# P vs NP and the Missing Axiom of Finite Measurement
## A Geofinitist Stress Test of the Clay Formulation

Kevin R. Haylett
Manchester,UK

June 2026
*Selected Communications*

# Contents

# Chapter 1

# The P vs NP Problem: A Geofinitist Lens

## Overview

The $P$ vs. $NP$ problem is one of the central open questions in theoretical computer science, traditionally framed in terms of asymptotic complexity over idealised computational models. While this abstraction has proven extraordinarily powerful for classification, it intentionally suppresses constants, noise, finite resources, and instance-dependent structure. This paper does not reject the classical formulation. Rather, it asks how these categories behave when computation is treated as a *measured physical process*. Within a Geofinitist framework—where all quantities are finite, measured, and accompanied by uncertainty—the question shifts from a metaphysical statement about infinite limits to an operational inquiry about *finite-regime separability*.

## A Motivating Analogy

Consider a locked safe. Given a candidate combination, verification is immediate: the safe opens or it does not. However, discovering the correct combination from scratch may require extensive search. The classical $P$ vs. $NP$ question asks whether this apparent asymmetry is fundamental: if verification is efficient, must discovery also be efficient?This intuitive distinction—between recognition and construction—has guided decades of research, yet remains unresolved.

## Classical Formulation

Formally, the $P$ vs. $NP$ problem concerns two classes of decision problems:

- $P$ **(Polynomial Time):** Problems solvable in time bounded by a polynomial function of input size $n$.
- $NP$ **(Nondeterministic Polynomial Time):** Problems for which proposed solutions (certificates) can be verified in polynomial time.

The central question is whether $P = NP$. If true, every efficiently verifiable problem would also be efficiently solvable. If false, there exist problems whose solutions can be checked quickly but not found quickly.Since the work of ***Cook/Levin*** and ***Karp*** in the early 1970s, this question has been formalised through NP-completeness, with the Boolean satisfiability problem (SAT) as a canonical example. Despite extensive study, no resolution has been found, and the problem remains one of the Clay Mathematics Institute's Millennium Prize Problems.

## A Measured Perspective

Classical complexity theory deliberately abstracts away many features of real computation. In practice:

- Computation occurs on finite machines with bounded memory and energy.
- Runtime exhibits variability across instances of equal size.
- Hardware introduces noise, latency, and architectural effects.
- Problem difficulty depends on structure, not merely size.

Geofinitism treats these not as nuisances but as primary features. Complexity becomes a *measured trajectory* rather than a single asymptotic classification.

## Geofinitist Pillars Applied to Complexity

### 1. Geometric Container Space

**Classical View:** Complexity is a scalar function $T(n)$.**Measured View:** Computation traces a trajectory through a high-dimensional space of:

- input structure (e.g. graph density, clause ratio),
- algorithmic pathways,
- hardware states.

Complexity is thus a path within a *container space*, not a point on a curve.

### 2. Approximations and Measurements

**Classical View:** $T(n)$ is exact up to asymptotic equivalence.**Measured View:** Runtime is a measured quantity with uncertainty:

$$T(n) \pm \varepsilon(n),$$

where $\varepsilon(n)$ reflects both hardware noise and distributional variation across instances.

### 3. Dynamic Flow of Computation

**Classical View:** A problem has a single complexity class.**Measured View:** Computation unfolds across stages:

$$C(n) = \frac{1}{K} \sum_{i=1}^{K} T_i(n),$$

capturing preprocessing, solving, verification, and post-processing. Classical complexity collapses these layers into a single scalar.

### 4. Useful Fiction

The equality $P = NP$ is a meaningful abstraction within asymptotic theory. However, it does not directly correspond to measurable reality. It is best treated as a *useful fiction*—a limiting statement whose operational relevance must be grounded.

### 5. Finite Reality

All computation occurs within finite bounds:

- finite input sizes,
- finite precision,
- finite time and energy.

Claims about infinite scaling must therefore be interpreted through finite observations.

