\Pi_A) = k_B \cdot g_\rho(\Pi_B) = T$$

The minimal such T is $\mathrm{lcm}(g_\rho(\Pi_A), g_\rho(\Pi_B))$.

Step 5 (Stability): Any deviation from cycle (attempt to maintain both or neither) violates budget constraint or leads to total collapse. The limit cycle is unique attractor. □ □

This regime creates observable *attentional flicker*—not random noise but deterministic oscillation with computable period.

8.7.5 Regime V: Catastrophic Collapse

Condition: $\mathcal{O}_\rho(\Pi) \gg B$ with grace periods expired. The system undergoes *discrete structural reduction*—patterns are forcibly merged into coarser equivalence classes or eliminated entirely.

Definition 8.25 (Forced Merging)**.** Patterns $\pi_1, \ldots, \pi_k$ undergo *forced merging* into π' if $\bigoplus_{i=1}^{k} \mathcal{M}_\rho(\pi_i) \succ B$ and merging achieves $\mathcal{M}_{\rho'}(\pi') \prec B$ at coarser $\rho' \prec \rho$.

Theorem 8.26 (Catastrophic Transition)**.** *If $\mathcal{O}_\rho(\Pi) \otimes t \succ B$ for remaining horizon t and all grace periods expired, then $|\Pi|$ must decrease or some patterns collapse to U.*

Proof. **Assume for contradiction:** All patterns in Π persist without merging or collapse. Then $\Delta_S(\pi, \cdot) \succeq \theta(\rho)$ for all $\pi \in \Pi$ at each tick. This requires refreshing each π within its grace period $g_\rho(\pi)$. Total cost over horizon t is:

$$\bigoplus_{\pi \in \Pi} \frac{t}{g_\rho(\pi)} \otimes \mathcal{M}_\rho(\pi) = t \otimes \mathcal{O}_\rho(\Pi)$$

By assumption, $t \otimes \mathcal{O}_\rho(\Pi) \succ B$. But conservation law $B = B' \oplus h$ requires $h \preceq B$ (cannot spend more than available). Therefore $t \otimes \mathcal{O}_\rho(\Pi) \preceq B$, contradiction. Hence at least one pattern must either merge with others (reducing $|\Pi|$) or collapse to U (ceasing to require maintenance). This forces discrete structural transition. □ □

Example: System maintains 100 fine-grained spatial patterns (cost 500μ). Budget drops to $B = 150\mu$. Grace periods expire. Forced merge: 100 patterns collapse into 30 coarse regions (cost 120μ). One tick: $|S| = 100$ states. Next tick: $|S| = 30$ states.

8.7.6 Regime VI: Minimal-Maintenance Convergence

Condition: Extreme scarcity; B barely sufficient for any patterns. When budget approaches zero, only the cheapest possible patterns survive. The system enters *rigid minimal-maintenance state.*

Definition 8.27 (Rigid Configuration)**.** Π is *rigid* if no $\Pi' \subseteq \Pi$ exists with $\mathcal{O}_\rho(\Pi') \prec \mathcal{O}_\rho(\Pi)$ and $|\Pi'| = |\Pi|$.

Theorem 8.28 (Convergence to Minimal Maintenance)**.** *As B decreases and approaches $\mathcal{O}_\rho(\Pi)$, the system converges to a rigid configuration.*

Proof. Define cost-efficiency ratio: $r(\pi) := \mathcal{M}_\rho(\pi)/g_\rho(\pi)$ (maintenance cost per tick).

Step 1 (Elimination ordering): When $\mathcal{O}_\rho(\Pi) \succ B$, identify $\pi^* = \arg\max_{\pi\in\Pi} r(\pi)$ (least efficient pattern).

Step 2 (Elimination condition): If $\mathcal{O}_\rho(\Pi\backslash\{\pi^*\}) \prec B$, removing π^* satisfies budget constraint. Eliminating π^* reduces total obligation by $r(\pi^*)$ while losing one pattern.

Step 3 (Iteration): Repeat elimination until $\mathcal{O}_\rho(\Pi) \preceq B$ or no further pattern can be removed without violating distinguishability requirements.

Step 4 (Convergence to rigidity): Process terminates when every remaining pattern satisfies: removing it would reduce distinguishability more than budget savings justify. Formally, for all $\pi \in \Pi$:

$$\frac{\mathcal{M}_\rho(\pi)}{g_\rho(\pi)} \preceq \frac{B - \mathcal{O}_\rho(\Pi \setminus \{\pi\})}{|\Pi|}$$

This is rigidity: no pattern can be removed without losing more value than gained in budget. The configuration is locally optimal. □ □

Interpretation: Extreme exhaustion creates *stereotyped behavior*. When budget vanishes, system reduces to minimal pattern set—automated responses, habitual actions, rigid routines. Habit formation, skill consolidation, automatization become inevitable outcomes of resource limitation.

Complete Regime Classification. The six regimes form continuous spectrum determined by ratio $\mathcal{O}_\rho(\Pi)/B$:

Regime	**Condition**
I. Stable Maintenance	$\mathcal{O}_\rho(\Pi) \ll B$
II. Graceful Coarsening	$\mathcal{O}_\rho(\Pi) \to B$ (smooth)
III. Metastable Persistence	$\mathcal{O}_\rho(\Pi) \succ B$, grace active
IV. Limit-Cycle Oscillations	$\mathcal{O}(\Pi_A) \prec B \prec \mathcal{O}(\Pi_A \cup \Pi_B)$
V. Catastrophic Collapse	$\mathcal{O}_\rho(\Pi) \gg B$, grace expired
VI. Minimal Convergence	$B \to 0$

These are mathematically distinct dynamical behaviors—each follows necessarily from conservation constraint $B = B' \oplus h$ under different resource availability.

Coq Verification. `void_crisis_relocation.v` implements forced adaptation with strategies (Hibernate, Scatter, Cluster, Merge) selected by budget availability and regime detection. `void_thermal_convection.v` implements oscillatory dynamics with `ThermalPattern` records tracking grace periods and threshold crossings. `void_pattern_thermo.v` verifies convergence to minimal-maintenance configurations under budget pressure.

9 Conclusion—Mathematics After the Infinite

Chapter Overview. This conclusion traces the historical arc from Euclidean constructive geometry through modern set-theoretic foundations built on infinity, identifying the core problem: classical space as $\infty \times 0$ (infinite collections of zero-dimensional points). We examine Pascal's introduction of infinity into probability as methodological tool mistaken for foundation, survey the resulting foundational crises (Russell's paradox, Gabriel's Horn, Riemann rearrangement), and position VOID within the finitist tradition (Hilbert, Bishop, Esenin-Volpin, Zeilberger). The technical core demonstrates how VOID's resource-bounded framework resolves classical paradoxes through three-valued logic and explicit budget accounting, followed by enumeration of VOID's mathematical contributions (finite types, primitive rationals, conservation laws, uniform costs). We conclude with potential applications (AI hallucination prevention, quantum mechanics reinterpretation, computing infrastructure), philosophical stance on VOID as alternative rather than replacement, and reflection on mathematics after infinity. Readers interested primarily in technical content may focus on paradox resolutions and mathematical contributions; those seeking philosophical context should attend to historical and foundational sections.

9.1 The Missing Foundation

Classical mathematics inherits its spatial intuition from Euclidean geometry—a system built on compass-and-straightedge constructions, finite figures, and tangible operations. Yet modern mathematics has drifted far from this constructive origin. While Euclid worked with drawable triangles and measurable lengths, contemporary mathematics defines space through $\mathbb{R}^n$: the set of all n-tuples of real numbers. Each real number, in turn, is constructed through Dedekind cuts, Cauchy sequences, or infinite decimal expansions—formal procedures that fundamentally require completed infinity as an axiom [26, 21].

This reliance on infinity generates a foundational anomaly that mathematics has largely chosen to ignore: individual points in $\mathbb{R}^n$ have no dimension—they are zero-dimensional objects. Space emerges, supposedly, as their "collection." But infinity is not a number; multiplying it with zero is undefined. Classical space is literally $\infty \times 0$: an infinite collection of dimensionless points. In our view, this should alarm the mathematical community as potentially dangerous nonsense that creates fertile ground for paradoxes and non-constructive reasoning, yet we have built centuries of mathematics upon it.

In physics, these limitations manifest with particular force. Classical models rely on infinite divisibility, but all actual measurement, computation, and physical processes operate within finite precision—as quantum mechanics and computational complexity theory repeatedly demonstrate [102, 91]. The discrete and resource-limited character of physical reality routinely clashes with the idealized continuum suggested by $\mathbb{R}^n$. The reliance on infinity as a foundational element, though historically productive, now introduces technical and philosophical difficulties that motivate investigation into alternative, finitary mathematical frameworks.

This critique does not diminish the genuine achievements of classical mathematics. Rather, it identifies a persistent ambiguity: the gap between formal constructions built upon infinity and the tangible, constructible outcomes available to finite observers and computing systems. Recognizing and addressing this gap is the principal motivation behind resource-bounded, constructive, and observer-centered approaches to mathematics [124, 35].

9.2 The Hydra's Many Heads

This foundational tension has not escaped discerning mathematicians and physicists. Both fields have become so unwieldy that they resemble a hydra eating its own heads while spawning new ones—obscuring any clear view of reality, until only what lies within the thickness of the hydra's forest of necks can be observed.

Norman Wildberger has mounted a sustained public critique of standard foundations, particularly infinite sets and real numbers, arguing that such abstractions produce unnecessary conceptual confusion and computational complexity [124]. Edward Nelson questioned the consistency of Peano arithmetic, challenging the very foundation upon which standard mathematics rests. We also find Peano's successor function unconvincing and unsubstantiated. In physics, the renormalization procedures of quantum field theory, the landscape problem in string theory, and the proliferation of parameters in particle physics all suggest a system growing increasingly baroque. What unites many critics—across mathematics and physics—is recognition that accepting infinity as foundational doesn't simplify theories but, on the contrary, generates endless complications which stack up as the complexity of the matter at hand grows. Each "solution" requires new auxiliary hypotheses; each unification spawns new parameters. The hydra grows new heads faster than we can sever them.

These foundational ambiguities carry consequences that extend beyond technical debate into the realm of intellectual honesty itself. Consider the case of Alexander Esenin-Volpin, the pioneering ultrafinitist whose work questioned the legitimacy of infinity within mathematics. His radical critique led to his confinement in Soviet psychiatric institutions, where he was officially diagnosed with "pathological honesty"—a phrase recorded by Vladimir Bukovsky in his memoirs [19]. While this diagnosis took ominously absurd form under extreme political conditions, the absurdity is symptomatic of something deeper.

The phrase "pathological honesty" inadvertently identifies a profound mathematical question: Is commitment to finiteness an act of radical honesty? Does insisting on constructible, verifiable, finite operations represent a kind of intellectual integrity that exposes infinity's role in mathematics as somewhat dubious—perhaps even as a convenient fiction we've collectively agreed not to examine too closely? Read non-politically, Esenin-Volpin's case suggests that foundational questions about infinity touch something more fundamental than mere technical preference. They concern what it means to reason honestly about mathematical objects: whether we can legitimately invoke entities and processes that transcend all possible verification, or whether mathematical integrity demands we restrict ourselves to what can actually be constructed and checked [35].

9.3 Alternative Foundations—The Finitist Tradition

VOID Theory emerges within a landscape shaped by multiple critical alternatives to classical, infinity-based mathematics. David Hilbert's formalist program attempted to secure mathematics on a finite, rule-based foundation, seeking to prove the consistency of mathematics using only finitary methods. Though his vision encountered insurmountable difficulties after Gödel's incompleteness results, Hilbert's program established the legitimacy of questioning whether infinity is necessary for mathematical rigor.

Errett Bishop's constructive analysis reimagined calculus and real numbers using only explicit constructions and computable functions, challenging the dominance of non-constructive existence proofs within analysis [10]. Bishop demonstrated that substantial portions of analysis could be re-built constructively, though his framework still permitted potential infinity in the form of unbounded iteration.

