THE ZERO PARADOX

ZP-R: A Cross-Category Account of the Self-Referential Fixed Point

Initial: July 2026 | Current: 1.29, November 2026

Synthesis / placement layer. The framework's self-application fixed point is a role, filled per face: in the fork / AFA face by ⊥ of the ZPSemilattice (the order-bottom, a fixed point and the only one by the class fields of AbstractSelfApp), in the computability face by a Kleene code, a term of another type. This document asks where the role is filled by a Lawvere fixed point, across three categories ("faces"): refuted in Set on every carrier except a one-element one (Cantor), in the monotone / domain regime (a genuine reflexive object there is the business of Scott's D_∞ and its generalizations) free of a monotone obstruction on complete lattices (Knaster–Tarski), with no reflexive object built here and no map from the two-element chain onto its monotone self-maps (the first finite stage of Soto-Andrade and Varela's two-element example; Trimble; Bergman; stated there for order-preserving maps, ours for any map), and realized in the computability face (Rogers / Kleene). Every theorem used is classical. Each result box (R1–R4) and each F1 property is backed by a machine-checked Lean 4 theorem, except one marked cited: that the computability fixed point is an instance of Lawvere's theorem. The rest is cited background or marked Reading. The contribution is the per-face placement of the framework's self-application role and the scope result. How the faces formalized here relate to Lawvere's theorem is collected in Section III, in the box "Lawvere's theorem, face by face"; ZP-R conjectures nothing global.

A recurring role in this framework is a fixed point of self-application — "self-referential" in the sense of being defined by pointing at itself. Its occupants differ by face: ⊥ of an AFA lattice, the occupant there of the Quine-atom role (a point that steps to itself and to nothing else; quineAtomHost_quine_role_iff, for the host built from the AFA structure), and a program equal to its own transform up to evaluation (a Kleene code). This document asks one checkable question: is a fixed point in that role an instance of Lawvere's fixed-point theorem (Lawvere 1969) — the categorical statement behind the diagonal arguments of Cantor, Russell, Gödel and Tarski, carried to the halting problem and the recursion theorem by Yanofsky (2003) — and if so, in which category? Lawvere states the theorem for cartesian closed categories (Theorem 1.1) and notes that it holds in any category with finite products (§2, Theorem 2.2), with weak point-surjectivity phrased as a map $A \times A \to B$ that represents every map $A \to B$. It yields a fixed point only where such a map exists (in the closed case, a weakly point-surjective $A \to B^A$; the reflexive case is $B = A$). So the honest form of the question is not "is it Lawvere" but "where is it Lawvere."

The answer is: yes in one adequate category, and not globally. We locate precisely where the realization fails and where it succeeds. The mathematics is classical and is invoked, not extended; the contribution is (i) the per-face placement of this framework's own self-application role relative to the Lawvere fixed point — its Lawvere realization refuted in Set on every carrier except a one-element one, the monotone obstruction absent on complete lattices in the monotone / domain face, with no reflexive object built there, the realization achieved in the computability face — and (ii) a scope result: the realization is category-relative — an existence statement in one face rather than a global identification. The general link between self-reference and Lawvere's theorem is Lawvere's (1969) and Yanofsky's (2003), not ours, and so is the obstruction the placement uses: that a fixed-point-free endomap of an object Y rules out, for every A, a

point-surjection $A \to Y^A$ is Lawvere's Corollary 1.2 (p. 5 of the 2006 TAC reprint). Of the suggestion of Soto-Andrade and Varela (1984, pp. 1 and 18) that every structure with the fixed point property (in the order and category sense used throughout ZP-R: every endomap of the structure has a fixed point) is a retract of a reflexive domain, ZP-R answers the question only in Set, where it holds trivially (Section II), and makes no claim about the categories they intend.