## A Formal Geofinitist Framework

Let $\Pi$ be a problem family. For each input size $n$, define a measured instance registry:

$$\mathcal{I}_n \subset \mathbb{M}^{d(n)},$$

with provenance describing generation and perturbation processes.A solver $\mathsf{A}$ produces measured runtime:

$$\mathsf{A}(I) = (v_T(I), \varepsilon_T(I), P_{\mathsf{A}}).$$

Define empirical runtime statistics:

$$\hat{T}(n) = \text{median}_{I \in \mathcal{I}_n} v_T(I), \quad \widehat{\sigma}(n), \quad \widehat{\text{IQR}}(n).$$

Define a finite scaling exponent:

$$\widehat{\alpha}(n) = \frac{\Delta \log \hat{T}(n)}{\Delta \log n}.$$

## Finite Separability

We define:

- **Polynomial behaviour:** $\hat{T}(n) \lesssim Cn^d$ within uncertainty bands.
- **Verification dominance:** $\hat{T}_V(n) \ll \hat{T}(n)$.
- **Finite separation:** persistent divergence of scaling exponents across observed ranges.

This reframes $P$ vs. $NP$ as a question of *empirical separability*:

**Finite Separability:** Do solver and verifier trajectories exhibit robust divergence over measured ranges

If not, the question is *underdetermined at the given resolution.*

## Robustness via Perturbation

Introduce perturbation operator $\mathsf{P}_\eta$. Define smoothed runtime:

$$\hat{T}_\eta(n) = \mathbb{E}[v_T(\mathsf{P}_\eta(I))].$$

- Stability under perturbation $\Rightarrow$ robust tractability.
- Persistent divergence $\Rightarrow$ robust hardness.

## Interpretation

Within this framework, complexity classes are not fixed sets but *regions in a measured scaling space.* The classical question:

$$P \stackrel{?}{=} NP$$

becomes: *Are solving and verification trajectories empirically indistinguishable across finite regimes?*

## Conclusion

The classical $P$ vs. $NP$ problem remains unresolved within its asymptotic formulation. Geofinitism does not attempt to resolve it in that domain. Instead, it reframes the question as a measurable, reproducible protocol grounded in finite computation.This shift replaces metaphysical classification with empirical geometry:

- from infinite limits to finite ranges,
- from exact classes to uncertainty bands,
- from static membership to dynamic trajectories.

In doing so, it provides a practical methodology for understanding computational difficulty as it is actually encountered: within finite systems, under measurable constraints, and across structured instance spaces.

The essays that follow take this Geofinitist lens into three concrete domains. First, the Travelling Salesman Problem is examined as a laboratory in which the abstract reframing becomes tangible: TSP is treated not merely as a combinatorial optimisation task but as a finite attractor-finding problem in a symbolic phase space. Second, the distinction between solving and verification is unpacked at the measurement boundary itself, revealing that these are not two timings of the same process but ontologically different kinds of finite symbolic trajectory. Finally, the Clay formulation of the $P$ vs. $NP$ problem is subjected to a detailed axiomatic analysis, exposing the missing axiom of measurement and culminating in the practical discipline of explicit trajectory initialisation. In each case the same insistence is maintained: classical mathematics is not rejected; it is returned to the care required once computation is treated as a measured, finite, physically instantiated process.

# Chapter 2

# The Travelling Salesman Problem as an Attractor-Finding Problem

## Overview

The Travelling Salesman Problem (TSP) provides a natural worked example for the Geofinitist treatment of the $P$ vs. $NP$ problem. The classical formulation asks for the shortest closed tour through a finite set of cities, visiting each city exactly once and returning to the starting point. The problem is easy to state, easy to verify once a candidate tour is supplied, but difficult to solve in general.This makes TSP a useful bridge between the classical complexity question and the measured, finite, trajectory-based framing developed in the previous chapter. In the classical view, TSP is a combinatorial optimisation problem. In the Geofinitist view, it is a finite symbolic trajectory problem in which the goal is to identify the stable attractor of a constrained cost landscape.The central shift is as follows:

$$\text{TSP as classical optimisation} \quad \longrightarrow \quad \text{TSP as finite attractor selection.}$$

This does not reject the classical theory. Rather, it asks what the classical result means when the computation is treated as a finite, measured, physically instantiated process.

