THE ZERO PARADOX

ZP-R Addendum: The Diagonal Family

Initial: July 2026 | Current: 1.0, July 2026

Addendum to ZP-R. Where ZP-R locates and realizes the framework's self-application fixed point ⊥ as a Lawvere fixed point, this addendum maps the complete family of that relationship: every variant of self-reference at ⊥, organized by the μ/ν fork — the wall faces where self-reference cannot close (no fixed point) and the floor faces where it does (a fixed point exists). Every entry is tied to ⊥ and carries a machine-checked Lean 4 witness. The diagonal-family unification is Lawvere (1969) / Yanofsky (2003), cited; the contribution is the formalization and the tie to the bottom element (core choice-free; the computability faces choice-carrying). It supersedes the earlier private "Zero as a Wall" working draft.

The self-referential paradoxes — Cantor, Russell, Gödel, Turing, Tarski — are one theorem seen from many sides: Lawvere's fixed-point theorem (Lawvere 1969), unified by Yanofsky (2003). ZP-R identifies the framework's bottom element ⊥ as an instance of that fixed point and locates where it is realized. This addendum takes the next step and lays out the whole family in one place, each variant with its Lean witness, so the framework can say plainly: every variant of the self-referential relationship at ⊥ is accounted for.

The organizing principle is a single discriminator, the μ/ν fork (ZP-R §I; ZP-P): the engine is negation, which has no fixed point, and Lawvere's theorem turns that one fact into the whole family. Where the self-reference relation is well-founded, no self-loop exists — the wall (μ): Cantor, Russell, Turing, Tarski, Curry. Where it is non-well-founded, a self-referential fixed point exists — the floor (ν): the Quine atom, the Kleene quine, the Löb sentence, and (with a twist) Rice. Most of the engine and the classical faces are already formalized in the framework's Wall module; this addendum completes the roster with the four faces added in the diagonal-family build (Tarski, Curry, Rice, Löb) and ties every entry to ⊥.

Section I: The Engine

The root of the entire family is one fact: negation has no fixed point. No proposition p satisfies $p \leftrightarrow \neg p$. Lawvere's theorem is the amplifier: a point-surjection $e : A \to (A \to B)$ forces every $f : B \to B$ to have a fixed point, so a fixed-point-free f (negation) refutes the surjection. Every wall face below is this contrapositive; every floor face is the same engine where the refutation is lifted (the relation is non-well-founded, or the diagonal is not constructible, so the fixed point returns).

The engine (Wall.lean)
negation_no_fixedpoint: $\neg (p \leftrightarrow \neg p)$ — negation is fixed-point-free (the root).
lawvere_fixedpoint: a point-surjection $A \to (A \to B)$ forces every $f : B \to B$ to have a fixed point (Lawvere 1969, the general engine).
The μ/ν fork is exactly its two regimes: μ = no fixed point (the wall); ν = a fixed point exists (the self-referential object).
Lean purity: choice-free. ✓

Section II: The Wall Faces (μ) — self-reference cannot close

On a well-founded relation there is no self-loop: no x with x pointing at itself (wf_no_selfloop). Under Foundation this is "no set is self-membered." Each wall face is the same obstruction in its own domain — self-reference tries to close and cannot, and the failure is the bottom element refusing to be internalized.

The wall, general (Wall.lean)

wf_no_selfloop: a well-founded relation admits no self-loop — the object-level core the metatheoretic boundary reduces to. no_quine_atom: under Foundation, no set is self-membered.

Lean purity: wf_no_selfloop choice-free; no_quine_atom [propext, Quot.sound]. ✓

Cantor / Russell / Turing (Wall.lean) — the classical wall faces

Cantor (cantor_via_engine): no g : A → (A → Prop) is surjective — no self-surjection onto predicates.

Russell (russell_via_engine): no membership relation realizes every predicate (naive comprehension is impossible).

Turing (no_self_decider): no g : A → (A → Bool) is surjective — the halting diagonal flips the decider.