Section I: The Fork and the Engine

Begin in order theory, where the picture is cleanest and constructive. Over a complete lattice, a monotone self-map f has a least fixed point (lfp f) and a greatest fixed point (gfp f) — Knaster–Tarski (Knaster 1928, existence in the power-set case; Tarski 1955). Existence of a fixed point is therefore free: it is never in question. What can vary is uniqueness, and there is a clean equivalence, the μ/ν fork: a specification pins a unique object exactly when the fork collapses. When the fork is open, the specification is met by more than one object; one can then only characterize, not single out. This is the order-theoretic form of a distinction the framework meets repeatedly — between an object and a property that characterizes it. It renames Knaster–Tarski; it is not a new theorem, and it is constructively choice-free.

R1 — the fork (fork_collapse_iff)
$\mathrm{lfp}\ f = \mathrm{gfp}\ f \iff \exists!\ x, f\ x = x$
A monotone self-map on a complete lattice has a unique fixed point if and only if its fixed-point interval [lfp f, gfp f] collapses to a point.
Witnesses: instance_pinnable_iff_fork_collapse (RequirementsGap.lean), instance_always_exists (MetaFork.lean).
Lean purity: choice-free. ✓

Lawvere's theorem, in curried form, says: if a map $e : A \to (A \to B)$ is point-surjective (every $g : A \to B$ equals $e\ a$ for some a), then every $f : B \to B$ has a fixed point. The fixed point it produces has a specific shape — it is $e\ a\ a$, the value of e at a diagonal point applied to that same point: self-application. Reading: the framework's selfApp names this operation abstractly; no declaration connects Lawvere's e to selfApp.

R2a — Lawvere's engine yields its fixed point in the shape e a a
Given a surjection $e : A \to (A \to B)$, every $f : B \to B$ has a fixed point of the form $e\ a\ a$ — e applied to a diagonal point, at that same point. The statement mentions no ZPSemilattice, no selfApp and no $\bot$: it is a claim about shape only.
Witness: lawvere_fixedpoint_selfApp (LawvereBridge.lean). Lean purity: choice-free. ✓

R2b — in the fork / AFA face, existence and uniqueness are class fields, not engine output
In the fork / AFA face, that $\bot$ of the ZPSemilattice is a fixed point of selfApp is the class field fixed_bot of AbstractSelfApp, and that it is the only one is the class field unique_fp. Both are commitments of the class; neither is supplied by Lawvere's engine. selfApp_pinnable packages the two fields as $\exists!\ x, \mathrm{selfApp}\ x = x$.

R2b — in the fork / AFA face, existence and uniqueness are class fields, not engine output

On a ZPSemilattice with two or more points, some map has a fixed point and not a unique one (existence_without_uniqueness). On a ZPSemilattice with a point other than ⊥, the engine's hypothesis fails outright: no surjection A → (A → L) exists (nontrivial_lattice_no_witness).

Witnesses: selfApp_pinnable, existence_without_uniqueness (LawvereBridge.lean); nontrivial_lattice_no_witness (Lawvere.lean). Lean purity: choice-free (selfApp_pinnable, existence_without_uniqueness). ✓

Section II: Locating the Obstruction, and the Crossing

To fill the self-application role with a Lawvere fixed point in a given face — to source the occupant from the engine rather than posit it — one needs Lawvere's hypothesis there: a weakly point-surjective map onto a function space (Theorem 1.1; the reflexive case is a map $A \to A^A$). Whether one exists is the entire question, and one condition rules it out cleanly: the presence of a fixed-point-free endomap.

R3-neg — in Set, no reflexive object on a carrier with a fixed-point-free endomap (Cantor); on a complete lattice every fixed-point-free endomap is non-monotone; on the empty set, which is not a complete lattice, the identity is fixed-point-free and monotone

A point-surjection $e : A \to (A \to B)$ would, by Lawvere, force every $f : B \to B$ to have a fixed point. For B two-valued, negation is fixed-point-free — a contradiction. This is Cantor's theorem, read as the contrapositive of Lawvere's own engine.

