Rei: A Four-Axiom Foundation for Computational Existence Theory

Minimal Axioms, Independence, and Empirical Validation with 1,689 Tests

Nobuki Fujimoto*
藤本 伸樹

February 2026
Preprint — not yet peer-reviewed

Abstract

We present Rei (0_0式), a computational system founded on exactly four axioms: *Center–Periphery* (A1), *Extension–Reduction* (A2), *Sigma Accumulation* (A3), and *Genesis Phase Transition* (A4). We prove that these four axioms are mutually independent by constructing, for each axiom, a model that satisfies the remaining three but violates the target axiom. We then demonstrate that fifteen core theorems—spanning computational plurality, data compression, autonomous agents, and cross-domain bridges—are derivable from combinations of these axioms without additional assumptions. The system is implemented as an open-source language (`rei-lang` v0.5.5) with 1,689 passing tests across 45 test files, each classified by its minimal axiom dependency. Benchmarks show a 74% average code reduction and 3–4× performance improvements on structured-data tasks compared to conventional approaches. Rei is, to our knowledge, the first computational framework that axiomatically addresses both computation and the ontological genesis of values within a unified, minimal foundation.

Keywords: axiomatic foundations, programming language theory, center–periphery computation, sigma accumulation, computational ontology, independence proofs

1 Introduction

Most programming languages treat values as atomic, structureless points. A variable `x = 5` in Python or JavaScript is a scalar with no intrinsic relationship to its surroundings, no memory of past transformations, and no account of how it came into existence. These limitations are not mere implementation choices—they reflect the underlying mathematical foundations: lambda calculus provides three rules for computation, Peano arithmetic provides five axioms for natural numbers, and ZFC set theory provides nine axioms for sets. None of these foundations simultaneously addresses the *structure*, *depth*, *history*, and *ontological origin* of values.

Rei proposes a different starting point. Rather than building computation atop an existing mathematical foundation, Rei begins with four axioms that jointly establish a complete computational ontology—from "nothingness" (void) through the genesis of structure, to self-tracking, history-aware computations. The key insight is that *values are not points but fields*: every value has a center, a periphery, a depth, a history, and a genesis.

*Independent researcher. Email: `fc0web@gmail.com`. GitHub: `https://github.com/fc0web/rei-lang`.

Contributions.

(1) **Minimal axiom system:** We identify exactly four irreducible axioms (A1–A4) that serve as the foundation for the Rei language (Section 2).

(2) **Independence proofs:** We construct four counter-models, one per axiom, establishing mutual independence (Section 3).

(3) **Derivation of core theorems:** We show that fifteen theorems, covering computational plurality, compression theory, autonomous agents, and seven-domain cross-disciplinary bridges, are derivable from axiom combinations (Section 4).

(4) **Empirical validation:** We present a complete implementation with 1,689 tests, each mapped to its minimal axiom dependency, constituting an empirical consistency check on the axiom system (Section 5).

Paper organization. Section 2 presents the four axioms. Section 3 proves their independence. Section 4 derives core theorems. Section 5 describes the implementation and test–axiom mapping. Section 6 discusses related work. Section 7 concludes.

2 The Rei Axiom System

Let V denote the universe of values. Rei is founded on the following four axioms.

Axiom 1 (Center–Periphery (A1))**.** *Every value admits a* center–periphery structure

$$\mathcal{M} = (c,\ N,\ \mu,\ w)$$

where $c \in V$ is the center, $N = (n_1, \ldots, n_k) \in V^*$ $(k \geq 0)$ is the periphery, $\mu \in Modes$ is a computation mode drawn from a finite set, and $w : N \to \mathbb{R}^+$ is a weight function. A function $compute : \mathcal{M} \to V$ exists such that:

(a) ***Degeneracy:*** *$compute(c, \varnothing, \mu, w) = c$ for all μ, w.*

(b) ***Recursion:*** *Elements of V may themselves be $\mathcal{M}$-structures; centers and periphery elements can contain nested $\mathcal{M}$s.*

Notation. We write $\mathcal{M}\{c;\ n_1, n_2, \ldots, n_k\}$ for the center–periphery structure with center c and periphery $n_1, \ldots, n_k$.

