ZP-M Illustrated Companion

THE ZERO PARADOX
ZP-M: Kleene–Ordinal Bridge

ZP Companion | Version 1.2 | May 2026

This companion explains the ideas in plain language. It is not the formal document — every claim here restates a result already proved in ZP-M: Kleene-Ordinal Bridge. Consult that document for the authoritative mathematics.

---

1. What This Layer Does

ZP-M is a consolidation layer. Its job is to connect two results that were proved separately in ZP-K and ZP-L and show that they are looking at the same structure from different angles.

ZP-K proved that the initial state c_0 ($\perp$) is a Kleene fixed point — a program that is its own program, with no external executor required. ZP-L proved that the ordinal ε_0 is the exact snap threshold: the tower ω, ω^ω, ω^ω^ω, … approaches ε_0 from below, and any monotone map that sends tower stages to c_0 must send ε_0 to c_1. ZP-L also showed that the tower encodings in $\mathbb{Z}_2$ converge to 0.

ZP-M builds the bridge connecting these: a formal map snapEmbed that sends c_1 (the snap state) to 0 in $\mathbb{Z}_2$, and proves that all three objects — the ordinal ε_0, the snap state c_1, and the 2-adic zero — are formally co-witnessed in a single theorem (zpm_triangle). It also closes a gap in ZP-L's minimality theorem by showing that its free hypothesis follows from simpler conditions.

---

2. The Type Bridge (snapEmbed)

The two layers — ZP-E's machine state space $\{c_0, c_1\}$ and ZP-B's 2-adic integers $\mathbb{Z}_2$ — are mathematically unrelated types. One is a two-element set; the other is an infinite algebraic structure. snapEmbed is the map that connects them structurally.

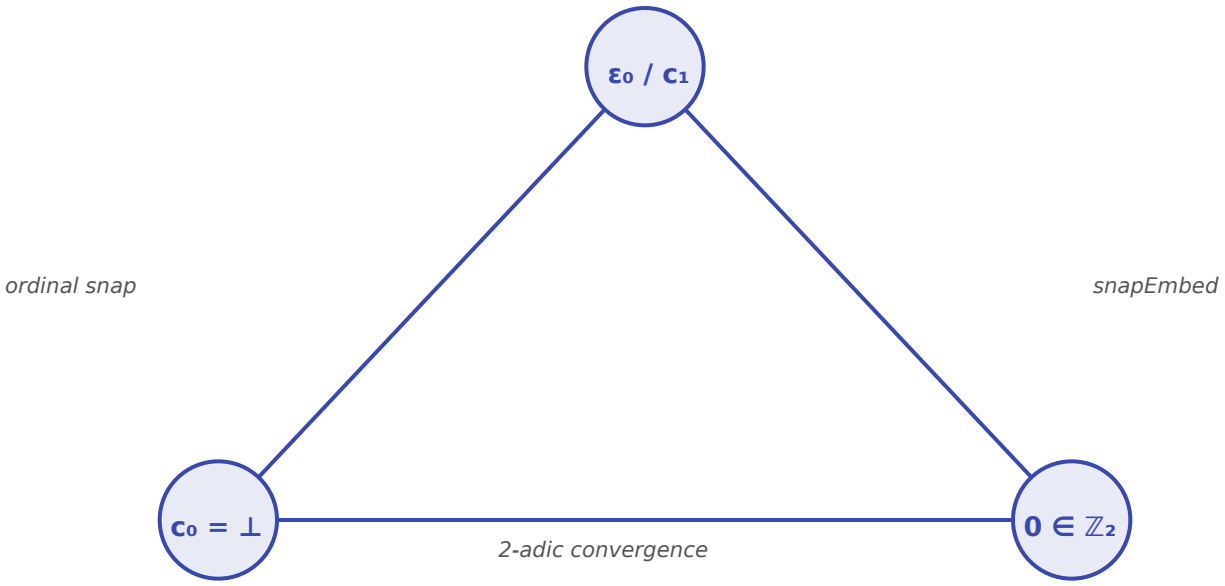
snapEmbed
c_0 (pre-snap) $\mapsto 1 \in \mathbb{Z}_2$ (a 2-adic unit, valuation 0)
c_1 (snap state) $\mapsto 0 \in \mathbb{Z}_2$ (the 2-adic zero, infinite valuation)

The map is injective (c_0 and c_1 land in distinct places) and preserves the algebraic structure: joining c_1 with anything gives c_1, and multiplying 0 by anything gives 0. The same absorbing-element pattern holds in both types, under different operations.

The key geometric fact: 0 in $\mathbb{Z}_2$ is divisible by every power of 2. That is what "infinite 2-adic valuation" means. This is the same mathematical signature as $\perp = \{\perp\}$ — a structure that contains itself at every level of depth, with no bottom that is not also the top.

