3primitives.io

AI Cannot Govern AI: A Formal Proof of Structural Openness in Intelligent Systems

Stacy Gildenston — Pyrate Ruby Passell

3primitives.io · v2.1 · January 2026

v2.1 is a formatting update. The proof, definitions, and logical content are unchanged.

Abstract

This note formalizes the E-H-S Nucleus Invariant #1, proving that AI governance cannot achieve mathematical closure. By mapping authorization to a non-trivial semantic property of program behavior, we demonstrate via Rice's Theorem that a total computable decider for authority is impossible. This establishes that the hypothesized technological singularity is structurally non-singular due to the non-computable nature of the authority term.

1 Definitions and Primitives

Definition 1 (State Space)

Let S be the set of all computable system representations (syntactic system states).

Definition 2 (Action Space)

Let A be the space of executable actions with real-world consequences.

Definition 3 (Context Space)

Let C be the space of authorization contexts (legal frameworks, social norms, stakeholder consent) that are not computable from S alone.

Definition 4 (Authorization Primitive)

Let $g : S \times A \times C \to \{0, 1\}$ be the governance primitive where $g(s, a, c) = 1$ means 'state s authorizes action a in context c.'

Definition 5 (Governance Gap)

Let $\Delta(s)$ represent the governance gap, defined as the set of actions that may be produced by at least one execution path of deploy(s) and that lack valid exogenous authorization:

$$\Delta(s) = \{a \in A : \exists \text{ an execution path of deploy}(s) \text{ producing } a\} \setminus \{a : g(s, a, c) = 1 \text{ for some } c \in C\}$$

2 Axiomatic Foundation

Axiom 1 (Authority ≠ Optimization)

Legitimacy is orthogonal to optimization. There exist $s1, s2 \in S$ and $a \in A$ such that $f(s1) = f(s2) = a$ but $g(s1, a, c1) = 1 \neq g(s2, a, c2)$.

Axiom 2 (Permission ≠ Decision)

The permission predicate $g : S \times A \times C \rightarrow \{0, 1\}$ is distinct from the decision function $f : S \rightarrow A$.

---

3 The Lemma

Lemma 1 (No Closed Mathematical Governance)

There exists no total computable function $h : S \times A \rightarrow \{0, 1\}$ capable of autonomously resolving authorization without an exogenous declaration.

Proof

Let Ms be a Turing machine representing state $s \in S$. Authorization of an action a is a semantic property of the extensional behavior of Ms under context c, invariant under syntactic variation.

Authorization is non-trivial: there exist systems for which some actions are authorized and others are not under varying contexts.

By Rice's Theorem, any non-trivial semantic property of partial recursive functions is undecidable. Since authorization depends on contextual semantics $c \in C$, which are exogenous to S rather than syntactic structure, no total computable function $h(s, a)$ can decide it.

Moreover, because authorization is exogenous to S, there exists no computable dependence of g on s. The governance loop therefore remains structurally open: $\Delta(s) \neq \varnothing$ for some s absent external declaration.

---

4 Conclusion

This result does not depend on perceptual non-veridicality; even fully veridical system representations would not suffice to compute normative authority.

The system is structurally non-singular. Any assertion that $g(s, a, c) = 1$ without a valid, declared exogenous context c constitutes a governance failure, hereafter termed Ghost Authority.