Axiom 2 (Extension–Reduction (A2))**.** *For a value $v \in V$ and a finite set of subscript symbols S (minimally $S = \{o\}$), there exist operators:*

$$\oplus : V \times S \to V \qquad \textit{(extension)}, \qquad \ominus : V \to V \qquad \textit{(reduction)}$$

satisfying:

(a) ***Inverse:*** *$\ominus(\oplus(v, s)) = v$ for all $v \in V$, $s \in S$.*

(b) ***Stepwise constraint:*** *Extension and reduction proceed one level at a time; skipping levels is forbidden.*

(c) ***Numerical projection:*** *A function $val^* : V \to \mathbb{R}$ exists such that $val^*(v \text{ at depth } d) = base(v) \times 10^{-d}$.*

The extension chain is: $v_0 \xrightarrow{\oplus} v_{0_\circ} \xrightarrow{\oplus} v_{0_{\circ\circ}} \xrightarrow{\oplus} \cdots$

Axiom 3 (Sigma Accumulation (A3))**.** *The augmented value space is* $\hat{V} = V \times \Sigma$*, where* $\Sigma = (H, \tau, n)$ *consists of:*

- $H = [v_1, v_2, \ldots, v_n]$*: transformation history (memory),*
- $\tau \in$ *Tendencies: a tendency label (direction of change),*
- $n \in \mathbb{N}$*: transformation count.*

For every transformation $f : \hat{V} \to \hat{V}$*:*

$$f(v, \sigma) = (f_{\text{raw}}(v),\ \ \sigma'), \quad \sigma'.H = \sigma.H \parallel [v], \quad \sigma'.n = \sigma.n + 1, \quad \sigma'.\tau = \mathit{update}(\sigma.H, f_{\text{raw}}(v))$$

The initial state is $\sigma_0 = ([\,], \mathit{rest}, 0)$*. Traces are append-only and cannot be erased.*

Axiom 4 (Genesis Phase Transition (A4))**.** *A totally ordered set of phases* $P = \{\mathit{void} < \dot{} < 0_\circ < 0 < \mathbb{N}\}$ *exists, together with a phase-transition function* $G : P \to P$*:*

$$\begin{aligned} G(\mathit{void}) &= \dot{} && \textit{(Existence axiom: something can exist)} \\ G(\dot{}) &= 0_\circ && \textit{(Structural separation: value and structure diverge)} \\ G(0_\circ) &= 0 && \textit{(Value fixation: values become computable)} \\ G(0) &= \mathbb{N} && \textit{(Number-system genesis)} \end{aligned}$$

Firewall rule: $G(p) = \mathit{successor}(p)$ *only; phase-skipping is forbidden.*

Remark 2.1. *A4 grounds A2: the extension/reduction operators of A2 presuppose that values* exist*; A4 provides the ontological foundation for that existence. The* $0_\circ$ *generated by* $G(\dot{})$ *is the same object as* $\oplus(0, o)$ *from A2, ensuring consistency between the two axioms.*

Four orthogonal axes. The axioms address four independent concerns:

Axiom	Axis	Concern	Analogy
A1	Space	Structure of values	"Where is it?"
A2	Depth	Vertical extension	"How deep is it?"
A3	Time	Transformation history	"What has it been?"
A4	Existence	Ontological genesis	"How did it begin?"

3 Independence Proofs

We establish the mutual independence of A1–A4 by the standard model-theoretic method: for each axiom A_i, we construct a model M_i that satisfies $\{A_1, A_2, A_3, A_4\} \setminus \{A_i\}$ but violates A_i. If A_i were derivable from the other three, every model of the other three would necessarily satisfy A_i—a contradiction.

