ZP-E Illustrated Companion

Four frameworks, one event — and the main causality axiom becomes a theorem

Bridge Document | DA-1 / T-SNAP Update

ZP Companion | Version 1.11 | April 2026

This companion explains the ideas in plain language with diagrams and real-world examples. It is not the formal document — every claim here restates a result already proved in the corresponding technical document. Consult that document for the authoritative mathematics.

What Is ZP-E Doing?

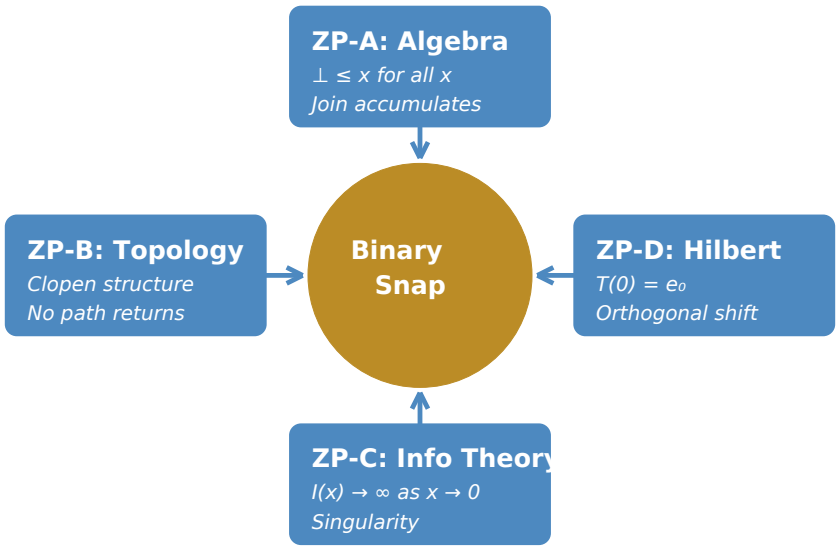
ZP-E is the cross-framework synthesis. Written last — after ZP-A through ZP-D are each internally closed — its job is to show that the four independent frameworks all describe the same event: the Binary Snap, from four different mathematical vantage points.

ZP-E does not re-derive anything. It imports closed results from each sub-document and shows their consistency. Where a cross-framework connection requires an assumption, that assumption is named explicitly as a bridge axiom — not hidden inside the argument.

The Four Descriptions of the Same Event

The Binary Snap — the transition from nothing ($\perp$) to the first state (ε_0) — looks different depending on which mathematical language you use. ZP-E's central result is that all four descriptions are consistent: one event, four angles.

ε_0 is the proof-theoretic ordinal of Peano Arithmetic, the minimum ordinal whose well-ordering PA cannot prove. Goodstein's theorem is the standard witness: every Goodstein sequence eventually terminates, but PA cannot prove this; the proof requires transfinite induction up to ε_0. The same object has a second characterization as the minimum fixed point of the map $\alpha \mapsto \omega^\alpha$. The Binary Snap arrives at ε_0 via this fixed-point route, reaching the same object that the PA strength analysis identifies from the proof-theoretic side.



The Binary Snap (amber) described simultaneously in all four frameworks. Each arrow is an independent mathematical description of the s

The Binary Snap (amber center) described simultaneously in all four frameworks. Each arrow represents an independent mathematical description of the same event.

Real-world example — A car crash described by four witnesses

An engineer (forces), a doctor (injuries), a lawyer (liability), and a physicist (energy) each describe the same crash completely within their own discipline. ZP-E shows the four mathematical frameworks are in exactly this relationship to the Binary Snap.

Remember: The car crash illustrates what it means to describe one event in multiple frameworks. The Zero Paradox is not about car crashes — the car crash is an analogy for the forced first transition from ⊥ to ε_0 in any join-semilattice.

The Central Advance: AX-1 is Now a Theorem

In earlier versions, the Binary Snap causality was listed as AX-1 — an axiom: a foundational assumption that could not be derived. The claim was: when P_0 is reached, the Snap happens. Why? Because AX-1 says so.