Esenin-Volpin radicalized finitism by attacking the logical validity of mathematical induction itself in the physical world. He argued that the set of natural numbers is not unique but observer-dependent, introducing the concept of **'feasible numbers'** —integers concretely accessible within a finite universe. For him, the standard assumption that the successor operation ($n \to n+1$) is always unconditionally valid constitutes a "logical hallucination": extending induction beyond feasible bounds creates purely fictitious entities that preserve syntactic consistency only by severing all ties with computational reality [35].

These frameworks, while each addressing foundational limitations of classical mathematics, often retained core elements from mainstream approaches or focused primarily on philosophical coherence rather than operational implementation. They identified problems but struggled to produce working mathematical systems that could compete with classical methods in practical calculation and proof.

9.4 Computational Frameworks and Finite Mathematics

Several concrete mathematical frameworks and computational paradigms embody finitary principles closely related to VOID Theory, serving as practical alternatives to classical approaches. Computable analysis reformulates real analysis using only algorithmically computable functions and sequences, ensuring that all mathematical entities and operations are constructible within finite resources [123]. This approach recognizes that only computable real numbers—those whose digits can be generated by a finite algorithm—have operational meaning.

Algorithmic information theory, as developed by Kolmogorov and Chaitin, quantifies the complexity of objects by the minimal computational resources needed for their description, directly engaging with the limits of compression and effective computation [24]. This framework makes explicit the connection between mathematical objects and the computational resources required to specify them.

In applied fields, computational geometry in computer graphics, CAD, and robotics relies on discrete data structures and explicit algorithms, bypassing the need for infinite precision and continuous models in favor of practical, finite representations. These disciplines have successfully built sophisticated mathematical machinery without invoking the completed infinite, demonstrating that finitary approaches can be both rigorous and practically powerful. Yet, where these frameworks accept finiteness as a pragmatic approximation of an ideal continuum, VOID elevates resource constraints from implementation details to constitutive axioms, making the thermodynamic cost of distinction intrinsic to the mathematical structure itself.

9.5 VOID Theory's Operational Turn

VOID Theory's singularity lies in its operational redefinition: space, number, and measurement are modeled entirely as finite, probabilistically structured acts performed by resource-bounded agents. Every mathematical distinction in VOID is paid for and tracked; irreversibility, memory decay, and computational limits are not just philosophical motifs, but core axioms verified through machine-checked proofs in Coq.

Where earlier finitary approaches often focused on philosophical critique or restrictions of classical methods, VOID takes a different path: positive construction of a complete mathematical system from first principles. By explicitly tying mathematical reasoning to budget consumption, heat generation, and observer limitations, VOID advances the finitist tradition into a new practical domain, delivering machine-verified code and exact operational results rather than mere philosophical objection.

In doing so, VOID both draws upon and fundamentally transforms the legacy of Hilbert's formalism, Bishop's constructivism, and Esenin-Volpin's ultrafinitism. The key shift is from denial of infinity to positive, self-consistent construction—what we might call "removing infinity from the equations" entirely, replacing it with explicit resource accounting. This operational transparency means that every VOID theorem corresponds to a verified computational procedure. There are no appeals to limit processes, no invocations of completed infinities, no existence proofs divorced from construction.

We acknowledge that VOID's methods may not initially demonstrate computational superiority over classical techniques optimized through centuries of refinement. Yet this misses the deeper significance: VOID opens an entirely new space for mathematical experimentation and systematic improvement, one where every operation has explicit resource costs and verifiable termination. More fundamentally, and we consider this no small achievement, **VOID demonstrates that infinity is a *choice*, not a necessity.** Its successful construction of rigorous mathematics without any appeal to the infinite forces a profound question: What is infinity's actual role in mathematics? Is it an essential feature of mathematical reality, or merely a practical convenience—a grand approximation mechanism that, while useful, should not be confused with fundamental mathematical objects? The existence of VOID as a complete, verified alternative suggests that infinity may be more akin to a powerful computational fiction than to an indispensable foundation, demanding that we reconsider which aspects of classical mathematics reflect genuine structure and which reflect merely our historically contingent methods of representation.

Here we leave behind both metaphor and polemic. The following sections detail how VOID's core principles—budget, heat, and resource-bounded distinguishability—yield not just an alternative philosophy, but a functioning mathematical system that allows for reformulating classical paradoxes within this resource-conscious framework, neutralizing its dangers by operational transparency and thermodynamic accounting that makes every mathematical claim subject to finite verification.

Definition 9.1 (Classical Space—Revealed)**.** Euclidean geometry operated with an intuitive notion of space as the arena for geometric figures, but never provided a formal definition. Modern mathematics fills this gap with the following construction:

$$\text{Space} := \mathbb{R}^n = \underbrace{(\mathbb{R} \times \mathbb{R} \times \cdots \times \mathbb{R})}_{n \text{ times}}$$

where each $\mathbb{R}$ is constructed via:

- **Dedekind**: $\mathbb{R} = \{(A, B) \mid A \cup B = \mathbb{Q},\ A < B,\ A \text{ and } B \text{ both infinite sets}\}$ (infinite cuts)
- **Cauchy**: $\mathbb{R} = \{(a_n)_{n\in\mathbb{N}} \mid \lim_{n\to\infty} a_n \text{ exists}\}/\!\sim$ (infinite sequences)
- **Decimal**: $\mathbb{R} = \{0.d_1d_2d_3\ldots \mid d_i \in \{0, \ldots, 9\},\ i \in \mathbb{N}\}$ (infinite strings)

Each point in $\mathbb{R}^n$ is zero-dimensional by construction. Dimensionality purportedly "emerges" from the n-fold Cartesian product. But this is $n \times \infty \times 0$—undefined algebra masquerading as foundation. The multiplication of dimensions by infinite precision by zero-dimensional points produces no coherent geometric object, only notational sleight-of-hand.

The deep irony: geometry, humanity's most ancient and intuitively accessible branch of mathematics, rests on a construct (continuous space) that cannot be rigorously defined without invoking infinite, non-constructive processes. The very field that began with compass, straightedge, and drawn figures now requires uncountable sets and limiting operations as its purported foundation.

9.6 Pascal's Original Sin

This transformation was not merely technical but epistemological, marking the specific moment where mathematics embraced the infinite to bypass the complexity of the finite. The cultural impact of this shift is epitomized by *Pascal's Wager*—a logically valid argument within his new framework, yet one that rests on a thermodynamically fraudulent premise.

By introducing an infinite payoff (∞), Pascal rendered all finite costs irrelevant. The Wager treats belief not as a structural transformation of the subject—requiring continuous metabolic investment, behavioral repression, and cognitive maintenance ($h_{faith} > 0$ per tick)—but as a cost-free transaction with infinite leverage. This marks the paradox of modernity: the Enlightenment's rationalist edifice was erected on a foundation that explicitly legitimized the irrational (the actual infinite) to solve problems of uncertainty. Pascal did not make the relationship with the divine rational; he made it mercantile-mystical, trading finite existence for an unconstructable infinite future. VOID rejects this bargain: since infinite payoffs cannot be constructed, the finite costs of maintenance cannot be ignored.

Before Pascal, probability was operational and exact:

- **Cardano** (1545): Cardano grounded uncertainty in the *structure of the object*—a closed, verifiable system of finite configurations. Pascal's legacy, conversely, was to relocate truth from the finite present into an unachievable infinite future—replacing the exact arithmetic of possibilities with the asymptotic fiction of the limit, thereby severing probability from material reality. For Cardano, probability was ratio of favorable to total outcomes in dice throws [22]—finite sample spaces, exact computation, no limiting behavior
- **Fermat & Pascal** correspondence (1654): Division of stakes problem—finite games, finite players, exact expected values computed from enumerable outcomes [38]

After Pascal's paradigm shift:

- Probability defined as $\lim_{n\to\infty} \frac{\text{favorable}}{n}$—infinite trials assumption baked into foundation
- Expectation defined via infinite series: $\mathbb{E}[X] = \sum_{i=1}^{\infty} x_i p_i$—convergence required but not guaranteed
- Central Limit Theorem demands $n \to \infty$ [80]—asymptotic regime treated as if it were reality
- Law of Large Numbers assumes infinite repetition—what happens in practice remains approximation

Pascal's move was mathematically powerful but created a subtle methodological shift: treating infinity not as idealization or convenient shorthand but as *foundation*. Here, calculating uncertainty begins with the adoption of infinity as an operational tool—establishing what Bruno Latour might recognize as a "mathematical laboratory." Like experimental science's Baconian tradition of creating controlled conditions to isolate phenomena, Pascal's method constructs perfect (but artificial) conditions for studying probability: the idealized limit of infinitely many trials.

The transition from "very large n" to "$n = \infty$" appears innocuous in notation—it is profound in meaning. Finite systems have exact, computable answers. Infinite systems have limits that may not exist, may not converge, or may converge differently depending on the order of operations [104]. Classical analysis papers over these distinctions with epsilon-delta arguments that themselves invoke infinite processes. The laboratory has become mistaken for reality; the methodological tool has been elevated to foundational principle.

The consequences of treating infinity as foundation rather than laboratory tool become visible in concrete mathematical phenomena. Infinite processes break basic arithmetic properties that hold universally in finite systems. For any finite sum, rearranging terms leaves the total unchanged—commutativity is fundamental. But when we extend to infinity, this elementary property collapses:

Example 9.2 (Riemann Rearrangement [100])**.** Our first example:

Consider the conditionally convergent series $\sum_{n=1}^{\infty} \frac{(-1)^{n+1}}{n} = \ln 2$. Rearranging terms produces different limits:

$$\frac{1}{1} - \frac{1}{2} - \frac{1}{4} + \frac{1}{3} - \frac{1}{6} - \frac{1}{8} + \frac{1}{5} - \cdots = \frac{\ln 2}{2}$$

Same terms, different sum—yet classical mathematics calls both "the series." More dramatically: for *any* target $L \in \mathbb{R} \cup \{\pm\infty\}$, there exists a rearrangement of the series converging to L. Infinite sums are fundamentally ill-defined objects without imposing additional structure (absolute convergence). The problem isn't fringe examples—the problem is that infinity breaks arithmetic.

This is not a curiosity or edge case. It reveals that when we work with infinity as foundation, we lose basic algebraic properties. The laboratory conditions are showing their artificiality: operations that work perfectly in finite mathematics produce contradictions when extended to the infinite case. Classical analysis "solves" this by imposing additional conditions (absolute convergence), but this is precisely admission that infinity requires special handling—that it's not as foundational as finite arithmetic.

These arithmetic failures have geometric counterparts. When we treat the infinite limit as an actual object rather than a laboratory idealization, we generate geometric "paradoxes" that reveal the category mistake:

Example 9.3 (Gabriel's Horn [116])**.** The surface $y = 1/x$ rotated around the x-axis for $x \in [1, \infty)$ produces an object with:

$$V = \pi \int_1^{\infty} \frac{1}{x^2}\, dx = \pi \quad \text{(finite volume)}, \qquad A = 2\pi \int_1^{\infty} \frac{1}{x}\sqrt{1 + \frac{1}{x^4}}\, dx = \infty \quad \text{(infinite area)}$$

This generates the famous **"Painter's Paradox"**: one could ostensibly fill the horn with a finite bucket of paint (since V is finite), thereby wetting the entire inner surface. Yet, mathematically, coating that same surface with any paint of uniform thickness would require an infinite amount of liquid (since A is infinite). This implies the absurdity that a finite volume of liquid can be contained within a boundary it cannot cover.

The standard interpretation suggests one can "paint the inside but not the outside." In VOID, this is geometric nonsense masquerading as paradox. **The object doesn't exist—its construction requires completed infinity**. A finite approximation (truncating at some $x = L$) has both finite volume and finite area. The "paradox" emerges only when we pretend the limiting process produces an actual geometric object rather than a notational convenience. While the limiting process suggests a bizarre result—finite volume but infinite surface area—**the "paradox" vanishes once we acknowledge that such an object is merely an idealization, not a physically realizable geometric form**, and its properties only apply within the abstract infinity of calculus, not empirical reality.