The obstruction witness — negation — has a structural property worth isolating: it is not monotone (it reverses the order False ≤ True). What separates the Set face from the monotone face is the class of maps, and what makes that restriction remove the obstruction is completeness: on a complete lattice every monotone self-map has a fixed point (Knaster–Tarski), so there every fixed-point-free endomap is non-monotone. Neither condition suffices alone: Bool with false ≤ true is a complete lattice on which negation is fixed-point-free, and Nat.succ is monotone and has no fixed point, on a carrier that is not a complete lattice. The empty set is not a complete lattice; its identity is monotone and fixed-point-free, and no map from it is onto its one-element function space (the LawvereBridge.lean § VII examples).

Witnesses: reflexive_object_refuted, not_monotone_not, and the anonymous examples after not_monotone_not in LawvereBridge.lean § VII, which back the Bool, Nat.succ and empty-set sentences. Lean purity: reflexive_object_refuted: no axioms; not_monotone_not: [propext]; the LawvereBridge.lean § VII examples are choice-free except the two that use Bool's complete-lattice instance (the instance example and the two-element-chain example), which carry Classical.choice from Mathlib's instance. The backing table's scoped Set statement also rests on the anonymous example after no_reflexive_object_bool in LawvereBridge.lean § VI and on fixedPointFree_of_nontrivial, and both carry Classical.choice. ✓

Absence of a fixed-point-free endomap is necessary for a reflexive object — it removes Lawvere's contradiction (his Corollary 1.2). Whether it is also sufficient depends on the face. In Set it is: a set with no fixed-point-free endomap has exactly one element and carries a surjection onto its endomaps, so it is a reflexive object. In the monotone face it is not: on the two-element chain false ≤ true every monotone self-map has a fixed point (Knaster–Tarski), yet no map from its two points onto its monotone self-maps is surjective. Both statements are anonymous Lean examples in ZeroParadox/Settheory/LawvereBridge.lean.

The second is not new: it is the first finite stage of Soto-Andrade and Varela's two-element example (1984, §2.5, pp. 15–16), which follows their remark that the converse of Lawvere's theorem fails for lattices with monotone maps; it is the case $Y = 2$ of Trimble's Cantor theorem for posets, that no poset map $Y \to 2^Y$ is surjective, and the case $P = 2$ of Bergman's Corollary 7 of arXiv:0801.0751 (Corollary 3.2 in the 2008 journal version), that no isotone map from a poset to its lattice of downsets is surjective (all three there for order-preserving maps; the Lean example allows any map, by counting two points against three monotone self-maps). A related question, about retracts rather than reflexive objects, is the suggestion of Soto-Andrade and Varela (1984, pp. 1 and 18) that every structure with the fixed point property is a retract of a reflexive domain. In any category a reflexive object is a retract of itself, so wherever every structure with the fixed point property is a reflexive object, each is also a retract of one. In Set the first of the two Lean examples above answers the retract question: a set with the fixed point property has exactly one element and carries a surjection onto its endomaps, so it is a reflexive object and trivially a retract of itself; being one-element, it is also in bijection with its own function space. The converse implication, from "every such structure is a retract of a reflexive object" to "every such structure is a reflexive object", is not shown here outside Set. ZP-R does not address the retract question in any other category; Björner (1985, Corollary, p. 100) shows that a finite poset which is a retract in Po of a reflexive domain is dismantlable and, citing finite non-dismantlable posets with the fixed point property, concludes that the suggestion has counterexamples in the category Po of partially ordered sets and monotone maps. Two categories are free of a restricted form of the obstruction, and only one of them is shown here to furnish a reflexive object. The reasons are not the same, and the difference is the point: the monotone regime is free of its monotone form literally — on a complete lattice, Knaster–Tarski bans a fixed-point-free monotone endomap, and the completeness hypothesis is load-bearing, since Nat.succ is a monotone endomap with no fixed point at all, and by Davis (1955, Theorem 2, p. 313) a lattice on which every increasing function has a fixpoint is complete — while the computable regime is free of it only up to eval — literally fixed-point-free total computable endomaps do exist.

R3-pos (monotone / domain) — no monotone obstruction on complete lattices; no reflexive object built here; no map from the two-element chain onto its monotone self-maps

On a complete lattice every monotone self-map has least and greatest fixed points (Knaster–Tarski), so fixed-point-free monotone endomaps are structurally banned — the monotone obstruction is absent. Existence is thereby guaranteed; uniqueness is not (the interval [lfp, gfp] need not collapse).