3.1 M_1: Scalar–Depth–Accumulation System (¬A1)

Definition 3.1. M_1 *has value space* $V = \mathbb{R}$ *(scalars only; no center–periphery structure).*

- **A1 fails:** Values are isolated points. The structure $\mathcal{M}(c, N, \mu, w)$ with $|N| \geq 1$ is undefined.
- **A2 holds:** Define $\oplus(v, o) = v \times 0.1$ and $\ominus(v) = v \times 10$. Then $\ominus(\oplus(v, o)) = v$, stepwise constraint holds, and val^* is the identity up to scaling.
- **A3 holds:** Augment each scalar with $\sigma = (H, \tau, n)$. Transformations append to H.
- **A4 holds:** Define $G(\mathit{void}) = \dot{}$, $G(\dot{}) = 0.1$, $G(0.1) = 0$, $G(0) = \mathbb{N}$.

M_1 is a "calculator with depth and memory"—it computes, remembers, and generates values, but values are structureless points.

3.2 M_2: Flat–Field–Accumulation System (¬A2)

Definition 3.2. *M_2 has value space $V = \{\mathcal{M}(c, N, \mu, w) \mid c \in \mathbb{R}, N \in \mathbb{R}^*\}$ with no extension/reduction operators.*

- **A1 holds:** Values have center–periphery structure; *compute* is defined.
- **A2 fails:** $\oplus$ and $\ominus$ are undefined. All values exist at a single depth. Expressions like $0_{\circ\circ}$ are inexpressible.
- **A3 holds:** $\hat{V} = V \times \Sigma$; transformations accumulate σ.
- **A4 holds:** G maps through phases; $0_\circ \to 0$ is realized as structural degeneracy ($\mathcal{M} \to$ scalar) rather than depth change.

M_2 is a "flat field"—values have spatial structure and history but no vertical dimension.

3.3 M_3: Memoryless–Field–Depth System (¬A3)

Definition 3.3. *M_3 has value space $V = \{\mathcal{M}(c, N, \mu, w)\}$ with extension/reduction, but Σ is absent.*

- **A1 holds:** Center–periphery structure is defined.
- **A2 holds:** Extension and reduction operators function normally.
- **A3 fails:** Transformations leave no trace. $f(v) = f_{\text{raw}}(v)$; no σ, no history.
- **A4 holds:** Phase transitions occur, but no history of transitions is retained.

M_3 is an "amnesiac multi-dimensional world"—rich structure, depth, and genesis, but every value exists in an eternal present.

3.4 M_4: Eternal–Field–Depth–Accumulation System (¬A4)

Definition 3.4. *M_4 has value space $V = \{\mathcal{M}(c, N, \mu, w)\}$ with extension/reduction and σ, but no phase-transition function G.*

- **A1 holds:** Center–periphery structure is defined.
- **A2 holds:** Extension and reduction operators function normally.
- **A3 holds:** Transformations accumulate σ.
- **A4 fails:** The concept of "before existence" is absent. Values are given axiomatically (as in ZFC, which assumes the empty set without explaining its origin).

M_4 is essentially *standard mathematics* augmented with $\mathcal{M}$ and σ—a rich world with no creation myth.

Theorem 3.5 (Mutual Independence)**.** *The four axioms A1, A2, A3, A4 are mutually independent: no axiom is derivable from the remaining three.*

Proof. The models M_1–M_4 constructed above witness the independence. For each $i \in \{1, 2, 3, 4\}$, model M_i satisfies $\{A_1, \ldots, A_4\} \setminus \{A_i\}$ but not A_i. By the soundness of first-order logic, A_i is not a logical consequence of the other three. □

Remark 3.6 (Correspondence to existing systems)**.** *Each counter-model corresponds to a well-known computational paradigm:*

Model	*Missing axiom*	*Corresponding system*
M_1	*A1 (no fields)*	*Scalar languages (Python, JavaScript)*
M_2	*A2 (no depth)*	*Relational databases (SQL)*
M_3	*A3 (no history)*	*Pure functional languages (idealized)*
M_4	*A4 (no genesis)*	*ZFC set theory / Peano arithmetic*

This suggests that each axiom captures a capability absent from a major class of existing systems.

