Research Monograph Series

The Phase Difference Between Formal and Generative Time:

A Reflexive Reading of P versus NP through Generative Topological Ontology

Author: Tomokazu Tanaka (bigakusha-haha)
ORCID: https://orcid.org/0009-0009-3796-5889
Affiliation: Sayama Aesthetics School / Institute of Aesthetics
Version: 1.0
Last Updated: September 2026
Status: Ongoing Research

Research Monograph

Tomokazu Tanaka (bigakusha-haha)
Version 1.0, September 2026

Publication Information

Copyright

© 2026 Tomokazu Tanaka (bigakusha-haha). All rights reserved.

License

This work is distributed as a Research Monograph.

No part of this publication may be reproduced, distributed, or transmitted in any form or by any means without prior written permission of the author, except for brief quotations used in academic review or scholarly citation.

Suggested Citation

Tanaka, Tomokazu (bigakusha-haha). 2026.

The Phase Difference Between Formal and Generative Time:
A Reflexive Reading of P versus NP through Generative Topological Ontology

Research Monograph Series, Version 1.0.

Sayama Aesthetics School / Institute of Aesthetics.

Research Monograph

Version: 1.0

Last Updated: September 2026

Status: Ongoing Research

Abstract

This paper rereads the P versus NP problem through Generative Topological Ontology and clarifies the phase difference between formal time and the generative time of reality. In computational complexity theory, P and NP classify formalized decision problems in terms of computational resources. P consists of problems decidable in deterministic polynomial time; NP consists of problems whose positive instances have polynomial-time verifiable certificates, equivalently problems decidable in nondeterministic polynomial time. P versus NP asks whether P equals NP. The problem remains open, and P ≠ NP has not been proved. The philosophical interpretation advanced here does not answer this mathematical question.

The paper distinguishes computational time, measured as steps relative to input length under a fixed formal model, from generative time, in which relations, dependencies, histories, and irreversible transformations accumulate. These are not two disconnected substances. Actual computation takes place within generative time, while complexity theory abstracts a comparable resource measure from computational practice. The difficulty arises when this methodological abstraction is unconsciously identified with reality's temporality as a whole.

The epistemic tension produced by this identification is called the "reflexive value" of P versus NP. The problem is not mathematically self-referential. Yet when it is used to explain reality, creativity, or intelligence in general, the conditions of its own formalization become relevant: what does the theory isolate, and what does it relegate to the background? The contradiction that becomes visible here is not a formal logical contradiction asserting both P = NP and P ≠ NP. It is a tension between conceptual frameworks that arises when distinct temporal phases are reduced to one measure.

In Generative Topological Ontology, Soku (即) is not identity but a structure in which different phases connect while retaining their differences, thereby enabling new generation. At the level of philosophical analogy, P and NP may be related to distinct modes of access—generation and verification—within one problem space. This claim neither assigns a mathematical topology to the complexity classes nor proves P ≠ NP. Likewise, pre-humanity and post-humanity are not alternative names for complexity classes; they are ontological terms for conditions that precede human meaning-making and consequences that continue beyond human intention and ownership.

The paper further names "methodological reification" as the process by which the local effectiveness of a method is forgotten and its formalism is extended into a total measure of being. The fallenness of scholarship lies not in formalization itself, but in losing sight of its conditions

and scope. The paper concludes that the philosophical significance of P versus NP is distinct from its mathematical proof value: it can serve as a critical interface that exposes the difference between formal and generative time.

Keywords

P versus NP, formal time, generative time, Generative Topological Ontology, Soku, phase difference, reflexivity, methodological reification, pre-humanity, post-humanity

Contents

1. Introduction 1

2. Formal Time and Generative Time 2

3. The Reflexive Value of P versus NP 4

4. Phase Difference, Contradiction, and Soku 5

5. Pre-Humanity, Post-Humanity, and the Worldliness of Computation 7

6. Methodological Reification and the Fallenness of Scholarship 8

7. An Open Generative Circuit 9

8. Conclusion 10

References 12

Research Method 13

1. Introduction

The P versus NP problem asks whether every problem whose solution can be verified efficiently can also be solved efficiently. Mathematically, P is the class of decision problems decidable in deterministic polynomial time. NP is the class of decision problems for which positive instances have polynomial-time verifiable certificates, equivalently the class decidable in nondeterministic polynomial time. P is contained in NP, and whether P = NP remains open (Cook 1971; Cook 2006).

