THE ZERO PARADOX
ZP-M: Kleene–Ordinal Bridge

Version 1.0 | May 2026

v1.0: Initial release. All theorems §I–§IV proved sorry-free in Lean 4. Closes the hfp free hypothesis (ZP-L) and the snapEmbed type bridge. Axiom footprint: [propext, Classical.choice, Quot.sound] throughout.

ZP-M bridges the two prior layers — ZP-K (computational grounding via Kleene's second recursion theorem) and ZP-L (ordinal snap threshold at ε_0) — with two formal results. First, the free hypothesis hfp in snap_exactly_at_epsilon_zero (ZP-L §VI) is proved to follow from a minimal alignment hypothesis plus monotonicity: hfp is not an independent condition. Second, the canonical type bridge snapEmbed : MachinePhase → $\mathbb{Z}_2$ formalizes the structural triangle connecting the pre-snap state c_0 (Kleene quine, ZP-K), the ordinal snap at ε_0 (ZP-L), and the 2-adic limit 0 (ZP-B).

The closing result (§IV) records that both the Kleene recursion theorem and the ordinal fixed-point ε_0 = ω^ε_0 are instances of the same diagonalization pattern: a self-referential operation produces a forced fixed point. The two domains — computability theory (codes and Gödel numbers) and ordinal theory (ω-tower iteration) — use no shared mathematical machinery, but the structural schema is identical. Remark R-M.1 tracks what this means for DA-1 Path 2 and the boundary of the diagonalization frame.

Section I: The snapEmbed Morphism

snapEmbed is the canonical type bridge from ZP-E's machine-phase state space to ZP-B's 2-adic integers. The pre-snap state c_0 maps to 1 (a 2-adic unit); the snap state c_1 maps to 0 (the 2-adic limit of the tower encodings). Under multiplication on $\mathbb{Z}_2$, 0 is absorbing — mirroring exactly the join structure on MachinePhase, where c_1 is the absorbing element.

Within the ZP framework, $0 \in \mathbb{Z}_2$ plays the role of ⊥ — it is the 2-adic valuation limit and the fixed point of multiplication by 2. snapEmbed makes this modelling commitment concrete: c_1 maps to 0 because c_1 IS the object at infinite 2-adic depth. This is a modelling commitment, not a ring-theoretic theorem.

Definition: snapEmbed (ZPM.lean §I)
snapEmbed : MachinePhase → $\mathbb{Z}_2$
snapEmbed c_0 = 1 (pre-snap: 2-adic unit)
snapEmbed c_1 = 0 (snap state: 2-adic zero = 2-adic limit of tower)
The morphism property: join on MachinePhase (c_1 is absorbing under join) corresponds to multiplication on $\mathbb{Z}_2$ (0 is absorbing under ×). Both structures share the absorbing-element pattern — the bridge is structural, not incidental.

Theorem: snapEmbed_injective (ZPM.lean §I)

Function.Injective snapEmbed

c_0 and c_1 map to distinct 2-adic integers ($1 \neq 0$ in $\mathbb{Z}_2$).

Proof: case split on MachinePhase; simp_all closes each case.

Lean purity: [propext, Classical.choice, Quot.sound]. ✓

Theorem: snapEmbed_mul_morphism (ZPM.lean §I)

∀ a b : MachinePhase, snapEmbed (join a b) = snapEmbed a × snapEmbed b

snapEmbed sends the join operation to multiplication.

Under this map, the absorbing-element structure is preserved:

join c_1 x = c_1 for all x ↔ $0 \times y = 0$ for all $y \in \mathbb{Z}_2$.

Proof: case split; simp [snapEmbed].

Lean purity: [propext, Classical.choice, Quot.sound]. ✓

Lemma: snapEmbed_c0_val (ZPM.lean §I)

(snapEmbed c_0).valuation = 0

c_0 maps to a 2-adic unit: valuation 0, norm 1.

Proof: simp via PadicInt.valuation_one.

Lean purity: [propext, Classical.choice, Quot.sound]. ✓

Lemma: snapEmbed_c1_dvd (ZPM.lean §I)

∀ n : ℕ, $(2 : \mathbb{Z}_2)$^n | snapEmbed c_1

c_1 maps to 0, which is divisible by all powers of 2.

$0 \in \mathbb{Z}_2$ is divisible by 2^n for every n — infinite 2-adic depth.

This is the formal signature of ⊥ = {⊥}: infinite 2-adic valuation.

Proof: simp [snapEmbed_c1].

Lean purity: [propext, Classical.choice, Quot.sound]. ✓

Remark: snapEmbed vs the Identification Conjecture

snapEmbed establishes the morphism property (join ↦ ×) and the absorbing-element correspondence. It does not derive the identification $\varepsilon_0 \leftrightarrow \bot$ as a type-theoretic theorem: establishing that snap_exactly_at_epsilon_zero's threshold IS $0 \in \mathbb{Z}_2$ via an order-preserving embedding would require ZPSemilattice morphisms between Ordinal and MachinePhase, which are not defined in this library.

