Coherent Comparison Costs from the d'Alembert Composition Law: Discrete Ledger Structure with a Lean 4 Formalization

Sebastian Pardo-Guerra,[*] Megan Simons,[†] Anil Thapa,[‡] Jonathan Washburn,[§] and Brett Werner[¶]

Recognition Physics Institute

Abstract

We study a calibrated multiplicative d'Alembert functional equation arising from the requirement that comparison costs compose coherently on ratios. The d'Alembert composition law, together with normalization and quadratic calibration at unity, uniquely determines the reciprocal cost functional $J(x) = \frac{1}{2}\left(x + x^{-1}\right) - 1$. Building on this result, we develop a discrete double-entry ledger model driven by this cost and derive several structural consequences, including atomic tick updates, closed-chain flux conservation, and the emergence of scalar potentials on connected components under an explicit cycle-closure assumption. The framework further implies a 2^d-tick periodicity constraint. A Lean 4 formalization of the core derivations is available, providing machine-verified support for the foundational results. We refer to the resulting cost-first formalism for discrete dynamics and structured comparison as *Recognition Science*.

[*] sebas@recognitionphysics.org

[†] msimons@recognitionphysics.org

[‡] athapa@recognitionphysics.org

[§] washburn@recognitionphysics.org

[¶] bwerner@recognitionphysics.org

I. INTRODUCTION

The question of what minimal structures are required to describe physical phenomena has motivated numerous theoretical frameworks. Traditional approaches begin with spacetime, fields, and equations of motion as primitive concepts. Recognition Science (RS) inverts this perspective by asking: what minimal informational constraints are necessary for recognition to occur, and what mathematical structures emerge from those constraints?

This paper develops the cost-first, ledger-based foundations of Recognition Science: starting from a composition law for comparison costs, we develop a discrete double-entry ledger model and record its main structural consequences (T1–T8), with machine-checked support in Lean 4. A companion paper isolates the keystone uniqueness theorem for the canonical reciprocal cost on $\mathbb{R}_{>0}$ in a self-contained functional-equation treatment.

A. Scope and assumptions

This manuscript is a mathematical development of a discrete ledger model driven by a reciprocal cost on ratios. All claims of "necessity" are *relative* to explicitly stated assumptions (see Definition 1).

- **Input theorem (cost forcing).** We use Theorem T5 (proved in the companion paper) to fix the canonical reciprocal cost J from the d'Alembert composition law plus normalization and quadratic calibration.

- **Ledger-model assumptions.** The ledger results assume (i) deterministic state-update semantics $S_{t+1} = U(S_t, \sigma_t)$ where σ_t is a sequence of events (Axiom L1), (ii) minimality (no intra-tick ordering metadata; Axiom L2), (iii) a conservation principle for total balance per tick, (iv) no external sources/sinks, (v) pairwise-local event updates (a single event affects only its two endpoints), and (vi) torsion-free quantized postings in $\delta\mathbb{Z}$. From (i) and (ii), we derive atomicity (Theorem T2): at most one event per tick.

- **Cycle-closure assumption (for potentials).** Double-entry and quantization constrain postings but do not, by themselves, force vanishing circulation on arbitrary cycles. For the potential results (T3–T4) we therefore additionally assume cycle closure (path-independence / no-arbitrage): the sum of postings around every directed cycle is zero.

- **Results established here.** Under these assumptions we derive: atomic tick updates (T2), balanced double-entry postings (Proposition: Double-entry constraint), algebraic consequences of quantization (Proposition T8), and, under cycle closure, cycle flux conservation (T3) and the existence/uniqueness of discrete scalar potentials on connected components (T4), as well as the minimal schedule period bound $T \geq 2^d$ with a Gray-code realization at $d = 3$ (T6–T7).

- **Additional hypotheses.** The $D = 3$ discussion (Section III J) combines the 2^d-period constraint with additional synchronization/linking hypotheses ("gap-45"/golden-angle motivation). It should be read as conditional on those extra hypotheses.

- **Out of scope.** We do not fit data, derive SI numerical constants, or claim physical completeness; links to particular physical models require additional bridging assumptions beyond the present paper.

The framework begins with a seemingly abstract question: if we wish to compare two quantities by their ratio, and we require that such comparisons compose coherently, what constraints does this impose? This question leads naturally to the d'Alembert functional equation, which encodes the requirement that comparison costs combine consistently under multiplication and division. Together with normalization and a quadratic calibration at unity, this constraint forces a unique reciprocal cost functional (Theorem T5).

Central to this manuscript is the canonical reciprocal cost functional

$$J(x) = \tfrac{1}{2}(x + x^{-1}) - 1$$

as the unique solution to the d'Alembert composition law. The full uniqueness proof of J is deferred to the companion paper; here we treat it as a keystone input and focus on the discrete ledger consequences under explicitly stated structural assumptions.

The remainder of this paper is organized as follows. Section II elucidates the philosophical connection between coherent comparison and the d'Alembert composition law, motivating why this functional equation is adopted as the primitive composition axiom. Section III develops the ledger-based framework, using the cost uniqueness theorem (T5) as input and deriving the remaining theorems as consequences. Section IV synthesizes the framework and discusses its scope. Appendix A provides Lean 4 module references for the core theorems.

II. MOTIVATION: FROM COHERENT COMPARISON TO THE D'ALEMBERT COMPOSITION LAW

The foundation of Recognition Science rests on a simple but profound insight: if recognition involves comparison, and comparison has a cost, then the requirement that comparisons compose coherently uniquely determines that cost structure. This section elucidates the philosophical and mathematical connection between the idea of coherent comparison and the d'Alembert functional equation.

A. The Primacy of Comparison

At its most fundamental level, recognition is a relational act: one entity recognizes another. This recognition involves some form of comparison—measuring similarity, difference, or correspondence. In Recognition Science, we formalize this by asking: if we compare two quantities by their ratio $x = a/b$, what "cost" or "defect" does this comparison incur?

Central to this framework is the idea that the comparison cost $F(x)$ should depend only on the ratio x itself, not on the absolute magnitudes of a and b. This reflects the intuition that recognition is fundamentally about *relationships*, not absolute values. Moreover, we require that $F(1) = 0$: when the ratio equals unity, there is perfect balance, no defect, and hence no cost. This will be stated as Axiom A1 (Normalization).

B. Coherent Composition: The d'Alembert Constraint

The key insight comes from requiring that comparisons compose coherently. Suppose we compare a to b, obtaining ratio $x = a/b$ with cost $F(x)$. Then we compare b to c, obtaining ratio $y = b/c$ with cost $F(y)$. What should be the cost of comparing a to c directly?

We have two routes to the comparison a to c:

- **Direct route:** Compare a to c directly, giving ratio $xy = (a/b)(b/c) = a/c$ with cost $F(xy)$.

- **Composed route:** Combine the two comparisons, which should somehow combine their costs.

For the framework to be coherent, these routes should be equivalent. But how do costs combine? The answer lies in the structure of the composition law itself. When we compose comparisons multiplicatively (xy), we also have access to the *relative* ratio between the two comparison ratios:

$$\frac{x}{y} = \frac{(a/b)}{(b/c)} = \frac{ac}{b^2}.$$

This quantity is not the direct ratio a/c (which is xy); rather, it compares the two intermediate ratios to each other.

To formalize coherence, we require that the cost of the composed route depends only on the costs $F(x)$ and $F(y)$ of the individual comparisons, and on the relationship between x and y (captured by x/y). Moreover, the composition should be *symmetric* in the sense that interchanging the roles of x and y (which sends $(x, y) \mapsto (y, x)$ and $(xy, x/y) \mapsto (xy, y/x)$) should yield a consistent constraint.

A natural requirement is that the composition law relates F evaluated on the pair $(xy, x/y)$ to F evaluated on x and y individually. The most general symmetric, bilinear form (in $F(x)$ and $F(y)$) that respects the multiplicative structure is:

$$F(xy) + F(x/y) = \alpha F(x)F(y) + \beta F(x) + \beta F(y) + \gamma,$$

where α, β, γ are constants. The symmetry requirement (invariance under $x \leftrightarrow y$) forces the coefficients of $F(x)$ and $F(y)$ to be equal, hence both are β.

To determine the constants, we impose natural constraints:

- **Consistency with normalization:** When $x = y = 1$, we have $F(1) + F(1) = \alpha F(1)^2 + 2\beta F(1) + \gamma$. With $F(1) = 0$ (Axiom A1), this gives $\gamma = 0$.

- **Reciprocity compatibility:** The equation should be consistent with the natural expectation that $F(x) = F(x^{-1})$ (reciprocity), which will be a consequence of the final form. This symmetry is encoded in the x/y term.

- **Scaling behavior:** For small deviations from unity, the composition should reduce appropriately. The choice $\alpha = 2, \beta = 2$ yields the standard d'Alembert form with the correct scaling properties.

