THE ZERO PARADOX

ZP-M: Kleene–Ordinal Bridge

Initial: May 2026 | Current: 1.21, November 2026

Synthesis bridge. All theorems §I–§IV proved sorry-free in Lean 4. Closes the hfp free hypothesis (ZP-L) and the snapEmbed type bridge. Axiom footprint: at most [propext, Classical.choice, Quot.sound], per declaration below.

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 (⊥ of MachinePhase; reading it as the Kleene quine is not an equation, since a Code and a MachinePhase state are different types, and that it runs is the occurrence commitment), the ordinal snap at ε_0 (ZP-L), and the 0 of $\mathbb{Z}_2$ (ZP-B), the limit of the tower's 2-adic images (ZP-L).

The closing result (§IV) proves two existence statements together: a code c with eval c n = eval c (encode c + n) for every n (a periodicity condition, proved via the recursion theorem, which constant codes also meet) and the least ordinal fixed point $\varepsilon_0 = \omega^{\varepsilon_0}$. It is a conjunction of two existence proofs over different types. The two domains — computability theory (codes and Gödel numbers) and ordinal theory (ω-tower iteration) — use no shared mathematical machinery; reading them as one pattern is the framework's interpretation, not what the theorem proves (Remark R-M.1). R-M.1 also tracks what this means for DA-1 Path 2.

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 limit of the tower's 2-adic images). Under multiplication on $\mathbb{Z}_2$, 0 is absorbing — mirroring exactly the join structure on MachinePhase, where c_1 is the absorbing element.

No equation between a MachinePhase state and a 2-adic integer is stated; they are different types. What c_1 and 0 share is a role, the die's Motion.SelfOnlyAt (ZeroParadox/Die.lean): a point that steps to itself and to nothing else; in each carrier its occupant is proved for the step chosen there. Two pairings, each through the map snapEmbed, stand in place of an identity (Incompleteness.lean §I-b). Under the join step a → join a z for some z, only c_1 occupies the role (machineJoinMotion_selfOnlyAt_iff); under the multiplication step x → x·w for some w on $\mathbb{Z}_2$, only 0 does (z2MulMotion_selfOnlyAt_iff); snapEmbed is a map of these two motions (snapEmbedHom). Its back clause fails at c_0 (1 steps to 2, and no state maps to 2), so the general transport of the role along such maps does not apply; at c_1 the role does carry over, because 0 has only itself as successor (snapEmbed_selfOnlyAt_c1, from Motion.selfOnlyAt_relHom). Under doubling on $\mathbb{Z}_2$, only 0 occupies the role (z2_double_selfOnlyAt_iff), and 0 is divisible by every power of 2 (snapEmbed_c1_dvd); but snapEmbed carries no motion with a step at c_0 into doubling steps (snapEmbed_doubling_noStepAt): the

snap $c_0 \to c_1$ goes to $1 \to 0$, which is not a doubling step.

Definition: snapEmbed (Incompleteness.lean §I)

snapEmbed : MachinePhase → $\mathbb{Z}_2$

snapEmbed $c_0 = 1$ (pre-snap: 2-adic unit)

snapEmbed $c_1 = 0$ (snap state: the 2-adic zero, the limit of the tower's 2-adic images)

The morphism property: join on MachinePhase (c_1 is absorbing under join) corresponds to multiplication on $\mathbb{Z}_2$ (0 is absorbing under ×). In the die this is a map of motions, from a → join a z for some z on MachinePhase to x → x·w for some w on $\mathbb{Z}_2$ (snapEmbedHom, §I-b).

Theorem: snapEmbed_injective (Incompleteness.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 (Incompleteness.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 × y = 0 for all y ∈ $\mathbb{Z}_2$.

Proof: case split; simp [snapEmbed].

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

Lemma: snapEmbed_c0_val (Incompleteness.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 (Incompleteness.lean §I)

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

Lemma: snapEmbed_c1_dvd (Incompleteness.lean §I)

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.

Proof: simp [snapEmbed_c1].

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

Results: snapEmbed in the die (Incompleteness.lean §I-b)

machineJoinMotion_selfOnlyAt_iff: under a → join a z for some z, y steps to itself and to nothing else iff $y = c_1$. Lean purity: [propext].

z2MulMotion_selfOnlyAt_iff: under x → x·w for some w on $\mathbb{Z}_2$, y steps to itself and to nothing else iff $y = 0$. Lean purity: [propext, Classical.choice, Quot.sound].

snapEmbedHom: snapEmbed is a strict map between these two motions (steps to steps, from snapEmbed_mul_morphism). An example beside it shows the back clause fails at c_0 (1 steps to 2, and no state maps to 2). Lean purity: [propext, Classical.choice, Quot.sound].