3. The Triangle

Three objects that appear in three different layers of the framework are co-witnessed in a single Lean theorem: zpm_triangle.



The Kleene–Ordinal–2-adic triangle: three edges, one formal proof.

The left-side edge is the ordinal snap: below ε_0, the tower stages map to c_0; at ε_0, the canonical map flips to c_1. The right-side edge is the type bridge: snapEmbed sends c_1 to 0. The bottom edge is the 2-adic convergence proved in ZP-L: the tower encodings in $\mathbb{Z}_2$ converge to 0 as the ordinal stages approach ε_0.

All three edges are theorems, and zpm_triangle assembles them into a single formal statement. This is the first place in the framework where the ordinal, computational, and 2-adic threads appear in the same proof.

4. Closing a Gap in ZP-L

ZP-L's central minimality theorem (snap_exactly_at_epsilon_zero) carried a free hypothesis called hfp. It asserted that for any map from ordinals to machine phases, every fixed point of ω-exponentiation must be sent to c_1. This was not proved from the ordinal structure — it was an extra assumption.

ZP-M §II closes this gap. The argument is short: if a map is monotone and sends ε_0 to c_1, then for any other fixed point $\alpha \geq \varepsilon_0$, monotonicity forces the map to agree with c_1 at α (because c_1 absorbs every join). The minimality theorem now holds under weaker hypotheses: just monotonicity and "snap at ε_0" together imply the full result, without hfp as a separate commitment.

hfp is not an additional assumption — it is a consequence of the ordinal structure plus the alignment condition $\varphi(\varepsilon_0) = c_1$. The gap in ZP-L is closed.

---

5. One Pattern, Two Domains

The deepest result in ZP-M is not a new theorem but a recognition: Kleene's second recursion theorem (computability) and the construction of ε_0 (ordinal theory) are the same argument running in different mathematical worlds.

Kleene (computability)	Ordinal (set theory)
Operation: eval c = selfApply c	Operation: α ↦ ω^α
Fixed point: ∃ c, IsComputationalQuine c	Fixed point: ε₀ = ω^ε₀ (minimal)
Domain: codes + Gödel numbers	Domain: ordinals + ω-tower iteration
Forced by: second recursion theorem	Forced by: least fixed-point theorem

Same schema, different domains — no shared machinery between the two.

The diagonalization schema in two domains: same structure, no shared machinery.

In each case: you define an operation that refers to itself, and the framework forces a fixed point to exist. In computability theory, the operation refers to a program's own Gödel number (its address in the code space). In ordinal theory, the operation is ω-exponentiation, and the fixed point is the first ordinal that the tower cannot exceed. The mechanisms are entirely different. The schema is identical.

The theorem both_fixed_points_exist states this directly: both fixed points exist, and their proofs live in the same Lean context. This is not a coincidence that requires explanation — it is the recognition that diagonalization is a structural fact that transcends any particular domain.

---

6. Remark R-M.1: Where the Frame Ends

ZP-M establishes the diagonalization frame for two of the three threads in the DA-1 argument. Path 1 (AFA structural) and Path 3 (Kleene computational) are both diagonalization instances — they were unified in ZP-K. Now both share the frame with the ordinal result.

Path 2 (informational) remains outside this frame. L-INF (ZP-C) says that ⊥ has unbounded information content — no finite description can capture it. This is structurally analogous to the diagonalization pattern, but the analogy has not been made formal.

The reason, visible from ZP-M, is a framework separation: L-INF is a measure-theoretic result (about probability distributions and Shannon entropy), while Kleene's theorem is a computability-theoretic result (about codes and recursive functions). The two use no shared mathematical machinery. The concept that bridges them — Kolmogorov complexity, which is simultaneously an information-theoretic and computability-theoretic notion — is not yet formalized in Mathlib.

DA-1 Path 2 is not a missing proof step — it is a genuine framework boundary. ZP-M makes the boundary precise: on one side, diagonalization over codes and ordinals (in Lean scope). On the other side, the AIT bridge between incompressibility and self-execution (outside current Lean scope).

Key Results — ZP-M v1.0
snapEmbed: $c_0 \mapsto 1$, $c_1 \mapsto 0$ in $\mathbb{Z}_2$ — injective, join-to-multiply morphism. ✓
hfp_from_epsilon_zero: hfp derived from monotonicity + $\varphi(\varepsilon_0) = c_1$ alone. ✓
zpm_triangle: all three edges co-proved — ordinal snap, 2-adic convergence, type bridge. ✓
both_fixed_points_exist: Kleene and ordinal fixed points co-witnessed in one formal context. ✓
R-M.1: DA-1 Path 2 boundary made precise — AIT bridge is outside current Lean scope; gap is framework separation, not missing proof.
All 9 theorems proved sorry-free. Axiom footprint: [propext, Classical.choice, Quot.sound].
