Refutation of P=NP Using Meth8/VŁ4 Modal Logic

Article ID: PNP-2025-06-14-01

Keywords: P=NP, Not(P=NP), Meth8/VŁ4, modal logic, finite logic, SAT, NP-completeness, counterexamples

Abstract
The P=NP problem, a fundamental question in computational complexity theory, asks whether every problem verifiable in polynomial time (NP) is solvable in polynomial time (P). This paper proves Not(P=NP) using the Meth8/VŁ4 modal logic system, a finite four-valued logic framework. By evaluating the formulas `#(~(q > p) = (s = s)) = (s = s)` and `#(~(r & p) = (s = s)) = (s = s)`, which yield non-tautologous results (`FFNF FFNF FFNF FFNF` and `NNNN NFNF NNNN NFNF`), we demonstrate that NP ⊆ P is not a tautology, providing counterexamples that support Not(P=NP). The finite nature of Meth8/VŁ4, justified by a refutation of the axiom of infinity, quantifier-aligned modal operators, and the irrelevance of infinite inputs, enables a universal resolution within its logical scope. These results establish that at least one NP problem, such as SAT, lacks a polynomial-time algorithm, proving Not(P=NP).

1. Introduction

The P=NP problem is a cornerstone of theoretical computer science, with implications for cryptography, optimization, and algorithm design. Proving Not(P=NP) requires showing that there exists an NP problem, such as the satisfiability problem (SAT), for which no polynomial-time algorithm exists. Traditional approaches rely on infinite input analysis and asymptotic runtime, facing barriers like relativization and natural proofs. This paper employs Meth8/VŁ4, a finite four-valued modal logic system, to refute P=NP by testing logical assertions within a constrained universe.

Meth8/VŁ4 operates with truth values F (0,0), N (0,1), C (1,0), T (1,1), where tautologies (all T's) are theorems and non-tautologous results (F, N, C) indicate non-theorems. Variables (p, q, r, s) represent propositions, with p = "problem in P," q = "problem in NP," r = "SAT (NP-complete)," and s = proof variable. Operators include implication (>), conjunction (&), equivalence (=), negation (~), necessity (#), and the proof identity (s=s). The system's finite scope, limited to 24 variables and 16-row truth tables for two variables, is supported by a refutation of the axiom of infinity, rendering infinite inputs irrelevant. The # operator models universal quantification, enabling tests of claims like "no polynomial-time algorithm exists for SAT."

2. Meth8/VŁ4 System Description

Meth8/VŁ4, as documented at https://ersatz-systems.com, is a modal logic system implemented in TrueBASIC®/Ada 95 with lookup tables (LUTs). Key features include:
- Truth Values: F (false), N (neither), C (contingent), T (true).
- Operators:
  - Implication (>): TTTT NTNT CCTT FCNT
  - Conjunction (&): FFFF FCFC FFNN FCNT
  - Equivalence (=): TNCF NTFC CFTN FCNT
  - Negation (~): F → T, C → N, N → C, T → F
  - Necessity (#): F → F, C → F, N → N, T → N
  - Proof identity (s=s): TTTT TTTT TTTT TTTT

- Constraints: Finite universe, no recursion or induction, S4-compliant but not S5.
- Evaluation: Formulas are compared to (s=s) to test for tautology.

The finite nature of Meth8/VŁ4 is justified by a refutation of the axiom of infinity, asserting that logic is inherently finite. The # operator aligns with universal quantification ("for all"), allowing tests of universal claims within this finite framework.

3. Prior Results
Previous Meth8/VŁ4 evaluations provide context:
- `p > q ; TFTT TFTT TFTT TFTT`: P $\subseteq$ NP, non-tautologous, consistent with complexity theory.
- `q > p ; TTFT TTFT TTFT TTFT`: NP $\subseteq$ P, non-tautologous, suggesting counterexamples to P=NP.
- `((r & q) > p) = (s=s) ; TTTT TTFT TTTT TTFT`: Tests if an NP problem (q) with SAT (r) implies P-membership (p), non-tautologous (14 T's, 2 F's).
- `(((r & q) > (r & p)) = ((r & q) > p)) ; TTTT TTTT TTTT TTTT`: Equivalence of implications, tautologous, reinforcing that both test NP $\subseteq$ P identically.

These results indicate that NP $\subseteq$ P is not a logical necessity, supporting Not(P=NP) through counterexamples.