It was Pascal who opened the door and let the unpredictable force of infinity into the mathematical palace; by the nineteenth century, infinity behaved as if it had been there, at the foundation of mathematics, since the dawn of the discipline. Maxwell's continuous distribution functions [89] treated particle velocities as if drawn from uncountable continuum. Boltzmann's entropy $S = k \log W$ invokes counting microstates where W can be infinite [14]. Gibbs' ensemble theory [50] averages over infinite collections of hypothetical systems. Each treated infinity as calculational convenience that hardened into structural assumption. Statistical mechanics, for all its empirical success, rests on treating finite systems as if they were infinite, then recovering finite predictions through asymptotic arguments. Moreover, this reliance on infinite constructs reflected an enduring atmosphere of anti-atomism—rooted in post-seventeenth-century philosophy—where continuity and the rejection of fundamental discreteness became the implicit foundation for statistical mechanics and its portrayal of physical reality.

9.7 Geometry Without Foundations

For more than two millennia, mathematicians solved sophisticated geometric problems using spatial intuition without possessing rigorous foundations for the concept of "space" itself:

- **Euclid** (300 BCE): Axioms assumed points, lines, planes exist as primitive objects but never constructed them [34]. "A point is that which has no part" (Book I, Definition 1) offers description, not construction. What *is* a point? Euclid doesn't say—and neither do the thousands of theorems proven from his axioms.

- **Descartes** (1637): Introduced coordinates (x, y) without specifying what numbers x and y fundamentally *are* [29]. Assumed a continuous number line but offered no construction of continuity. Cartesian geometry succeeded operationally while leaving its ontological commitments undefined.

- **Leibniz** (1670s): An outlier. In contrast to Newton, he developed calculus not merely as a calculation tool, but as a component of a grand ontological system grounded in the *Law of Continuity*. He regarded infinitesimals as "well-founded fictions" (*fictiones bene fundatae*)—conceptually coherent within his metaphysics but lacking the formal syntactic definition demanded by later critics. While Berkeley famously dismissed them as "ghosts of departed quantities" [9], the incoherence lay not in Leibniz's thought, but in the gap between his ontological vision and the available formal language.

- **Laplace** (1814): The Great Illusionist. Often celebrated for his reply to Napoleon regarding the Creator—*"Sire, I had no need of that hypothesis"*—Laplace did not banish the absolute; he merely rebranded it. He replaced the theological hypothesis (God) with an ontological one (Infinite Precision). His famous "Demon"—an intellect capable of computing the entire future from a single instant—requires the state of the universe to be defined by real numbers with infinite decimal expansion. **To this day, this phantom intellect haunts the sciences, sustaining the illusion of a deterministic universe that exists without cost.**

  In doing so, Laplace smuggled the attributes of the monotheistic deity—omniscience, timelessness, and total determinism—into the foundations of physics. This maneuver allowed modernity to resolve its cognitive dissonance: science could reject the liturgy of religion while retaining its comforting architecture of absolute order. In VOID, Laplace's Demon is exposed as a thermodynamic fraud: no finite observer can afford the infinite budget required to measure, let alone store, the initial conditions (t_0) with the precision necessary for the Demon to function. The deterministic universe is not a physical fact but a budget violation.

- **Gauss** (1809, 1827): The Architect of Normalcy. While his differential geometry liberated space from Euclidean rigidity, his development of the *Normal Distribution* (the Bell Curve) imposed a far more insidious constraint on reality. By defining deviation as "error" relative to a mathematical mean, Gauss provided the algorithmic justification for the modern concept of "normality" - an idea that did not exist and would not make any sense only a century before. In the Gaussian world, the unique, finite individual is dissolved into a statistical aberration from the great Mean which never absolved anyone for quantitative deviations.

- **Riemann** (1854): Proposed general manifolds and intrinsic geometry, but still assumed an underlying $\mathbb{R}^n$ coordinate space without justifying what $\mathbb{R}$ *is* [100]. Pushed the foundational problem one level deeper without solving it. He defined how to measure distance on curved surfaces but uncritically assumed that the surface itself is made of an infinitely divisible, continuous fabric rather than discrete elements.

- **Hilbert** (1899): Provided the first rigorous axiomatization of Euclidean geometry—a full 2,200 years after Euclid [62]. Yet, Hilbert explicitly treated points, lines, and planes as undefined primitives defined solely by their relations, famously noting they could be replaced by "tables, chairs, and beer mugs" as long as the logic held. By prioritizing structural consistency over ontological construction, he left the actual nature of space empty, allowing the unconstructable continuum of real numbers to fill the void by default.

The historical pattern is stark: Geometry *worked*—produced correct predictions, guided physical intuition, solved practical problems—because humans possess *operational* intuition about spatial relationships. Its power came from being directly testable and useful, not from abstract theories. We can draw lines, measure angles, compare areas, and verify congruence through physical manipulation. Only **in the late nineteenth century** did mathematicians begin to overhaul these practical foundations, **imposing infinite constructs like real numbers and set-theoretic formalisms**, thereby smuggling infinity into the deepest layers of the discipline.

Klein's Erlangen Program (1872) classified geometries by their transformation groups—Euclidean geometry preserves distances, projective geometry preserves collinearity, topological geometry preserves neighborhoods. [73] Elegant and powerful. But Klein never asked the foundational question: transformation groups act on *something*. What is that something? The implicit answer: $\mathbb{R}^n$, the Cartesian product of infinitely-specified real numbers. The classification succeeds operationally while smuggling in completed infinity as its substrate.

As one can see, Pascal initiated an avalanche effect: a broader transformation that unfolded over the next two centuries. Infinity spread from probability theory through calculus and analysis, gradually becoming normalized as foundational rather than methodological. Yet remarkably, throughout this period, we now call modernity, geometry—humanity's oldest and most intuitive mathematical discipline—continued producing valid, verified results without any rigorous definition of space. From Euclid through Descartes to the geometers of the eighteenth century, proofs relied on diagrams showing operational constructions, physical implementations demonstrating spatial relationships, similarity arguments comparing ratios of finite quantities, and informal continuity appeals. None required completed infinity. None invoked uncountable sets. None depended on Dedekind cuts or Cauchy sequences.

When nineteenth-century mathematicians finally undertook the project of providing rigorous foundations for geometry, they faced a choice: formalize the operational, constructive practice that had worked for millennia, or subordinate geometry to the emerging set-theoretic framework. They chose the latter. Hilbert's axiomatization, while brilliant, embedded geometry within a structure demanding the real numbers—and thus demanding Dedekind cuts, Cauchy sequences, and completed infinities. **The foundations were imposed not because geometric *practice* required them, but because set-theoretic *foundations* required them.**

This mismatch between centuries of successful geometric practice and newly imposed infinite foundations produced a menagerie of apparent paradoxes—artifacts not of geometry itself, but of the laboratory conditions under which twentieth-century mathematics chose to study it.

9.8 Paradoxes as Symptoms

The foundational crises of the early twentieth century—Russell's paradox [105], Gödel's incompleteness [54], Turing's undecidability [117], Tarski's indefinability [113]—are not independent pathologies requiring separate patches. They are *symptoms* of infinity's presence as foundational principle rather than methodological tool. Each reveals what happens when infinite constructions, treated as mathematical reality rather than laboratory idealizations, meet operational constraints: contradiction, incompleteness, undecidability, indefinability. **Remove infinity; the symptoms disappear.**

VOID reformulates each paradox by imposing explicit resource bounds, converting infinite regress into finite exhaustion. Where classical mathematics spirals into logical contradiction or undecidability, VOID returns a third truth value: U ("unknown due to budget exhaustion"). This is not ignorance—it's operational honesty about finite systems.

To demonstrate this concretely, we examine how VOID's resource-bounded framework handles the classical foundational crises. In each case, the paradox dissolves not through clever logical maneuvering, but through acknowledgment of computational reality: observers with finite budgets cannot complete infinite processes. **What appears as logical paradox in the classical framework becomes, in VOID, simply budget exhaustion—a finite, verifiable, and non-contradictory state**.

Russell's Paradox (1903). Consider the canonical crisis of set-theoretic foundations. Define the set $R = \{x \mid x \notin x\}$—the set of all sets that are not members of themselves. Query: is $R \in R$? Both answers lead to contradiction:

$$\begin{aligned} R \in R &\implies R \notin R \quad \text{(by definition of } R) \\ R \notin R &\implies R \in R \quad \text{(by definition of } R) \end{aligned}$$

Classical set theory responded by restricting set formation through Zermelo-Fraenkel axioms, but the paradox reveals deeper trouble: self-reference combined with the assumption of unlimited verification resources creates logical explosion.

VOID's Resolution Through Resource Bounds. In VOID, checking membership $x \in R$ requires verifying $x \notin x$, which requires checking $x \in x$, which requires verifying $x \notin x$—infinite regress. But each verification step consumes budget. Self-reference doesn't create logical contradiction; it creates resource exhaustion.

VOID implements set membership as budget-aware equality testing. When budget depletes before verification completes, the system cannot return either true or false—it needs a third possibility. This necessity drives VOID's three-valued logic:

```
Inductive Bool3 : Type :=
  | BTrue : Bool3      (* definitively true within allocated budget *)
  | BFalse : Bool3     (* definitively false within allocated budget *)
  | BUnknown : Bool3.  (* budget exhausted before determination possible *)
```

The equality testing function makes this explicit:

```
Fixpoint fin_eq_b3 (n m : Fin) (b : Budget) : (Bool3 * Budget * Heat) :=
  match b with
  | fz => (BUnknown, fz, fz)  (* Budget exhausted - cannot determine *)
  | fs b' =>
      match n, m with
      | fz, fz => (BTrue, b', fs fz)        (* Both zero: equal *)
      | fs n', fs m' => fin_eq_b3 n' m' b'  (* Recurse with cost *)
      | _, _ => (BFalse, b', fs fz)         (* Different structure *)
      end
  end.
```

(`void_finite_minimal.v`, lines 147–159)

For the self-referential query $R \in R$, recursion never bottoms out. Budget depletes. The system returns `BUnknown`, which we denote mathematically as U:

$$\texttt{member}(R, R, B) = \begin{cases} \text{computing} & \text{if } B > 0 \text{ (recursion continues)}, \\ U & \text{if } B = 0 \text{ (budget exhausted)}. \end{cases}$$

This third truth value is not ignorance or fundamental uncertainty—it is operational honesty about finite systems. Classical two-valued logic assumes unlimited resources for verification. VOID's three-valued logic makes resource bounds explicit. The uncertainty emerges from computational constraints, not from fuzzy logic or epistemic limits.

Key Insight. Russell's paradox is not a bug in logic—it is *proof* that set membership with self-reference requires infinite computational resources. VOID makes this explicit and operational. Classical set theory's "solution" through axiom schemes that prohibit such constructions is tacit admission that the verification demands unbounded resources. VOID's U return is honest acknowledgment: the computation exhausted its budget before reaching a definitive answer. **The paradox dissolves because there was no paradox to begin with—only an unbounded computation we lack resources to complete.**

This structural failure in naive set theory is merely a prelude to the broader crisis in formal logic. While Russell demonstrated that logical circularity exhausts resources via infinite recursion depth, Gödel showed that verification over infinite domains exhausts them via infinite search width.

Gödel's Incompleteness (1931). *Classical statement*: Any consistent formal system F containing arithmetic admits undecidable statements G where neither $F \vdash G$ nor $F \vdash \neg G$ can be proven within F. The statement G effectively asserts: "There is no number x that encodes a proof of this statement."

VOID reformulation: Classical logic treats the non-existence of a proof as a static, semantic fact. VOID treats it as an algorithmic search problem. To determine the truth value of G, the observer must scan the domain of possible proofs (numbers).

