

Article

Not peer-reviewed version

Multitime Barriers for P vs NP: Why Some Reasons May Not Travel in Polynomial Time

[Rogério Figurelli](#) *

Posted Date: 27 January 2026

doi: 10.20944/preprints202601.2011.v1

Keywords: P vs NP; SAT; reductions; proof logging; DRAT/LRAT/FRAT; temporal state machines (TSM); multitime; receipts; abstain-gated kernels; governance; no-reopen; auditability



Preprints.org is a free multidisciplinary platform providing preprint service that is dedicated to making early versions of research outputs permanently available and citable. Preprints posted at Preprints.org appear in Web of Science, Crossref, Google Scholar, Scilit, Europe PMC.

Copyright: This open access article is published under a [Creative Commons CC BY 4.0 license](#), which permit the free download, distribution, and reuse, provided that the author and preprint are cited in any reuse.

Disclaimer/Publisher's Note: The statements, opinions, and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions, or products referred to in the content.

Article

Multitime Barriers for P vs NP: Why Some Reasons May Not Travel in Polynomial Time

Rogério Figurelli

Independent Researcher, Brazil; figurelli@gmail.com

Abstract

P vs NP is often approached as a question about algorithms and bounds: either exhibit a polynomial-time decider for an NP-complete problem or prove that no such decider exists. This paper proposes a different lens: treat $P = NP$ as a transport property — do “reasons” (the operational structure that makes a solver succeed) travel across polynomial reductions in a way that remains auditable and stable? We frame this through the Temporal State Machine (TSM) kernel of Compositional Clock Theory: decisions occur under multiple clocks (execution, verification, audit), and progress must be governed by admissibility, receipts, and commit depth. We introduce abstain-gated decision kernels under a no-reopen discipline, where the system may refuse to commit unless the evidence is replayable and the recovery budget is feasible. Within this governance-first framing, “transport” becomes a commutation requirement between reductions and receipts: success is not only solving instances, but carrying a verifiable explanation of why the instance is solved that survives encoding changes and can be checked under declared costs. We instantiate the transport test on canonical NP-complete domains (3-SAT and Sudoku, as representatives), not to claim a proof of $P = NP$, but to define a falsifiable program: either discover a stable, general, receipt-carrying polynomial strategy (supporting $P = NP$), or demonstrate systematic transport failure that resists any admissible repair (supporting $P \neq NP$). The payoff is a structured research agenda that aligns complexity theory with governance and audit, clarifying what would count as credible progress under Clay-level scrutiny.

Keywords: P vs NP; SAT; reductions; proof logging; DRAT/LRAT/FRAT; temporal state machines (TSM); multitime; receipts; abstain-gated kernels; governance; no-reopen; auditability

I. Introduction

The Clay Mathematics Institute frames P vs NP as the problem of whether every problem whose solutions can be verified in polynomial time can also be solved in polynomial time [2].

The canonical route through this landscape uses NP-completeness: show that a single NP-complete language (such as SAT) is in P to obtain $P = NP$, or prove no polynomial-time algorithm exists for any NP-complete language to obtain $P \neq NP$ [3–6]. The difficulty is not only algorithmic invention, but credibility: to be accepted at Clay level, a result must be unambiguous, formally correct, and robust to adversarial scrutiny [2,7,8].

This paper proposes a continuation-layer framing consistent with the Temporal State Machine (TSM) kernel of Compositional Clock Theory [1]. In TSM, “time” is not merely wall-clock; it is a vector of operational clocks tied to transition families: execution, verification, audit, rollback, and legitimacy.

A system is unsafe not only when it is wrong, but when it cannot be signed under scrutiny and cannot be unwound within a declared recovery budget [1]. If we take that seriously, then “progress toward $P = NP$ ” is not just finding solvers that work on benchmarks. It is producing artifacts that remain valid under verification and audit clocks, with receipts that are replayable and admissible.

Within that governance-first stance, we introduce $P = NP$ as a transport test: do the operational “reasons” for success (the structural basis a method uses to decide) travel across polynomial reductions in a way that preserves checkability and stability? NP-completeness reductions guarantee existence of mappings between instances, but do not guarantee that a solver’s internal explanation

structure is preserved. If “reasons” do not transport, then an apparent breakthrough may be an encoding artifact, a fragile heuristic, or a non-general exploitation of structure.