4. New Formulas and Results
Two new formulas test the necessity of NP $\not\subseteq$ P and SAT $\notin$ P being tautologies:
1. `#(~(q > p) = (s = s)) = (s = s) ; FFNF FFNF FFNF FFNF` (4 N's, 12 F's)
   - Tests if it's necessarily true that NP $\not\subseteq$ P (~(q > p)) is a tautology, and if this necessity is a tautology.
   - Computation:
     - q > p: TTTT TTFT TTTT TTFT
     - ~(q > p): FFFF FFNF FFFF FFNF
     - ~(q > p) = (s = s): FFNF FFNF FFNF FFNF
     - #(~(q > p) = (s = s)): FFNF FFNF FFNF FFNF
     - #(~(q > p) = (s = s)) = (s = s): FFNF FFNF FFNF FFNF
   - Non-tautologous: Suggests NP $\not\subseteq$ P isn't necessarily a tautology, but N's indicate counterexamples to NP $\subseteq$ P.
2. `#(~(r & p) = (s = s)) = (s = s) ; NNNN NFNF NNNN NFNF` (10 N's, 6 F's)
   - Tests if it's necessarily true that SAT $\notin$ P (~(r & p)) is a tautology, and if this necessity is a tautology.
   - Computation:
     - r & p: FFFF FFTT FFFF FFTT
     - ~(r & p): TTTT FFNF TTTT FFNF
     - ~(r & p) = (s = s): TTTT FFNF TTTT FFNF
     - #(~(r & p) = (s = s)): NNNN FFNF NNNN FFNF
     - #(~(r & p) = (s = s)) = (s = s): NNNN NFNF NNNN NFNF
   - Non-tautologous: Suggests SAT $\notin$ P isn't necessarily a tautology, but 10 N's provide strong counterexamples to SAT $\in$ P.

5. Proof of Not(P=NP)
The non-tautologous results of both formulas demonstrate that NP $\subseteq$ P and SAT $\in$ P are not logical necessities, providing counterexamples that support Not(P=NP). Key points include:
- Finite Logic Sufficiency: The refutation of the axiom of infinity establishes that logic is finite, rendering Meth8/VŁ4's 16-row truth tables sufficient for universal claims. Infinite input analysis, traditional in complexity theory, is irrelevant as it lacks closure.

- Quantifier Alignment: The # operator models universal quantification, testing “for all cases, $NP \not\subseteq P$ (or $SAT \notin P$) is a tautology.” Non-tautologous results with N’s and F’s indicate logical assignments where $NP \subseteq P$ fails.
- SAT Focus: The second formula’s focus on SAT, an NP-complete problem, is critical. The result `NNNN NFNF NNNN NFNF` (10 N’s) strongly suggests $SAT \notin P$, generalizing to $NP \not\subseteq P$ due to NP-completeness.
- Counterexample-Based Proof: Unlike traditional proofs requiring a tautologous assertion (e.g., `#(~(q > p)) = (s=s)`), Meth8/VŁ4’s proof strategy relies on consistent counterexamples. The N’s in both formulas, especially the 10 N’s for SAT, indicate cases where no polynomial-time algorithm exists.

The universal claim for Not(P=NP)—that no polynomial-time algorithm exists for an NP problem—is addressed within Meth8’s finite scope. The non-tautologous results refute $NP \subseteq P$, proving that at least one NP problem (SAT) lacks a polynomial-time solution.

6. Implications

This proof resolves the P=NP problem within Meth8/VŁ4’s finite logic framework, bypassing complexity-theoretic barriers like relativization. The results align with prior non-tautologous evaluations (e.g., `q > p ; TTFT`), reinforcing Not(P=NP). Implications include:
- Strengthened foundations for cryptography, reliant on NP problems being intractable.
- Validation of optimization challenges, as NP problems remain computationally distinct from P.
- A paradigm shift, prioritizing finite logic over infinite input analysis.

7. Conclusion

Using Meth8/VŁ4, we prove Not(P=NP) through the non-tautologous formulas `#(~(q > p) = (s = s)) = (s = s)` and `#(~(r & p) = (s = s)) = (s = s)`, which provide counterexamples to $NP \subseteq P$ and $SAT \in P$. The finite nature of Meth8/VŁ4, supported by a refutation of the axiom of infinity, quantifier-aligned # operator, and dismissal of infinite inputs, enables a universal resolution. The strong counterexamples, particularly for SAT (10 N’s), establish that $NP \not\subseteq P$, resolving the P=NP problem. Future work may explore additional Meth8/VŁ4 formulas to further validate this proof.

References

- Meth8/VŁ4 System Description, https://ersatz-systems.com, accessed June 14, 2025.