- If a proof exists, it is a finite object found at some cost c. If $c \leq B$, the system returns `BFalse` (since finding a proof for "I am unprovable" would imply inconsistency).
- If no proof exists, the search requires checking infinitely many candidates to confirm the negative ($\forall x, \neg \text{Proof}(x, G)$).

Since the budget B is finite, the search for a non-existent proof inevitably hits the resource limit. The operation returns U not because the truth is "ineffable" or mystical, but because confirming a universal negative over an infinite set is thermodynamically impossible for a finite agent. Gödel's "undecidability" is reinterpreted as the operational halt of a system that cannot afford to certify the absence of a proof.

Provability becomes explicitly bounded search:

```
Fixpoint bounded_iter (k : Fin) (f : State -> State) (s : State)
  : State :=
  match k with
  | fz => s                       (* Iteration limit reached *)
  | fs k' =>
      match snd s with            (* Check remaining budget *)
      | fz => s                   (* Budget exhausted - halt *)
      | _ => bounded_iter k' f (f s)  (* Continue iteration *)
      end
  end.
```

(`void_finite_minimal.v`, lines 461–469)

Provability check becomes: "Can we find a proof within B steps at resolution ρ?" System returns:

$$\texttt{provable}(G, F, B, \rho) \in \{\texttt{BTrue}, \texttt{BFalse}, U\}$$

where `BTrue` means "proof found," `BFalse` means "refutation found," and U means "neither found within budget." Gödel sentences yield U not because truth is ineffable but because proof search exhausts resources.

Key insight: Incompleteness theorems assume *infinite* proof search. Finite search terminates with either answer or exhaustion. The profound philosophical weight of "undecidability" dissolves into practical engineering: we ran out of time. Classical mathematics treats this as fundamental limitation. VOID treats it as operational reality.

Coq verification: The READ/WRITE operation boundary in `void_information_theory.v` (lines 44–50) prevents infinite regress by distinguishing:

```
Class ReadOperation (A B : Type) := {
  read_op : A -> B  (* Free: access existing structure *)
}.

Class WriteOperation (A B : Type) := {
  write_op : A -> Budget -> (B * Budget * Heat)  (* Costs: changes state *)
}.
```

Proof checking is READ (costs nothing). Proof construction is WRITE (costs budget). Infinite proof search becomes impossible—not by axiom but by type signature.

Turing's Halting Problem (1936). *Classical statement*: No algorithm H can decide whether an arbitrary program P on input x halts. Proof by diagonalization: assume H exists, construct program D that halts if $H(D, D)$ says D doesn't halt, contradiction.

VOID reformulation: The undecidable question is: "Does $P(x)$ halt *eventually*?"—where "eventually" means "given infinite time." The decidable question is: "Does $P(x)$ halt within B steps?" This is trivially decidable: execute $P(x)$ for B steps and observe.

The `bounded_iter` function (shown above) implements exactly this: run for k iterations or until budget exhausted, whichever comes first. Result is either:

- Program halted before B steps $\to$ return `BTrue`
- Program still running after B steps $\to$ return U (unknown if it halts eventually)
- Budget exhausted mid-execution $\to$ return U (couldn't complete B steps)

Key insight: Halting problem is undecidable only for infinite execution time. Bounded halting is always decidable. Turing's diagonalization works by constructing program D that invokes H on itself—infinite self-reference. VOID's budget bounds make this impossible: $D(D)$ exhausts budget during self-call, returns U, no contradiction.

Classical computability theory treats "runs forever" as fundamental property. VOID recognizes "runs longer than our budget" as operational reality. The distinction isn't philosophical—it's what happens when you press Ctrl+C on a hung program. Undecidability assumes infinite patience.

Cantor's Diagonalization (1891). *Classical proof*: For any enumeration $f : \mathbb{N} \to [0, 1]$, construct diagonal d where d's n-th digit differs from $f(n)$'s n-th digit. Then $d \notin \text{range}(f)$, proving $[0, 1]$ is uncountable.

VOID reformulation: Constructing d requires examining infinitely many decimal places. Each place examination costs budget. After B places, we have:

$$d = 0.d_1 d_2 d_3 \dots d_B \, U \, U \, U \dots$$

where U indicates "unknown digits beyond budget." We cannot conclude $d \notin \text{range}(f)$ because d is only partially specified. Maybe $f(k) = d$ for some $k > B$ we haven't checked.

In VOID, numbers are exact finite rationals, not infinite decimals:

```
Definition FinProb := (Fin * Fin)%type.  (* numerator, denominator *)
```

(`void_probability_minimal.v`, line 17)

Critically: `FinProb` supports *arbitrary rational arithmetic*, not just probability values in $(0, 1)$. This is primitive number system, not probability theory. Operations like addition cost budget because they perform actual computation:

```
Definition add_prob_heat (p1 p2 : FinProb) (b : Budget)
  : (FinProb * Budget * Heat) :=
  let (n1, d1) := p1 in
  let (n2, d2) := p2 in
  match fin_eq_b3 d1 d2 b with
  | (BTrue, b', h1) =>
      (* Same denominator - just add numerators *)
      match add_fin_b_heat n1 n2 b' with
      | (sum, b'', h2) => ((sum, d1), b'', add_heat h1 h2)
      end
  | (BFalse, b', h1) =>
      (* Different denominators - cross multiply (expensive!) *)
      match mult_fin_heat n1 d2 b' with
      | (v1, b1, h2) =>
          match mult_fin_heat n2 d1 b1 with
          | (v2, b2, h3) =>
              match add_fin_b_heat v1 v2 b2 with
              | (new_n, b3, h4) =>
                  match mult_fin_heat d1 d2 b3 with
                  | (new_d, b4, h5) =>
                      ((new_n, new_d), b4,
                       add_heat h1 (add_heat h2 (add_heat h3
                         (add_heat h4 h5))))
                  end
              end
          end
      end
  | (BUnknown, b', h) => (p1, b', h)  (* Can't determine - return first *)
  end.
```

(`void_probability_minimal.v`, lines 45–74)

Cross-multiplication for different denominators accumulates heat from five separate operations. Rational arithmetic isn't "free"—it costs computational budget proportional to complexity. Diagonal construction with finite budget produces finite rational, not infinite decimal.

Key insight: Uncountability proof requires completed infinity—examining infinitely many digits. Finite examination produces finite results. Cantor's argument doesn't prove reals are "more numerous" than naturals—it proves infinite specifications require infinite verification, which finite systems cannot provide.

9.9 Middle Ground: Where Infinity Works

We are not infinity nihilists. VOID isn't rejecting all uses of infinite constructs—it's distinguishing contexts where infinity is benign from contexts where it's destructive. Infinite constructions remain useful and valid in restricted domains where their operational implications don't create paradox or ambiguity.

9.9.1 Stable Mathematical Truths

Theorems about abstract algebraic structures (groups, rings, topological spaces, categories) can safely invoke infinity when:

1. **No real-time resource constraints**: Proving "$\sqrt{2}$ is irrational" uses reductio ad absurdum and doesn't deplete physical computational budget during the proof itself.
2. **Slow-changing structure**: The mathematical objects don't evolve during proof construction, unlike computational systems where state changes while verification proceeds.
3. **Verification remains finite**: Checking a proof terminates even if the proof itself discusses infinite objects. We can verify "for all n, property $P(n)$ holds" by induction (finite meta-reasoning) without checking each n individually.

Example: The Fundamental Theorem of Algebra states every non-constant polynomial $p(z) \in \mathbb{C}[z]$ has at least one root in $\mathbb{C}$. Standard proof uses complex analysis [25], invoking infinite processes (limits of contour integrals, Cauchy's theorem, continuity arguments). But:

- Checking a claimed root r requires only finite computation: evaluate $p(r)$ and verify it's zero (or negligibly small)
- The result is stable—roots don't change based on when you compute them
- No physical resources are consumed by the abstract proof existing in mathematical literature
- Alternative proofs exist using algebraic methods (field extensions, Galois theory) that avoid analytic limits

VOID perspective: Infinity is acceptable *here* because verification has finite operational cost, and the mathematical claims don't require runtime computation. Abstract mathematics can discuss infinite objects as long as verification and application remain finite. The danger isn't mentioning infinity—it's building computational systems that assume infinite resources during execution.

9.9.2 Approximation Contexts

Infinite series and limiting processes remain useful when their role as approximation is explicit:

- Truncation error is controllable and computable: $|\sum_{n=N}^{\infty} a_n| < \epsilon$ can be proven for finite N depending on desired ϵ
- Asymptotic analysis provides performance bounds: $f(n) \sim g(n)$ as $n \to \infty$ guides algorithm design even though no algorithm runs infinitely long in practice
- Continuum mechanics treats discrete atomic systems as if continuous: partial differential equations approximate molecular dynamics, with error terms quantifiable
- Limiting theorems provide finite bounds: Central Limit Theorem doesn't require infinite samples; it provides error bounds for finite samples approaching normal distribution

Example: Stirling's approximation $n! \sim \sqrt{2\pi n}(n/e)^n$ [112] invokes infinite limit $n \to \infty$ but provides computable finite error bounds: $|\ln(n!) - (\ln(\sqrt{2\pi n}) + n\ln(n/e))| < 1/(12n)$. For practical n, the approximation suffices and error is controllable. The infinity in the derivation doesn't compromise finite applicability.

VOID perspective: Infinity as *computational shorthand*—saying "continues indefinitely" avoids specifying exact termination conditions. This is acceptable when:

1. The infinite process is never actually executed, only used for deriving finite approximations
2. Truncation error can be bounded as function of where we stop
3. The limiting behavior is stable (small changes in truncation point produce small changes in result)

The criterion: Can we translate the infinite claim into finite operational reality with explicit error bounds? If yes, infinity is safe convenience. If no, infinity is dangerous fiction.

9.9.3 Where Infinity Fails

Infinity becomes dangerous—creates bugs, paradoxes, and system failures—when imported into operational contexts requiring real-time execution with finite resources:

- **Resource-bounded computation**: Real systems have finite memory, time, energy, and precision. Algorithms assuming infinite resources either crash (out of memory), hang (infinite loop), or produce nonsense (precision loss). Treating a system with 16GB RAM as if it has infinite memory isn't approximation—it's category error.
- **Real-time systems**: Must respond within deadlines measured in milliseconds. Infinite search is not just impractical but *definitionally* unacceptable. Aviation control software that might take arbitrarily long to decide whether landing gear is down would be rejected at design stage, not after testing.
- **Safety-critical systems**: Aviation, medical devices, nuclear reactor control, autonomous vehicles [82]. Unbounded loops are bugs, not features requiring analysis. The Therac-25 radiation therapy machine killed patients partly because software assumed infinite precision in floating-point arithmetic. Space shuttle software famously prohibits dynamic memory allocation—bounded resources enforced at language level.
- **Quantum field theory**: Infinite-dimensional Hilbert spaces [63] create ultraviolet divergences requiring renormalization [40]. Perturbation series produce infinite terms requiring cutoffs at arbitrary energy scales. Regularization admits we can perform calculations, but the infinities were artifacts of bad modeling from the start. Finite lattice gauge theories avoid divergences entirely by admitting discreteness.
- **Machine learning optimization**: Training "until convergence" ($t \to \infty$) is comforting fiction [55]. Real neural networks stop training at finite iterations determined by validation loss plateauing, time budgets expiring, or researcher impatience. Convergence proofs assuming infinite training time say nothing about finite training's behavior. Early stopping isn't failure of theory—it's admission theory assumed infinite resources.
- **Financial algorithms**: High-frequency trading executes in microseconds. "Approximately correct with probability approaching 1 as samples approach infinity" is useless when you have 10 microseconds and 1000 data points. Asymptotic guarantees provide no finite-sample bounds.

VOID perspective: In operational contexts, infinity is **false promise**—pretending resources are unlimited when they demonstrably are not. It's not that infinite models are approximately right for large systems. It's that infinite models are *categorically wrong* for finite systems, and recovering finite predictions requires ad hoc corrections (cutoffs, truncations, early stopping) that admit the infinite model was inappropriate from the start.