This leads to the d'Alembert-type functional equation:

$$F(xy) + F(x/y) = 2F(x)F(y) + 2F(x) + 2F(y). \tag{1}$$

This equation encodes the requirement that the cost of comparing a to c via the composed route $(F(xy) + F(x/y))$ equals the cost of combining the individual comparison costs $(2F(x)F(y) + 2F(x) + 2F(y))$. The specific form of the right-hand side—combining both multiplicative $(F(x)F(y))$ and additive $(F(x) + F(y))$ terms—reflects the dual nature of composition: comparisons can be chained (multiplicative) or compared to each other (additive with interaction). The factor of 2 in each term ensures proper scaling and symmetry. For background on d'Alembert-type functional equations, see e.g. [1, 2].

This is Axiom A2 (Composition Law). While A2 can be derived from more fundamental principles about symmetric bilinear composition operators (as sketched above), for the purposes of this manuscript we take it as a primitive axiom. The key insight is that the d'Alembert form is the unique symmetric, bilinear composition law (up to the normalization and scaling choices above) that respects the multiplicative structure of ratios while maintaining coherence. Alternative composition laws can be studied, but the uniqueness theorem we use (T5) is specific to A2 together with A1 and A3.

C. Calibration and Uniqueness

The d'Alembert equation alone does not uniquely determine the cost function. We require one additional constraint to fix the scale. Axiom A3 (Quadratic calibration) specifies that in log coordinates $t = \ln x$, the cost has unit quadratic behavior at unity:

$$\lim_{t\to 0} \frac{2F(e^t)}{t^2} = 1.$$

If F is twice differentiable at unity, this is equivalent to normalizing the second derivative of the log-lift $G(t) = F(e^t)$ via $G''(0) = 1$. This condition fixes the overall scale of the cost.

Together, these three axioms (A1: Normalization, A2: Composition Law, A3: Quadratic calibration) uniquely determine the cost functional. This is the content of Theorem T5, the keystone theorem of Recognition Science. The unique solution is:

$$J(x) = \frac{1}{2}(x + x^{-1}) - 1. \tag{2}$$

Interpretation: coherent comparison together with normalization and calibration eliminates functional-form freedom; under these axioms, the cost is uniquely determined.

Primitive Axioms (A1–A3) $\to$ **T5** (Cost Unique, Keystone)

$\to$ **Law of Existence** $\to$ **T1** (Meta-Principle)

$\to$ **Ledger Structure** $\to$ **T2** (Atomic Tick)

$\to$ **T8** (Ledger Units) $\to$ **T3** (Cycle flux; under cycle closure) $\to$ **T4** (Potential)

$\to$ **T6–T7** (Minimal period) $\to$ **D=3** (conditional).

FIG. 1. High-level cost-first dependency chain used in this manuscript (assumptions made explicit in Section I A).

D. From Cost to Existence

Once the cost functional is established, a cascade of implications follows. The cost function $J(x)$ has the property that $J(x) \geq 0$ with equality only when $x = 1$. This yields the *Law of Existence*: a configuration exists (has zero cost) if and only if $x = 1$, i.e. perfect balance.

As $x \to 0^+$ or $x \to \infty$, the cost diverges: $J(x) \to \infty$. We formalize the corresponding boundary consequence as Theorem T1 (Boundary divergence / Meta-Principle) in the next subsection.

The cost function also exhibits *reciprocity*: $J(x) = J(x^{-1})$ for all $x > 0$. This symmetry is compatible with representing recognition events in reversible pairs. In this manuscript, the balanced (double-entry) posting rule is obtained from explicit ledger-model assumptions (conservation, no sources/sinks, and pairwise-local events), as developed below.

The remainder of this paper develops the discrete ledger consequences (T2–T8) under explicitly stated structural assumptions.

III. MATHEMATICAL FRAMEWORK

This section develops the ledger-based framework of Recognition Science. The keystone cost uniqueness theorem (T5) is stated here and proved in the companion paper; we then use it as an input to derive the remaining ledger-structure results under explicit structural assumptions.

The logical structure follows the *cost-first* foundation (Figure 1).

A. Notational Convention: Mathematical Necessity

To ensure precision, we establish the following convention for claims of mathematical necessity:

Definition 1 (Relative Necessity) *In this paper, a claim that property P is* mathematically necessary *means: P is provable in Lean 4 from the explicitly listed axioms, definitions, and structural assumptions stated in this document. Every such claim must reference either (i) a theorem with explicitly stated hypotheses, (ii) a lemma with citation to the Lean module and exact statement, or (iii) an explicitly labeled assumption or axiom.*

This convention ensures that claims of necessity are verifiable and not based on hidden premises. When we state that a structure is "forced" or "required," we will either prove it as a theorem (with explicit hypotheses), state it as an explicit assumption, or reference the Lean 4 verification.

B. Taxonomy: Primitive Axioms, Derived Theorems, and Structural Assumptions

To clarify what is assumed versus what is derived, we classify the foundational elements of Recognition Science according to the *cost-first* foundation. The key insight: the d'Alembert composition law is *primitive*, and everything else—including the Meta-Principle—is derived.

Notational convention: We use **A1–A3** to denote the three *cost axioms* (primitive assumptions about the cost functional), and **L1–L2** to denote the two *ledger axioms* (structural assumptions about state updates). This distinction is maintained throughout the manuscript.

This taxonomy reflects the *cost-first* foundation: the three primitive axioms (A1–A3) fix the unique cost functional J (T5), and Theorem T1 (Meta-Principle) is treated as a derived boundary theorem (Section II).

C. The Primitive Foundation: Cost Functional and Uniqueness (T5)

We begin with the keystone theorem that establishes the unique cost functional. This theorem is the foundation from which all other structures derive.

TABLE I. Taxonomy of foundational elements in Recognition Science

Category	Elements
Primitive Cost Axioms (A1–A3)	A1 (Normalization): $F(1) = 0$
	A2 (Composition Law): $F(xy) + F(x/y) = 2F(x)F(y) + 2F(x) + 2F(y)$
	A3 (Quadratic calibration): $\lim_{t\to 0} \frac{2F(e^t)}{t^2} = 1$
Derived Theorems	T5: Cost Uniqueness: $J(x) = \frac{1}{2}(x + x^{-1}) - 1$ (from A1–A3)
	Law of Existence: x exists $\Leftrightarrow J(x) = 0$
	T1 (Meta-Principle): $J(0^+) \to \infty$ (Nothing costs infinity)
	T2: Atomic Tick (from Axioms L1 + L2: minimality + non-commutativity)
	T3: Equivalence of cycle closure and path-independence (under cycle-closure assumption)
	T4: Potential Uniqueness (from T3 + discrete Poincaré lemma)
	T6: Minimal period 2^d (eight ticks for $d = 3$) (from T2 + scheduler constraints)
	T7: Coverage Lower Bound (from T6 + pigeonhole principle)
	T8: Ledger Units (proposition: algebraic consequences of quantization / discreteness assumptions)
Ledger Axioms (L1–L2)	L1: Deterministic state-update semantics ($S_{t+1} = U(S_t, \sigma_t)$)
	L2: Minimality of ledger structure (no ordering metadata)
	(From L1 + L2, we derive T2: atomicity)
Additional Structural Assumptions	Conservation principle: Total balance invariant per tick
	No external sources/sinks
	Pairwise locality of events (single event affects only its two endpoints)
	Cycle closure (path-independence / no-arbitrage)
	Discreteness: No torsion in ledger structure
	Lossless interface: Discrete-continuous mapping preserves information
Definitions	Recognition event: $(a, b) \in A \times B$
	Ledger state: $S_t \in \mathcal{S}$
	Tick: Minimal temporal unit for one state update
	Posting function: $\Delta(e, t) \in \delta\mathbb{Z}$
	Recognition structure: Directed graph $G = (X, E)$

Note: The following three axioms (A1–A3) are the *cost axioms*, distinct from the ledger axioms (L1–L2) introduced in Section III E. The cost axioms determine the functional form of the cost J, while the ledger axioms govern the discrete update structure.

Axiom A1 (Normalization). The cost at unity is zero: $F(1) = 0$. Perfect balance is free.

Axiom A2 (Composition Law). For all $x, y > 0$:

$$F(xy) + F(x/y) = 2F(x)F(y) + 2F(x) + 2F(y). \tag{3}$$

This is the d'Alembert functional equation in multiplicative form. As motivated in Section II, this form is the unique symmetric, bilinear composition law (up to normalization) that respects the multiplicative structure of ratios while ensuring coherent composition of comparison costs. It ensures that costs combine coherently under multiplicative composition of ratios.

Axiom A3 (Quadratic calibration). In log coordinates $t = \ln x$, define $G(t) = F(e^t)$. We require

$$\lim_{t\to 0} \frac{2G(t)}{t^2} = \lim_{t\to 0} \frac{2F(e^t)}{t^2} = 1.$$

This fixes the overall scale (and, when G is twice differentiable at 0, coincides with $G''(0) = 1$).

Theorem 2 (T5: Cost Uniqueness) *Let $F : \mathbb{R}_{>0} \to \mathbb{R}$ satisfy:*

1. ***Normalization (A1):*** $F(1) = 0$
2. ***Composition Law (A2):*** $F(xy) + F(x/y) = 2F(x)F(y) + 2F(x) + 2F(y)$
3. ***Quadratic calibration at unity:*** $\lim_{t\to 0} \frac{2F(e^t)}{t^2} = 1$

Then, F is uniquely determined:

$$F(x) = \frac{1}{2}(x + x^{-1}) - 1. \tag{4}$$

We denote this unique cost functional by J.