Lean purity: choice-free (all three, off the engine). ✓

Tarski (Tarski.lean) — the truth face

tarski_no_internal_truth: no internal universal truth-naming exists — truth is not definable inside a system that can name all its own predicates (Tarski 1936). The liar sentence (tarski_liar_from_naming) is the explicit diagonal; the T-schema at the liar is absurd (tarski_Tschema_liar_absurd).

The bottom relationship: no floor — tarski_no_truth_bottom (¬ HasLawvereWitness Prop). Truth is the wall dual of Gödel's provability.

Lean purity: choice-free. ✓

Curry (Curry.lean) — the explosion face

curry_paradox: a self-referential p ↔ (p → C) proves C, for ANY C — a strict generalization of the engine (the liar is the C = False case, liar_is_curry_at_false). curry_from_naming: a surjective internal comprehension proves everything.

The bottom relationship: no floor, and pretending otherwise explodes — curry_no_bottom (¬ Surjective naming, via explosion at C = False).

Lean purity: choice-free. ✓

Section III: The Floor Faces (ν) — self-reference closes

Where the self-reference relation is non-well-founded, or the fixed-point-free diagonal is not constructible, the fixed point returns: self-reference closes on a bottom element. These are the ν faces — the framework's keystone, seen in each domain.

The Quine atom (SetTheoryAFA.lean) — the set-theoretic floor

$\bot = \{\bot\}$: the self-containing bottom. t_exec / T-EXEC (t_exec_iff): every Quine atom IS $\bot$ (IsQuineAtom q $\leftrightarrow$ q $= \bot$) — location and uniqueness of the floor. Lives in ZF + AFA (walled off under Foundation, Section II).

Lean purity: choice-free. ✓

The Kleene quine (Lawvere.lean) — the computability floor

computability_face_fixedPoint: every computable self-map on codes has a fixed point — Kleene's recursion theorem, the reflexive object of the computable category (the crossing realized in ZP-R). This is the one face where the fixed point is genuinely Lawvere-sourced.

Lean purity: [propext, Classical.choice, Quot.sound] — Mathlib recursion theory. ✓

Löb (Loeb.lean) — the provability floor; and Gödel's second incompleteness

loeb: given a Löb diagonal ($\psi \leftrightarrow (\Box\psi \to A)$), $\vdash \Box A \to A$ yields $\vdash A$. The Löb sentence is a genuine fixed point (loeb_sentence_is_fixedpoint) — the provability face has a floor.

godel_two: a consistent system cannot prove $\Box\bot \to \bot$ — its own consistency (Gödel's second incompleteness, the A = $\bot$ corollary). Built from scratch (no modal logic in Mathlib): an abstract ProvabilityLogic typeclass with the Hilbert–Bernays–Löb derivability conditions.

Lean purity: choice-free. ✓

Rice (Rice.lean) — the exists-but-undecidable floor

The floor exists but membership at it is undecidable. quine_exists_yet_rice: the Kleene quine EXISTS (recursion theorem) yet every non-trivial semantic property is UNDECIDABLE (rice_face; Rice 1953, cited from Mathlib). rice_face_has_bottom: the recursion-theorem fixed point is the computability bottom.

This is the pivot face: the fixed point is not refuted (as on the wall) but present (ν) — the failure migrates from existence to decidability.

Lean purity: [propext, Classical.choice, Quot.sound] — Mathlib recursion theory. ✓

Section IV: The Complete Map

The whole family in one table. Gödel's first incompleteness ($G \leftrightarrow \neg\mathrm{Prov}(G)$) sits between the columns — the diagonal via the engine, incompleteness rather than inconsistency. Every row is a machine-checked Lean witness tied to $\bot$.