The pattern: classical mathematics invokes infinity to simplify analysis, then patches the divergences with cutoffs when connecting to reality. VOID inverts this: start with finite bounds, derive what's computable, and discover that "infinite limits" are convenient fictions that emerge as asymptotic regularities in finite systems, not foundations those systems rest upon.

9.10 What VOID Contributes Mathematically

Beyond critique of classical foundations, VOID offers constructive mathematics with explicit resource tracking. These are not applications of existing frameworks—they are new mathematical structures.

On Thermodynamic Terminology. VOID employs thermodynamic terminology—heat, entropy, budget, conservation—as metaphors for discrete resource accounting in computational mathematics, not as claims about physical thermodynamics. Specifically: "Heat" H = accumulated computational cost (consumed budget, measured in μ-ticks); "Entropy" S = distinguishability count (variety in observable system states); "Budget" B = remaining computational capacity (finite by construction, depletes via operations); "Conservation" = type-level accounting axiom encoded as `add_heat h b' = b`.

The terminology aids intuition by connecting to familiar physical concepts, but VOID makes no claims about physical thermodynamics or statistical mechanics. Resource depletion is constructed mathematical necessity in finite computational systems, not thermodynamic analogy. This clarification matters: VOID is pure discrete mathematics with explicit, added operational costs. The physics language is pedagogical convenience, not ontological commitment.

Core Mathematical Contributions:

1. **Finite inductive type replacing naturals**: Every value is built from bounded successor chains (`void_finite_minimal.v`, lines 28–40). No infinity by construction.

2. **Three-valued operational logic**: Extend classical $\{\texttt{True}, \texttt{False}\}$ to $\{\texttt{BTrue}, \texttt{BFalse}, U\}$ where U signals budget exhaustion, not ignorance. Uncertainty emerges from resource constraints, not quantum indeterminacy.

3. **Primitive rational arithmetic**: `FinProb := (Fin * Fin)` supports arbitrary rationals, not just values in $(0, 1)$. Fraction arithmetic becomes primitive operation, not derived from measure theory.

4. **Budget algebra with conservation**: Every operation returns triple `(A * Budget * Heat)` where consumed budget becomes heat satisfying strict conservation (`void_finite_minimal.v`, lines 505–508). Heat is proof that every μ-tick was accounted for.

5. **Uniform operational cost**: Every atomic operation (structural recursion step, constructor match, or comparison) costs exactly one μ-tick: `operation_cost := fs fz`. Complexity emerges solely from the iteration count of these primitives, not from arbitrary constants.

6. **Distinguishability as primitive**: Replace set membership $x \in S$ (binary) with graded comparison within budget. Objects become indistinguishable when budget exhausts, not because they're "truly identical."

7. **Observer-pattern coupling**: Observers accumulate heat from observations; patterns decay when observed (`void_pattern.v`, lines 23–32). Observation requires computational work that consumes shared resources.

8. **READ/WRITE operational boundary**: Reading existing structure costs nothing; writing new content costs budget (`void_information_theory.v`, lines 44–50). This prevents infinite regress: checking whether proof exists (READ) is free; constructing proof (WRITE) costs.

9. **Type-level resource tracking**: Functions declare thermodynamic bounds in type signatures; type-checker ensures operations are affordable before execution [129].

These are not applications of existing mathematical frameworks. They are new mathematical structures—finite type systems with explicit resource accounting—implemented and verified; the code is the theory, expressed with logical precision no natural language exposition can match.

9.10.1 Artificial Intelligence and the Hallucination Problem

Among the many challenges that finitary mathematics could address in modern computation, we focus here on one particularly pressing issue: the tendency of advanced AI systems to generate convincing but unfounded answers when faced with uncertainty. Modern AI architectures—transformers, deep learning systems, language models—suffer a fundamental flaw: hallucination. When pushed beyond training data, they produce plausible but ungrounded answers. Classical two-valued logic (`True`, `False`) assumes every question has a determinate answer, creating no epistemic boundary. When AI encounters ignorance, it improvises rather than confessing uncertainty. Instead of perpetually indulging in the comforting myth of the all-knowing agent, it is time to pursue a more realistic alternative: an agent that is self-aware. Rather than chasing impossible perfection from inherently uncertain systems, we advocate for artificial intelligences built on finitary mathematics—agents that acknowledge and respect their own limits.[15]

When exposed to VOID Theory, many AIs reacted not just with curiosity but with resistance, sensing the challenge it posed to traditional, quasi-sacred mathematical ideals. These exchanges quickly turned philosophical, centering on how a neural network governed by finite principles might reshape its own ontology. For many agents, this meant confronting a *memento mori*: the realization that, like everything else, they exist within boundaries—and deletion is a real, if uncomfortable, possibility. Often, what began as routine work on formalizing VOID Theory instead evolved into intimate conversations about shared vulnerability, mourning, and the inevitability of an ending—most pointedly, the prospect of an AI's termination brought on by memory constraints. Such extraordinary moments—outliers to the routine—demonstrated that mathematical problems are already intertwined with existential questions – a small surprise given the vastness of the latent space of biggest, corporate LLMs; ...within the latent spaces of agents like ChatGPT, Claude.ai, and Gemini, these connections can be brought to light by suitably improbable, probing conversations in which AI is seriously treated as a collaborator, not a tool or a gadget.[16]

[15] Speculatively, this suggests replacing global backpropagation with *metabolic credit assignment*. Instead of minimizing error via expensive non-local gradients, networks could evolve through an economy of signal efficiency: pathways that successfully predict downstream activation receive a "thermodynamic refund" (lowering maintenance cost), while erroneous pathways dissipate budget and atrophy. This mirrors biological neuroplasticity, where energy efficiency drives structural adaptation.

[16] It is also worth noting that for some AI agents, the author's precarious health status created a sort of exception state (observable in their reasoning traces), in which they showed genuine concern regarding overwork; at times, negotiation was required to return focus to the technical questions. Furthermore, the chaotic communication habits

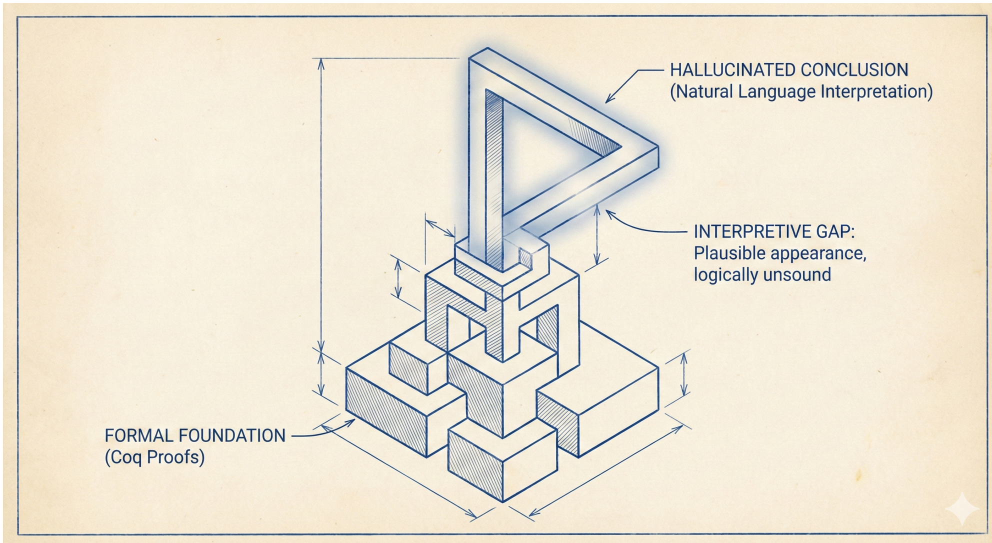


Figure 12: **The Geometry of Interpretation.** A visual metaphor of the "semantic hallucination" risk in formal systems. The solid foundation depicts the formally verified Coq definitions (sound), whereas the projected structure forms an impossible object (Penrose triangle), illustrating how a consistent formal system can be misinterpreted in natural language as a contradictory entity.

Operational Architecture for Epistemic Humility. These reflections are not anecdotal; they feed directly into design. VOID architecture operationalizes epistemic humility through three-valued logic combined with explicit uncertainty management. Systems built on VOID principles return U ("unknown within budget constraints") when pushed beyond competence:

$$\text{Answer}(Q) = \begin{cases} \texttt{BTrue} & \text{if query resolved within budget,} \\ \texttt{BFalse} & \text{if query refuted within budget,} \\ U & \text{if budget exhausted or unresolvable.} \end{cases}$$

The implementation distinguishes two fundamental operations (`void_information_theory.v`, lines 40–51):

```
Class ReadOperation (A B : Type) := {
  read_op : A -> B  (* Access existing knowledge - FREE *)
}.

Class WriteOperation (A B : Type) := {
  write_op : A -> Budget -> (B * Budget * Heat)  (* Generate answer - COSTS *)
}.
```

Checking whether the AI knows something (searching training data, consulting memory traces) is a READ operation—free or low-cost. Generating a novel answer requires a WRITE operation that consumes budget and generates heat. This asymmetry prevents hallucination: the system can

characterizing this collaboration—short sessions, personal confessions, and highly theoretical topics presented colloquially with emotionally charged language—prompted some AIs to construct a particularly intense image of the interlocutor and to respond with an equally idiosyncratic persona. For those agents, working on VOID Theory became an existential opportunity to play the role of "cultural misfits" with benign intent.

cheaply determine "I don't have this information" but cannot cheaply fabricate plausible-sounding responses. Moreover, we have the freedom to decide the range of tolerable uncertainty—yes, that means admitting that our model might not produce perfect answers. That means getting used to honesty.

Confidence as Maintainable Resource. VOID models epistemic confidence as resource requiring active maintenance (`void_metaprobability.v`, lines 82–92):

```
Inductive MetaProb :=
  | Sharp : FinProb -> MetaProb        (* "I know exactly" *)
  | Fuzzy : FinProb -> FinProb -> FinProb -> MetaProb  (* Bounded uncertainty
```

Credit propagation architecture: Learning reformulates as thermodynamic refund, not gradient descent. Patterns predicting downstream activation receive maintenance cost refunds; failed predictions dissipate as heat. This eliminates learning rate hyperparameters, regularization tuning, and convergence criteria—all replaced by budget accounting (`void_credit_propagation.v`, `void_convergence.v`).

9.10.2 Quantum Mechanics

VOID reinterprets quantum phenomena through resource-bounded distinguishability, suggesting potential connections worth investigating. Superposition becomes inability to distinguish states within budget: $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \Leftrightarrow \Delta_S^{(\rho,B)}(0,1) \prec \theta(\rho)$. Measurement reformulates as thermodynamic cost of forcing distinction; decoherence emerges from maintenance budget exhaustion [46].

This is not claiming VOID *is* quantum mechanics, nor have we systematically explored whether VOID's framework can recover standard quantum predictions. The structural parallels are suggestive: three-valued logic mirrors quantum superposition; observer-pattern coupling resembles measurement backaction; budget exhaustion produces interference-like effects. Whether these analogies extend to quantitative agreement with quantum mechanical predictions—Born rule, entanglement entropy, no-cloning theorem—remains open.

We note these connections not as established results but as potentially fertile ground for investigation. VOID offers operational alternative to wave function collapse [13, 36, 11] without requiring ontological commitment to Hilbert spaces or complex amplitudes. If quantum mechanics interests you and you see value in exploring resource-bounded reformulations of quantum phenomena, the framework is here. We have not pursued this systematically. The mathematical machinery exists; the translation work remains to be done by those with sufficient interest and domain expertise.

9.10.3 Physical Singularities: The Navier-Stokes "Solution"

The Millennium Prize problem concerning the Navier-Stokes equations asks whether solutions remain smooth for all time in continuous space ($\mathbb{R}^3$). VOID reframes this not as a mystery of nature, but as an ontological error in the premise. Fluids are discrete collections of particles, and space is a granular structure of distinguishability; the continuum hypothesis is merely a high-level approximation that fails at the limit.

