THE ZERO PARADOX

ZP-J: Executability of Self-Reference

Version 1.0 | April 2026

v1.0: Initial release — Theorem T-EXEC: the Quine atom Q = {Q} is provably the bottom element ⊥ of any ZP-A lattice with AFA grounding. CC-1 (ZP-A) is derived as a structural consequence, not committed as a modelling choice. All ZPJ.lean theorems verify axiom-free in Lean 4.

This document establishes Theorem T-EXEC (Executability of Self-Reference): in any ZP-A join-semilattice with AFA (Anti-Foundation Axiom) grounding, the unique Quine atom Q — the element satisfying Q = {Q} — is provably the bottom element ⊥. This closes the last open bridge between the set-theoretic and order-theoretic layers of the framework. CC-1 from ZP-A, which stated "$S_0 = ⊥$" as a modelling commitment, becomes a derived theorem.

The key is a single structural field in the AFAStructure typeclass: bot_self_mem, which encodes that the bottom element is self-containing. With this, the proof of T-EXEC is three lines. No bridge axiom. No freestanding commitment. The identification ⊥ = {⊥} — implicit in the framework since ZP-E's DA-1 Path 1 — is now a verified structural prerequisite, not an asserted coincidence.

Section I: The Open Question — CC-1 as a Modelling Commitment

---

I. CC-1 in ZP-A

ZP-A Conditional Claim CC-1 states: if the state sequence is initialised at ⊥, then ⊥ ≤ S(n) for all n. This follows trivially from T2 (⊥ is the global minimum), so its formal content is essentially: we are choosing ⊥ as the initial state S_0.

The label "Conditional Claim" (CC) marks this as a modelling commitment — an explicit choice not derivable from the axioms A1–A4 alone. ZP-A's axioms give us a semilattice with a bottom element. They do not say which instantiation of the semilattice should start at that bottom. CC-1 asserts: ours does. This is well-motivated, but it is a choice.

The question ZP-J investigates: is this choice forced? Is there a structural reason — derivable from the framework's foundational commitments — that any well-grounded instantiation of ZP-A must begin at ⊥? The answer is yes, provided the lattice has AFA grounding.

II. The Implicit Identification

ZP-E's DA-1 Path 1 already states: ⊥ = {⊥} (Quine atom, ZF+AFA). This identification — the bottom element is the unique self-containing set under AFA — was present informally but never formally bridged to CC-1. It appeared as motivation for why ⊥ is the right starting point, not as a derivation of it.

The gap: ZP-A's lattice order is abstract (defined by axioms A1–A4). AFA is a set-theoretic axiom. There is no automatic connection between "x is self-containing" and "x is the lattice bottom." Connecting them requires a bridge — and that bridge was the missing piece. ZP-J provides it.

> Open question entering ZP-J: Is CC-1 ($S_0 = \bot$) forced by the framework's foundational structure, or is it an independent modelling choice? If forced, what is the structural reason? Answer (ZP-J T-EXEC): it is forced — by the AFA identification $\bot = \{\bot\}$, encoded structurally in the AFAStructure typeclass.

Section II: AFA Machinery — Self-Membership and the Quine Atom

---

I. The Anti-Foundation Axiom

Standard set theory (ZF) includes the Foundation Axiom: every non-empty set S contains an element disjoint from S. A consequence is that no set can contain itself: $x \in x$ is impossible in ZF. This rules out self-referential sets by fiat.

The Anti-Foundation Axiom (AFA, Aczel 1988) replaces Foundation with a universal existence and uniqueness result: every accessible pointed graph (APG) has a unique set-theoretic decoration. Under AFA, self-containing sets are not only possible but uniquely characterised. The resulting theory ZF+AFA is consistent and expressive — it is the natural set-theoretic home for fixed-point and self-referential structures.

II. The Quine Atom

Under AFA, the equation $x = \{x\}$ has a unique solution. This solution is called the Quine atom, denoted Q. Q is the unique set that contains itself as its sole member: $Q = \{Q\}$. It is not the empty set ($\varnothing = \{\}$ contains nothing); it is not any well-founded set (those cannot contain themselves by Foundation's analogue in the well-founded universe). Q is the minimal non-trivial AFA set — it contains exactly one thing, and that thing is itself.