Proof. See the companion paper *Uniqueness of the Canonical Reciprocal Cost* [5] for a self-contained proof. The corresponding Lean statement is `CostUniqueness.T5_uniqueness_complete`. □

The cost functional has the following key properties:

- **Reciprocity:** $J(x) = J(x^{-1})$ for all $x > 0$
- **Non-negativity:** $J(x) \geq 0$ with equality iff $x = 1$
- **Boundary divergence:** $J(x) \to \infty$ as $x \to 0^+$ or $x \to \infty$

The boundary divergence is *not assumed*—it is a consequence of the unique functional form. This is why the Meta-Principle is derived, not primitive.

1. *Unique zero-cost configuration (Law of Existence)*

The unique cost functional immediately yields the *Law of Existence*:

Definition 3 (Perfect Balance predicate) *A configuration $x > 0$ is* in perfect balance *(in the RS sense) if and only if its defect collapses to zero, that is,*

$$\mathit{Bal}(x) \iff J(x) = 0. \tag{5}$$

The predicate $\mathrm{Bal}(x)$ is a *technical consistency predicate* in the RS framework, not a claim about ontological presence. Note that only $x = 1$ satisfies $\mathrm{Bal}(x)$ (zero defect). Indeed, starting from the definition of J:

$$J(x) = \frac{1}{2}\left(x + x^{-1}\right) - 1,$$

we have that

$$\begin{aligned} J(x) = 0 &\iff \frac{1}{2}\left(x + x^{-1}\right) - 1 = 0 \\ &\iff x + x^{-1} = 2 \\ &\iff x^2 - 2x + 1 = 0 \qquad \text{(multiply by } x > 0) \\ &\iff (x-1)^2 = 0 \\ &\iff x = 1. \end{aligned}$$

In this way, configurations with $x \neq 1$ have $J(x) > 0$ and are *recognizable* precisely because their nonzero defect is quantifiable and enables comparison and composition via the ledger rules.

Definition 4 *We say that a configuration* exists *in the RS framework if and only if it satisfies finite-cost admissibility (existence in the RS sense).*

$$\mathrm{Exists}(x) \;:\!\iff\; J(x) < \infty.$$

Therefore, every configuration $x > 0$ *exists* in this sense, as it has finite cost. In contrast, the boundary regimes $x \to 0^+$ and $x \to \infty$ correspond to cost blow-up and are thus excluded from admissible configurations. In the next subsection, we show that these divergent limits admit a natural interpretation as "nothingness" within the Recognition Science framework, marking the breakdown of recognizability rather than the presence of any physical or informational configuration.

Near equilibrium ($x = 1$), the cost function exhibits quadratic behavior. Let $x = e^\epsilon$ for small ϵ. Then

$$J(e^\epsilon) = \frac{1}{2}(e^\epsilon + e^{-\epsilon}) - 1 = \cosh(\epsilon) - 1 = \frac{\epsilon^2}{2} + \frac{\epsilon^4}{24} + \cdots \approx \frac{1}{2}\epsilon^2, \tag{6}$$

reproducing a Euclidean metric in log-space. This local quadratic structure ensures well-behaved optimization near equilibrium.

As a modeling choice, consider the recurrence equation $x_{n+1} = 1 + 1/x_n$ as a simple self-similar update rule. Fixed points satisfy $x = 1 + 1/x$, yielding the quadratic equation

$$x^2 - x - 1 = 0 \quad \Rightarrow \quad \phi = \frac{1+\sqrt{5}}{2} \approx 1.618.$$

At ϕ, the additive (self) and reciprocal (other) components balance. The recognition cost evaluates to

$$J(\phi) = \frac{1}{2}\left(\phi + \frac{1}{\phi}\right) - 1 = \phi - \frac{3}{2} \approx 0.118.$$

By reciprocity, $J(\phi) = J(\phi^{-1})$, so both ϕ and its reciprocal $1/\phi \approx 0.618$ lie at the same cost. Thus, under this update rule, ϕ marks a natural self-similar scale where the cost function exhibits special symmetry.

Lemma 5 *If $f(x_1) = f(x_2)$ with $f(x) = x + x^{-1}$, then $x_1 = x_2$ or $x_1 = 1/x_2$.*

Proof. From $x_1 + x_1^{-1} = x_2 + x_2^{-1}$, multiply by $x_1 x_2$ and factor to get $(x_1 - x_2)(x_1 x_2 - 1) = 0$. Therefore, either $x_1 = x_2$ or $x_1 x_2 = 1$, which implies $x_1 = 1/x_2$. □

The quantity $J_{\text{bit}} = \ln\phi \approx 0.481$ is a convenient log-scale reference associated with this self-similar fixed point. In applications it can be interpreted as a characteristic scale for multiplicative deviations measured in log-coordinates. No claim is made here that ϕ (or J_{bit}) is forced without additional dynamical/self-similarity hypotheses beyond the cost axioms.

D. Boundary divergence (Meta-Principle)

The Meta-Principle—"Nothing cannot recognize itself"—is not assumed but *derived* from the cost functional. Once J is established by T5, the Meta-Principle becomes a derived boundary consequence:

Definition 6 *Given sets A (recognizer) and B (recognized), a* recognition event *is an ordered pair $(a, b) \in A \times B$. This represents the minimal relational structure assumed between recognizer and recognized. We write* $\mathrm{Recognition}(A, B) = A \times B$ *for the set of all recognition events. If either set is empty, then* $\mathrm{Recognition}(A, B) = \emptyset$.

Remark 7 *To connect the ratio model $x = a/b$ to the set-theoretic statement*

$$\mathrm{Recognition}(A, B) = A \times B,$$

interpret a and b as nonnegative "availability/size" functionals of the underlying domains, e.g. $a := \mu(A)$ and $b := \mu(B)$, with $\mu(\emptyset) = 0$ and $\mu(\cdot) > 0$ for nonempty domains in the intended class. With this identification, taking $x = a/b$ and holding $b > 0$ fixed, the limit $x \to 0^+$ is precisely $\mu(A) \to 0^+$, i.e. the recognizer domain A is depleted toward emptiness/zero-substrate. In that regime there are no admissible recognition events, since $\mu(A) = 0$ corresponds to $A = \emptyset$ (or "no available elements"), hence $A \times B = \emptyset$ and in particular $\mathrm{Recognition}(\emptyset, \emptyset) = \emptyset$.

Theorem 8 (T1: Boundary divergence (Meta-Principle)) Nothing cannot recognize itself*:*

$$\mathrm{Recognition}(\emptyset, \emptyset) = \emptyset.$$

This is a logical tautology (the Cartesian product of empty sets is empty). In the ratio model, the corresponding boundary statement is that approaching $x \to 0^+$ incurs infinite cost:

$$\lim_{x \to 0^+} J(x) = +\infty. \tag{7}$$

In particular, the $x \to 0^+$ limit lies outside the finite-cost regime of the model.

Proof. From the uniqueness theorem (T5), $J(x) = \frac{1}{2}(x + x^{-1}) - 1$. Now, rewrite J as

$$J(x) = \frac{x}{2} + \frac{1}{2x} - 1.$$

Then, as $x \to 0^+$, each term tends to

$$\frac{1}{2x} \to +\infty, \qquad \frac{x}{2} \to 0, \qquad -1 \to -1,$$

so the sum diverges to $+\infty$:

$$\lim_{x \to 0^+} J(x) = +\infty.$$

Therefore, by Remark 7, any configuration approaching "Nothing" has unbounded cost. □

The Meta-Principle is thus no longer a mysterious pre-logical axiom, but rather a *derived consistency constraint*: the formalism assigns infinite cost to the limiting regime $x \to 0^+$ (with $J(x) \to \infty$), thereby excluding "nothingness" configurations from the domain of finite-cost recognition dynamics.

E. Single-event updates and atomic ticks

The atomic-tick principle emerges from the requirement to record recognition events unambiguously. We model recognition dynamics through a *ledger*: a sequential record updated once per tick. When a spatial carrier is assumed later in the paper, we use hypercubic graphs Q_d as a convenient family for analyzing scheduling and coverage constraints; the arguments below do not depend on a specific embedding of the ledger in $\mathbb{R}^d$.

To derive atomicity, we must specify the minimal structural constraints on how the ledger records recognition events. We introduce the following axioms:

Axiom L1 (Deterministic State-Update Semantics). The ledger state S_t at tick t evolves deterministically according to a function $U : \mathcal{S} \times \mathcal{E}^* \to \mathcal{S}$, where $\mathcal{S}$ is the state space, $\mathcal{E}$ is the set of recognition events, and $\mathcal{E}^*$ denotes finite sequences of events (including the empty sequence). The state update rule is:

$$S_{t+1} = U(S_t, \sigma_t), \tag{8}$$

where $\sigma_t \in \mathcal{E}^*$ is a finite sequence of recognition events at tick t (possibly empty, possibly containing multiple events). The function U is deterministic: for fixed S_t and σ_t, the resulting state S_{t+1} is uniquely determined.