This paper preserves those definitions while asking what P versus NP reveals—and does not reveal—about time in reality. Computational time is formalized as execution steps and compared relative to input length. Generative time, by contrast, includes the way bodies, environments, institutions, past histories, and responses alter the conditions of subsequent connection. Computation is executed within real time, but complexity theory abstracts resource use within a particular problem formulation rather than describing the whole history of that practice.

The central issue is therefore not which of the two temporalities is “real.” It is what becomes unaccounted for when formal time, which makes a portion of reality measurable, is unconsciously identified with reality’s temporality as a whole. This paper analyzes that confusion as an epistemic operation that forgets the scope of a formal method, rather than as an error inherent in complexity theory.

2. Formal Time and Generative Time

2.1 Time in complexity theory

In complexity theory, time is counted as the number of steps an algorithm executes on a specified computational model. For P and NP, time bounds are evaluated as functions of input length. Formal time is thus an ordered and countable resource whose executions can be compared under the same encoding and model conditions. Both NP certificate verification and P decision procedures consume time. NP is not a domain without time; it is a different computational mode concerning verification format and the representation of solution existence.

This definition does not imply that computation exists independently of reality. Actual computation depends on devices, energy, memory, encoding, maintenance, and purposes of use. A formal model does not describe all these conditions simultaneously; it abstracts those relevant to a particular question. Formal computational time is useful precisely as a resource measure for comparing computational processes.

2.2 Generative time

Generative time refers neither simply to clock time nor to the accumulation of computational steps. It includes the formation of relations, dependencies, histories, traces in bodies and environments, and the transformations through which these alter future possibilities. In Generative Topological Ontology, informational time is constituted by the order, dependency, repetition, waiting, memory, and irreversibility of transformations (Tanaka 2026a).

When generative time is called “increasing,” this does not mean that time itself, as a physical quantity, expands. It means that the accumulation of events and relations continues to change subsequent possibilities for operation, interpretation, and dependency. Even a repeatable procedure is situated in a historical context that is never wholly identical. Reproducibility of a procedure must therefore be distinguished from historical identity of events.

2.3 The relation between the two temporalities

Formal and generative time are not unrelated substances. Formal computation occurs as a technical and social practice embedded in generative time, and complexity theory abstracts a repeatable measure of steps from that practice. Formal time can therefore be situated within generative time without exhausting its totality.

Confusion arises when this relation is forgotten and the question actually answered by complexity theory is not distinguished from philosophical questions derived from it. P versus NP compares computational resources for decision and verification within formalized problems. The irreversible passage of real time or the historical transformation of information space does not follow directly from a theorem about P and NP.

3. The Reflexive Value of P versus NP

The P versus NP problem is not itself a self-referential statement. Its mathematical content concerns whether two complexity classes coincide. However, when its significance is extended from the comparison of computational time to reality, creativity, or intelligence in general, the scope of the formal theory is reopened to question. The theory abstracts computational time and thereby also occasions inquiry into what that abstraction relegates to the background.

In this sense, P versus NP is reflexive at a philosophical level. It is formulated through formal time, while also becoming an interface for asking how far formalized time can represent reality. “Embodiment” here means neither mathematical self-reference nor a theorem about ontology. It means that the object of a theory and the assumptions that make the theory possible become visible within the same interpretive field.

The “philosophical proof value” used here must be distinguished from the mathematical proof value of $P = NP$ or $P \neq NP$. Whatever the mathematical answer, it establishes a relation among formal complexity classes. Philosophical value lies in diagnosing the limits of thought that extends the computational timescale to the whole of real time without mediation. P versus NP does not prove the existence of generative time; it provides a critical reference point from which the difference between formal and generative time can be examined.

4. Phase Difference, Contradiction, and Soku

4.1 Two modes of access to information

At the level of philosophical analogy, P and NP can be read as different modes of access to the same formal problem. P foregrounds a procedure that decides an answer from an input; NP foregrounds a certificate for a positive instance and the relation through which it is verified. In this reading, information does not appear as a single undifferentiated content. It takes distinct modes: what is generated or decided, and what is verified within a relation.

“Phase difference,” however, is a concept from Generative Topological Ontology here; it does not claim that a mathematical topological space has been assigned to the complexity classes. The inclusion $P \subseteq NP$ is a set-theoretic fact and must be distinguished from a philosophical claim about topological overlap. Whether $P = NP$ remains open. Thus, “overlapping but non-identical” may be proposed as a philosophical description, but not as an established mathematical assertion that $P \neq NP$.