The AFA uniqueness guarantee is the key property: there is at most one Quine atom. Any two self-containing elements are equal. This means the self-membership property uniquely identifies an element — a "fingerprint" that belongs to exactly one object in the universe.

AFA Uniqueness (AFAStructure.quine_unique)
For any type L with AFA structure: if $x, y \in L$ both satisfy selfMem(x) and selfMem(y), then $x = y$.
Informally: the Quine atom is unique. Self-containment is a property held by at most one element of any AFA-structured type.
Lean: AFAStructure.quine_unique — encoded as a typeclass field, not a theorem, because AFA is a foundational axiom and cannot be derived from type-theoretic principles alone. It is a structural prerequisite for any AFA-grounded lattice.

III. Self-Membership as a Lattice Predicate

In ZP-J, the AFA machinery is abstracted minimally. We do not need the full apparatus of accessible pointed graphs, bisimulation, or set decoration. We need only two things: a predicate selfMem on the lattice elements, and the guarantee that it is held by at most one element (quine_unique). The third structural field — bot_self_mem — is where the bridge is built.

The AFAStructure typeclass in ZPJ.lean captures exactly this. Its three fields are sufficient to derive T-EXEC with no additional axioms. The full AFA machinery (APGs, bisimulation, decoration) provides the informal justification for why any concrete lattice grounded in ZF+AFA satisfies these three fields — but the formal derivation requires only the fields themselves.

Section III: AFAStructure — The Structural Bridge

I. The Three Fields

The AFAStructure typeclass for a ZP-A semilattice L has three fields. The first two encode standard AFA properties. The third is the bridge.

AFAStructure Typeclass (ZPJ.lean § I)
class AFAStructure (L : Type*) [ZPSemilattice L] with:
(1) selfMem : L → Prop — the self-membership predicate. selfMem(x) means x contains itself as a member in the AFA sense.
(2) quine_unique : ∀ x y : L, selfMem(x) → selfMem(y) → x = y — AFA uniqueness. At most one element of L is self-containing.
(3) bot_self_mem : selfMem(⊥) — the bridge field. The bottom element of the lattice is self-containing. This is the formal encoding of ⊥ = {⊥}.

II. What bot_self_mem Says

bot_self_mem is the single structural claim that connects the order-theoretic world (ZP-A's lattice) to the set-theoretic world (AFA). It says: the bottom element ⊥ of the lattice is self-containing — it satisfies selfMem(⊥).

In set-theoretic terms: $\bot \in \bot$, i.e. $\bot = \{\bot\}$. In the framework: the null state contains itself as its sole content. This is not a new claim — it is the identification that ZP-E's DA-1 Path 1 already invokes informally. ZP-J encodes it as a typeclass field, making it a verifiable structural prerequisite rather than a narrative motivation.

Any concrete lattice L that claims to be AFA-grounded must prove bot_self_mem as part of its AFAStructure instance. If it cannot, it is not genuinely AFA-grounded — the identification $\bot = \{\bot\}$ is part of what "AFA-grounded" means.

III. Why a Typeclass Field Rather than a Freestanding Axiom

In the stub version of ZPJ.lean (commit 629a534), the bridge was a freestanding axiom: ax_j1_quine_join_identity. It stated directly that the Quine atom satisfies the join-identity. This compiled, but the purity check showed T-EXEC depending on ax_j1_quine_join_identity — a named axiom floating outside any typeclass.

Encoding the commitment as a typeclass field is philosophically and formally cleaner. A freestanding axiom is a global assertion that the proof checker accepts without verification. A typeclass field is a requirement: any instance of AFAStructure must supply a proof of bot_self_mem. The commitment is not asserted once and forgotten — it is checked at every instantiation. Concrete lattices must earn their AFA status.

> The distinction: a freestanding axiom says "trust me, this is true." A typeclass field says "prove it for your specific lattice, or it does not compile." ZP-J uses the second. The philosophical commitment ($\bot = \{\bot\}$) becomes a proof obligation at every concrete instantiation.