We do not claim a solution to P vs NP. Instead, we formalize a falsifiable program: define transport operators, enforce receipts and admissibility under no-reopen, and evaluate whether any candidate strategy can satisfy both (i) empirical effectiveness and (ii) Clay-grade obligations: completeness, polynomial worst-case bounds, and verifiable proof artifacts.

What this enables: a disciplined bridge from “solver success” to “auditable progress,” separating empirical performance from proof-grade commitments.

II. Background: NP-Completeness, Reductions, and Why Transport Matters

NP is the class of languages whose membership proofs can be verified in polynomial time; P is the class decidable in polynomial time [7,8]. SAT is NP-complete by Cook–Levin, and reductions propagate NP-completeness broadly [3–6]. The standard meta-logic is: if $\text{SAT} \in \text{P}$ then every NP problem $\in \text{P}$; if an NP-complete problem is not in P then $\text{P} \neq \text{NP}$ [5–8].

However, reductions preserve decision equivalence, not the solver’s explanatory anatomy. A reduction $r: A \rightarrow B$ ensures $x \in A \Leftrightarrow r(x) \in B$, with r computable in polynomial time [5–7]. It does not ensure that an algorithmic strategy that “works well” on B provides stable, interpretable, verifiable reasons that can be mapped back to A without exploding the verification budget. In practice, encodings can alter locality, constraint graph structure, propagation strength, and proof trace morphology. That is a governance problem: if your claimed progress depends on fragile encoding quirks, you cannot sign it under scrutiny.

This motivates the transport test: define what a “reason” is (operationally), define what it means to transport it across r , and then empirically and formally probe whether transport can be made stable under admissibility constraints.

What this enables: a crisp target for “generalization” that is stronger than cross-benchmark success and closer to proof-grade robustness.

III. TSM Compatibility: Enriched State, No-Reopen, and Abstain-Gated Kernels

We adopt the TSM kernel as the governing semantics for discovery and claims [1]. A TSM models system evolution by transitions over enriched state, not by assuming a single privileged timeline. For a candidate proof program, we track at minimum:

$$x = (w, b, a, c, r)$$

w: operational world-state (instances, encodings, solver configuration, reduction maps)

b: belief/model state (hypotheses, inferred structure, learned heuristics)

a: admissibility state (what moves are allowed; what evidence counts)

c: commit depth (how irreversible the claim is becoming; how costly it is to retract)

r: receipts (the replayable log that makes the claim checkable and auditable)

No-reopen discipline: once a claim is closed under a declared tribunal (admissibility rules), it may not be reopened except under an explicitly declared revision protocol. This prevents “LLM circularity” and protects against p-hacking dynamics where the search mutates the rules until something looks like a result [1,30,31]. In complexity terms: no-reopen forces monotone accumulation of obligations satisfaction.

Abstain-gated decision kernels: the system may output abstain (reject option) instead of committing to SAT/UNSAT unless the receipts meet declared standards (replayable, checkable, within verifier budgets). Abstention is not a failure; it is a controlled output that respects the audit clock [27–29]. This matters because many heuristic SAT methods are incomplete; they may succeed often but do not provide completeness guarantees or UNSAT certificates [9,10,19]. Under abstaining, such methods can still contribute, but they cannot force a commit unless a verifiable closure artifact exists.

What this enables: a control-plane semantics where discovery can run fast, but commitment is permitted only under auditable closure.

IV. Formalizing “Reasons” and Transport Operators

We define a “reason” as a structured witness of decisional progress that (i) can be logged, (ii) can be checked, and (iii) constrains future search in a way that is stable under declared invariants.

Examples of reasons (operational classes):

- Proof traces for UNSAT: clausal proofs checkable by independent verifiers (e.g., DRAT/LRAT/FRAT-style logs) [23–26].
- Verifiable witnesses for SAT: an assignment plus a checker; optionally augmented with a derivation trace that supports why the assignment was found under the kernel’s policy [7,8].
- Structure certificates: a bounded-width decomposition, a backdoor set, or other tractability witness that makes an instance decidable by a known polynomial-time method conditioned on the certificate [13–16,21].
- Reduction receipts: mappings r and (when needed) their inverses or interpretation maps that allow the decision and its reason to be transported back.