What the bridge achieves: c_1 (the snap state assigned at ε_0) maps to 0 (the 2-adic limit) under snapEmbed. The triangle is formally exhibited. The conjectured stronger bridge — Classical.choice forced by ZP metric collapse — is deferred to §V (future work).

Section II: Closing the hfp Gap

snap_exactly_at_epsilon_zero (ZP-L §VI) carries a free hypothesis hfp asserting that for any map ϕ : Ordinal → MachinePhase, every fixed point of ω^· maps to c_1. This hypothesis is required for the upper-bound direction of the minimality argument (ε_0 is the first snap point). The canonical witness in epsilon_zero_snap_canonical (ZP-L §VII) satisfies hfp directly by case analysis, but hfp is not derived from the structural axioms.

ZP-M §II closes this gap. Given monotonicity and $\phi\ \varepsilon_0 = c_1$ as a minimal alignment hypothesis, hfp follows. The argument: for any fixed point α of ω^·, epsilonZero_le_fixedPoint gives $\varepsilon_0 \le \alpha$; monotonicity then chains $\phi\ \varepsilon_0 \le \phi\ \alpha$; substituting c_1 and using c_1's absorbing property gives $\phi\ \alpha = c_1$.

Theorem: hfp_from_epsilon_zero (ZPM.lean §II)

For any ϕ : Ordinal → MachinePhase satisfying:

hmono: $\forall\ \alpha \le \beta$, join $(\phi\ \alpha)\ (\phi\ \beta) = \phi\ \beta$

$h\varepsilon_0$: ϕ epsilonZero $= c_1$

we have: $\forall\ \alpha,\ \omega^\wedge\alpha = \alpha \to \phi\ \alpha = c_1$.

Proof: epsilonZero_le_fixedPoint gives $\varepsilon_0 \le \alpha$; hmono gives join $c_1\ (\phi\ \alpha) = \phi\ \alpha$; join $c_1\ x = c_1$ for all x (absorbing property of c_1), so $c_1 = \phi\ \alpha$.

Lean purity: [propext, Classical.choice, Quot.sound]. ✓

Theorem: snap_unconditional (ZPM.lean §II)

For any ϕ satisfying hmono, h0 (tower stages ↦ c_0), and $h\varepsilon_0$ ($\phi\ \varepsilon_0 = c_1$):

$\phi\ \varepsilon_0 = c_1 \land \forall\ \alpha,\ \phi\ \alpha = c_1 \to \varepsilon_0 \le \alpha$.

ε_0 is the minimal snap threshold under minimal hypotheses.

Proof: snap_exactly_at_epsilon_zero with hfp supplied by hfp_from_epsilon_zero.

Lean purity: [propext, Classical.choice, Quot.sound]. ✓

What §II achieves: hfp is no longer a free hypothesis in the minimality theorem. Given $\phi\ \varepsilon_0 = c_1$ (the alignment condition: the map snaps at ε_0) plus monotonicity, the full minimality result follows from the ordinal structure alone. hfp is a derived consequence, not an additional commitment.

Section III: The Kleene–Ordinal Triangle

ZP-K established: $c_0 = \bot$ is a Quine atom (da1_closed_concrete : IsQuineAtom (⊥ : MachinePhase)). ZP-L established: the canonical snap map assigns c_1 exactly at ε_0, with tower encodings in $\mathbb{Z}_2$ converging to 0. The triangle connects all three objects formally:

Edge	Objects connected	Formal content
A ↔ C (ordinal snap)	$\varepsilon_0 \leftrightarrow c_1$	Canonical snap map: stages below $\varepsilon_0 \mapsto c_0$; $\varepsilon_0 \mapsto c_1$ (ZP-L §VII)
A ↔ B (2-adic convergence)	Tower $\to 0 \in \mathbb{Z}_2$	cnfToZp2(towerNONote n) $\to 0$ in $\mathbb{Z}_2$ (ZP-L §IV)
B ↔ C (type bridge)	$c_1 \leftrightarrow 0 \in \mathbb{Z}_2$	snapEmbed $c_1 = 0$ (§I above)

Theorem: snap_state_zp2_is_zero (ZPM.lean §III)
snapEmbed ((λ α : Ordinal, if $\alpha < \varepsilon_0$ then c_0 else c_1) ε_0) = 0
The canonical snap map assigns c_1 at ε_0; snapEmbed sends c_1 to 0.
The snap state at the ordinal threshold maps to the 2-adic zero.
Proof: simp (the if-condition evaluates to False at $\alpha = \varepsilon_0$).
Lean purity: [propext, Classical.choice, Quot.sound]. ✓