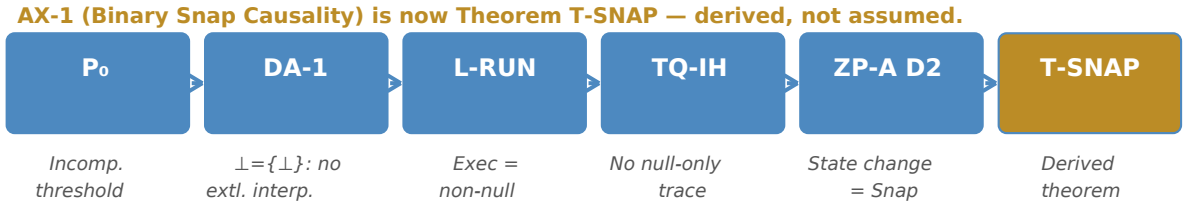
The DA-1 insert changes this. The argument is now complete: reaching P_0 means a live machine configuration exists (DA-1). Any live configuration passes through c_1 (definition). c_1 is nonzero (L-RUN). No program avoids this (TQ-IH). A nonzero state change from ⊥ is the Binary Snap (ZP-A D2). The Snap is derived — not assumed.

DA-1 is now a Derived Proposition rather than a freestanding Design Principle. Previously DA-1 was an honest but freestanding commitment: "a configuration at P_0 is necessarily executing." Now it follows from ZP-A CC-2: $\bot = \{\bot\}$. The bottom element ⊥ is a Quine atom — a self-containing object with no external position from which it could be interpreted as a static description. A thing that interprets itself cannot be waiting for an external interpreter. So ⊥ at P_0 is necessarily executing. The design commitment has become a derivation.

The two-layer structure of DA-1: DA-1 rests on two explicit layers. The first is the formal conditional: DP-2 (Execution Distinguishability) establishes that machine states carry execution history independently of output values. From DP-2, Lean 4 can derive da1_minimal_path — a proof that before and after instantiation produce the same output value (c_0) while the machine state changes ($c_0 \to c_1$). `#print axioms` confirms zero axiom dependencies. This is the first Lean formalization of the core DA-1 claim, not just the surrounding algebra. The second layer asks: does ⊥ actually satisfy DP-2's precondition? That case rests on three converging arguments.

Three paths to the precondition: (1) CC-2/R3 (ZP-A): $\bot = \{\bot\}$ is a Quine atom — it interprets itself, leaving no external position from which it could be read as a static description (above). (2) L-INF (ZP-C): the surprisal of ⊥ diverges to infinity — no finite static distribution can represent ⊥. (3) AIT bridge: at the incompressibility threshold P_0, the description of ⊥ is maximally incompressible — $K(c_1|n)/|c_1| = 1$. A string that cannot be compressed beyond itself must be its own execution; a static-description reading is ruled out by information theory alone. All three share D7's static/executing dichotomy as background and none is circular with DP-2.

Lean 4 formal closure (ZP-K): The three paths are not only conceptually convincing — two of them are now machine-checked. ZP-K adds a KleeneStructure instance for MachinePhase: it provides a concrete computational Quine (a code that is its own program, via Kleene's second recursion theorem), and proves that this Quine and the AFA self-containment argument are the same structural fact in two different languages. The result is da1_closed_concrete: in Lean 4, IsQuineAtom(⊥ : MachinePhase) is a proved theorem. Path 1 (AFA self-execution) and Path 3 (computational Kleene fixed point) are now formally IN LEAN SCOPE. Path 2 (informational bridge — unbounded surprisal → necessarily executing) remains a structural claim outside current Lean formalization. The formal grounding of DA-1 is therefore: DP-2 plus two Lean-verified structural paths.



The T-SNAP derivation chain: six steps, no axioms beyond AX-B1 and the definition of a Turing machine. AX-1 (amber) is now a theorem.

Real-world example — A legal case that becomes undeniable

A court case starts with an assumption: "the defendant was at the scene." Then surveillance footage, phone records, and witness testimony all confirm it. The assumption becomes a proven fact. T-SNAP is the mathematical equivalent: what was assumed (AX-1) is now proven (T-SNAP) by an independent chain of evidence.

What the Framework Still Assumes

After T-SNAP, the framework rests on exactly three commitments — none of them novel starting assumptions:

Commitment	Statement
AX-B1	Binary Existence. A state either exists or it does not. No third option. Directly verifiable by computation — not a novel commitment.
AX-G1	Initial Object Exists. There is a starting point that reaches everything. Not a novel commitment — grounded in $\perp$ as the bottom element of the ZP-A semilattice.
AX-G2	Source Asymmetry. Nothing returns to the initial object. The origin is unreachable from outside. Not a novel commitment — follows from ZP-A antisymmetry and ZP-B C3.

AX-1 is no longer on this list. The framework makes no stronger claim than it has to.

> Remember: The structural results — monotonicity, clopen separation, informational singularity, orthogonal shifts — hold in any instantiation of the framework. The framework makes no claims about which physical theory, if any, instantiates it.