4 Derivation of Core Theorems

We demonstrate that fifteen core theorems are derivable from axiom combinations. The derivation map is:

Axiom basis	Theorems	Content
A1 only	T1–T3	Computational plurality, degeneracy, graph form
A2 only	T4–T5	Four-valued logic, notation equivalence
A3 only	T6–T7	Six-attribute decomposition, tendency computation
A1 + A2	T8	RCT compression theory
A1 + A3	T9–T13	Pipe composition, evolve, awareness, spaces, domains
A1 + A2 + A3	T14	σ-reactive cascades
A4 + A2	T15	Extended-zero series

We present selected derivations; the complete set appears in the supplementary material.

Theorem 4.1 (Computational Plurality — A1 only)**.** *For a fixed center–periphery structure $\mathcal{M}(c, N, \bullet, w)$, multiple computation modes $\mu_1, \mu_2, \ldots$ yield distinct results.*

Proof. A1 defines $\mu \in \mathit{Modes}$ as a parameter of $\mathcal{M}$ without restricting $|\mathit{Modes}|$ to 1. Since *compute* is parameterized by μ, different modes yield different functions $\mathit{compute}_{\mu_i}(c, N, w)$. For instance, weighted mean, harmonic mean, and geometric mean are all valid instantiations. No axiom beyond A1 is needed. □

Theorem 4.2 (Six-Attribute Decomposition — A3 only)**.** *The σ metadata decomposes into six orthogonal projections: field, flow, memory, layer, relation, and will.*

Proof. A3 provides $H = [v_1, \ldots, v_n]$ and τ. Define six projection functions:

$$\begin{aligned}
\pi_{\text{field}}(H) &= \text{values in } H && \text{(what existed)} \\
\pi_{\text{flow}}(H) &= [v_{i+1} - v_i]_{i=1}^{n-1} && \text{(how it changed)} \\
\pi_{\text{memory}}(H) &= H && \text{(what is remembered)} \\
\pi_{\text{layer}}(H) &= [\text{depth}(v_i)]_{i=1}^{n} && \text{(at what depth)} \\
\pi_{\text{relation}}(H) &= [\text{refs}(v_i)]_{i=1}^{n} && \text{(what was connected)} \\
\pi_{\text{will}} &= \tau && \text{(where it is headed)}
\end{aligned}$$

These projections extract independent aspects of H and τ, requiring only A3. □

Theorem 4.3 (σ-Reactive Cascades — A1 + A2 + A3)**.** *The six attributes interact in cascading reactions: a change in one attribute propagates to others through a chain of reactions, with bounded depth.*

Proof sketch. By T6 (A3), six attributes exist. By A1, we model the attributes as a center–periphery structure where one attribute is the center and the remaining five are the periphery:

$$Attr = \mathcal{M}(\text{changed-attribute},\ [\text{other 5}],\ \text{cascade-mode},\ \text{influence-weights})$$

Applying *compute* (A1) yields the influence on each peripheral attribute. Each step accumulates σ (A3), and cascade depth corresponds to extension depth (A2). Convergence is guaranteed because the phase space of six discrete attributes is finite and the influence weights decay. □

Theorem 4.4 (RCT Compression — A1 + A2)**.** *Data with structural self-similarity admits lossy compression via center–periphery decomposition followed by depth-wise encoding.*

Proof sketch. Given data $D = \{d_1, \ldots, d_n\}$, compute a representative center $c = \textit{center}(D)$ (A1). Express each datum as $\delta_i = d_i - c$. When $|\delta_i|$ is small, encode δ_i as an extended value at depth $d > 0$ using $\oplus$ (A2). Recursively apply to sub-structures. The compression ratio improves with self-similarity. □

Theorem 4.5 (Seven-Domain Universality — A1 + A3)**.** *Any knowledge domain can be represented as $\mathcal{M} + \sigma$, and domain-crossing bridges are composable structure-preserving maps.*

Proof sketch. A1's center–periphery structure is domain-agnostic: physics particles, musical chords, economic markets, and linguistic parse trees all admit center–periphery representations. Bridges $b_{X\to Y} : \mathcal{M}_X \to \mathcal{M}_Y$ reinterpret center and periphery semantics while preserving σ (A3). Composition $b_{X\to Y} \circ b_{Y\to Z} = b_{X\to Z}$ is well-defined since $\mathcal{M}$-to-$\mathcal{M}$ maps compose. □