Section IV: Theorem T-EXEC — The Quine Atom is Bottom

Theorem T-EXEC — Executability of Self-Reference

Statement: Let L be a ZP-A semilattice with AFAStructure. If q : L is a Quine atom (selfMem(q) and q is unique among self-containing elements), then $q = \bot$.

Lean: ZeroParadox.ZPJ.t_exec — proved in ZPJ.lean. Purity: does not depend on any axioms. ✓

I. The Proof

The proof of T-EXEC is immediate from the typeclass fields. It has three steps:

- hq.2 states: every self-containing element of L equals q. (This is the uniqueness half of IsQuineAtom q.)
- AFAStructure.bot_self_mem states: $\bot$ is self-containing. (This is the bridge field — $\bot = \{\bot\}$ encoded structurally.)
- Applying hq.2 to $\bot$ using bot_self_mem gives: $\bot = q$. By symmetry: $q = \bot$. QED.

The entire proof is one line in Lean 4: (hq.2 bot AFAStructure.bot_self_mem).symm. No appeal to the join operation. No bridge axiom. No DA-2. Just AFA uniqueness applied at $\bot$.

Remark R-J.1

The proof does not use da2_bottom_characterization (from ZP-E). That result — "($\forall$ x, join S x = x) $\leftrightarrow$ S = $\bot$" — is used in the derived theorem J1 (Section V), but T-EXEC itself is purely order-theoretic: it uses only quine_unique and bot_self_mem.

II. IsQuineAtom bot

A corollary of T-EXEC is that $\bot$ itself is a Quine atom. This is the converse direction: not only does the Quine atom equal $\bot$ (T-EXEC), but $\bot$ equals the Quine atom (bot_is_quine_atom).

Proposition — bot_is_quine_atom

In any AFAStructure lattice L: IsQuineAtom(⊥).

Proof: ⊥ is self-containing by bot_self_mem. Any other self-containing x satisfies x = ⊥ by quine_unique(x, ⊥, selfMem(x), bot_self_mem).

Lean: ZeroParadox.ZPJ.bot_is_quine_atom — does not depend on any axioms. ✓

III. The Full Biconditional

Combining T-EXEC and bot_is_quine_atom yields the full biconditional: IsQuineAtom(q) ↔ q = ⊥. Being the Quine atom and being the bottom element are the same property, stated in set-theoretic and order-theoretic language respectively.

Theorem t_exec_iff — Full Equivalence

For any q : L in an AFAStructure lattice:

IsQuineAtom(q) ↔ q = ⊥ ↔ ∀ x : L, join q x = x.

The three conditions are mutually equivalent: Quine atom (set-theoretic), bottom element (order-theoretic), and join-identity element (algebraic). They are three formulations of one structural role.

Lean: ZeroParadox.ZPJ.t_exec_iff — does not depend on any axioms. ✓

Section V: Derived Results — J1 and CC-1 as Theorems

I. J1 — QuineJoinIdentity (Derived)

In the initial stub of ZPJ.lean, the claim "the Quine atom satisfies the join-identity" was stated as a freestanding axiom (ax_j1_quine_join_identity). ZP-J replaces it with a theorem.

Theorem J1 — QuineJoinIdentity (formerly Axiom AX-J1)

In any AFAStructure lattice L, if q is the Quine atom, then ∀ x : L, join q x = x.

Proof: q = ⊥ by T-EXEC. Then join q x = join ⊥ x = x by A4 (bot_join, ZP-A). ✓

Status: DERIVED THEOREM. Was axiom ax_j1_quine_join_identity in the stub — now proved from T-EXEC + ZP-A A4. No freestanding axiom remains.

Lean: ZeroParadox.ZPJ.j1_quine_join_identity — does not depend on any axioms. ✓

The elimination of AX-J1 is the payoff of bot_self_mem. In the stub, we could prove T-EXEC from AX-J1 (via da2_bottom_characterization), but AX-J1 was itself an axiom — the buck stopped there. With bot_self_mem, the buck stops at the typeclass definition: a structural requirement, not a global assertion.