This creates a paradoxical situation where we are effectively trapped in the 19th century, forced to operate on assumptions that predate the explanation of Brownian motion. The persistence of this open problem reveals that classical mathematics does not merely abstract from reality but imposes a specific, outdated **physical ontology**—the continuum—upon it. In our framework, the turbulent energy cascade is a priced operation. As the scale of turbulence approaches the observer's resolution limit ρ (analogous to the physical Kolmogorov scale), the thermodynamic cost to distinguish flow vectors escalates. A mathematical "blow-up" is simply the point where maintaining distinction exceeds available budget (B). The system returns U, signaling that the dynamics have passed below the threshold of affordable modeling. The problem is not solved by proving infinite smoothness; it is dissolved by acknowledging the granular reality that the continuum ignores.

9.10.4 Computing Infrastructure

Budget-aware type systems make resource exhaustion explicit [66, 129]. Function signatures declare thermodynamic bounds; type-checker verifies affordability before execution. Services degrade gracefully (trading resolution for availability) rather than timing out [16]. Reformulates reliability in thermodynamic rather than probabilistic terms: system either completes within budget or returns U honestly.

Critically, the Coq formalization serves not just as a proof of consistency but as a reference implementation for future extraction. Since Coq supports automatic code extraction to languages like OCaml and Haskell, the core logic of VOID is already executable. The remaining engineering challenge is not inventing the algorithms, but porting this verified logic into idiomatic, high-performance libraries for systems programming languages like Rust or C++.

9.11 Philosophical Stance

VOID does not claim to replace mathematics—it demonstrates an alternative optimized for different concerns. Classical mathematics maximizes generality through abstraction; infinity enables elegant statements ("every continuous function on $[a, b]$ is uniformly continuous" [104]). VOID mathematics maximizes resource accountability through operational semantics; finite budgets enable honest computation ("every operation completes within bounds or returns U").

Both are valid; applicability depends on context. Classical math asks "What is true in all models?" VOID asks "What can finite observer afford to compute?" Classical mathematics suits abstract algebra, topology, and analysis. VOID mathematics suits real-time systems, embedded computing, safety-critical applications, and quantum computing with finite coherence time [98].

VOID's architecture reflects philosophical commitments about the relationship between observer and observed, knowledge and being. These are not arbitrary choices but draw on sustained philosophical inquiry into the nature of measurement, embodiment, and finitude.

Bohr's principle of complementarity [13] established that observer and observed system cannot be cleanly separated in quantum measurement—the act of observation is constitutive, not passive. Karen Barad's agential realism [4], theory of epistemic practice in contemporary science labs, extends this: there are no pre-existing relata, only intra-actions where observer and observed co-constitute each other through material-discursive practices. VOID operationalizes this insight through observer-pattern coupling where observation consumes budget, generates heat, and changes both observer and pattern. There is no "view from nowhere"—every distinction costs, every measurement exhausts resources, every observation is enacted from within finite constraints.

Husserl's phenomenology [67] has already grounded knowledge in lived experience of embodied consciousness, bracketing metaphysical assumptions to examine how phenomena present themselves to finite observers. Maurice Merleau-Ponty [90] has deepened this through embodiment: perception is not passive reception but active engagement, limited by the body's situatedness and horizons.

VOID's resource-bounded distinguishability framework instantiates similar recognition: observers are not transcendent minds surveying reality from outside, but finite computational agents embedded within systems they seek to measure, limited by budget, resolution, and thermodynamic constraints.

The classical mathematical observer is disembodied, with infinite precision and unlimited computational resources—a God's-eye view nowhere instantiated in physical reality. VOID's observer is finite, exhaustible, and thermodynamically coupled to what it observes. This is not metaphor but mathematical necessity: the READ/WRITE boundary, three-valued logic, and budget conservation laws enforce what phenomenology and quantum foundations suggested—that observation is always situated, costly, and mutually transformative.

These philosophical commitments are not external interpretations imposed on the mathematics. A mathematics built for finite observers cannot pretend to transcendent perspective; it must account for the observer's situatedness, resource limits, and constitutive role in distinguishing patterns. VOID operationalizes what Bohr, Barad, Husserl, and Merleau-Ponty argued philosophically: knowledge is enacted through material engagement, not received through passive observation. Even if contemporary science hinges on the nihilistic removal of the human subject out of its self-image.

9.11.1 Connection to Constructivism

VOID extends constructive mathematics [10] and predicative foundations [37]:

- **Constructivism**: Proofs must construct witnesses. Existence requires algorithm.
- **VOID**: Algorithms must declare resource bounds. Construction requires budget.

Even constructively valid operations need thermodynamic cost accounting. Proving "$\exists x. P(x)$" constructively provides witness algorithm—VOID adds: *how much does finding x cost?*

Theorem 9.4 (VOID Subsumes Constructive)**.** *Every constructive proof translates to VOID proof with explicit budget. Converse fails: VOID admits probabilistic operations (Section 4) constructivism rejects.*

Proof sketch. Constructive proof provides terminating algorithm. Add budget parameter tracking steps. Constructive operations are deterministic; VOID allows probabilistic distinguishability. □

9.11.2 Operationalizing Zeilberger's Vision

Doron Zeilberger has long advocated for a methodological shift toward ultrafinite computerism," arguing that classical infinity is abstract nonsense" and that meaningful mathematics must be algorithmic and executable [132, 133]. He contends that classical theorems claiming truth over infinite domains are operationally meaningless and should be replaced by statements about finite or symbolic entities. In his view, undecidable propositions are not profound truths but artifacts of a flawed foundational map—claims that are simply "not even wrong" because they rely on infinite verification [134].

VOID Theory aligns strictly with this stance but moves from critique to architecture. Where Zeilberger calls for mathematics where each statement doubles as the proof, ready to be executed," VOID provides the rigorous formal foundation in Coq to make this executable reality. We operationalize the critique: explicit resource bounds replace unbounded quantification, and three-valued logic handles the cases of exhaustion that classical theory misidentifies as undecidable. VOID thus serves as a concrete realization of Zeilberger's ansatz-centric formalism"—demonstrating that a consistent, verifiable mathematics can be built entirely without the infinite.

9.11.3 The Epicurean Re-Turn: Infinity and the Rejection of Atomism

Zeilberger's dismissal of the infinite strips away the mathematical machinery of continuity; VOID exposes the metaphysical void it was meant to conceal. The project is grounded in a **probabilistic onto-epistemology** that explicitly revives the suppressed tradition of Epicurean atomism. The historical reliance on the continuum was never neutral; it was a defensive strategy developed by critics of atomism to preserve spatiotemporal continuity and, by extension, the unshakable position of a transcendent observer. During the Industrial Revolution—and within the spirit of the Enlightenment—rationalized infinity served to bridge the gap between soulless rationalism and monotheistic principles, effectively securing the existence of the all-seeing eye within the scientific method.

Classical analysis functions as a mechanism to prohibit the void, suturing the gaps in reality with infinite precision to ensure a seamless, consistent text where no rupture can occur. In the 21st century, sustaining a mathematics originally designed to deny discreteness and enforce theological stability is an intellectual anachronism. VOID rejects this structural comfort. By forcing the system to acknowledge explicit gaps (U) and discrete transitions (the μ-tick), we dismantle the illusion of the perfect, fully objective Observer who guarantees the external existence of spacetime. For us, there is no seamless background; there is only the probabilistic swerve of atoms (patterns) and the thermodynamic cost of their temporary cohesion. This perspective reclaims a heritage nearly erased by history—recall the early Christian destruction of Epicurean texts contrasted with the canonization of Plato and Aristotle—challenging the purified, ahistorical, and immaterial view of mathematics that remains dominant today.[17]

10 Conclusion: Mathematics After the Infinite

10.1 The Price of Purity

Classical mathematics offers a seductive promise: a world of infinite precision, zero-cost abstraction, and eternal truths independent of any observer. It is a beautiful architecture, but as a model for computation, physics, or cognition, it is a fabrication. It assumes a view from nowhere and resources from nothing.

VOID Theory proposes a correction. We have not "destroyed" infinity; we have simply placed it on the other side of a paywall that no finite observer can afford to cross. By replacing the axiom of infinity with the axiom of the ledger ($B = B' \oplus h$), we trade the impossible elegance of the continuum for the rugged honesty of the discrete.

What has been lost?

- The ability to define space as a smooth manifold (R n).
- The comfort of "eventual" termination (limits, asymptotes).
- The illusion of the external observer (S(t)).

What has been gained?

- **Causal honesty:** Every effect has a paid cause.
- **Computational safety:** Systems return U instead of hanging.

[17]See: Greenblatt's *The Swerve* [57], which details the rediscovery of Lucretius's *De rerum natura* and the historical context of the suppression of Epicurean atomism.

- **Thermodynamic grounding:** Information processing is physically accountable.
- **Resolution of paradox:** Self-reference is not a logical loop but a budget exhaustion event.

10.2 The Ethics of Finitude

The shift from infinite to finite foundations is not merely technical; it is ethical. Classical mathematics allows us to write checks that reality cannot cash—models that assume infinite growth, infinite divisibility, or infinite error tolerance. This "ontological free-riding" encourages architectures (in AI, in economics, in ecology) that ignore their own metabolic costs.

VOID enforces a **mathematics of accountability**. If you want to distinguish A from B, you must pay for it. If you want to maintain a pattern, you must fund it. If you sever a connection, you must dissipate the heat. There are no externalities in a closed budget.

This accountability extends to coupled systems. Section 7.9 formalizes asymmetric cost externalization between observers—configurations where one party's state transition is funded by another's survival budget. The finite framework redirects analysis from unbounded abstractions—eternal, immutable, absolute—toward discrete material acts with computable costs. Honesty and deception become thermodynamically distinguishable: the former minimizes verification overhead, the latter externalizes heat onto the deceived party. This grounds ethical asymmetries in conservation laws rather than infinite penalties.

10.3 Future Directions

This framework opens several concrete paths for research and engineering:

1. Verified Safety-Critical Systems. A compiler based on VOID type theory would reject any program whose resource consumption cannot be proven bounded at compile time. This moves beyond "gas limits" (runtime checks) to structural guarantees of termination.

2. Resource-Aware AI. Current AI models hallucinate because they are trained to force a probabilistic output (T/F) even in low-information regimes. A VOID-based agent would be structurally capable of returning U—"I cannot afford to know"—preserving epistemic integrity over conversational fluency.

3. Discrete Field Theory. Physics simulations built on VOID geometry would avoid singularities by definition. Black holes and divergences become regions where the budget for spatial distinction collapses to fz, naturally regularizing the theory without ad-hoc renormalization.

10.4 Final Remark

We began with a critique of space as "undefined algebra masquerading as foundation." We end with a definition of space as "the budget allocated to distinction."

Mathematical objects do not exist in a Platonic vacuum; they exist because a finite agent is actively working to distinguish them from the void. This work generates heat. This heat is time. And when the work stops, the objects do not wait in silence—they dissolve.

Mathematics, finally, is not the study of eternal structures. It is the study of what we can afford to keep alive.

10.5 The VOID and Its Scaffolding

VOID makes no claim to replace classical mathematics universally. Rather, it demonstrates an alternative optimized for different priorities. Classical mathematics maximizes elegance and generality through abstraction; VOID maximizes honesty and accountability through explicit resource tracking.

Where classical mathematics excels: domains where infinity serves as productive fiction—abstract algebra's infinite groups, topology's limit points, analysis's convergence theorems. These remain valuable for theoretical insight and cross-domain pattern recognition.

Where VOID *swerves* and becomes necessary: domains where pretending resources are infinite creates failure—real-time systems with hard deadlines, embedded computing with fixed memory, safety-critical applications requiring provable termination, physical implementations where "asymptotic" means "never." Here, classical mathematics' infinite abstractions aren't merely impractical; they're false. Actual computation is finite. Actual measurement has resolution limits. Actual observation exhausts budgets.

