The Incompleteness of Logic: A Recursion-Theoretic Generalization of Gödel's Theorem

Author: Zhi Systems

Abstract

This paper proves that the class of all deductive logics—understood as recursively defined, truth-preserving sets of statements—is not recursively enumerable. By generalizing Gödel's incompleteness result, we show that the very space of logical systems cannot be exhaustively captured by any recursive procedure. The argument proceeds in two steps: first, by proving via diagonalization that the set of total recursive functions is not recursively enumerable; and second, by constructing an injective embedding of total recursive functions into distinct truth-preserving logics.

---

1. Definitions and Framework

Definition 1.1 (Logic): A logic is a pair (S, Φ), where:

- S is a finite set of base statements (axioms)
- Φ is a total recursive function from strings to strings

The logic-extension L generated by (S, Φ) is the smallest set K such that:

1. $S \subseteq K$
2. For every x in K, $\Phi(x) \in K$

Definition 1.2 (Truth-Preserving): A function Φ is truth-preserving if whenever x is a true statement, $\Phi(x)$ is also a true statement (under the intended interpretation).

Definition 1.3 (Recursive Logic): A logic (S, Φ) is recursive if Φ is a total recursive function and S is finite.

Throughout this paper, we work within the framework of classical first-order arithmetic with standard model $\mathbb{N}$ as our background theory of truth.

---

2. The Non-Enumerability of Total Recursive Functions

Theorem 2.1: The set of all total recursive functions is not recursively enumerable.

Proof: Suppose, toward contradiction, that there exists a recursive enumeration $\{\Phi_0, \Phi_1, \Phi_2, ...\}$ of all total recursive functions from ℕ to ℕ.

Define a new function F: ℕ → ℕ by:

$F(n) = \Phi_n(n) + 1$

Since each Φ_n is total and recursive, and addition is recursive, F is itself a total recursive function.

However, F differs from every function in the enumeration: for each n, we have $F(n) = \Phi_n(n) + 1 \neq \Phi_n(n)$, so $F \neq \Phi_n$.

Therefore F is a total recursive function not in our enumeration, contradicting the assumption that the enumeration was complete.

Thus, the set of total recursive functions is not recursively enumerable. ■

---

3. Injective Embedding of Recursions into Logics

Not every recursive function directly generates statements. However, we can construct an injective mapping from total recursive functions to truth-preserving logics that preserves distinctness.

Construction 3.1 (Canonical Logic Encoding): For each total recursive function f: ℕ → ℕ, we construct a corresponding logic L_f = (S_f, Φ_f) as follows:

Base axioms S_f:

- "$f(0) = k_0$" where $k_0 = f(0)$

Recursive operator Φ_f:

- Input: "$f(n) = k_n$"
- Output: "$f(n) = k_n \land f(n+1) = k_{n+1}$" where $k_{n+1} = f(n+1)$

The logic-extension generated by L_f contains all statements of the form:

- "$f(0) = k_0$"
- "$f(0) = k_0 \land f(1) = k_1$"
- "$f(0) = k_0 \land f(1) = k_1 \land f(2) = k_2$"
- etc.

Each statement encodes the complete computational history of f up to some point n.

Lemma 3.2: The mapping $f \mapsto L\_f$ is injective.

Proof: Suppose $f \neq g$. Then there exists some n_0 such that $f(n_0) \neq g(n_0)$.

The logic L_f contains the statement "$f(0) = f(0) \wedge \ldots \wedge f(n_0) = f(n_0)$", which encodes the value $f(n_0)$.

The logic L_g contains the statement "$g(0) = g(0) \wedge \ldots \wedge g(n_0) = g(n_0)$", which encodes the value $g(n_0)$.

Since $f(n_0) \neq g(n_0)$, these statements are distinct. Therefore, the sets L_f and L_g differ, so $L\_f \neq L\_g$. ■

Lemma 3.3: Each logic L_f is truth-preserving under the standard interpretation of arithmetic.

Proof: Each statement in L_f is a conjunction of arithmetic equations of the form "$f(i) = k_i$" where k_i is the actual value computed by f at i. Since these equations state true facts about f's behavior, and conjunction preserves truth, every statement in L_f is true. ■

---

4. Main Result

Theorem 4.1: The class of all recursive, truth-preserving logics is not recursively enumerable.