Transport test requirement (commutation):

Let $r: A \rightarrow B$ be a polynomial-time reduction. Let Reason_B be the reason object produced on $r(x)$. A transport operator τ_r must map Reason_B into Reason_A such that:

$$\text{Verify}_A(x, \text{Reason}_A) = \text{Accept}$$

and

$$\text{Decision}_A(x) = \text{Decision}_B(r(x))$$

with verification and translation costs remaining within declared budgets (in particular, polynomial in $|x|$ under the intended claim).

This is stronger than correctness. It demands that the explanation structure survive crossing domains.

Representative domains: we use 3-SAT as the canonical NP-complete SAT form [5–7], and Sudoku as a widely studied NP-complete puzzle family (in generalized form) with standard SAT encodings [34,35]. The point is not the domains themselves, but whether transport survives across meaningfully different constraint geometries.

What this enables: an explicit, checkable bridge between “I solved it there” and “I can justify it here.”

V. Proof Logging, Receipts, and Admissibility Tribunals

A Clay-grade claim must be checkable by others. For SAT/UNSAT, this naturally leads to proof logging and independent checking pipelines [23–26]. We define a tribunal (admissibility policy) that specifies:

- Which reason types count as closure artifacts (SAT witnesses, UNSAT proofs, structure certificates).
- Which proof formats and checkers are admissible (e.g., DRAT-family logs or successor formats), and what constitutes a valid receipt.
- Maximum verifier complexity and resource assumptions.
- Rules for abstention and for safe escalation (when incomplete solvers may be used as proposal engines but not as commit engines).
- A no-reopen protocol for revisions: what is allowed to change (hypotheses, heuristics, encodings) without invalidating previously closed obligations.

Receipts are the minimal replayable record: inputs, reduction maps, solver configuration, intermediate artifacts, proof logs, and check results. The governance stance is: if it cannot be replayed, it cannot be claimed.

What this enables: a shared language for “closure,” making progress contestable rather than narrative.

VI. Strategy Space: Why Rankings Help, and Why They Do Not Solve Clay

Empirical ranking — like the DPLL/WalkSAT/GSAT style populations commonly explored in solver engineering — can be useful as a discovery engine [9,10,19,20]. Such rankings estimate fitness under windows: solve rate, time, steps, and stability under perturbations. They can also identify candidates that generate short UNSAT proofs or reliably find witnesses, which matters for receipts.

But rankings do not meet Clay requirements by themselves. The Clay gap is categorical:

1. Completeness: heuristic local search methods (e.g., WalkSAT/GSAT) are not complete deciders [19].
2. Worst-case polynomial bound: even complete methods like DPLL variants do not come with known polynomial worst-case bounds on SAT [7,9,10].
3. General transport: success on one encoding does not imply stable success across reductions; transport must be demonstrated under admissibility constraints (Sections IV–V).
4. Formal proof artifacts: for $P = NP$, one needs a polynomial-time algorithm with a proof of its bound; for $P \neq NP$, one needs a lower-bound proof against all polynomial-time algorithms — famously difficult [2,7,8].

Therefore, rankings help as: (i) a generator of hypotheses about tractable structure (backdoors, width, parity structure), (ii) a source of receipts (proof logs), and (iii) a way to tune abstain-gates. They do not substitute for the required theorems.

What this enables: a clean separation between “search for ideas” and “commit to claims.”

VII. Transport Stress Tests and Entropy-Style Stability Measures

To avoid circular progress, we define stress tests that intentionally disrupt encoding and structure while preserving decision equivalence:

- Encoding families: multiple reductions from Sudoku-like CSPs to SAT; multiple 3-SAT normalizations.
- Perturbations: variable renamings, clause shuffles, gadget alternatives, and structure-preserving transformations.
- Cross-domain replay: run candidate strategy on $B = r(x)$, then transport the reason back and verify on A .