Axiom L2 (Minimality of Ledger Structure). The ledger records only final states at each tick: S_t contains no event-ordering metadata beyond the tick index itself. Recognition events do not commute in general—there exist recognition sequences $\sigma, \sigma' \in \mathcal{E}^*$ such that $U(S, \sigma) \neq U(S, \sigma')$ even when σ and σ' contain the same events in different orders. Thus, the ledger includes no structure beyond what is necessary for unambiguous recording under Axiom L2.

1. Ledger semantics: events, postings, and state updates

To make subsequent statements (atomic ticks, quantized postings, and cycle flux) mathematically precise, we adopt the following minimal ledger semantics.

Definition 9 (Recognition structure) *Fix a directed graph $G = (X, E)$, where X is the set of nodes and $E \subseteq X \times X$ is the set of directed edges. We assume E is closed under reversal: if $(u \to v) \in E$ then $(v \to u) \in E$.*

Definition 10 (Ledger state as balances) *Fix a nonzero increment $\delta > 0$. A ledger state at tick t is a balance function*

$$S_t \equiv b_t : X \longrightarrow \delta\mathbb{Z}.$$

We write the total balance as

$$\mathcal{B}(S_t) \; := \; \sum_{x \in X} b_t(x),$$

assuming this sum is well-defined (e.g. X finite, or b_t has finite support).

Definition 11 (Event-to-posting map) *By Theorem T2, at most one recognition event occurs per tick. Therefore, we can write the state update as $S_{t+1} = U(S_t, e_t)$ where $e_t \in E$ is a single oriented edge (or the empty event if no recognition occurs). Given this deterministic update rule, define the induced* node postings *(balance increments) as*

$$\mathsf{Post}(S_t, e_t)(x) \; := \; \big(U(S_t, e_t)\big)(x) \; - \; S_t(x) \in \delta\mathbb{Z}.$$

Equivalently, $S_{t+1} = S_t + \mathsf{Post}(S_t, e_t)$ as functions $X \to \delta\mathbb{Z}$.

Definition 12 (Edge postings and cycle flux) *Under the pairwise-locality assumption introduced below (so that only the endpoints (u, v) can change at tick t), the posting is determined by a single magnitude*

$$\Delta_t \; := \; \mathsf{Post}(S_t, e_t)(v) \; = \; -\mathsf{Post}(S_t, e_t)(u) \in \delta\mathbb{Z}.$$

We then define the corresponding edge-posting function *$\Delta(\cdot, t) : E \to \delta\mathbb{Z}$ by*

$$\Delta(u \to v, t) = \Delta_t, \qquad \Delta(v \to u, t) = -\Delta_t, \qquad \Delta(e, t) = 0 \text{ for all other } e \in E.$$

For any directed cycle $\gamma = (e_1, \ldots, e_n)$ in G, define the cycle flux

$$\Phi(\gamma, t) \; := \; \sum_{i=1}^{n} \Delta(e_i, t).$$

Remark 13 (Relation to more general flow formalisms) *More general finitary flow formalisms (local finiteness, inflow/outflow sums, and "closed-chain sums") can be used to relate event structure to conservation and exactness on large or infinite graphs. The definitions above are the minimal finite-support specialization needed for the present manuscript.*

We now state the main result:

Theorem 14 (T2: Atomic tick) *Under Axioms L1 and L2, at most one recognition event is processed per tick. There are no concurrent recognitions.*

Proof. Suppose, for contradiction, that multiple recognition events can occur in a single tick t. Then σ_t contains at least two events.

By Axiom L2, the ledger state S_t contains no event-ordering metadata beyond the tick index t itself. However, also by Axiom L2, recognition events do not commute in general: there exist sequences $\sigma, \sigma' \in \mathcal{E}^*$ containing the same events in different orders such that $U(S, \sigma) \neq U(S, \sigma')$ for some state S.

Now consider two cases for the sequence σ_t:

Case 1: The order of events in σ_t matters, i.e., there exists a permutation $\pi(\sigma_t)$ of the events in σ_t such that $U(S_t, \sigma_t) \neq U(S_t, \pi(\sigma_t))$. In this case, to determine S_{t+1} unambiguously from S_t and the tick index t alone, the ledger would need to record the order in which events occurred within tick t. But Axiom L2 explicitly forbids such intra-tick ordering metadata. This is a contradiction.

Case 2: The order of events in σ_t does not matter, i.e., for all permutations $\pi(\sigma_t)$ of the events in σ_t, we have $U(S_t, \sigma_t) = U(S_t, \pi(\sigma_t))$ for all states S_t. This would mean that the events in σ_t commute. However, Axiom L2 asserts that non-commutativity occurs *in general*—there exist event sequences where order matters. If we allow arbitrary sequences σ_t with multiple events, we cannot guarantee that the events in σ_t will always commute. Since the ledger must work unambiguously for all possible event sequences (subject to the constraints), and we cannot rely on commutativity holding in general, we must exclude sequences with multiple events.

Therefore, to ensure unambiguous state updates without ordering metadata (as required by Axiom L2), we must have σ_t containing at most one event. This completes the proof. □

Theorem T2 establishes discrete temporal order: time advances in atomic steps. The proof shows that atomicity is a necessary consequence of the minimality constraint (Axiom

L2) when combined with the fact that events do not commute in general. As a corollary, we can restrict the domain of U to $\mathcal{S} \times \mathcal{E}$ (single events) rather than $\mathcal{S} \times \mathcal{E}^*$ (sequences), since sequences of length greater than one are excluded by T2.

2. Balanced postings (double-entry)

Theorem T2 establishes *atomicity* but not the posting *structure*. We now show that, under explicit structural assumptions (conservation, no sources/sinks, and pairwise-local events), each recognition event must be recorded as a balanced debit–credit pair (double-entry). The reciprocity property $J(x) = J(x^{-1})$ is then naturally compatible with reversing the orientation of an event without changing its cost.

Structural Assumption: Conservation Principle. The total ledger balance is invariant at each tick: if $\mathcal{B}(S_t)$ denotes the total balance (sum over all nodes) of state S_t, then $\mathcal{B}(S_{t+1}) = \mathcal{B}(S_t)$ for all t.

Structural Assumption: No External Sources or Sinks. Postings are the only state-changing operations. There are no auxiliary fields, external flows, or hidden variables that can absorb or supply balance.

Structural Assumption: Pairwise locality of events. Each recognition event e_t designates an ordered pair of nodes (u, v), and the update $S_{t+1} = U(S_t, e_t)$ can change balances only at those two nodes.

Remark 15 *Why pairwise locality is required for "exactly two postings". Conservation alone implies only that the* net *change in total balance per tick is zero. Without a locality condition, the deterministic update rule $U(S_t, e_t)$ could (in principle) redistribute balance across many nodes in response to a single event input e_t, while still preserving the total sum. Therefore, the "exactly two postings" conclusion requires a modeling commitment that a recognition event is* pairwise *at the ledger level (an ordered pair of nodes, i.e. a directed edge), so that only the event's endpoints may change on that tick.*

Proposition 16 (Double-entry constraint) *Under the following assumptions:*

1. *Atomicity: At most one recognition event per tick (Theorem T2)*
2. *Conservation: Total balance is invariant per tick*

3. *No external sources/sinks: Postings are the only balance-changing operations*

4. *Self-contained state updates: The state S_{t+1} depends only on S_t and the recognition event e_t (Axiom L1, with atomicity from T2)*

5. *Pairwise locality of events (structural assumption above)*

each recognition event must be self-balancing: it records exactly two postings of equal magnitude and opposite sign on the participating nodes, $+\Delta_t$ and $-\Delta_t$ (credit and debit). If postings are quantized in $\delta\mathbb{Z}$, then $\Delta_t \in \delta\mathbb{Z}$.

Proof. Fix a tick t and suppose a recognition event e_t occurs (if no event occurs, the claim is vacuous). By pairwise locality, only two node balances can change in passing from S_t to S_{t+1}; call them u and v. By conservation, the net change in total balance is zero, so the balance change at u must be the negative of the balance change at v. Writing the change magnitude as Δ_t, the event therefore records exactly two opposite postings, $-\Delta_t$ at u and $+\Delta_t$ at v. □

Therefore, under the explicitly stated structural assumptions, double-entry accounting (balanced debit–credit pairs) is required. When we additionally assume quantization in $\delta\mathbb{Z}$ (Section I A), each event records $+\Delta_t$ and $-\Delta_t$ with $\Delta_t \in \delta\mathbb{Z}$. The reciprocity property $J(x) = J(x^{-1})$ supports treating the reversed event (v, u) as the same-cost counterpart of (u, v).