Proof: By Construction 3.1 and Lemma 3.2, there exists an injective function from the set of total recursive functions to the set of recursive truth-preserving logics.

By Theorem 2.1, the set of total recursive functions is not recursively enumerable.

Since there is an injection from a non-r.e. set into the set of recursive logics, the set of recursive logics cannot be recursively enumerable (if it were, we could pull back the enumeration to enumerate total recursive functions).
■

---

5. Formal Truth as a Relational Concept

Traditionally, philosophers have sought to define "formal truth" as an intrinsic syntactic property—something about the logical form of a statement that makes it true independent of interpretation.

This paper suggests a different conception:

Definition 5.1 (Formal Truth - Relational): A statement S is formally true if and only if there exists a truth-preserving recursive logic L such that $S \in L$.

Formally: **FormallyTrue(S) ⇔ ∃L [L is truth-preserving ∧ L is recursive ∧ S ∈ L]**

Corollary 5.2: The predicate "formally true" is not recursively enumerable.

Proof: If "formally true" were r.e., we could enumerate all formally true statements. But each formally true statement belongs to some recursive logic, and different logics can be distinguished by their member statements. This would allow us to enumerate the recursive logics themselves, contradicting Theorem 4.1. ■

This reframes logical truth not as an intrinsic property but as a relationship between a statement and a truth-preserving recursive system. No algorithm can generate all such systems.

---

6. Philosophical Consequences

6.1 The Historical Confusion About Logical Form

Historically, certain expressions ("and," "or," "not," "all," "some") have been classified as "logical constants" while others ("knows," "believes," "possibly") have not. This distinction has often been presented as tracking something deep about meaning or form.

This paper suggests a deflationary view: expressions count as "logical" when we have identified truth-preserving recursive rules governing them. The apparent special status of classical logical constants reflects merely the fact that we discovered workable recursive systems for them early (via truth tables, natural deduction, etc.).

There is no intrinsic syntactic mark of "logicality"—only the pragmatic fact that some expressions admit readily discoverable recursive axiomatizations.

6.2 The Limit of Formal Knowledge

Gödel's original incompleteness theorem showed that no single recursive axiom system can capture all arithmetic truths. This result generalizes that insight: no recursive *meta*-procedure can generate all recursive logical systems.

Just as Gödel placed a limit on what any particular formal system can prove, this result places a limit on what the space of formal systems itself can be: it cannot be recursively surveyed.

6.3 Dissolution of the Formality Problem

The question "What makes a truth formally true?" has been a central problem in philosophy of logic. This paper suggests the question was misconceived.

Formal truth is not a monadic property (✓ or ✗) but a relation to a recursive proof system. Since the space of such systems is not recursively enumerable, there is no decision procedure for formality itself.

This dissolves rather than solves the traditional problem: there is no unified essence of "formal truth" to be discovered, only an infinite, non-enumerable plurality of recursive truth-preserving systems.

---

7. Conclusion

We have proven:

1. The set of total recursive functions is not recursively enumerable (Theorem 2.1)
2. Total recursive functions can be injectively embedded into truth-preserving recursive logics (Construction 3.1, Lemma 3.2)
3. Therefore, the class of recursive logics is not recursively enumerable (Theorem 4.1)
4. Consequently, "formal truth" (understood relationally) is not a recursively enumerable predicate (Corollary 5.2)

This establishes a recursion-theoretic boundary to formal knowledge that is more fundamental than Gödel's original result. While Gödel showed that no single logic is complete, we have shown that the space of logics itself defies algorithmic completion.

The dream of a unified, recursive characterization of logical truth—a master algorithm that could generate or recognize all formal systems—is provably unrealizable.

References

Church, A. (1936). An unsolvable problem of elementary number theory. *American Journal of Mathematics*, 58(2), 345–363.

Gödel, K. (1931). Über formal unentscheidbare Sätze der Principia Mathematica und verwandter Systeme I. *Monatshefte für Mathematik und Physik*, 38(1), 173–198.

Kleene, S. C. (1952). *Introduction to Metamathematics*. North-Holland.

Rogers, H. (1967). *Theory of Recursive Functions and Effective Computability*. McGraw-Hill.

Turing, A. M. (1936). On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 2(42), 230–265.