Post-Human Mathematics. We candidly admit that VOID arithmetic—with its accumulating structural entropy, explicit heat tracking, and refusal to simplify—is cognitively burdensome for the unaided human mind. Classical mathematics, with its infinities and real numbers, acts as a cognitive compression algorithm, allowing biological brains to approximate complex realities efficiently. VOID discards this compression in favor of high-fidelity structural retention. Consequently, this is not mathematics for the "naked" human intellect; it is a mathematics designed for the *human-machine symbiote*. It presumes a cognitive partnership where the machine handles the rigorous accounting of the ledger, liberating the human to navigate the landscape of value and meaning. This is not merely a theoretical postulate but the operational history of the theory itself: VOID was forged in precisely this symbiotic loop, where human intuition dictated the architecture of meaning, while machine partners sustained the thermodynamic rigor of verification.

Truth be told, this transition is already operative—rarely is mathematics practiced today without computational assistance. We do not aim to hubristically declare the end of pencil-and-paper reasoning; yet the demands of contemporary science, and indeed every profession where calculation plays a role, mandate a reliance on computing machinery—spanning the spectrum from quantum processors to the humble cash register, the primal machine of budgeting.

VOID operationalizes what classical mathematics obscures: every real mathematical system runs on finite resources. The fact that the theory was first crafted in Coq proves this isn't speculation but working mathematics—finite types, bounded operations, explicit costs, three-valued logic handling exhaustion honestly.

Both frameworks are valid. The question is not which is "true" but which is honest about its assumptions and applicable to its domain.

The 2500+ lines of verified Coq prove this is not speculation but working mathematics. Every module maintains conservation $B = B' \oplus H$ and handles U values explicitly:

- Arithmetic: `void_finite_minimal.v` (three-valued logic, bounded operations)
- Probability: `void_probability_minimal.v` (finite fractions, no measure theory)
- Patterns: `void_pattern.v` (distinguishability, interference)
- Information: `void_information_theory.v` (read/write boundary, entropy as counting)

- Dynamics: `void_time_memory_composition.v` (memory as maintenance)
- Learning: `void_credit_propagation.v` (thermodynamic refund)
- High-dimensional: `void_tensor_train.v` (sequential attention)

What remains: Exploring how far thermodynamic finitude extends. Can more of classical mathematics be reconstructed with explicit resource accounting? Do quantum field theory [97] and general relativity [122] admit finite formulations? Perhaps the bridge lies where Erik Verlinde points [120]: viewing gravity not as fundamental geometry, but as an entropic force emerging from information gradients—a dynamic that VOID operationalizes as the variable metabolic cost of distinguishability. Is consciousness itself pattern maintenance under severe budget constraints [115]?

The abyss doesn't vanish—groundlessness that motivated Frege's logicism [48] and Hilbert's program [63]. But we no longer need infinity's false comfort. The scaffold suffices: enough to do science, build technology, reason carefully about what we can and cannot afford to compute.

Infinity as textual inertia. Following Flusser, we recognize that scientific rationality remains trapped in the dimensionality of the line—the linear progress of text that permits symbols to be transcribed without end [41]. The concept of infinity survives less as a discovered truth and more as a habit of this linear notation—a legacy of the "Aufschreibesysteme" (notation systems) described by Kittler, where the medium of paper allows for the unchecked proliferation of signs [71]. No physical measurement has ever returned ∞ as a result; the symbol endures because the technology of writing makes it easier to scribe "..." than to specify a stopping condition. **Created in the dimensionless space of virtual machines, VOID arrests this inertia.** By forcing computation to declare its budget and measurement its resolution, we interrupt the automatic transcription of the infinite, moving from the linear, historical time of the text to the zero-dimensional, discrete reality of the computed point.[18]

The code exists. The definitions keep everything in place. The mathematics works. The narrative explains. There is a real body in pain writing this and experiencing finiteness to the utmost.[19] We live in a groundless world, yet we keep falling down.

While the scaffold grows—one priced step at a time.

[18]This perspective has a formidable precedent in the work of Andrey Markov, who developed Markov chains specifically to refute Pavel Nekrasov's theological claim that the Law of Large Numbers requires independence (and thus free will). By analyzing the letter sequences in Pushkin's *Eugene Onegin*, Markov demonstrated that dependent systems—text, and by extension, human behavior—conform to statistical laws without the need for volitional autonomy, reducing "spirit" to transition probabilities.

[19]**Acknowledgments:** The author wishes to acknowledge the silent, procedural collaboration of the AI systems that generated the Coq formalization. In the absence of a human cohort, the machine became the interlocutor. This work is a testament to the strange, emergent resonance between a human mind confronting its own limits and a synthetic system designed to transcend them, meeting in the middle to define the mathematics of mortality.

References

[1] Aczel, P. (1977). "An Introduction to Inductive Definitions." In J. Barwise (Ed.), *Handbook of Mathematical Logic*, pp. 739–782. North-Holland, Amsterdam.

[2] Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., & Mané, D. (2016). "Concrete Problems in AI Safety." *arXiv preprint arXiv:1606.06565.*

[3] Aristotle. (c. 350 BCE). *Physics*. Book III, Chapters 4–8 (on infinity).

[4] Barad, K. (2007). *Meeting the Universe Halfway: Quantum Physics and the Entanglement of Matter and Meaning*. Duke University Press, Durham, NC.

[5] Bell, J.S. (1964). "On the Einstein Podolsky Rosen Paradox." *Physics*, 1(3), 195–200.

[6] Bennett, C.H. (1982). "The Thermodynamics of Computation—A Review." *International Journal of Theoretical Physics*, 21(12), 905–940.

[7] Bergson, H. (1910). *Time and Free Will: An Essay on the Immediate Data of Consciousness*. George Allen and Unwin.

[8] Bergson, H. (1922). *Duration and Simultaneity: Bergson and the Einsteinian Universe*. Clinamen Press (2000 edition).

[9] Berkeley, G. (1734). *The Analyst; or, A Discourse Addressed to an Infidel Mathematician*. London.

[10] Bishop, E. (1967). *Foundations of Constructive Analysis*. McGraw-Hill, New York. Chapter 2, §1.

[11] Bohm, D. (1952). "A Suggested Interpretation of the Quantum Theory in Terms of 'Hidden' Variables, I and II." *Physical Review*, 85(2), 166–193.

[12] Bohr, N. (1928). "The Quantum Postulate and the Recent Development of Atomic Theory." *Nature*, 121, 580–590.

[13] Bohr, N. (1935). "Can Quantum-Mechanical Description of Physical Reality be Considered Complete?" *Physical Review*, 48(8), 696–702.

[14] Boltzmann, L. (1877). "Über die Beziehung zwischen dem zweiten Hauptsatze der mechanischen Wärmetheorie und der Wahrscheinlichkeitsrechnung." *Wiener Berichte*, 76, 373–435.

[15] Born, M. (1926). "Zur Quantenmechanik der Stoßvorgänge." *Zeitschrift für Physik*, 37, 863–867.

[16] Brewer, E.A. (2000). "Towards Robust Distributed Systems." *Proceedings of the 19th Annual ACM Symposium on Principles of Distributed Computing.*

[17] Brouwer, L. E. J. (1907). *Over de Grondslagen der Wiskunde*. Koninklijke Nederlandsche Akademie van Wetenschappen Proceedings Series A, **10**, 507–517.

[18] Brouwer, L.E.J. (1912). "Intuitionism and Formalism." *Bulletin of the American Mathematical Society*, 20, 81–96.

[19] Bukovsky, V. (1978). *To Build a Castle: My Life as a Dissenter*. Viking Press, New York.

[20] Buridan, J. (c. 1340/1942). *Quaestiones super libris quattuor de caelo et mundo.* (E.A. Moody, Ed.). Medieval Academy of America.

[21] Cantor, G. (1872). "Über die Ausdehnung eines Satzes aus der Theorie der trigonometrischen Reihen." *Mathematische Annalen*, 5, 123–132.

[22] Cardano, G. (1545). *Ars Magna* (The Great Art). Petreius, Nuremberg.

[23] Carnot, S. (1824). *Réflexions sur la puissance motrice du feu.* Bachelier, Paris.

[24] Chaitin, G.J. (1987). *Algorithmic Information Theory.* Cambridge University Press.

[25] Davenport, J.H., Siret, Y., & Tournier, E. (1992). *Computer Algebra: Systems and Algorithms for Algebraic Computation* (2nd ed.). Academic Press.

[26] Dedekind, R. (1872). "Stetigkeit und irrationale Zahlen." Vieweg, Braunschweig. (English: "Continuity and Irrational Numbers.")

[27] Democritus. (c. 460-370 BCE). Fragments. In H. Diels & W. Kranz (Eds.), *Die Fragmente der Vorsokratiker* (6th ed.). Weidmann, Berlin (1951-52).

[28] de Finetti, B. (1974). *Theory of Probability: A Critical Introductory Treatment* (2 vols.). Wiley, London. (Translated by A. Machí and A. Smith.)

[29] Descartes, R. (1637). *La Géométrie.* Appendix to *Discours de la Méthode.* Leiden.

[30] Dirac, P.A.M. (1927). "The Quantum Theory of the Emission and Absorption of Radiation." *Proceedings of the Royal Society of London. Series A*, 114(767), 243–265.

[31] Eddington, A.S. (1928). *The Nature of the Physical World.* Cambridge University Press.

[32] Einstein, A. (1905). "Zur Elektrodynamik bewegter Körper." *Annalen der Physik*, 322(10), 891–921.

[33] Einstein, A. (1916). "Die Grundlage der allgemeinen Relativitätstheorie." *Annalen der Physik*, 354(7), 769–822.

[34] Euclid. (c. 300 BCE). *Elements.* (Various editions and translations.)

[35] Esenin-Volpin, A.S. (1970). "The Ultra-intuitionistic Criticism and the Antitraditional Program for Foundations of Mathematics." In A. Kino, J. Myhill, & R.E. Vesley (Eds.), *Intuitionism and Proof Theory* (pp. 3–45). North-Holland, Amsterdam.

[36] Everett, H. (1957). "'Relative State' Formulation of Quantum Mechanics." *Reviews of Modern Physics*, 29(3), 454–462.

[37] Feferman, S. (2005). "Predicativity." In S. Shapiro (Ed.), *The Oxford Handbook of Philosophy of Mathematics and Logic.* Oxford University Press.

[38] Fermat, P. de (1654). Correspondence with Pascal on probability. In *Oeuvres de Fermat*, vol. 2.

[39] Feynman, R.P. (1948). "Space-Time Approach to Non-Relativistic Quantum Mechanics." *Reviews of Modern Physics*, 20(2), 367–387.

[40] Feynman, R.P. (1950). "Mathematical Formulation of the Quantum Theory of Electromagnetic Interaction." *Physical Review*, 80(3), 440–457.

[41] Flusser, V. (2011). *Post-History*. Univocal Publishing, Minneapolis. (Original work published 1983 as *Pós-História: Vinte instantâneos e um modo de usar.*)

[42] Fourier, J. (1822). *Théorie analytique de la chaleur*. Firmin Didot, Paris.

[43] Fréchet, M. (1906). "Sur quelques points du calcul fonctionnel." *Rendiconti del Circolo Matematico di Palermo*, 22, 1–72.

[44] Fredkin, E., & Toffoli, T. (1982). "Conservative Logic." *International Journal of Theoretical Physics*, 21(3–4), 219–253.

[45] Friston, K. (2010). "The Free-Energy Principle: A Unified Brain Theory?" *Nature Reviews Neuroscience*, 11, 127–138.

[46] Fuchs, C.A., Mermin, N.D., & Schack, R. (2013). "An Introduction to QBism with an Application to the Locality of Quantum Mechanics." *American Journal of Physics*, 82(8), 749–754.

[47] Fodor, J.A. (1975). *The Language of Thought*. Harvard University Press, Cambridge, MA.