Variant	Face	What self-reference does	Lean witness
the engine	—	negation has no fixed point (the root)	negation_no_fixedpoint, lawvere_fixedpoint (Wall.lean)
Cantor	μ	no self-surjection onto predicates	cantor_via_engine (Wall.lean)

Variant	Face	What self-reference does	Lean witness
Russell	μ	no membership realizes every predicate	russell_via_engine (Wall.lean)
Turing	μ	the halting decider is flipped	no_self_decider (Wall.lean)
Tarski	μ	truth is not internally definable	tarski_no_internal_truth (Tarski.lean)
Curry	μ	self-reference proves everything (explosion)	curry_paradox, curry_no_bottom (Curry.lean)
Gödel 1st	between	self-reference is true-but-unprovable	via lawvere_fixedpoint (Wall.lean)
Quine atom	ν	⊥ = {⊥}: self-containing bottom	t_exec / t_exec_iff (SetTheoryAFA.lean)
Kleene quine	ν	a program computes its own code	computability_face_fixedPoint (Lawvere.lean)
Löb / Gödel 2nd	ν	the provability diagonal closes; consistency unprovable	loeb, godel_two (Loeb.lean)
Rice	ν	the floor exists but is undecidable	rice_face_has_bottom, quine_exists_yet_rice (Rice.lean)

The map is a placement, not a new theorem. The unification of the diagonal family is Lawvere (1969) and Yanofsky (2003); each face is a classical result (or a direct instance of the engine), formalized here axiom-free where the logic is pure and choice-carrying where it inherits Mathlib's recursion theory. What the framework adds is the tie to ⊥ and the completeness of the roster — every variant of the one relationship, in one place.

Fence: the cross-face identity is a type boundary

The faces are the same relationship, not the same object. The Quine atom (a set), the Kleene quine (a code), the Löb sentence, the 2-adic 0 — these are terms of different types in different categories; "x = y" across them is not false, it is not well-formed (ZP-P hard fence; ZP-R F2). What is claimed is that each domain forks at its own self-referential contact point, and that ⊥ is the framework's name for that point in each. The global identification — that these are one object — is held as a fenced conjecture, here as in ZP-R and ZP-P.

Relation to Prior Work

The diagonal-family unification via one fixed-point theorem is Lawvere, "Diagonal Arguments and Cartesian Closed Categories," LNM 92 (1969), and Yanofsky, "A Universal Approach to Self-Referential Paradoxes, Incompleteness and Fixed Points," Bull. Symbolic Logic 9(3) (2003). The individual faces are classical: Cantor; Russell (1901); Gödel (1931); Turing (1936); Tarski (1936); Curry (1942); Löb (1955); Rice (1953); Kleene's recursion theorem. The point-surjective theorem is in Mathlib (Function.exists_fixed_point_of_surjective); the framework's engine re-derives it axiom-free for a self-contained family. The contribution of this addendum is the formalization of the family (core choice-free; the computability faces choice-carrying) and its tie to the bottom element — a placement, not an extension.

Axiom Purity

Axiom Purity
The engine and the wall faces (negation_no_fixedpoint, lawvere_fixedpoint, cantor_via_engine, russell_via_engine, no_self_decider, wf_no_selfloop, Tarski, Curry) and the pure-logic floor (Quine atom t_exec; Löb, godel_two) are choice-free — the diagonal family needs no Axiom of Choice.
The computability floor faces (Kleene: computability_face_fixedPoint; Rice: rice_face_has_bottom, quine_exists_yet_rice) carry [propext, Classical.choice, Quot.sound], inherited from Mathlib's classical recursion theory — disclosed, not claimed otherwise.
Core choice-free, computability realization choice-carrying — the framework's standing pattern. Zero sorry. Verified: lake build, July 2026.

End of ZP-R Addendum | The Diagonal Family | one engine (negation has no fixed point) | wall faces μ (Cantor, Russell, Turing, Tarski, Curry) | floor faces ν (Quine atom, Kleene, Löb / Gödel 2, Rice) | every variant tied to $\perp$ and Lean-witnessed | cross-face identity a type boundary | the map is a placement, not a new theorem.