No reflexive object is built in this face here: obtaining one in the order / domain world is the business of Scott's D_∞ (Scott-continuous maps on a directed-complete order, by inverse limits) — a construction we do not carry out and do not need, since the computability face below already realizes the crossing. What this face provides here is the fork.

Witnesses: monotone_regime_derives_pinned (LawvereBridge.lean), for the fork; the last anonymous example of LawvereBridge.lean § VII, for the two-element chain (the first finite stage of Soto-Andrade and Varela's two-element example, 1984, §2.5, pp. 15–16; Trimble, "No surjective map $Y \to 2^Y$"; Bergman, Corollary 7 of arXiv:0801.0751, Corollary 3.2 in the journal). Lean purity: monotone_regime_derives_pinned choice-free; in the two-element-chain example the no-surjection half is [propext], and Classical.choice enters through the Knaster–Tarski half (OrderHom.lfp on Mathlib's complete-lattice instance on Bool). ✓

R3-pos (computability) — the reflexive object is realized: the crossing
The category of numbered sets carries a canonical reflexive object: the universal machine. The evaluation map eval is point-surjective onto the partial computable functions — every partial computable function has an index (Rogers 1967). And Rogers' fixed-point theorem gives every total computable endomap of codes a fixed point up to eval — two codes computing the same function.
That is weaker than "no fixed-point-free endomap", and the strong form is false: fun c => Code.pair c c is total and computable and returns its own input for no c (code_pair_diag_fixedPointFree, ZeroParadox/Category/Lawvere.lean). The obstruction is absent at the level of eval, which is where the reflexive object lives — and the reason is that the effective category admits fewer maps, not that eval lands in a different type: no computable self-map on codes is eval-fixed-point-free, so the diagonal the Set refutation runs has no computable representative.
So the reflexive structure is present and the self-referential fixed point exists there: in Rogers' form, for computable g : Code → Code a code c with eval (g c) = eval c; and in Kleene's two-place form (the second recursion theorem), for partial computable F a code c with eval c = F c.
Witnesses: eval_point_surjective, computable_fixedpoint_up_to_eval, selfref_fixedpoint_exists_computable (ComputableCrossing.lean). Lean purity: choice-carrying (Mathlib computability). ✓

That the recursion theorem, in Rogers' form (a total computable h has an n with $\varphi_{h(n)} = \varphi_n$), is an instance of Lawvere's scheme is standard: the derivation is Yanofsky (2003) Theorem 5 (p. 18 of arXiv:math/0305282v1), within his unified treatment. Bauer (2017) proves a version of Lawvere's theorem for multi-valued maps, in synthetic computability, and derives the Kleene–Rogers theorem from it (Theorem 5.2, Corollary 5.3). The Lean here checks the classical fixed point and the point-surjectivity of eval; the Lawvere-form derivation is cited, and this is the citation the later sections point to. Lawvere (1969) supplies the engine and raises the recursive case as an open question rather than deriving it (his §2, p. 9 of the 2006 TAC reprint); the reflexive structure of the computable category is the subject of the Turing-category / partial-combinatory-algebra literature (Cockett–Hofstra 2008; Longley 1994). Collecting the three faces:

Face	Reflexive object	Mechanism	Status
Set	refuted on every carrier except a one-element one, the empty set included (a one-element set is reflexive)	a fixed-point-free map blocks it (Cantor): on two or more points, the map the proof of fixedPointFree_of_nontrivial constructs, one point b_0 sent to a second point b_1 and every other point to b_0 (a swap when there are exactly two points, negation in the two-valued case); on the empty set the identity. On a complete lattice the blocking map is non-monotone; on the empty set, not a complete lattice, the identity is monotone	the wall
Monotone / domain	not furnished here (route: Scott D_∞, unbuilt)	monotone obstruction absent on complete lattices (Knaster–Tarski); existence via the fork, uniqueness = fork collapse	fork story; no Lawvere realization built here