We measure:

- Transport success rate: fraction of instances where reasons transport and verify.
- Receipt size growth: how the proof/trace scales across transport.
- Verifier time: independent checking cost.
- Stability index s : sensitivity of outcomes to admissible perturbations.
- “Entropy-style” dispersion: whether the reason distribution fragments across encodings (a proxy for brittleness). Entropy language here is operational: dispersion of receipts across perturbations, not a claim of a new information-theoretic bound.

This connects to known concerns about selection bias and overfitting in empirical evaluation: a method that “wins” by silently adapting to a benchmark distribution is not credible as a universal claim [30,31].

What this enables: evidence that a method’s success is structural, not an encoding coincidence.

VIII. Interpreting Outcomes: What Would Count as “Trend” Evidence for $P = NP$ or $P \neq NP$

Under this program, outcomes are interpreted as trends in admissible evidence:

Evidence trend toward $P = NP$ (not a proof):

- Discovery of a candidate decider whose reasons transport across reductions with stable receipts, and whose runtime and verifier costs appear bounded by an analyzable polynomial; plus a path to a formal proof of the bound.

Evidence trend toward $P \neq NP$ (not a proof):

- Systematic transport failure: no matter how the tribunal is strengthened (without cheating), reasons fragment across reductions or require superpolynomial blowup in receipts/verifier time, suggesting that “general polynomial solving” resists admissible stabilization.

But neither trend is a Clay proof. For $P = NP$, one still must provide a polynomial-time algorithm and prove its bound. For $P \neq NP$, one must prove a lower bound that rules out all polynomial-time deciders — an area with deep barriers [2,7,8]. The value of the transport test is that it turns vague intuitions (“it doesn’t generalize”) into logged, replayable failure modes that can guide theory.

What this enables: a research agenda where empirical work produces audit-grade artifacts usable by theory.

IX. Limitations and Failure Modes

1. Tribunal gaming: if admissibility is weak, the program devolves into storytelling. No-reopen and explicit revision protocols are required [1,30].
2. Proxy success: methods may exploit superficial structure that does not transport; stress tests must be adversarial.
3. Proof-format dependence: proof logging and checking are powerful but can hide complexity in tooling; independent verifiers and multiple checkers reduce this risk [23–26].
4. Misinterpretation risk: transport failure is not a proof of $P \neq NP$; transport success is not a proof of $P = NP$. The program is diagnostic, not decisive.

What this enables: honest scoping — progress signals without overclaiming.

X. Conclusion

We reframed $P = NP$ as a transport question: not only “can we decide NP-complete languages efficiently,” but “do the reasons for decision remain stable, checkable, and auditable as we move across reductions?”

This reframing aligns naturally with the TSM kernel, where correctness alone is insufficient; admissibility, receipts, and recovery-to-okay feasibility determine whether a claim can be signed under scrutiny [1]. Under abstain-gated kernels and no-reopen discipline, heuristic success becomes a source of hypotheses and receipts — but not an excuse to commit.

The transport test offers a disciplined path between empirical solver engineering and Clay-grade obligations. It forces commutation between reductions and receipts, elevating “generalization” from benchmark performance to verifiable structure.

If a polynomial-time decider exists, we should expect reasons to transport with bounded overhead and stable receipts; if no such decider exists, we should expect systematic transport brittleness that resists admissible repair. Either way, the program yields replayable artifacts: proof logs, failure modes, and stability maps that can guide theory rather than decorate narratives.

What this enables: a governance-first research pipeline in which “progress” is not declared — it is logged, replayed, and signed.

License and Declarations

License. © 2025 Rogério Figurelli. This preprint is licensed under the Creative Commons Attribution 4.0 International License (CC BY 4.0). You are free to share (copy and redistribute the material in any medium or format) and adapt (remix, transform, and build upon the material) for any purpose, even commercially, provided that appropriate credit is given to the author. A link to the license and an indication of any changes made must be included when reusing the material. The full license text is available at: <https://creativecommons.org/licenses/by/4.0/>.

Author Contributions. This work was conceived, structured, and written in full by Rogério Figurelli. No other authors qualify for authorship under the ICMJE recommendations. Contributions from institutions, collaborators, or organizations that supported background research, data organization, or conceptual framing are acknowledged separately in related project documentation.