Theorem: zpm_triangle (ZPM.lean §III)
All three edges of the triangle, co-proved:
(A ↔ C) ∀ n : ℕ, fundamentalSeq n $< \varepsilon_0$
(A ↔ C) (λ α, if $\alpha < \varepsilon_0$ then c_0 else c_1) $\varepsilon_0 = c_1$
(A ↔ B) Filter.Tendsto (λ n, cnfToZp2 (towerNONote n)) Filter.atTop (nhds 0)
(B ↔ C) snapEmbed (λ α, if $\alpha < \varepsilon_0$ then c_0 else c_1) $\varepsilon_0 = 0$
Proof: ⟨epsilonZero_tower_lt, if_neg (lt_irrefl ε_0), tower_converges_to_zero, snap_state_zp2_is_zero⟩.
Lean purity: [propext, Classical.choice, Quot.sound]. ✓

Section IV: Shared Diagonalization Pattern

Both the Kleene recursion theorem (ZP-K) and the ordinal fixed-point ε_0 (ZP-L) arise from the same structural schema: a self-referential operation has a forced fixed point. In computability theory, the diagonal applies to codes and their Gödel numbers. In ordinal theory, the diagonal applies to the tower iteration $\alpha \mapsto \omega\hat{}\alpha$. The two domains share no mathematical machinery, but the diagonalization pattern is identical.

Domain	Self-referential operation	Forced fixed point
Computability (ZP-K)	eval c = selfApply c (c runs on its own Gödel number)	∃ c : Code, IsComputationalQuine c
Ordinal theory (ZP-L)	ω^α = α (α is a fixed point of ω-exponentiation)	ε_0 = nfp (ω^·) 0, the least such α

Theorem: both_fixed_points_exist (ZPM.lean §IV)

(∃ c : Code, ∀ n, eval c n = eval c (encode c + n)) ∧

(∃ α : Ordinal, ω^α = α ∧ ∀ β, ω^β = β → α ≤ β)

Both diagonalization patterns produce fixed points, co-proved in one formal context.

Left: Kleene quine (period = Gödel number of c), via computational_quine_exists.

Right: $\varepsilon_0 = \omega\hat{}\varepsilon_0$, minimal; via epsilonZero_fixedPoint and epsilonZero_le_fixedPoint.

The co-existence is the formal content: two forced fixed points, two domains, one theorem.

Lean purity: [propext, Classical.choice, Quot.sound]. ✓

Remark R-M.1: DA-1 Path 2 and the Limits of the Diagonalization Frame

both_fixed_points_exist establishes that Kleene diagonalization (a code on its own Gödel number) and ordinal diagonalization ($\varepsilon_0 = \omega\hat{}\varepsilon_0$, least fixed point) are the same structural pattern in two domains. Both are textbook instances of the diagonalization schema.

L-INF (ZP-C §II: ⊥ has divergent surprisal — unbounded information content, incompressible by any finite description) is structurally analogous. Whether it fits the same diagonalization schema formally is the open question.

The formally unconnected instance is L-INF. This gap is DA-1 Path 2 — the informational bridge. The reason Path 2 resisted formalization is now visible from this layer: L-INF is a measure-theoretic statement (Shannon entropy, probability distributions) while the Kleene quine is a computability-theoretic statement (partial recursive functions, Gödel encoding). The two frameworks share no mathematical machinery.

The diagonalization frame established here provides the common conceptual ground, but the formal bridge requires the Kolmogorov complexity connection: incompressibility is the concept that lives in both worlds simultaneously. That bridge is outside the current Lean scope, pending AIT infrastructure not yet in Mathlib.

Remark R-M.1: DA-1 Path 2 and the Limits of the Diagonalization Frame

DA-1 Path 2 was recharacterized in ZP-E/ZP-K as a foundational commitment rather than a missing proof. The diagonalization frame here makes the boundary precise: the gap is not a skipped proof step but a genuine framework separation — measure-theoretic surprisal on one side, computability-theoretic encoding on the other, with no shared machinery in current Mathlib.

Theorem Summary

Theorem	Section	Status
snapEmbed_injective	§I	Proved ✓
snapEmbed_mul_morphism	§I	Proved ✓
snapEmbed_c0_val	§I	Proved ✓
snapEmbed_c1_dvd	§I	Proved ✓
hfp_from_epsilon_zero	§II	Proved ✓
snap_unconditional	§II	Proved ✓
snap_state_zp2_is_zero	§III	Proved ✓
zpm_triangle	§III	Proved ✓
both_fixed_points_exist	§IV	Proved ✓

Axiom Purity

All theorems carry axiom footprint: [propext, Classical.choice, Quot.sound].

These are standard Mathlib infrastructure axioms, inherited from ordinal theory (ZP-L), 2-adic analysis (ZP-B), and computability theory (ZP-K).

Classical.choice is load-bearing in both the computability fixed-point (Kleene's theorem) and the ordinal fixed-point (nfp). Its presence is expected and documented.

Zero sorry in ZPM.lean. Verified: lake build, May 2026.

End of ZP-M v1.0 | Kleene–Ordinal Bridge | snapEmbed type bridge | hfp gap closed | zpm_triangle co-proved | both_fixed_points_exist | R-M.1: DA-1 Path 2 boundary | All ZPM.lean theorems verified. Axioms: [propext, Classical.choice, Quot.sound].