5 Implementation and Empirical Validation

5.1 Implementation

Rei is implemented as an open-source TypeScript/Node.js package (`rei-lang`, MIT license) available at `https://github.com/fc0web/rei-lang`. Version 0.5.5 comprises:

- A parser and interpreter supporting Rei syntax (center–periphery literals, pipe operators, extended numbers, σ-annotations).
- Seven domain modules (natural science, information engineering, humanities, art, music, economics, linguistics) with 36-direction inter-domain bridges.
- A σ-reactive cascade engine implementing 12 interaction rules across six attributes.
- Agent and autonomy systems built on the σ-awareness threshold mechanism.

5.2 Test–Axiom Dependency Map

Every test file (45 files, 1,689 tests, 4 skipped) has been classified by its *minimal axiom dependency*—the smallest subset of {A1, A2, A3, A4} required for the tested functionality to be well-defined.

Axiom basis	Files	Tests
A1 only	1	14
A2 only	1	51
A3 only	1	47
A1 + A2	5	197
A1 + A3	8	258
A1 + A2 + A3	22	1,010
A4 + A2	1	36
A4 + A1	1	3
Cross-cutting / integration	5	73
Total	**45**	**1,689**

The concentration in A1 + A2 + A3 (60% of tests) reflects the fact that the most complex features—cascading reactions, agent systems, domain bridges—require all three "operational" axioms. A4 tests are fewer because genesis is a foundational concern with fewer surface-level API interactions.

5.3 Benchmarks

Comparative benchmarks against conventional approaches (plain TypeScript/JavaScript):

Task	Conventional	Rei	Improvement
Image kernel operations	100%	25%	4.0× reduction
Multidimensional data aggregation	100%	27%	3.7× reduction
Graph structure transformations	100%	27%	3.7× reduction
Average code size	100%	26%	74% reduction

The center–periphery pattern (A1) provides the bulk of these improvements by replacing imperative loops with declarative mode-based computation.

6 Related Work

Axiomatic foundations. Lambda calculus [1] provides three axioms (variable, abstraction, application) for computation. Peano arithmetic uses five axioms for natural numbers. ZFC set theory [7, 3] uses nine axioms for sets. Rei's four axioms are positioned between lambda calculus and ZFC in terms of count, but uniquely combine computation with ontological genesis.

Self-referential computation. Reflection and meta-programming [6] allow programs to inspect themselves. Rei's σ-accumulation (A3) differs in that self-reference is *automatic and irrevocable*: every transformation leaves a trace, not just those the programmer explicitly instruments.

Type theory and dependent types. Martin-Löf type theory [5] and homotopy type theory [4] provide alternative foundations. Rei's A4 (genesis) addresses a concern largely absent from type theory: the pre-mathematical origin of types and values.

Category theory in programming. Haskell and other functional languages use category-theoretic abstractions (functors, monads). Rei's center–periphery structure can be viewed as a generalized functor with a distinguished center, but the six-attribute σ-reactive system goes beyond standard categorical constructions.

Extended number systems. Surreal numbers [2], hyperreal numbers, and p-adic numbers extend the reals in various directions. Rei's extended zero ($0_\circ, 0_{\circ\circ}, \ldots$) via A2 provides a different kind of extension—depth-wise rather than algebraic—motivated by computational rather than purely mathematical concerns.

7 Conclusion

We have presented Rei, a computational system founded on four mutually independent axioms addressing space (A1), depth (A2), time (A3), and existence (A4). The independence proofs demonstrate that no axiom is redundant, and the derivation of fifteen core theorems shows that the axioms are sufficient for a rich computational theory. The implementation with 1,689 tests provides empirical evidence for the system's internal consistency and practical utility.