Face	Reflexive object	Mechanism	Status
Computability	realized	universal machine point-surjective; Rogers / Kleene	crossed

R4 — in the computability face, the self-application role is filled by a Lawvere fixed point (cited)

In the computability face the self-application fixed-point role is filled by a Kleene code, and that fixed point is a Lawvere fixed point (cited). The code is a term of another type than ⊥ of the ZPSemilattice. Existence is machine-checked in Rogers' form, eval (g c) = eval c for computable g : Code → Code (computable_fixedpoint_up_to_eval), which is the form Yanofsky and Bauer state, and in the two-place form, eval c = F c for partial computable F (selfref_fixedpoint_exists_computable). That it is an instance of Lawvere's theorem is cited (Section II). Since Lawvere's theorem holds in any category with finite products (his §2), nothing obliges the realization to be in Set or in a Scott domain.

What is proved about the framework's two structural regimes is that a self-loop rules out well-foundedness (mu_nu_branch_exclusion).

Witnesses: the computability fixed point is proved in R3-pos above (computable_fixedpoint_up_to_eval, selfref_fixedpoint_exists_computable, ComputableCrossing.lean; choice-carrying); mu_nu_branch_exclusion (LawvereBridge.lean; choice-free). ✓

Section III: Scope and Fences

The result is deliberately narrow, and its limits are structural — part of the statement rather than caveats to it. The sharpest way to state them is to track which face carries which property.

F1 — what is measured, per face

• Set / fork face. On a ZPSemilattice L with a point a ≠ ⊥, no surjection A → (A → L) exists (nontrivial_lattice_no_witness : a ≠ ⊥ → ¬HasLawvereWitness L). That ⊥ of the ZPSemilattice is the unique fixed point of selfApp follows from the class fields fixed_bot and unique_fp (selfApp_pinnable); every Quine atom equals it (t_exec, from the AFAStructure class fields).

• Computability face. For computable g : Code → Code, some code c has eval (g c) = eval c (computable_fixedpoint_up_to_eval). For partial computable F, the codes c with eval c = F c form an infinite set (fixed_points_infinite). This does not exclude a selfApp on Code with a literally unique fixed point.

• Monotone / domain face. No occupant of the self-application role is claimed here; what is measured for this face is in Section II (R3-pos).

F2 — representation is not transfer; the self-application role carries its face

The crossing is face-local. That a Kleene code fills the self-application role as a Lawvere fixed point in the computability face does not make a Lawvere fixed point available in Set on the same carrier, which has two or more elements and where the same construction remains a contradiction (Cantor; reflexive_object_refuted). What does not transfer is the Lawvere realization, not definability: the least fixed point of a monotone map on a powerset lattice, for instance, is definable in ZFC.

Reading: the framework treats the self-application fixed point as a role carried with its face, characterized rather than constructed as one global object. Its choice of a non-well-founded foundation (ZFC^{-} + AFA: Foundation replaced by AFA, Choice kept; Aczel 1988, p. xviii; Barwise & Moss 1996, p. 72, whose ZFC^{-} on p. 28 also has urelements) is a separate commitment, which this face-locality does not by itself require. A theorem stated for every suitable category can be witnessed in whichever category has Lawvere's hypothesis; the occupant it produces is defined only relative to that category.

Lawvere's theorem, face by face.