Acknowledgments. The author(s) acknowledge prior foundational work and intellectual contributions that have shaped the conceptual and methodological background of this manuscript. Relevant influences include advances in causal reasoning and modeling, developments in artificial intelligence safety and governance, insights from decision sciences and behavioral theories, reinforcement learning frameworks, and broader discussions in information ethics. The author(s) also recognize the value of open-innovation initiatives, collaborative research environments, and public knowledge projects that provide resources, context, and inspiration for ongoing inquiry. Visualizations, figures, and supporting materials were prepared using openly available tools and may have benefited from institutional or laboratory collaborations aimed at promoting reproducibility and transparency.

Data Availability. No proprietary or sensitive datasets were used in the preparation of this manuscript. All conceptual models, equations, and figures were generated as part of the author(s)' own research process. Any supplementary materials, such as simulation data, analytical scripts, or graphical code used to produce figures, are available upon request and may be deposited in an open-access repository in accordance with FAIR data principles. Authors encourage reuse and adaptation of such resources, provided appropriate credit is given.

Ethics Statement. This research does not involve experiments with humans, animals, or plants and therefore did not require ethics committee approval. The work is conceptual, methodological, and computational in scope. Where references to decision-making domains such as healthcare, finance, and retail are made, they serve as illustrative vignettes rather than analyses of proprietary or sensitive datasets.

Conflicts of Interest. The author declares no conflicts of interest. There are no financial, professional, or personal relationships that could inappropriately influence or bias the work presented.

Use of AI Tools. AI-assisted technology was used for drafting, editing, and structuring sections of this manuscript, including the generation of visual prototypes and narrative expansions. In accordance with Preprints.org policy, such tools are not considered co-authors and are acknowledged here as part of the methodology. All conceptual contributions, final responsibility, and authorship remain with the author, Rogério Figurelli.

Withdrawal Policy. The author understands that preprints posted to Preprints.org cannot be completely removed once a DOI is registered. Updates and revised versions will be submitted as appropriate to correct or expand the work in response to community feedback.

Contributions

- Defines “transport of reasons” as a testable property across NP-completeness reductions.
- Aligns transport testing with the TSM kernel: admissibility, receipts, commit depth, and no-reopen governance.
- Introduces abstain-gated decision kernels as control-plane semantics for safe commitment under verification/audit clocks.

- Specifies evaluation metrics emphasizing stability, replayability, proof logging, and verifier cost.
- Proposes a round-based discovery protocol that separates empirical solver fitness from Clay-grade proof obligations.