To illustrate the preceding ideas, we recall the definition of a *recognition structure* (Definition 9) and consider the following example:

Example 17 (Recognition structure) *Consider a recognition structure $G = (X, E)$ with four nodes $X = \{a, b, c, d\}$ and directed edges representing recognition relations:*

- *$a \to b$: node a recognizes node b*
- *$b \to c$: node b recognizes node c*
- *$c \to d$: node c recognizes node d*
- *$d \to a$: node d recognizes node a*
- *$a \to c$: node a recognizes node c*

This forms a directed graph with a cycle ($a \to b \to c \to d \to a$) and an additional edge ($a \to c$) creating a shortcut. At each tick t, the ledger assigns postings $\Delta(e, t) \in \delta\mathbb{Z}$ to each edge e. For instance, at tick $t = 1$, we might have:

$$\Delta(a \to b, 1) = +\delta,$$
$$\Delta(b \to a, 1) = -\delta,$$
$$\Delta(e, 1) = 0 \quad \textit{for all other edges } e \in E.$$

The double-entry rule ensures that for each recognition event, if node x recognizes node y with posting $+\delta$ on edge ($x \to y$), then node x records a debit and node y records a credit, maintaining balance.

F. Quantized posting units (δ)

The atomic tick structure with double-entry raises a fundamental question: what is the minimal unit δ? In this manuscript, *quantization* (postings taking values in $\delta\mathbb{Z}$ for some $\delta > 0$, together with torsion-free uniqueness of representation) is treated as an explicit structural assumption of the discrete ledger model (Section I A). Proposition T8 records the resulting algebraic consequences.

1. Quantization as a discreteness assumption

Assume postings take values in $\delta\mathbb{Z}$ for some $\delta > 0$ and the ledger has no torsion (unique integer representation). The following proposition records the algebraic structure that follows from this quantization assumption:

Proposition 18 (T8: Ledger Units) *Under the quantization assumption (postings in $\delta\mathbb{Z}$ with $\delta > 0$ and no torsion), the set of all ledger increments*

$$\Delta = \{k\delta \mid k \in \mathbb{Z}\}$$

forms a cyclic additive group $(\Delta, +)$ isomorphic to $\mathbb{Z}$ under the mapping $k \mapsto k\delta$.

Moreover, all ledger values are integer multiples of δ:

$$x = n\delta, \qquad n \in \mathbb{Z}$$

with unique representation (quantization).

Proof. Formally, consider the set of all possible ledger increments:

$$\Delta = \{k\delta \mid k \in \mathbb{Z}\}.$$

This set forms an additive group under the natural addition operation. The mapping $k \mapsto k\delta$ provides a group homomorphism from $\mathbb{Z}$ to Δ.

Since $\delta \neq 0$ and the ledger structure has no torsion (no finite-order elements other than zero), this homomorphism is injective. Moreover, by construction, every element of Δ is of the form $k\delta$ for some $k \in \mathbb{Z}$, so the homomorphism is surjective. Therefore, $(\Delta, +) \simeq \mathbb{Z}$ as additive groups.

The uniqueness of representation follows from the absence of torsion: if $k_1\delta = k_2\delta$ for $k_1, k_2 \in \mathbb{Z}$, then $(k_1 - k_2)\delta = 0$. Since $\delta \neq 0$ and there is no torsion, we must have $k_1 = k_2$. Therefore, every ledger value x has a unique representation as $x = n\delta$ for some $n \in \mathbb{Z}$.

This quantization is an explicit structural assumption of the discrete ledger model. The algebraic structure ensures that all ledger operations occur in discrete, countable units, providing a foundation for conservation-style statements. □

Notice that the algebraic structure $(\Delta, +) \simeq \mathbb{Z}$ forbids fractional ledger amounts: every recognition event posts exactly $\pm\delta$ (or integer multiples thereof), all balances are integer multiples of δ, and the isomorphism ensures each amount has a unique integer representation. Each ledger step corresponds to one unit of recognition, guaranteeing unique integer counts for all ledger states. The ledger's arithmetic parallels, at a structural level, how certain physical quantities (e.g., electric charge) appear in discrete units.

G. Cycle flux conservation

Double-entry and quantization constrain how values can be posted, but they do not by themselves force vanishing *circulation* around every graph cycle. To obtain a scalar-potential representation, we therefore impose an additional cycle-closure (path-independence / no-arbitrage) assumption.

Structural Assumption: Cycle closure. For each tick t and every directed cycle γ in the recognition graph, the cycle flux is zero: $\Phi(\gamma, t) = 0$.

Remark 19 (Deriving cycle closure in richer settings) *In more detailed formalisms, vanishing closed-chain sums ("exactness") can be derived from additional structural hypothe-*

ses (for example, well-foundedness / absence of directed cycles in a suitable evolution relation). In this manuscript, we surface the needed hypothesis directly as the explicit cycle-closure assumption above.

Before stating Theorem T3, we formalize the notion of path-independence. For a fixed tick t, let $P_{x\to y}$ denote a directed path from node x to node y in the recognition graph. Define the path sum

$$\Phi(P_{x\to y}, t) := \sum_{e\in P_{x\to y}} \Delta(e, t),$$

where the sum is taken over edges in the order they appear along the path.

Definition 20 (Path-independence) *The edge posting function $\Delta(\cdot, t)$ is* path-independent *if for any two nodes x, y in the same connected component and any two directed paths $P_{x\to y}$ and $P'_{x\to y}$ from x to y, we have $\Phi(P_{x\to y}, t) = \Phi(P'_{x\to y}, t)$.*

Theorem 21 (T3: Equivalence of cycle closure and path-independence) *Fix a tick t and assume the recognition structure is connected. The following are equivalent:*

1. ***Cycle closure:*** *For every directed cycle γ, $\Phi(\gamma, t) = 0$.*
2. ***Path-independence:*** *For any nodes x, y and any two directed paths $P_{x\to y}$, $P'_{x\to y}$ from x to y, we have $\Phi(P_{x\to y}, t) = \Phi(P'_{x\to y}, t)$.*

Proof. **(1) $\Rightarrow$ (2):** Suppose cycle closure holds. Fix nodes x, y and two directed paths $P_{x\to y}$ and $P'_{x\to y}$ from x to y. Consider the closed walk formed by following $P_{x\to y}$ from x to y, then following $P'_{x\to y}$ in reverse from y back to x.

More formally: let $\overline{P'_{x\to y}}$ denote the reverse of path $P'_{x\to y}$ (same edges in reverse order with reversed orientation). The concatenation $\gamma = P_{x\to y} \circ \overline{P'_{x\to y}}$ forms a closed walk from x back to x. By Definition 9, the recognition structure is closed under reversal, and by Definition 12, we have $\Delta(v \to u, t) = -\Delta(u \to v, t)$ for any edge. Therefore, $\Phi(\overline{P'_{x\to y}}, t) = -\Phi(P'_{x\to y}, t)$.

The closed walk γ can be decomposed into a collection of directed cycles. This is a standard graph-theoretic fact: any closed walk in a directed graph can be decomposed into edge-disjoint cycles, possibly with some edges appearing in both directions, in which case their contributions cancel (see e.g., standard graph theory texts). By the cycle-closure assumption, each directed cycle in this decomposition has zero flux. Therefore, $\Phi(\gamma, t) = \Phi(P_{x\to y}, t) - \Phi(P'_{x\to y}, t) = 0$, which implies $\Phi(P_{x\to y}, t) = \Phi(P'_{x\to y}, t)$.

(2) $\Rightarrow$ (1): Suppose path-independence holds. Let $\gamma = (v_0, e_1, v_1, e_2, \ldots, e_n, v_n)$ be a directed cycle with $v_n = v_0$. Consider the path $P_{v_0 \to v_0}$ that follows γ once (from v_0 along the cycle back to v_0), and the trivial path $P'_{v_0 \to v_0}$ that remains at v_0 with no edges. The trivial path has $\Phi(P'_{v_0 \to v_0}, t) = 0$ (empty sum). By path-independence, $\Phi(P_{v_0 \to v_0}, t) = \Phi(P'_{v_0 \to v_0}, t) = 0$. Since $P_{v_0 \to v_0}$ following γ has flux $\Phi(\gamma, t)$, we conclude $\Phi(\gamma, t) = 0$. $\square$

Theorem T3 establishes that cycle closure is equivalent to path-independence, which is the discrete analogue of a "curl-free" condition. In continuum settings, curl-free vector fields are precisely those that admit scalar potentials, and the discrete version of this connection is made explicit in Theorem T4.

Remark 22 (Heuristic continuum limit) *With additional scaling assumptions and a chosen continuum embedding, node-level conservation can be related to a continuity equation of the schematic form*

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \mathbf{J} = 0.$$

This remark is interpretive and not used in the discrete derivations.

Example 23 (Cycle Flux Conservation) *Consider the cycle $\gamma = (a \to b \to c \to d \to a)$ from Example 17. At a fixed tick t, suppose the edge postings are:*

$$\Delta(a \to b, t) = +2\delta,$$
$$\Delta(b \to c, t) = +\delta,$$
$$\Delta(c \to d, t) = -3\delta,$$
$$\Delta(d \to a, t) = 0.$$

The cycle flux is:

$$\Phi(\gamma, t) = (+2\delta) + (+\delta) + (-3\delta) + (0) = 0.$$

By Theorem T3 (cycle closure), this must always be zero. The example illustrates a choice of edge postings whose signed sum around the closed loop vanishes. If $\Phi(\gamma, t) \neq 0$, it would violate the cycle-closure condition required for the potential representation in Theorem T4.