The fences F1 and F2 are not hedging: they mark the face each result belongs to. Each per-face result — the Set wall, the fork, the computability crossing — is a full theorem within its own face. Lawvere's theorem is the translation key between the faces formalized here, not a mechanism every face instantiates: each face is read against its one arrow, a point-surjection onto a function space implies that every endomap has a fixed point, and the faces differ in which way, if either, they run it. A wall face runs it by the contrapositive, which is Lawvere's own reading ("The famed ‘diagonal argument’ is of course just the contrapositive of our theorem," p. 5 of the 2006 TAC reprint): a fixed-point-free endomap of an object Y rules out every point-surjection $A \to Y^A$ (his Corollary 1.2), and Cantor is the two-valued case. In Set every carrier except a one-element one has such an endomap, the empty set included, where the identity is fixed-point-free because the set has no elements; a one-element set has none and is the trivial forward case (Section II). The computability face runs it forward, up to eval: eval is point-surjective onto the partial computable functions, and every total computable endomap of codes has a fixed point up to eval (the R3-pos computability box, Section II); that this is a Lawvere instance is cited (Yanofsky 2003; Bauer 2017). In the fork / AFA face the self-application fixed point, $\bot$ of the ZPSemilattice, is supplied by the class fields fixed_bot and unique_fp (selfApp_pinnable), not by the engine, whose hypothesis fails on every ZPSemilattice with an element other than $\bot$ (nontrivial_lattice_no_witness). On every ZPSemilattice with an element other than $\bot$, that face is linked to the computability face through the die's Quine-atom role, whose occupancy is settled chart by chart: in the code chart no single code occupies it under the eval step (evalStepMotion_not_selfOnlyAt), and a class of codes up to equal eval does, for computable eval-respecting maps (codes_vs_classes). KleeneStructure supplies a code meeting a periodicity condition (one that constant codes also meet) and equates it with nothing. The reading it was built for, that this code is the computational witness of $\bot$, splits: its identity half is retired, and its running half, that $\bot$ executes, is the occurrence commitment (ZeroParadox/Computability/Kleene.lean §II). On the one-element ZPSemilattice, as for the one-element set, the forward case holds trivially. The monotone face, in posets with monotone maps, runs it in neither direction on the two-element chain: on a complete lattice no monotone endomap is fixed-point-free (Knaster–Tarski), so the contrapositive has no monotone input, and on the two-element chain no map onto the monotone endomaps is surjective (Section II). In chain-complete posets, Soto-Andrade and Varela (1984, pp. 15–16) construct a larger poset isomorphic to its maps into the two-element chain; ZP-R reads that construction with continuous maps, their category CCPo (§2.2, p. 9), since with all monotone maps Trimble's theorem excludes it. So the domain half differs: there the two-element chain is labelled by a larger object, not by itself. No occupant is identified with another: an equation between an element of a ZPSemilattice and a code is ill-typed (MC-1). ZP-R conjectures nothing global.