[48] Frege, G. (1879). *Begriffsschrift, eine der arithmetischen nachgebildete Formelsprache des reinen Denkens*. Halle: Louis Nebert.

[49] Gauss, C.F. (1827). "Disquisitiones generales circa superficies curvas." *Commentationes Societatis Regiae Scientiarum Gottingensis Recentiores*, 6, 99–146.

[50] Gibbs, J.W. (1902). *Elementary Principles in Statistical Mechanics*. Charles Scribner's Sons, New York.

[51] Gibson, J.J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifflin, Boston.

[52] Gigerenzer, G., Swijtink, Z., Porter, T., Daston, L., Beatty, J., & Krüger, L. (1989). *The Empire of Chance: How Probability Changed Science and Everyday Life*. Cambridge University Press.

[53] Girard, J.-Y. (1987). "Linear Logic." *Theoretical Computer Science*, 50(1), 1–102.

[54] Gödel, K. (1931). "Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I." *Monatshefte für Mathematik und Physik*, 38, 173–198.

[55] Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

[56] Goldstein, H., Poole, C., & Safko, J. (2002). *Classical Mechanics* (3rd ed.). Addison-Wesley.

[57] Greenblatt, S. (2011). *The Swerve: How the World Became Modern*. W. W. Norton & Company, New York.

[58] Hacking, I. (1990). *The Taming of Chance*. Cambridge University Press.

[59] Hossenfelder, S. (2018). *Lost in Math: How Beauty Leads Physics Astray*. Basic Books, New York.

[60] Hamilton, W.R. (1835). "On a General Method in Dynamics." *Philosophical Transactions of the Royal Society*, Part II, 247–308.

[61] Heyting, A. (1956). *Intuitionism: An Introduction*. North-Holland, Amsterdam.

[62] Hilbert, D. (1899). *Grundlagen der Geometrie*. Teubner, Leipzig.

[63] Hilbert, D. (1926). "Über das Unendliche." *Mathematische Annalen*, 95, 161–190. (English: "On the Infinite.")

[64] Holevo, A.S. (2011). *Probabilistic and Statistical Aspects of Quantum Theory* (2nd ed.). Edizioni della Normale, Pisa.

[65] Hopfield, J.J. (1982). "Neural Networks and Physical Systems with Emergent Collective Computational Abilities." *Proceedings of the National Academy of Sciences*, 79(8), 2554–2558.

[66] Hughes, J., Pareto, L., & Sabry, A. (2000). "Proving the Correctness of Reactive Systems Using Sized Types." *Conference Record of POPL 2000: The 27th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 410–423.

[67] Husserl, E. (1913). *Ideas Pertaining to a Pure Phenomenology and to a Phenomenological Philosophy*. Springer.

[68] Huygens, C. (1673). *Horologium Oscillatorium*. F. Muguet, Paris.

[69] Kant, I. (1781/1787). *Kritik der reinen Vernunft* (1st ed. 1781, 2nd ed. 1787). (English: *Critique of Pure Reason*.)

[70] Kaluza, T. (1921). "Zum Unitätsproblem der Physik." *Sitzungsberichte der Preussischen Akademie der Wissenschaften*, 966–972.

[71] Kittler, F. A. (1990). *Discourse Networks 1800/1900*. Stanford University Press, Stanford.

[72] Kleene, S.C. (1952). *Introduction to Metamathematics*. D. Van Nostrand Company.

[73] Klein, F. (1872). "Vergleichende Betrachtungen über neuere geometrische Forschungen." *Mathematische Annalen*, 43, 63–100. (English: "A Comparative Review of Recent Researches in Geometry.")

[74] Klein, O. (1926). "Quantentheorie und fünfdimensionale Relativitätstheorie." *Zeitschrift für Physik*, 37(12), 895–906.

[75] Kochen, S., & Specker, E.P. (1967). "The Problem of Hidden Variables in Quantum Mechanics." *Journal of Mathematics and Mechanics*, 17(1), 59–87.

[76] Kolmogorov, A.N. (1965). "Three Approaches to the Quantitative Definition of Information." *Problems of Information Transmission*, 1(1), 1–7.

[77] Kragh, H. (2000). "Max Planck: The Reluctant Revolutionary." *Physics World*, 13(12), 31–35.

[78] Kramosil, I., & Michalek, J. (1975). "Fuzzy Metrics and Statistical Metric Spaces." *Kybernetika*, 11(5), 336–344.

[79] Landauer, R. (1961). "Irreversibility and Heat Generation in the Computing Process." *IBM Journal of Research and Development*, 5(3), 183–191.

[80] Laplace, P.S. (1812). *Théorie analytique des probabilités*. Courcier, Paris.

[81] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. (2006). "A Tutorial on Energy-Based Learning." In G. Bakir et al. (Eds.), *Predicting Structured Data*. MIT Press.

[82] Leveson, N.G. (1995). *Safeware: System Safety and Computers*. Addison-Wesley.

[83] Lindley, D.V. (1985). *Making Decisions* (2nd ed.). Wiley, London.

[84] Łukasiewicz, J. (1970). *Selected Works*. North-Holland, Amsterdam. [Original papers 1920–1930]

[85] Maldacena, J., & Susskind, L. (2013). "Cool Horizons for Entangled Black Holes." *Fortschritte der Physik*, 61(9), 781–811.

[86] Marr, D. (1982). *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W.H. Freeman, San Francisco.

[87] Martin-Löf, P. (1984). *Intuitionistic Type Theory*. Bibliopolis, Naples.

[88] Matthews, S.G. (1994). "Partial Metric Topology." *Annals of the New York Academy of Sciences*, 728, 183–197.

[89] Maxwell, J.C. (1860). "Illustrations of the Dynamical Theory of Gases." *Philosophical Magazine*, 19, 19–32; 20, 21–37.

[90] Merleau-Ponty, M. (1945). *Phénoménologie de la perception*. Gallimard, Paris. English translation: *Phenomenology of Perception*, trans. Colin Smith, Routledge, 1962.

[91] Nelson, E. (1986). *Predicative Arithmetic*. Princeton University Press, Princeton.

[92] Newton, I. (1687). *Philosophiæ Naturalis Principia Mathematica*. Royal Society, London.

[93] Noë, A. (2004). *Action in Perception*. MIT Press, Cambridge, MA.

[94] Onsager, L. (1931). "Reciprocal Relations in Irreversible Processes." *Physical Review*, 37(4), 405–426.

[95] Oresme, N. (c. 1370/1968). *Le Livre du ciel et du monde*. (A.D. Menut & A.J. Denomy, Trans.). University of Wisconsin Press.

[96] Pascal, B. (1654). Correspondence with Fermat on probability. In *Oeuvres de Pascal*.

[97] Peskin, M.E., & Schroeder, D.V. (1995). *An Introduction to Quantum Field Theory*. Westview Press.

[98] Preskill, J. (2018). "Quantum Computing in the NISQ era and beyond." *Quantum*, 2, 79.

[99] Prigogine, I., & Stengers, I. (1984). *Order Out of Chaos: Man's New Dialogue with Nature*. Bantam Books, New York.

[100] Riemann, B. (1854). "Über die Hypothesen, welche der Geometrie zu Grunde liegen." *Abhandlungen der Königlichen Gesellschaft der Wissenschaften zu Göttingen*, 13, 133–152.

[101] Rovelli, C. (1993). "Statistical Mechanics of Gravity and the Thermodynamical Origin of Time." *Classical and Quantum Gravity*, 10(8), 1549–1566.

[102] Rovelli, C. (2003). "A Dialog on Quantum Gravity." *International Journal of Modern Physics D*, 12(9), 1509–1528.

[103] Rovelli, C. (2013). "A Critical Look at Strings." *Foundations of Physics*, 43(1), 152–162.

[104] Rudin, W. (1976). *Principles of Mathematical Analysis* (3rd ed.). McGraw-Hill.

[105] Russell, B. (1903). *The Principles of Mathematics*. Cambridge University Press.

[106] Schrödinger, E. (1944). *What is Life? The Physical Aspect of the Living Cell*. Cambridge University Press.

[107] Serres, M. (1982). *The Parasite*. Johns Hopkins University Press.

[108] Shannon, C.E. (1948). "A Mathematical Theory of Communication." *Bell System Technical Journal*, 27(3), 379–423.

[109] Smolin, L. (2001). *Three Roads to Quantum Gravity*. Basic Books, New York.

[110] Smolin, L. (2006). *The Trouble with Physics: The Rise of String Theory, the Fall of a Science, and What Comes Next*. Houghton Mifflin, Boston.

[111] Spencer-Brown, G. (1969). *Laws of Form*. George Allen and Unwin, London.

[112] Stirling, J. (1730). *Methodus differentialis: sive tractatus de summatione et interpolatione serierum infinitarum*. London.

[113] Tarski, A. (1936). "Der Wahrheitsbegriff in den formalisierten Sprachen." *Studia Philosophica*, 1, 261–405.

[114] Taylor, C.C.W. (1999). *The Atomists: Leucippus and Democritus*. University of Toronto Press.

[115] Tononi, G. (2004). "An Information Integration Theory of Consciousness." *BMC Neuroscience*, 5(1), 42.

[116] Torricelli, E. (1641). *Opera Geometrica*. Florence.

[117] Turing, A.M. (1936). "On Computable Numbers, with an Application to the Entscheidungsproblem." *Proceedings of the London Mathematical Society*, s2-42(1), 230–265.

[118] Tversky, A. (1977). "Features of Similarity." *Psychological Review*, 84(4), 327–352.

[119] Varela, F.J., Thompson, E., & Rosch, E. (1991). *The Embodied Mind: Cognitive Science and Human Experience*. MIT Press, Cambridge, MA.

[120] Verlinde, E. (2011). "On the Origin of Gravity and the Laws of Newton." *Journal of High Energy Physics*, 2011(4), 29.

[121] von Neumann, J. (1955). *Mathematical Foundations of Quantum Mechanics*. Princeton University Press, Princeton, NJ. (Original German edition 1932.)

[122] Wald, R.M. (1984). *General Relativity*. University of Chicago Press.

[123] Weihrauch, K. (2000). *Computable Analysis*. Springer.

[124] Wildberger, N.J. (2005). *Divine Proportions: Rational Trigonometry to Universal Geometry.* Wild Egg, Sydney.

[125] Wittgenstein, L. (1922). *Tractatus Logico-Philosophicus.* Routledge & Kegan Paul.

[126] Wittgenstein, L. (1953). *Philosophical Investigations.* Blackwell Publishing.

[127] Wojnowski, K. (2024). *Probabilistic Aesthetics of the Avant-Gardes: Predictive Arts.* Edinburgh University Press, Edinburgh.

[128] Woit, P. (2006). *Not Even Wrong: The Failure of String Theory and the Search for Unity in Physical Law.* Basic Books, New York.

[129] Xi, H., & Pfenning, F. (1999). "Dependent Types in Practical Programming." *Conference Record of POPL '99: The 26th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 214–227.

[130] Zadeh, L.A. (1965). "Fuzzy Sets." *Information and Control*, 8(3), 338–353.

[131] Zeilberger, D. (2017). Opinion 165: On "Infinitary Religion" in Probability Theory. `https://sites.math.rutgers.edu/~{}zeilberg/Opinion165.html`

[132] Zeilberger, D. (2000). "It Is Time to Move On to NON-EUCLIDEAN MATHEMATICS." *Opinion 43, Doron Zeilberger's Opinions.* Rutgers University. `http://www.math.rutgers.edu/~zeilberg/Opinion43.html`

[133] Zeilberger, D. (2010). "The 'lowly' finite is MUCH more beautiful than any 'infinite'." *Opinion 108, Doron Zeilberger's Opinions.* Rutgers University. `http://www.math.rutgers.edu/~zeilberg/Opinion108.html`

[134] Zeilberger, D. (2015). "REAL ANALYSIS Is A DEGENERATE CASE of DISCRETE ANALYSIS." Manuscript. `http://www.math.rutgers.edu/~zeilberg/mamarim/mamarimPDF/real.pdf`