H. Discrete potential representation (potential uniqueness)

By Theorem T3, the cycle-closure assumption is equivalent to path-independence: the sum of postings along any open path depends only on its endpoints, not on the specific route taken. This is the discrete analogue of a curl-free vector field descending from a scalar potential.

Formally, the ledger postings form a 1-cochain on the recognition structure: each oriented edge $e = (x \to y)$ carries a posting $\Delta(e,t) \in \delta\mathbb{Z}$ at tick t. Under the cycle-closure assumption, Theorem T3 guarantees that this cochain is closed (path-independent): for every cycle γ, the sum $\Phi(\gamma,t) = \sum_{e\in\gamma} \Delta(e,t)$ vanishes. The discrete Poincaré lemma provides the existence and uniqueness of a potential function that generates these postings.

Lemma 24 (Antisymmetry from cycle closure) *Assume the recognition structure is closed under reversal (if $(x \to y) \in E$ then $(y \to x) \in E$). If cycle closure holds at tick t for all directed cycles, then for every edge $(x \to y) \in E$ we have*

$$\Delta(y \to x, t) = -\Delta(x \to y, t).$$

Proof. Fix an edge $(x \to y) \in E$. By reversal-closure, $(y \to x) \in E$ as well. The directed 2-cycle $\gamma = (x \to y \to x)$ is a cycle, so by cycle closure

$$0 = \Phi(\gamma,t) = \Delta(x \to y, t) + \Delta(y \to x, t),$$

which implies $\Delta(y \to x, t) = -\Delta(x \to y, t)$. □

Definition 25 *A potential function at tick t on a connected component $\mathcal{C} \subseteq X$ is a map $p_t : \mathcal{C} \to \delta\mathbb{Z}$ such that for each edge $e = (x \to y)$ in $\mathcal{C}$, the edge difference reproduces the posting: $\Delta(x \to y, t) = p_t(y) - p_t(x)$. This is the standard definition of a discrete gradient.*

Lemma 26 (Discrete Poincaré lemma) *Let $G = (X, E)$ be a connected graph and let $\omega : E \to \delta\mathbb{Z}$ be an antisymmetric function: $\omega(y \to x) = -\omega(x \to y)$. If the sum of ω around every cycle is zero, then there exists $p : X \to \delta\mathbb{Z}$ such that $\omega(x \to y) = p(y) - p(x)$. The function p is unique up to an additive constant.*

Proof. **Existence:** Fix a spanning tree T of G and a root $v_0 \in X$. For any $v \in X$, there is a unique simple path $P_{v_0 \to v}$ in T from v_0 to v. Define

$$p(v) := \sum_{e \in P_{v_0 \to v}} \omega(e) \in \delta\mathbb{Z}, \quad p(v_0) := 0.$$

This is well-defined because T is a spanning tree, so the path $P_{v_0 \to v}$ is unique.

Verification for tree edges: For an edge $e = (x \to y)$ in T, the unique paths $P_{v_0 \to x}$ and $P_{v_0 \to y}$ differ by exactly the edge e. More precisely, $P_{v_0 \to y} = P_{v_0 \to x} \circ (x \to y)$ (concatenation). Therefore,

$$p(y) = \sum_{f \in P_{v_0 \to y}} \omega(f) = \sum_{f \in P_{v_0 \to x}} \omega(f) + \omega(e) = p(x) + \omega(e),$$

so $p(y) - p(x) = \omega(e)$ for all tree edges.

Verification for non-tree edges: If $e = (x \to y) \notin T$, then adding e to T creates a unique fundamental cycle C (since T is a spanning tree, there is exactly one cycle containing e). This cycle consists of e plus the unique path in T from y to x, call it $P^T_{y \to x}$. By hypothesis, the sum of ω around this cycle is zero:

$$0 = \sum_{f \in C} \omega(f) = \omega(e) + \sum_{f \in P^T_{y \to x}} \omega(f).$$

Since $P^T_{y \to x}$ is a path in T from y to x, and by antisymmetry $\omega(y \to x) = -\omega(x \to y)$ for edges in T, we have

$$\sum_{f \in P^T_{y \to x}} \omega(f) = -\sum_{f \in P^T_{x \to y}} \omega(f) = -(p(y) - p(x)),$$

where $P^T_{x \to y}$ is the unique path in T from x to y. Therefore,

$$0 = \omega(e) - (p(y) - p(x)),$$

which implies $\omega(e) = p(y) - p(x)$.

Uniqueness: Suppose $\tilde{p} : X \to \delta\mathbb{Z}$ also satisfies $\omega(x \to y) = \tilde{p}(y) - \tilde{p}(x)$ for all edges $(x \to y) \in E$. Then for any edge $(x \to y)$,

$$(\tilde{p}(y) - \tilde{p}(x)) - (p(y) - p(x)) = \omega(x \to y) - \omega(x \to y) = 0,$$

so $(\tilde{p} - p)(y) = (\tilde{p} - p)(x)$ for all adjacent vertices. Since G is connected, this implies $\tilde{p} - p$ is constant on X. Setting the constant by choosing $\tilde{p}(v_0) = p(v_0) = 0$ (or any fixed value) determines $\tilde{p}$ uniquely up to this choice. $\square$

Applying the discrete Poincaré lemma to the ledger postings $\Delta(\cdot, t)$ and using Theorem T3 (cycle closure) yields the following result. Antisymmetry of $\Delta(\cdot, t)$ follows from cycle closure together with reversal-closure of the recognition structure (Lemma above).

Theorem 27 (T4: Potential Uniqueness) *Fix a tick t and a connected component $\mathcal{C} \subseteq X$. Under Theorem T3, there exists a potential*

$$p_t : \mathcal{C} \longrightarrow \delta\mathbb{Z}$$

such that for each edge $e = (x \to y)$ in $\mathcal{C}$,

$$\Delta(e, t) = p_t(y) - p_t(x).$$

Moreover, p_t is unique up to an additive constant on $\mathcal{C}$.

Proof. Theorem T3 asserts cycle closure: for every cycle γ, $\Phi(\gamma, t) = 0$. This means the ledger postings $\Delta(\cdot, t)$ form a closed 1-cochain: the sum around any closed cycle is zero.

The discrete Poincaré lemma (proved above) provides the key tool: if a 1-cochain ω is closed (all cycle sums vanish), then there exists a potential function p such that $\omega = \delta p$, where δp denotes the discrete gradient (edge differences of p).

Applying this to the ledger postings: since $\Delta(\cdot, t)$ is closed by T3, and is antisymmetric by Lemma (antisymmetry from cycle closure), the discrete Poincaré lemma guarantees the existence of a potential p_t such that $\Delta(x \to y, t) = p_t(y) - p_t(x)$ for all edges.

Uniqueness up to an additive constant follows from the fact that if $\tilde{p}_t$ also satisfies $\Delta(x \to y, t) = \tilde{p}_t(y) - \tilde{p}_t(x)$, then $(\tilde{p}_t - p_t)(y) - (\tilde{p}_t - p_t)(x) = 0$ for all edges, implying $\tilde{p}_t - p_t$ is constant on each connected component.

The proof is constructive: fix a spanning tree, choose a root vertex, and define the potential by summing postings along tree paths. The cycle condition (T3) ensures this definition is consistent for all edges. □

Theorem T4 establishes that every admissible pattern of recognitions arises from a scalar potential. This potential is unique up to an additive constant on each connected component, reflecting the gauge freedom familiar in classical physics. In the present framework, the potential representation is a direct consequence of (i) antisymmetry of postings under edge reversal and (ii) the cycle-closure condition (T3), which together encode path-independence.

Note that if an edge carries a single posting $\Delta(e, t) = \pm\delta$, then $p_t(y) - p_t(x) = \pm\delta$. More generally, $\Delta(e, t) = k\delta$ implies $p_t(y) - p_t(x) = k\delta$ for some integer k.

Example 28 (Potential Function on a Small Graph) *Consider the recognition structure from Example 17 with nodes $\{a, b, c, d\}$ and the cycle $(a \to b \to c \to d \to a)$ plus edge*

$(a \to c)$. At tick t, suppose the edge postings are:

$$\begin{aligned}
\Delta(a \to b, t) &= +2\delta, \\
\Delta(b \to c, t) &= +\delta, \\
\Delta(c \to d, t) &= -3\delta, \\
\Delta(d \to a, t) &= 0, \\
\Delta(a \to c, t) &= +3\delta.
\end{aligned}$$

Since $\Phi(a \to b \to c \to d \to a, t) = 0$ (as verified in Example 23), Theorem T4 guarantees a potential exists. Following the constructive proof of the discrete Poincaré lemma, choose a as the reference vertex and set $p_t(a) = 0$. Then:

$$\begin{aligned}
p_t(b) &= p_t(a) + \Delta(a \to b, t) = 0 + 2\delta = 2\delta, \\
p_t(c) &= p_t(b) + \Delta(b \to c, t) = 2\delta + \delta = 3\delta, \\
p_t(d) &= p_t(c) + \Delta(c \to d, t) = 3\delta + (-3\delta) = 0.
\end{aligned}$$

We verify that $p_t(d) - p_t(a) = 0 - 0 = 0 = \Delta(d \to a, t)$, confirming the cycle closes. For the shortcut edge $(a \to c)$, we check: $p_t(c) - p_t(a) = 3\delta - 0 = 3\delta = \Delta(a \to c, t)$, which is consistent. The potential is unique up to an additive constant: if we had chosen $p_t(a) = k$ instead of 0, all values would shift by k, but the edge differences would remain unchanged.