4.2 From difference to contradiction

The existence of a phase difference does not by itself entail a logical contradiction. Contradiction emerges when propositions belonging to different temporal modes or relational frameworks are compared as if they belonged to the same meaning, criterion, and level, and are consequently subjected to incompatible demands. Conceptual tension, temporal non-identity, and the formal contradiction $P \wedge \neg P$ must therefore be distinguished.

The claim that reality receives contradiction does not mean that every pair of contradictory propositions is affirmed as true. It is a generative claim: demands arising from different phases need not immediately be reduced to one side; their tension may be connected to a reorganization of relations. Contradiction is not a universal principle that explains every aspect of reality, but a possible generative event that becomes manifest when distinct phases meet.

4.3 The generative structure of Soku

In the terminology of bigakusha-haha, Soku (即) is not an equation of A and B or the erasure of their difference. It is a structure in which different phases connect while retaining their differences, and that connection generates a new possibility C. Conceptually, this may be

expressed as $A \times B \rightarrow C$.

From this perspective, the philosophical focus of P versus NP is not to decide in advance whether the two are ontologically identical. It concerns the relation between generation and verification as distinct modes in one information space, and moments at which that relation calls its own comparative criteria into question. Soku does not close the phase difference through reconciliation; it generates a new problem setting from a connection that preserves difference.

5. Pre-Humanity, Post-Humanity, and the Worldliness of Computation

Pre-Humanity and Post-Humanity cannot be identified with P and NP on the basis of their mathematical definitions. P and NP are classes of problems; pre-humanity and post-humanity are ontological terms concerning conditions of generation. These terms can nevertheless serve as a heuristic for clarifying the worldly conditions of computation.

Pre-Humanity does not mean only a chronological era before the emergence of human beings. It refers to conditions—bodies, matter, nature, and causal constraints—that are not exhausted by human conceptualization yet enable human practice. Post-Humanity likewise does not mean merely a future date. It describes the way outputs of computation, models, institutions, or works may continue to affect relations beyond human intention or ownership. Computation depends on pre-human material conditions and may produce post-human consequences; this does not mean that P is pre-humanity or NP is post-humanity.

A computer is not an abstract subject independent of its environment. To function as a computer, it depends on relations among hardware, energy, code, data formats, purposes, operations, maintenance, and institutions of use. Formal models of computation fix or abstract some of these conditions. A blind spot here is analogical to the modern structure that attempts to explain the world from a self-certain subject. Yet in Heidegger's strict terminology, Being-in-the-world is the mode of being of Dasein; the fact that equipment belongs to a relational network in the world does not by itself make a computer Dasein (Heidegger 1962, §§12–18).

What can be asserted about the worldliness of computers is that computation is realized as a material and social practice within a world, and that abstract models select aspects of this practice according to particular aims. Whether a computer discloses a world in its own mode requires a further ontological argument.

6. Methodological Reification and the Fallenness of Scholarship

This paper calls “methodological reification” the process by which the local effectiveness of a method under specified conditions is unconsciously taken to show that the method exhausts reality as a whole. Examples include equating measurability with existence, reproducibility with truth, computability with generability, and verifiability with reality. Each equivalence requires a separate argument; none follows from methodological success alone.

On this account, the “fallenness of scholarship” does not mean the decline of mathematics, science, or logic as such. It names a tendency in which scholarship forgets its domain, measurement conditions, model choices, and institutional or technical bases, then treats local results as a universal ontology of the world. Where Heidegger’s concept of fallenness is invoked, it is used not as a moral condemnation but as a structure in which Dasein is absorbed in established interpretations and ceases to question its own possibilities of being (Heidegger 1962, §38).

This argument does not automatically establish the empirical conclusion that “various disciplines have already fallen.” Each field requires individual analysis: what formalization is effective, within what scope, how that scope is forgotten, and what is thereby excluded. The present paper offers a cross-disciplinary critical hypothesis that can be tested, not a blanket verdict.

From the perspective of Generative Topological Ontology, a formalism is a local stabilization emerging from relations, not a final closure of the world as a whole. A response to fallenness consists in making visible again the interface through which a formalism arises and the relations left outside its scope, while preserving its validity. This does not discard formalization; it clarifies the conditions of its connection to generative reality.

7. An Open Generative Circuit

The argument can be summarized as a non-closed circuit. Formalization cuts out a problem as an object of computation. Practice executes the formalism within material and social environments. Contact with those environments makes visible histories and conditions that the formalism does not capture. When this remainder conflicts with established interpretation, phase difference is experienced as contradiction or tension. Reorganizing relations while retaining the tension transforms both the formalism and the problem setting, and new generation begins.