snapEmbed_selfOnlyAt_c1: the image of c_1 steps to itself only in the multiplication motion, carried along snapEmbedHom's steps by Motion.selfOnlyAt_relHom. Lean purity: [propext, Classical.choice, Quot.sound].

z2_double_selfOnlyAt_iff: under doubling on $\mathbb{Z}_2$, y steps to itself and to nothing else iff $y = 0$. Lean purity: [propext, Classical.choice, Quot.sound].

snapEmbed_doubling_noStepAt: if snapEmbed sends every step of a motion M on MachinePhase to a doubling step, c_0 has no step in M. Lean purity: [propext, Classical.choice, Quot.sound]. ✓

Remark: snapEmbed and the bottom role

snapEmbed establishes the morphism property (join ↦ ×) and the absorbing-element correspondence. Relative to the ordinals' floor, the bottom-role occupant 0 and the ε_0 occupant differ (epsilon0_ne_bot); against the 2-adic 0 an identity with ε_0 is not a well-formed statement (it fails to elaborate, Ordinal vs $\mathbb{Z}_2$), and the link runs through maps: snapEmbed sends c_1 to 0, the bottom-role occupant of the Scale.lean chart, whose join with $\bot = 0$ is chosen (ZeroParadox/Valuation/Scale.lean §V). The canonical threshold map Ordinal → MachinePhase is order-non-decreasing in ZP-A's join order (snap_map_mono), and no map Ordinal → MachinePhase is compatible with the CNF → $\mathbb{Z}_2$ map along the tower: for every g : MachinePhase → $\mathbb{Z}_2$ the square fails, because MachinePhase has two values and the images of stages 1, 2, 3 have three distinct 2-adic valuations (the example after c1_epsilon_zero_identification in ZeroParadox/Ordinal/Gentzen.lean); through snapEmbed it fails at every stage $n \geq 1$, because snapEmbed takes only the values 1 and 0, and stage $n \geq 1$ maps to 2^n, which is neither (ZeroParadox/Ordinal/Incompleteness.lean § II).

What the bridge achieves: c_1 (the snap state assigned at ε_0) maps to 0 of $\mathbb{Z}_2$, the limit of the tower's 2-adic images, under snapEmbed. The triangle is formally exhibited. Whether Classical.choice is forced by the ZP metric collapse is a separate open question (ZeroParadox/Ordinal/SyntacticCollapse.md).

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 (Incompleteness.lean §II)

For any ϕ : Ordinal → MachinePhase satisfying:

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

hε_0: ϕ epsilonZero = c_1

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

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 (Incompleteness.lean §II)

For any ϕ satisfying hmono, h0 (tower stages ↦ c_0), and hε_0 ($\phi\ \varepsilon_0 = c_1$):

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

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 and every tower stage sent to c_0 (h0), the full minimality result follows (snap_unconditional). 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 ($\bot$: MachinePhase)). ZP-L established: the canonical snap map assigns c_1 exactly at ε_0, with tower images 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 (Incompleteness.lean §III)
snapEmbed $((\lambda\ \alpha : \text{Ordinal}, \text{if } \alpha < \varepsilon_0 \text{ then } c_0 \text{ else } c_1)\ \varepsilon_0) = 0$
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 (Incompleteness.lean §III)
All three edges of the triangle, co-proved:
(A ↔ C) $\forall\ n : \mathbb{N}$, fundamentalSeq $n < \varepsilon_0$
(A ↔ C) $(\lambda\ \alpha, \text{if } \alpha < \varepsilon_0 \text{ then } c_0 \text{ else } c_1)\ \varepsilon_0 = c_1$
(A ↔ B) Filter.Tendsto (λ n, cnfToZp2 (towerNONote n)) Filter.atTop (nhds 0)
(B ↔ C) snapEmbed $((\lambda\ \alpha, \text{if } \alpha < \varepsilon_0 \text{ then } c_0 \text{ else } c_1)\ \varepsilon_0) = 0$
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: The Kleene and Ordinal Fixed Points

The Kleene recursion theorem (ZP-K) gives a code fixed up to evaluation, and ε_0 (ZP-L) is the least fixed point of ω^·; the two come from different constructions. In computability theory the construction is a diagonal argument over codes and their Gödel numbers. In ordinal theory ε_0 is reached by iterating the normal function $\alpha \mapsto \omega^\alpha$ from 0 and taking the supremum ($\varepsilon_0 =$ nfp (ω^·) 0, epsilonZero_eq_nfp in ZeroParadox/Ordinal/Gentzen.lean), the fixed-point lemma for normal functions rather than a diagonal argument. The two domains share no mathematical machinery. The theorem below records the ordinal fixed point and, on the computability side, the computability fixed point the recursion theorem yields: a code c with eval c = selfApply c (IsComputationalQuine), a predicate constant codes also satisfy, so it does not pin self-reference (Remark R-M.1).