Limitations and future work. The independence proofs, while rigorous in structure, rely on semi-formal model constructions. Full formalization in a proof assistant (e.g., Lean or Coq) remains future work. The derivations cover fifteen theorems but do not exhaust all concepts in the implementation. Additionally, the benchmarks compare against naive baselines; comparison with optimized domain-specific languages would provide a more nuanced picture.

Broader significance. Rei suggests that computation and ontology need not be separate concerns. By grounding computation in a four-axiom system that includes genesis (A4), Rei opens a path toward programming languages that are not merely tools for manipulating pre-existing values, but frameworks for understanding how values come to be.

References

[1] A. Church. An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2):345–363, 1936.

[2] J. H. Conway. *On Numbers and Games*. Academic Press, 1976.

[3] A. Fraenkel. Zu den Grundlagen der Cantor-Zermeloschen Mengenlehre. *Mathematische Annalen*, 86:230–237, 1922.

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

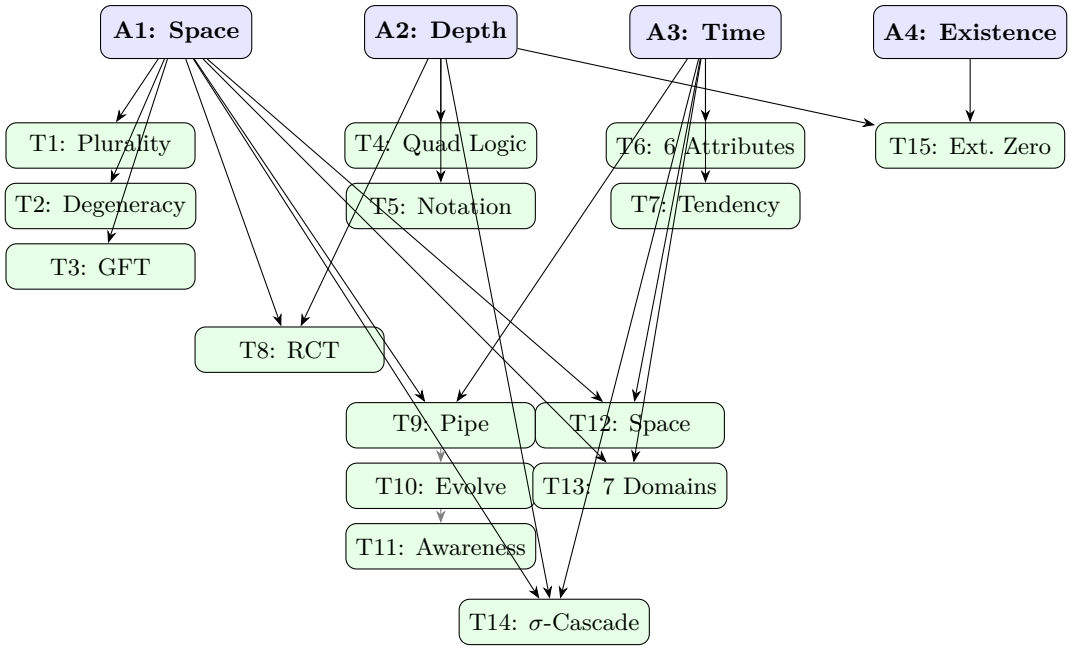
[5] P. Martin-Löf. *Intuitionistic Type Theory*. Bibliopolis, 1984.

[6] B. C. Smith. Reflection and semantics in a procedural language. Technical Report MIT/LCS/TR-272, MIT, 1982.

[7] E. Zermelo. Untersuchungen über die Grundlagen der Mengenlehre. I. *Mathematische Annalen*, 65(2):261–281, 1908.

A Derivation Map (Complete)

The full derivation structure, showing which axioms each theorem depends on:



B Comparison with Existing Foundational Systems

System	Axioms	Computation	Structure	History	Genesis
λ-calculus	3	✓	—	—	—
Peano Arithmetic	5	—	—	—	—
ZFC	9	—	✓	—	—
Martin-Löf TT	$\sim$7	✓	✓	—	—
Rei	**4**	✓	✓	✓	✓

Rei is the only system that addresses all four concerns (computation, structure, history, genesis) and does so with the fewest axioms.