II. CC-1 Derived

ZP-A CC-1 is now a theorem. If the initial state S_0 of a state sequence is the Quine atom Q, then $S_0 = \bot$ — a consequence of T-EXEC, not a modelling choice.

Theorem CC-1 (Derived) — cc1_derived
Let L be an AFAStructure lattice. Let $S : \mathbb{N} \to L$ be a state sequence (ZP-A D3) and Q : L a Quine atom. If S(0) = Q, then S(0) = ⊥.
Proof: S(0) = Q (hypothesis). Q = ⊥ by T-EXEC. Therefore S(0) = ⊥.
The modelling commitment "we choose ⊥ as the initial state" is replaced by "the Quine atom IS ⊥, structurally." Starting at Q and starting at ⊥ are not two choices — they are the same state, identified by structure.
Lean: ZeroParadox.ZPJ.cc1_derived — does not depend on any axioms. ✓

III. Uniqueness of the Bottom Role

A further consequence is algebraic uniqueness: in any ZP-A semilattice (without even requiring AFA structure), at most one element can satisfy the join-identity. This is a clean corollary of da2_bottom_characterization (ZP-E).

Theorem — bot_unique
For any ZP-A semilattice L (no AFA required): if x, y : L both satisfy ∀ z, join x z = z and ∀ z, join y z = z, then x = y.
Proof: da2_bottom_characterization gives x = ⊥ and y = ⊥. Therefore x = y.
Lean: ZeroParadox.ZPJ.bot_unique — does not depend on any axioms. ✓

Section VI: Implications — What Was the Commitment?

I. The Remaining Foundation

T-EXEC and its corollaries carry zero freestanding axioms in the Lean 4 purity check. The entire derivation traces to the fields of two typeclasses: ZPSemilattice (from ZP-A) and AFAStructure (new in ZP-J). No standalone axiom statement appears anywhere in the ZPJ.lean proof obligations.

The foundational commitments are the typeclass fields themselves. In ZPSemilattice: A1–A4 (the semilattice axioms). In AFAStructure: selfMem (a predicate), quine_unique (AFA uniqueness), and bot_self_mem (the bridge). These are the structural prerequisites for any lattice that claims ZP-A + AFA grounding. They are not axioms floating in the ambient theory — they are proof obligations that any concrete instance must discharge.

II. Where the Philosophy Lives Now

The philosophical content of $\bot = \{\bot\}$ has not disappeared. It has moved. Previously it lived in a narrative comment in ZP-A (CC-1's informal motivation) and in ZP-E's DA-1 Path 1 (one of three informal arguments for instantiation as execution). Now it lives in the definition of AFAStructure itself — specifically in bot_self_mem.

This is the correct location for a foundational claim. It is not hidden in a proof; it is stated in the typeclass definition where any reader can see it. Any lattice that wants to use ZP-J's results must provide a proof that its bottom element is self-containing. If it cannot, it is simply not an AFA-grounded lattice in the sense required by this framework.

III. The Formal Chain is Now Closed

The derivation chain entering ZP-J had one remaining gap: CC-1 was a committed starting point, not a derived one. The chain from AFA structure to T-SNAP now has no such gaps. Every node is either a proved theorem or a typeclass field:

- ZP-A axioms A1–A4: semilattice structure (ZPSemilattice fields).
- ZP-B: 2-adic topology and irreversibility (proved from Mathlib).
- ZP-C: information theory, L-INF, L-RUN (proved axiom-free or from Mathlib).
- ZP-D: state layer, orthogonality (proved from ZP-A and ZP-B).
- ZP-E: T-SNAP, DA-1, DA-2 (T-SNAP and DA-2 proved axiom-free; DA-1 Path 3 outside Lean scope).
- ZP-J: AFAStructure fields (selfMem, quine_unique, bot_self_mem). T-EXEC, J1, CC-1: proved axiom-free. ✓

DA-1 Path 3 (the AIT/Kolmogorov complexity argument in ZP-E) remains outside Lean scope for the same reason it always has: Kolmogorov complexity is uncomputable and absent from Mathlib. This is a Lean tooling limitation, not a gap in the mathematics. ZP-J does not affect DA-1's status.