Relation to Prior Work

The mathematics used here is classical and is invoked, not extended. Lawvere's fixed-point theorem and the unification of the diagonal arguments: Lawvere, "Diagonal Arguments and Cartesian Closed Categories," LNM 92 (1969) 134–145, cited by page in its reprint, Reprints in Theory and Applications of Categories 15 (2006); Yanofsky, Bull. Symbolic Logic 9(3) (2003), cited by page in arXiv:math/0305282v1. Lattice fixed points: Knaster, "Un théorème sur les fonctions d'ensembles," Ann. Soc. Polon. Math. 6 (1928) 133–134, existence of a fixed point for a monotone set function, the power-set case (Lemma (L), p. 133), reporting results obtained jointly with Tarski; Tarski, "A lattice-theoretical fixpoint theorem and its applications," Pacific J. Math. 5(2) (1955) 285–309; Davis, "A characterization of complete lattices," Pacific J. Math. 5(2) (1955) 311–319. The computability fixed points: Rogers, Theory of Recursive Functions and Effective Computability (1967); Kleene's second recursion theorem; a version of Lawvere's theorem for multi-valued maps, in synthetic computability, from which the recursion theorem follows: Bauer, "On fixed-point theorems in synthetic computability," Tbilisi Math. J. 10(3) (2017). The reflexive structure of the computable category: Cockett & Hofstra, "Introduction to Turing categories," Annals of Pure and Applied Logic 156 (2008) 183–209; Longley, Realizability Toposes and Language Semantics, PhD thesis, University of Edinburgh, 1994 (issued as LFCS report ECS-LFCS-95-332). Fixed points and reflexive domains: Soto-Andrade and Varela, "Self-reference and fixed points: a discussion and an extension of Lawvere's theorem," Acta Applicandae Mathematicae 2(1) (1984) 1–19, doi:10.1007/BF01405490, which notes that the converse of Lawvere's theorem fails for lattices with monotone maps (§2.5, p. 15) and conjectures that every structure with the fixed point property is a retract of a higher reflexive domain from which it inherits that property (pp. 1 and 18), proving it for chain-complete posets with continuous morphisms; Björner, "Reflexive domains and fixed points," Acta Applicandae Mathematicae 4(1) (1985) 99–100, doi:10.1007/BF02293493, which shows that if R is a retract of a reflexive domain then R^R has the fixed point property (Proposition, p. 99) and derives counterexamples to the conjecture in Po (Corollary, p. 100); Trimble, "Cantor's theorem for posets," nLab personal web (ncatlab.org/toddtrimble), revised 5 October 2016, which proves that no poset map $Y \to 2^Y$ is surjective, so Knaster–Tarski is not a corollary of Lawvere's theorem through a surjection onto a function space in posets; and Bergman, "On lattices and their ideal lattices, and posets and their ideal posets," Tbilisi Math. J. 1 (2008) 89–103 (arXiv:0801.0751), whose Corollary 7 in the arXiv numbering (Corollary 3.2 in the journal) states that no isotone map from a subset of a poset P to the lattice of all downsets of P is surjective. Non-well-founded sets characterized by unique solutions and by corecursion: Aczel, Non-Well-Founded Sets (1988); Barwise & Moss, Vicious Circles (1996), where AFA is stated as the Solution Lemma (p. 72) and the Corecursion Theorem (p. 250) is given as their version of Aczel's Special Final Coalgebra Theorem (their p. 264). Both books also construct: Aczel builds a model of $\mathrm{ZFC}^- + \mathrm{AFA}$ (Theorem 3.10, p. 37) and Barwise and Moss a model of ZFA (Chapter 9, pp. 103–115 in their index). Reading the self-referential object as characterized rather than constructed is this document's reading, not theirs.

Against this, the contribution of ZP-R is not a theorem but a placement: the per-face placement of the framework's self-application role relative to the Lawvere fixed point (its Lawvere realization refuted in Set on every carrier except a one-element one, the monotone obstruction absent on complete lattices in the monotone / domain face, with no reflexive object built there, the realization achieved in the computability face), and the scope result of Section III. That Kleene's recursion theorem is a Lawvere instance is due to the sources above, not to this document, and so is the obstruction the placement uses (Lawvere's Corollary 1.2). The retract question, Soto-Andrade and Varela's suggestion, sits beside the placement; ZP-R answers it only

in Set (Section II).

What Remains Open

Open items
ZP-R holds no global conjecture: how the faces formalized here relate to Lawvere's theorem is collected in Section III, in the box "Lawvere's theorem, face by face".
The domain-face route — a Scott D_∞ reflexive object — is not carried out; the computability face suffices for the existence claim, so it is not needed, but it would be a second realization.
Location and uniqueness (that selfApp's fixed point is ⊥ of the ZPSemilattice, and unique) are established — in the fork / AFA face, from class fields (fixed_bot; unique_fp) — not open.

Machine-Checked Backing

Each result box (R1–R4) and each F1 property is backed by a compiling Lean 4 (+ Mathlib) theorem, sorry-free, except the one marked cited: that the computability fixed point is an instance of Lawvere's theorem (Section II). F2's first mathematical sentence rests on reflexive_object_refuted (the R3-neg row), and its second, on the least fixed point of a monotone map, on Knaster–Tarski (Knaster 1928, existence in the power-set case; Tarski 1955; both cited); sentences marked Reading are interpretation and have no row. The fork schema (R1, monotone_regime_derives_pinned) is choice-free; the anonymous example after no_reflexive_object_bool in LawvereBridge.lean § VI, fixedPointFree_of_nontrivial and nontrivial_lattice_no_witness (both in ZeroParadox/Category/Lawvere.lean), and the examples using Bool's complete-lattice instance carry Classical.choice; the computability material inherits Classical.choice from Mathlib's computability library (Kleene's and Rogers' theorems use it), which is disclosed and not claimed otherwise.