Domain	Operation	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 (Incompleteness.lean §IV)

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

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

Two existence statements, co-proved in one formal context.

Left: a code periodic with period its own Gödel number, via computational_quine_exists; constant codes also satisfy it, so the left conjunct alone does not pin self-reference.

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

The co-existence is the formal content: two existence statements, 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 records, on one side, a code periodic with period its own Gödel number (obtained by Kleene diagonalization, though constant codes also meet the condition) and, on the other, the ordinal $\varepsilon_0 = \omega^{\varepsilon_0}$, the least fixed point and the supremum of the iterates of ω^· from 0. The statement is a conjunction of two existentials, so it establishes that both exist, not that they are one structural pattern; reading them as the same pattern is the framework's interpretation, and no equation between a Code and an Ordinal is well-formed. The Kleene fixed point comes from a diagonal argument (Lawvere 1969; Yanofsky 2003, Theorem 5, places the recursion theorem under that schema); ε_0 comes from the fixed-point lemma for normal functions, by iteration to a supremum.

The informational argument has two charts. Depth chart, ZP-C L-INF: surprisal is unbounded above in the ball-hierarchy depth, each value finite (l_inf). Point chart: at 0 of $\mathbb{Q}_2$, the one point in every ball of the chain centred at 0, the 2-adic additive valuation is ⊤ (padic_addVal_bot), the point to which c_0, ⊥ of MachinePhase, corresponds in ZP-C. As of 2026-11-01 no single declared map from MachinePhase into $\mathbb{Z}_2$ or $\mathbb{Q}_2$ sends c_0 to 0: searching the .lean files under ZeroParadox/ for both type names finds snapEmbed (ZeroParadox/Ordinal/Incompleteness.lean) and two negative examples (ZeroParadox/Ordinal/Gentzen.lean, ZeroParadox/Valuation/SemilatticeInstance.lean). A composite does: the order isomorphism from MachinePhase onto the two-point Pole (c_0 to zero), exhibited in an anonymous example in ZeroParadox/DieCannotBe.lean and read there as order shape only, followed by poleToSphere (ZeroParadox/Valuation/PoleCornersBridge.lean), which sends zero to 0 of $\mathbb{Q}_2$ in its one-point compactification; that composite is not declared. § I's snapEmbed is a different map, sending c_0 to 1 and c_1 to 0; the two are not composed. L-INF is structurally analogous to the Kleene fixed point. Whether it fits Kleene's diagonalization schema formally is the open question this remark tracks.

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

L-INF stays formally unconnected. It is a measure-theoretic statement (surprisal under a probability measure on ZP-B's binary space), while the Kleene quine is a computability-theoretic statement (partial recursive functions, Gödel encoding), and neither this remark nor both_fixed_points_exist links them. ZP-C uses the surprisal measure and Kolmogorov complexity as two separate arguments (ZP-C Remark R-BRIDGE).

DA-1 Path 2 is a different matter. In ZP-E (DA-1 insert § IV) it is the step from unbounded surprisal to executing: a bridge principle of its own, a missing principle and not a missing proof, which the framework does not adopt and which is not the occurrence commitment (ZeroParadox/Ordinal/Incompleteness.md, Remark R-M.1).

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 ✓
machineJoinMotion_selfOnlyAt_iff	§I-b	Proved ✓
z2MulMotion_selfOnlyAt_iff	§I-b	Proved ✓
snapEmbedHom	§I-b	Defined ✓
snapEmbed_selfOnlyAt_c1	§I-b	Proved ✓
z2_double_selfOnlyAt_iff	§I-b	Proved ✓
snapEmbed_doubling_noStepAt	§I-b	Proved ✓

Axiom Purity

Axiom footprint: at most [propext, Classical.choice, Quot.sound], per declaration (machineJoinMotion and machineJoinMotion_selfOnlyAt_iff carry [propext] only).

propext, Classical.choice and Quot.sound are Lean's own axioms; here they are reached through Mathlib's ordinal theory (ZP-L), 2-adic analysis (ZP-B), and computability theory (ZP-K).

Axiom Purity

Classical.choice is carried by the statements of the ordinal fixed-point (nfp) results as written (the statement controls printed in ZeroParadox/Ordinal/Epsilon0MinMax.lean; ZeroParadox/Category/ChoiceCannotBe.md), and present in the computability fixed-point (Kleene's theorem). ZP-K Section IV tabulates the per-declaration measurements.

The footprints listed here are copies; to re-measure one, run #print axioms on the named declaration.

End of ZP-M | 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 Incompleteness.lean theorems verified. Axioms: at most [propext, Classical.choice, Quot.sound], per declaration.