I. Minimal schedule period 2^d

Having established atomic single-event updates (Theorem T2), quantization (Proposition T8), and the cost function (Theorem T5), we now examine combinatorial constraints that link a discrete carrier (modeled here by Q_d) to discrete time. In particular, we seek lower bounds on the period required to visit all spatial positions under atomic updates.

For the purposes of this section, we treat d as an abstract dimension parameter indexing the hypercube family Q_d. A separate (conditional) discussion of selecting $d = 3$ is given later.

Thus, the fundamental structure is the d-dimensional hypercube Q_d, which at $d = 3$ (denoted Q_3) provides the minimal cell for ledger-compatible dynamics. The hypercube combinatorics are:

Object	Formula	$d = 3$
Vertices	2^d	8
Edges	$d \cdot 2^{d-1}$	12
Faces	$2d$	6

TABLE II. Combinatorics of the d-cube at $d = 3$. The Q_3 hypercube has 8 vertices, 12 edges, and 6 faces.

1. Ledger-Compatible Walk Constraints

Atomic single-event updates impose strict constraints on how recognition events can be scheduled across the spatial network. To characterize the minimal period required to visit all spatial positions, we introduce the concept of a *ledger-compatible walk*: a temporal sequence of recognition events satisfying atomicity and spatial completeness.

Under Theorem T2, each tick processes at most one recognition event. In the scheduler model below, we represent the system by a sequence of *active vertices* $(v_t)_{t=0}^{T-1}$ in Q_d such that each tick advances along one edge (so v_{t+1} is adjacent to v_t). Concretely, we take the single event at tick t to be the directed edge traversal

$$e_t := (v_t \to v_{t+1}),$$

so there is one edge-event per tick, while v_t is the canonical vertex label for tick t. This aligns the Gray-code "8-tick" walker picture.

1. **Atomicity:** At most one event per tick; in this scheduler model we traverse one edge per tick (no concurrent traversals).

2. **Spatial Completeness:** All vertices of Q_d appear at least once among the active-vertex labels (v_t) over one period.

3. **Timestamp Uniqueness:** Over one period, the active-vertex labels are all distinct: $v_t \neq v_{t'}$ for $t \neq t'$.

These constraints ensure that the ledger update is both *atomic* (no concurrency) and *complete* (all spatial positions are visited), while maintaining temporal ordering.

As a straightforward consequence, we have the next:

Theorem 29 (T6: Minimal period 2^d (eight ticks for $d = 3$)) *Let C be the vertex set of a d-dimensional hypercube Q_d, with $|C| = 2^d$, and let T be the scheduler period for a ledger-compatible walk.*

1. ***(Sufficiency)*** *If $T \geq 2^d$, then there exists a cyclic sequence of active vertices $(v_t)_{t=0}^{T-1}$ that is spatially complete and timestamp-unique (each vertex appears exactly once among the labels v_t), with v_{t+1} adjacent to v_t for each t (one edge traversal per tick). For $d = 3$, the Gray code Hamiltonian cycle realizes this minimal period: $000 \to 001 \to 011 \to 010 \to 110 \to 111 \to 101 \to 100 \to 000$.*

2. ***(Necessity)*** *If $T < 2^d$, then T ticks are insufficient to assign a distinct active-vertex label to each of the 2^d vertices. By the pigeonhole principle, some vertex label must repeat, so the walk cannot be both spatially complete and timestamp-unique.*

Proof. **Sufficiency:** We prove that for any $d \geq 1$, there exists a Hamiltonian cycle on Q_d (a cycle visiting all 2^d vertices exactly once). The proof is by induction on d.

Base case ($d = 1$): Q_1 has 2 vertices connected by an edge, forming a trivial 2-cycle.

Inductive step: Assume Q_d has a Hamiltonian cycle $H_d = (v_0, v_1, \ldots, v_{2^d-1}, v_0)$ for some $d \geq 1$. The hypercube Q_{d+1} can be constructed as two copies of Q_d (call them Q_d^0 and Q_d^1) with corresponding vertices connected. Let H_d^0 and H_d^1 be the Hamiltonian cycles in each copy (isomorphic to H_d).

To construct a Hamiltonian cycle in Q_{d+1}, start at a vertex v_0^0 in Q_d^0, follow H_d^0 until reaching a vertex v_i^0 that is adjacent to some vertex v_j^1 in Q_d^1. Since every vertex in Q_d^0 is adjacent to exactly one vertex in Q_d^1 (its copy in the hypercube construction), such a pair exists. Traverse the edge (v_i^0, v_j^1), then follow H_d^1 from v_j^1 until reaching a vertex v_k^1 adjacent back to Q_d^0, traverse back to complete the cycle. This yields a Hamiltonian cycle in Q_{d+1}.

For $d = 3$, the Gray code provides an explicit construction: $000 \to 001 \to 011 \to 010 \to 110 \to 111 \to 101 \to 100 \to 000$. This is a Hamiltonian cycle visiting all $2^3 = 8$ vertices exactly once, with v_{t+1} adjacent to v_t for each t, satisfying all three constraints.

Necessity: By constraint (3) (timestamp uniqueness), each of the 2^d vertices must appear exactly once in the sequence $(v_t)_{t=0}^{T-1}$. Therefore, $T \geq 2^d$. If $T < 2^d$, then by the pigeonhole principle, some vertex label must repeat, violating timestamp uniqueness. Therefore, $T \geq 2^d$ is necessary. $\square$

Therefore, the minimal period compatible with Theorem T2 for a d-dimensional hypercube is exactly

$$T_{\min} = 2^d. \tag{9}$$

For $d = 3$, this yields the eight-tick period: $T_{\min} = 2^3 = 8$ (within the scheduler model above).

Theorem T6 establishes the minimal period for a ledger-compatible walk, but it does not address whether this period is sufficient to distinguish all possible patterns. This leads to a complementary result about coverage:

Theorem 30 (T7: Coverage Lower Bound) *Let Q_d be a d-dimensional hypercube with 2^d vertices, and let T be the period of a ledger-compatible walk in the scheduler model above (one active vertex per tick). If $T < 2^d$, then the walk cannot cover all 2^d vertices within one period without repetition.*

Proof. In the scheduler model above, each tick carries one active-vertex label $v_t \in Q_d$. Therefore a period-T schedule can label at most T distinct vertices. If $T < 2^d$, the schedule cannot cover all 2^d vertices within one period without repetition. □

Together, Theorems T6 and T7 show that $T = 2^d$ is both necessary and sufficient for the scheduler model stated above: $T \geq 2^d$ is sufficient via a Hamiltonian cycle (Gray code at $d = 3$), while $T < 2^d$ is insufficient for covering all vertices without repetition. For $d = 3$, this yields the eight-tick period $T = 8$.

J. Conditional dimension selection: $d = 3$

Warning: This subsection presents *conditional* arguments that select $d = 3$ under additional hypotheses beyond the core ledger framework. These arguments are not part of the main derivation chain (T1–T8) and should be read as exploratory extensions requiring explicit additional assumptions.

We now present a conditional argument selecting $d = 3$ under additional hypotheses beyond the scheduling model. Specifically, we combine (i) the 2^d-tick counting structure (from T6), (ii) a "gap-45" synchronization criterion (an additional modeling hypothesis), and (iii) a linking-based distinguishability requirement (a topological constraint). Each of these, if adopted, independently selects $d = 3$.

Theorem 31 (Conditional dimensional rigidity) *Let $d \in \mathbb{N}$. Under the following* additional hypotheses *(beyond the core framework):*

1. ***(Gap-45 synchronization hypothesis)*** *The ledger period 2^d and a reference period of 45 ticks must synchronize with a common period of 360 ticks:* $\operatorname{lcm}(2^d, 45) = 360$.

2. ***(Linking requirement hypothesis)*** *Dimensions $d < 3$ are excluded because nontrivial topological linking (required for certain distinguishability properties) is only possible in $d \geq 3$.*

Then $d = 3$ is uniquely determined.

Proof. From hypothesis (1): $\operatorname{lcm}(2^d, 45) = 360 = 8 \cdot 45$. Since $360 = 2^3 \cdot 3^2 \cdot 5$ and $45 = 3^2 \cdot 5$, we have $\operatorname{lcm}(2^d, 45) = 2^{\max(d,3)} \cdot 3^2 \cdot 5 = 360$. This requires $\max(d, 3) = 3$, hence $d \leq 3$.

From hypothesis (2): $d \geq 3$ (as argued in the linking discussion below).