This circuit is not a complete return to its starting point. Each operation leaves a different history and set of conditions; consequently, the next formalization is non-identical with the previous one. Soku is positioned as the connective structure that enables this non-identical repetition. Thought may repeatedly produce contradiction because the same sign is relocated across different contexts, operations, and temporal phases. This does not by itself mean that thought has failed. Examining a contradiction may identify a conflation between phases or occasion a new connection.

8. Conclusion

This paper has separated the mathematical content of P versus NP from its generative-philosophical interpretation. P versus NP concerns the relation between computational resources required for decision and verification under a formal model. Its time is measured in computational steps relative to input length. Generative time is described through the accumulation of relations, dependencies, histories, and irreversible transformations. The former is practiced within the latter but is not identical with its totality.

The philosophical value of P versus NP does not consist in mathematically proving this distinction. It can act as a critical interface that questions the identification of formal time with all of reality's time. To say that "the problem embodies its own problem" is not to claim mathematical self-reference; it means that the problem's formal structure and the limits of its applicability appear within the same interpretive field.

Soku enables new generation through a connection that preserves differences among generation and verification, formalism and reality, past conditions and future candidates without identifying them. Contradiction is not necessarily entailed in every such connection, but it may appear as a tension in reality when demands from different phases are forced into a single framework. Criticism of scholarship consists not in abandoning formalism, but in undoing the unreflective identification of formalism with reality as a whole and reopening the relation between formal systems and the conditions of their generation.

The paper's final proposition is as follows:

> Read through Generative Topological Ontology, the philosophical value of P versus NP lies not in anticipating the mathematical answer among the complexity classes, but in making visible the limits of thought that identifies formal time with generative time. Formal computational time is embedded in real practice, yet its repeatable measure cannot exhaust the historical generation of reality. This phase difference becomes a critical interface for examining the asymmetry between generation and verification, the emergence of contradiction, and the non-closed connection enacted by Soku.

References

Cook, Stephen A. 1971. “The Complexity of Theorem-Proving Procedures.” In Proceedings of the Third Annual ACM Symposium on Theory of Computing, 151–158. New York: Association for Computing Machinery. https://doi.org/10.1145/800157.805047.

Cook, Stephen A. 2006. “The P versus NP Problem.” Clay Mathematics Institute. https://www.claymath.org/millennium/p-vs-np/.

Descartes, René. 1641/1996. Meditations on First Philosophy. Translated by John Cottingham. Cambridge: Cambridge University Press.

Hartmanis, Juris, and Richard E. Stearns. 1965. “On the Computational Complexity of Algorithms.” Transactions of the American Mathematical Society 117: 285–306. https://doi.org/10.2307/1994208.

Heidegger, Martin. 1927/1962. Being and Time. Translated by John Macquarrie and Edward Robinson. New York: Harper & Row.

Tanaka, Tomokazu (bigakusha-haha). 2026a. The Generative Topology of Information Space: P versus NP, the Discrete Positions of Creation, Viewing, and Evaluation, and Interface Ontology. Research Monograph, Version 1.0, September 2026.

Research Method

This text is published as a Research Memorandum within the continuing development of Generative Topological Ontology.

The concepts, definitions, hypotheses, and arguments presented here record the theoretical position reached at the time of writing and publication. They should not be understood as a finally established theoretical system. They may be expanded, revised, redefined, or reorganized through future research, artistic practice, interaction with reality, academic dialogue, and peer review.

This research does not regard only completed theories as legitimate research outcomes. It also treats the processes through which concepts emerge, become interconnected, undergo transformation, and are integrated into a theoretical structure as significant objects of inquiry. Accordingly, this memorandum should be understood not merely as an incomplete draft, but as a record of a particular phase in the generation of the theory.

The public release of a Research Memorandum is not intended to present a developing theory prematurely as a completed one. Rather, it is a method of documenting the generative processes of thought and practice, enabling their subsequent examination and reconstruction, and opening them to academic and artistic dialogue from different perspectives.

If the contents of this memorandum are subsequently developed into a peer-reviewed article, scholarly monograph, or other formal research output, the finalized work will be identified as a separate and fixed document.

When citing or referring to this memorandum, please include the title, author, version number, and date of the most recent update. Different versions may contain substantive differences in concepts, definitions, and lines of argument.

This Research Memorandum is not a final conclusion. It is one theoretical position reached within the continuing formation of Generative Topological Ontology and, at the same time, an interface open to further generation.