References

1. R. Figurelli, "Compositional Clock Theory: Temporal State Machines and Multitime," Jan. 14, 2026.
2. S. Cook, "The P versus NP Problem," Clay Mathematics Institute, Millennium Prize Problems, 2000.
3. S. A. Cook, "The complexity of theorem-proving procedures," in Proc. 3rd ACM Symp. Theory of Computing (STOC), 1971, pp. 151–158.
4. L. A. Levin, "Universal search problems," Problemy Peredachi Informatsii, vol. 9, no. 3, pp. 115–116, 1973.
5. R. M. Karp, "Reducibility among combinatorial problems," in Complexity of Computer Computations, R. E. Miller and J. W. Thatcher, Eds. New York, NY, USA: Plenum, 1972, pp. 85–103.
6. M. R. Garey and D. S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness. San Francisco, CA, USA: W. H. Freeman, 1979.
7. C. H. Papadimitriou, Computational Complexity. Reading, MA, USA: Addison-Wesley, 1994.
8. M. Sipser, Introduction to the Theory of Computation, 3rd ed. Boston, MA, USA: Cengage Learning, 2012.
9. M. Davis and H. Putnam, "A computing procedure for quantification theory," J. ACM, vol. 7, no. 3, pp. 201–215, 1960.
10. M. Davis, G. Logemann, and D. Loveland, "A machine program for theorem-proving," Commun. ACM, vol. 5, no. 7, pp. 394–397, 1962.
11. B. Aspvall, M. F. Plass, and R. E. Tarjan, "A linear-time algorithm for testing the truth of certain quantified Boolean formulas," Inf. Process. Lett., vol. 8, no. 3, pp. 121–123, 1979.
12. W. F. Dowling and J. H. Gallier, "Linear-time algorithms for testing the satisfiability of propositional Horn formulae," J. Logic Program., vol. 1, no. 3, pp. 267–284, 1984.
13. N. Robertson and P. D. Seymour, "Graph Minors. II. Algorithmic aspects of tree-width," J. Algorithms, vol. 7, no. 3, pp. 309–322, 1986.
14. H. L. Bodlaender, "A linear-time algorithm for finding tree-decompositions of small treewidth," SIAM J. Comput., vol. 25, no. 6, pp. 1305–1317, 1996.
15. B. Courcelle, "The monadic second-order logic of graphs. I. Recognizable sets of finite graphs," Inf. Comput., vol. 85, no. 1, pp. 12–75, 1990.
16. R. G. Downey and M. R. Fellows, Parameterized Complexity. Berlin, Germany: Springer, 1999.
17. M. Soos, "Enhanced Gaussian elimination in DPLL-based SAT solvers," in Pragmatics of SAT, 2010.
18. C.-S. Han and J.-H. R. Jiang, "When Boolean satisfiability meets Gaussian elimination in a simplex way," in Proc. Int. Conf. Computer Aided Verification (CAV), 2012.
19. C. P. Gomes, B. Selman, and H. Kautz, "Boosting combinatorial search through randomization," in Proc. AAAI Conf. Artificial Intelligence (AAAI), 1998, pp. 431–437.
20. M. Luby, A. Sinclair, and D. Zuckerman, "Optimal speedup of Las Vegas algorithms," Inf. Process. Lett., vol. 47, no. 4, pp. 173–180, 1993.
21. N. Nishimura, P. Ragde, and S. Szeider, "Detecting backdoor sets with respect to Horn and binary clauses," in Proc. Int. Conf. Theory and Applications of Satisfiability Testing (SAT), 2004.
22. A. Biere, M. Heule, H. van Maaren, and T. Walsh, Eds., Handbook of Satisfiability. Amsterdam, The Netherlands: IOS Press, 2009.
23. M. J. H. Heule, W. A. Hunt Jr., and N. Wetzler, "Bridging the gap between easy generation and efficient verification of unsatisfiability proofs," Softw. Test. Verif. Reliab., 2014.
24. M. J. H. Heule, M. Jarvisalo, and M. Suda, "DRAT-trim: Efficient checking and trimming using expressive clausal proofs," in Proc. Int. Conf. Theory and Applications of Satisfiability Testing (SAT), 2013.
25. C. Baek, M. Carneiro, and M. J. H. Heule, "A flexible proof format for SAT solver–elaborator communications," 2022.
26. T. Laitinen, "Extending clause learning SAT solvers with complete parity reasoning," 2012.
27. C. K. Chow, "On the optimum recognition error and reject tradeoff," IEEE Trans. Inf. Theory, vol. 16, no. 1, pp. 41–46, 1970.

28. Y. Geifman and R. El-Yaniv, "Selective classification for deep neural networks," in Proc. Adv. Neural Inf. Process. Syst. (NeurIPS), 2017.
29. R. El-Yaniv, "On the foundations of noise-free selective classification," J. Mach. Learn. Res., 2010.
30. G. C. Cawley and N. L. C. Talbot, "On over-fitting in model selection and subsequent selection bias in performance evaluation," J. Mach. Learn. Res., vol. 11, pp. 2079–2107, 2010.
31. J. P. A. Ioannidis, "Why most published research findings are false," PLoS Med., vol. 2, no. 8, e124, 2005.
32. R. D. Peng, "Reproducible research in computational science," Science, vol. 334, no. 6060, pp. 1226–1227, 2011.
33. G. C. Necula, "Proof-carrying code," in Proc. ACM SIGPLAN Symp. Principles of Programming Languages (POPL), 1997.
34. T. Yato and T. Seta, "Complexity and completeness of finding another solution and its application to puzzles," IEICE Trans. Fundamentals, 2003.
35. I. Lynce and J. Ouaknine, "Sudoku as a SAT problem," in Proc. Int. Symp. Artificial Intelligence and Mathematics (AIMath), 2006.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.