Therefore, $d = 3$ is the unique solution. □

Remark 32 (On the gap-45 hypothesis) *The gap-45 synchronization criterion is an* additional modeling hypothesis *motivated by considerations outside the scope of this paper (e.g., connections to angular periodicity or golden-angle structures). It is not derived from the cost axioms or ledger structure. Readers who do not adopt this hypothesis should treat the $d = 3$ selection as conditional on the linking argument alone, or on other independent constraints.*

1. *The Linking Argument*

The deeper reason $d = 3$ is special involves *topological linking*:

- $d = 2$: In the plane, "linking" of two closed curves is not a nontrivial topological invariant in the same way as in three dimensions (curves can be separated in $\mathbb{R}^2$).

- $d = 3$: In three dimensions, disjoint closed curves can be linked; the linking number is a topological invariant (e.g. via the Hopf link).

- $d \geq 4$: For embeddings of 1-dimensional loops, the classical three-dimensional notion of linking is not stable in the same way; additional ambient dimension generically permits unlinking moves that are forbidden in $d = 3$.

TABLE III. Three arguments selecting $d = 3$ (conditional)

Argument	**Constraint**	**Selects**
Linking	Non-trivial knot theory	$d = 3$
2^d-tick	$2^d = 8$ (hypercube counting)	$d = 3$
Gap-45	$\mathrm{lcm}(2^d, 45) = 360$	$d = 3$

Therefore, $d = 3$ is the *unique* dimension with non-trivial linking of closed curves in the classical sense. Interpreting "distinguishability via linking" as a model requirement yields a $d = 3$ selection under that hypothesis.

Remark 33 (Linking penalty as an additional modeling hypothesis) *The cost and ledger derivations in this manuscript do not, by themselves, assign a numerical "cost of linking" to linked configurations, nor do they identify a minimum crossing cost. Incorporating a* linking penalty *therefore requires an additional hypothesis specifying how topological linking is mapped into the ledger/cost formalism. A natural dimensionless scale one might use is the log-scale reference $J_{bit} = \ln\phi$ introduced above, but this identification is not derived here.*

2. *Summary: Three Independent Arguments for $d = 3$*

The conditional $d = 3$ selection rests on three independent constraint routes, each sufficient on its own:

The convergence of these constraints motivates the following conditional uniqueness statement.

Theorem 34 (Uniqueness of RS-compatible dimension (conditional)) *There exists a unique $d \in \mathbb{N}$ such that RSCompatibleDimension(d). That dimension is $d = 3$.*

IV. CONCLUSION

This paper develops the cost-first, ledger-based foundations of Recognition Science (RS). The foundation is not a metaphysical decree but a constraint on coherent comparison: the d'Alembert functional equation. The canonical reciprocal cost functional $J(x) = \frac{1}{2}(x +$

$x^{-1}) - 1$ is *uniquely forced* by the d'Alembert composition law together with normalization and quadratic calibration (Theorem T5; see the companion paper for a self-contained proof). From this keystone theorem, the remaining structures follow under the stated structural assumptions:

- The **Law of Existence**: a configuration exists iff $J(x) = 0$, which holds uniquely at $x = 1$.
- The **Meta-Principle** (T1): "Nothing cannot recognize itself" is *derived*—it reflects the mathematical fact that $J(0^+) \to \infty$.
- The **ledger structure**: J-symmetry ($J(x) = J(x^{-1})$) motivates a reversible event representation and is compatible with a balanced (double-entry) bookkeeping model under the stated ledger assumptions.
- **Discrete time** (T2), **quantized units** (T8), **cycle flux conservation** (T3), **scalar potentials** (T4), and the **minimal period constraint** (T6–T7) follow from the cost function and structural assumptions.
- The **golden ratio** ϕ: introduced here as the unique positive fixed point of the simple self-similar update $x \mapsto 1 + 1/x$, and used to define the log-scale reference $J_{\text{bit}} = \ln \phi$. Any stronger claim that ϕ is *forced* requires additional explicit self-similarity hypotheses beyond the cost axioms.
- **Three-dimensional space** ($d = 3$): a conditional selection argument is given under additional synchronization/linking hypotheses (Section III J).

All discrete statements in this manuscript are intended to be supported by machine-checked proofs implemented in Lean 4, subject to the stated assumptions and the scope limitations noted in Section I A. The present work focuses on the forcing chain up through discrete ledger consequences (T1–T8) and their formal verification; quantitative connections to particular physical models or constants, when desired, require additional bridging assumptions and are outside the scope of this paper.

---

[1] J. Aczél, *Lectures on Functional Equations and Their Applications*. Academic Press, 1966.

[2] M. Kuczma, *An Introduction to the Theory of Functional Equations and Inequalities*, 2nd ed. Birkhäuser, 2009.

[3] L. de Moura and contributors, *Lean 4 Theorem Prover*. https://leanprover.github.io/.

[4] The mathlib community, *Mathlib: the Lean mathematical library*. https://github.com/leanprover-community/mathlib4.

[5] J. Washburn and M. Zlatanović, *Uniqueness of the Canonical Reciprocal Cost*. Preprint, 2026.

Appendix A: Proofs of Core Theorems

All theorems T1–T8 have corresponding Lean 4 statements in the `IndisputableMonolith` repository, with machine-checkable proofs for the core forcing chain. We will provide a public archive with version control, build instructions, and a fixed commit hash corresponding to the version referenced by this manuscript.

Note on reproducibility: For proper scientific reproducibility, the archive will include: (i) a public repository URL (e.g., GitHub/GitLab) with version control, (ii) a specific commit hash corresponding to the verification status reported here, (iii) build instructions and dependency specifications (e.g., `lean-toolchain` and `lakefile.lean`), (iv) continuous integration (CI) configuration to verify builds, and (v) documentation for reproducing the verification.

1. Logic decomposition flow

For a high-level dependency summary (A1–A3 $\rightarrow$ T5 $\rightarrow$ T1 $\rightarrow$ T2–T8), see Figure 1 in the main text. This appendix focuses on pinpointing where each theorem is formalized in Lean.

2. Proof of Proposition T8: Ledger Units

Proof. **Lean Reference:** `LedgerUnits.equiv_delta_one`, `LedgerUnits.quantization`
Status: Proved
Strategy: Conservation tracking on discrete postings requires integer δ increments; no torsion forces unique representation. □

3. Proof of Theorem T4: Potential Uniqueness

Proof. **Lean Reference:** `Potential.unique_on_component`

Status: Proved

Strategy: Discrete exactness: closed 1-forms are exact on each connected component; the potential is unique up to an additive constant. □

4. Proof of Theorem T6: Minimal period 2^d (eight ticks for $d = 3$)

Proof. **Lean Reference:** `EightTick.minimal_and_exists`

Status: Proved (100% complete)

Certificate: `EightBeatCert`, `EightBeatHypercubeCert`, `GrayCodeCycleCert`. □

5. Proof of Theorem T7: Coverage Lower Bound

Proof. **Lean Reference:** `T7_nyquist_obstruction`, `T7_threshold_bijection`

Status: Proved □

6. Lean Repository Information

All proofs for theorems T1–T8 are verified in Lean 4 within the `IndisputableMonolith` repository. The verification follows the *cost-first foundation* described in this paper.

Repository structure (key modules):

- `IndisputableMonolith/Foundation/` — Cost-first foundation
  - `CostAxioms.lean` — Primitive axioms A1–A3
  - `LawOfExistence.lean` — Existence predicate, $J(0^+) \to \infty$
  - `LedgerForcing.lean` — Double-entry ledger structure (under explicit ledger assumptions)
  - `PhiForcing.lean` — Self-similarity and ϕ fixed-point analysis
  - `InevitabilityStructure.lean` — The forcing chain
  - `DerivationNarrative.lean` — Complete derivation chain

- `IndisputableMonolith/Cost/` — Cost functional uniqueness
  - `FunctionalEquation.lean` — d'Alembert equation analysis
- `IndisputableMonolith/CostUniqueness.lean` — T5 uniqueness proof
- `IndisputableMonolith/Verification/Tier1Cert.lean` — T1–T8 certificate bundle

Key Lean theorems for the cost-first foundation:

- **T5 (Keystone):** `CostUniqueness.T5_uniqueness_complete`
- **Law of Existence:** `Foundation.LawOfExistence.defect_zero_iff_one`
- **T1 (MP derived):** `Foundation.LawOfExistence.nothing_cannot_exist`
- ϕ **unique:** `PhiSupport.phi_unique_pos_root`
- **T6–T7:** `Patterns.cover_exact_pow`, `Patterns.min_ticks_cover`
- $d = 3$: `Verification.Dimension.onlyD3_satisfies_RSCounting_Gap45_Absolute`

Repository specifications:

- **Lean Version:** 4.3+
- **Mathlib Version:** Latest Mathlib4
- **Build:** `lake build` from repository root
- **Verification Status:** Core theorems proven, some scaffolds remain

All proofs produce executable certificates that can be verified independently. The cost-first foundation is fully formalized, with the Meta-Principle derived as a theorem rather than assumed as an axiom.