Key Results — ZP-J v1.0
T-EXEC: In any AFAStructure lattice, IsQuineAtom(q) $\leftrightarrow$ q = $\bot$. Self-reference and bottom-hood are the same structural role.
J1 (Derived): The Quine atom satisfies the join-identity $\forall$ x, join q x = x. Was axiom ax_j1 in the stub — now a theorem from T-EXEC + A4.
CC-1 (Derived): $S_0 = Q \Rightarrow S_0 = \bot$. The modelling commitment is replaced by a structural consequence.
Lean purity: all ZPJ.lean theorems verify "does not depend on any axioms." ✓

Traceability Register — ZP-J v1.0

Claim	Grounded In	New axiom?	Status
T-EXEC: Quine atom = $\bot$	AFAStructure.quine_unique + AFAStructure.bot_self_mem	None — typeclass fields, not freestanding axioms	Lean: t_exec — does not depend on any axioms ✓
J1: Quine join-identity	T-EXEC + ZP-A A4 (bot_join)	None	Lean: j1_quine_join_identity — axiom-free ✓ (was axiom ax_j1 in stub)

Claim	Grounded In	New axiom?	Status
CC-1 (Derived): $S_0 = Q \Rightarrow S_0 = \bot$	T-EXEC	None	Lean: cc1_derived — axiom-free ✓ (was ZP-A conditional claim)
bot_is_quine_atom: IsQuineAtom(⊥)	AFAStructure.bot_self_mem + quine_unique	None	Lean: bot_is_quine_atom — axiom-free ✓
t_exec_iff: IsQuineAtom(q) ↔ q = ⊥	T-EXEC + bot_is_quine_atom	None	Lean: t_exec_iff — axiom-free ✓
bot_unique: join-identity is unique	ZPE.da2_bottom_characterization	None	Lean: bot_unique — axiom-free ✓ (no AFA required)
AFAStructure.bot_self_mem	AFA (ZF+AFA) — ⊥ = {⊥}	Typeclass field — proof obligation at instantiation	Not a theorem; a structural prerequisite. Must be discharged by each concrete instance.

Open Items Register — ZP-J v1.0

Item	Status	Description
CC-1 (ZP-A) derivability	CLOSED — T-EXEC (ZP-J)	CC-1 is now a theorem. The Quine atom = ⊥ is structurally derived. No freestanding axiom. No modelling commitment beyond AFAStructure typeclass fields.
AX-J1 bridge axiom	CLOSED — J1 derived	The stub version of ZPJ.lean had ax_j1 as a freestanding axiom. The final version derives J1 as a theorem from T-EXEC + A4. Axiom eliminated.
AFAStructure concrete instances	OPEN — future work	ZPJ.lean defines AFAStructure abstractly. Concrete instances (e.g. the MachinePhase semilattice from ZP-E, the Q_2 model from ZP-B) must discharge bot_self_mem explicitly. Defining these instances is a natural next step.
DA-1 Path 1 formalisation	OPEN — outside Lean scope	DA-1 Path 1 (ZP-E) invokes ⊥ = {⊥} informally. ZP-J now provides the formal grounding for that identification. A future revision of ZP-E could cite ZP-J T-EXEC as the formal basis for Path 1, closing the last informal step in the DA-1 three-path argument.
ZP-A CC-1 label update	OPEN — editorial	ZP-A still labels CC-1 as a Conditional Claim. With T-EXEC established in ZP-J, the label can be updated to Derived Theorem (citing ZP-J). This is an editorial update, not a mathematical change. Other ZP documents are not modified by ZP-J.

End of ZP-J v1.0 | Theorem T-EXEC: Executability of Self-Reference | CC-1 derived — no freestanding axioms | All ZPJ.lean theorems: does not depend on any axioms | Remaining foundation: ZPSemilattice (A1–A4) and AFAStructure (selfMem, quine_unique, bot_self_mem)