Claim	Lean witness (file)
R1 (the fork)	instance_pinnable_iff_fork_collapse (ZeroParadox/Settheory/RequirementsGap.lean); instance_always_exists (ZeroParadox/Settheory/MetaFork.lean)
R2a / R2b (the shape e a a; existence and uniqueness as class fields)	lawvere_fixedpoint_selfApp, selfApp_pinnable, existence_without_uniqueness (ZeroParadox/Settheory/LawvereBridge.lean); nontrivial_lattice_no_witness (ZeroParadox/Category/Lawvere.lean)
R3-neg (Set, every carrier except a one-element one: refuted; every fixed-point-free endomap of a complete lattice is non-monotone; on the empty set, not a complete lattice, the identity is fixed-point-free and monotone)	reflexive_object_refuted, not_monotone_not, the anonymous example after no_reflexive_object_bool (§ VI) and the anonymous examples after not_monotone_not (§ VII) (ZeroParadox/Settheory/LawvereBridge.lean); fixedPointFree_of_nontrivial, which supplies the fixed-point-free map on two or more points (ZeroParadox/Category/Lawvere.lean)

Claim	Lean witness (file)
R3-pos monotone	monotone_regime_derives_pinned and the last anonymous example of § VII, the two-element chain (ZeroParadox/Settheory/LawvereBridge.lean)
R3-pos computability (the crossing)	eval_point_surjective, computable_fixedpoint_up_to_eval, selfref_fixedpoint_exists_computable (ZeroParadox/Computability/ComputableCrossing.lean)
R4 (a self-loop rules out well-foundedness)	mu_nu_branch_exclusion (ZeroParadox/Settheory/LawvereBridge.lean)
F1 location (fixed point = ⊥ of the ZPSemilattice)	t_exec, t_exec_iff (ZeroParadox/Settheory/SetTheoryAFA.lean)
F1 uniqueness (⊥ of the ZPSemilattice the only fixed point)	quine_atom_unique (ZeroParadox/Settheory/SetTheoryAFA.lean); AbstractSelfApp.unique_fp (ZeroParadox/Computability/SelfApp.lean)
F1 computability face (infinitely many eval-fixed points of F)	fixed_points_infinite (ZeroParadox/Computability/Kleene.lean)
F1 Set / fork face (no surjection onto L→L when a ≠ ⊥)	nontrivial_lattice_no_witness (ZeroParadox/Category/Lawvere.lean)

Axiom Purity
Order / fork spine: choice-free. instance_pinnable_iff_fork_collapse, instance_always_exists, monotone_regime_derives_pinned: [propext, Quot.sound]; not_monotone_not: [propext]; lawvere_fixedpoint_selfApp, selfApp_pinnable, existence_without_uniqueness, reflexive_object_refuted, mu_nu_branch_exclusion: no axioms.
Set-theoretic location / uniqueness (SetTheoryAFA: t_exec, quine_atom_unique): choice-free — derived from the ZPSemilattice / AFAStructure class fields alone.
nontrivial_lattice_no_witness and fixedPointFree_of_nontrivial (Lawvere.lean) carry [propext, Classical.choice, Quot.sound], from the classical construction of a fixed-point-free map.
Computability face (ComputableCrossing, Kleene): [propext, Classical.choice, Quot.sound] — Classical.choice inherited from Mathlib's classical recursion theory (Kleene / Rogers), disclosed and not claimed otherwise.
Core choice-free, realization choice-carrying — the framework's standing pattern.
The footprints listed here are copies; to re-measure one, run #print axioms on the named declaration.

End of ZP-R | A Cross-Category Account of the Self-Referential Fixed Point | the fork (choice-free) | Set, every carrier except a one-element one: refuted (Cantor) | monotone / domain: fork, no reflexive object built, no map from the two-element chain onto its monotone self-maps (Scott D_∞ unbuilt) | computability: crossed (Rogers / Kleene) | F1: per face, what is measured | F2: the self-application role carries its face | Lawvere the translation key between the faces (Section III) | no global conjecture.