## Classical Formulation of the Travelling Salesman Problem

Let

$$V = \{v_1, v_2, \ldots, v_n\}$$

be a finite set of cities, and let

$$D = (d_{ij})$$

be a matrix of distances or costs, where $d_{ij}$ is the cost of travelling from city $v_i$ to city $v_j$.A tour is an ordering of the cities,

$$\pi = (\pi_1, \pi_2, \ldots, \pi_n),$$

where $\pi$ is a permutation of the city labels. The total length of the tour is

$$L(\pi) = \sum_{i=1}^{n} d_{\pi_i \pi_{i+1}}, \quad \pi_{n+1} = \pi_1.$$

The optimisation problem is to find

$$\pi^\star = \arg\min_{\pi \in S_n} L(\pi),$$

where $S_n$ is the set of all permutations of $n$ cities.For the symmetric TSP, where $d_{ij} = d_{ji}$, the number of distinct tours is

$$\frac{(n-1)!}{2}.$$

This factorial growth is the first visible source of difficulty. Even though each tour is finite and each tour length can be measured, the number of possible symbolic trajectories grows so rapidly that direct exhaustive search becomes impractical for large $n$.The decision version of TSP asks whether there exists a tour whose length is at most a declared bound $B$:

$$\exists \pi \in S_n \quad \text{such that} \quad L(\pi) \leq B.$$

This decision form is the version most directly connected to $NP$. Given a candidate tour $\pi$, one can verify in polynomial time that it visits each city once and that its length does not exceed $B$. The construction of such a tour, however, is not known to be possible in polynomial time for the general problem.

## TSP as Recognition and Construction

The previous chapter framed $P$ vs. $NP$ through the distinction between recognition and construction. A proposed solution may be easy to recognise, while the route to discovering that solution may remain difficult.TSP expresses this distinction clearly.Verification follows a short symbolic trajectory:

$$\mathsf{Verify}_{TSP}(\pi, B) \sim \text{check tour validity} + \text{sum edge costs} + \text{compare with } B.$$

In a finite measured setting this may be written as

$$\mathsf{Verify}_{TSP}(\pi, B) \sim P_V(n) \mid (C, \alpha, H, \delta),$$

where $P_V(n)$ is a polynomial verification trajectory, $C$ is the consensus constraint, $\alpha$ is the finite symbolic or Alphonic limit, $H$ is the historical and procedural layer, and $\delta$ is the uncertainty associated with measurement and representation.Solving, by contrast, requires a search through a large symbolic landscape:

$$\mathsf{Solve}_{TSP}(D) \sim \text{navigate } S_n \text{ under } L(\pi).$$

Thus the central measured distinction becomes

$$\mathsf{Solve}_{TSP}(D) \not\sim \mathsf{Verify}_{TSP}(\pi, B),$$

not as a metaphysical statement, but as an observed and testable divergence of finite trajectories.This is precisely the finite separability question developed earlier:

Do solver and verifier trajectories exhibit robust divergence over measured ranges?

TSP gives this question a concrete landscape in which it can be tested.

## The TSP Phase Space

In a nonlinear dynamical systems framing, the relevant object is not merely the distance matrix. The relevant object is the phase space of possible tours.Define the tour space

$$\mathcal{S}_n = \{\pi : \pi \text{ is an admissible Hamiltonian cycle over } V\}.$$

Each element of $\mathcal{S}_n$ is a finite symbolic trajectory. The cost function

$$L : \mathcal{S}_n \to \mathbb{M}$$

maps each tour to a measured length, where $\mathbb{M}$ denotes the space of measured finite quantities.A search algorithm may then be treated as a dynamical process

$$\Phi : \mathcal{S}_n \to \mathcal{S}_n,$$

where $\Phi$ moves from one tour to another according to some admissible transformation rule. Examples include city swaps, edge reversals, 2-opt moves, 3-opt moves, Lin-Kernighan style transformations, genetic recombination, simulated annealing transitions, branch-and-bound decisions, or cutting-plane refinements.In this framing, a tour is not merely a candidate answer. It is a state in a finite symbolic dynamical system.

$$(\mathcal{S}_n, L, \Phi)$$

is therefore the core TSP dynamical object.

## Local and Global Attractors

A local attractor is a tour that cannot be improved under a declared local transformation rule. Let $\mathcal{R}$ be a neighbourhood rule, such as all allowed 2-opt moves. A tour $\pi$ is locally stable under $\mathcal{R}$ if

$$L(\pi) \leq L(\rho(\pi)) \quad \text{for all} \quad \rho \in \mathcal{R}.$$

This defines a local attractor:

$$A_{\text{local}} = \Big\{\pi \in \mathcal{S}_n : L(\pi) \leq L(\rho(\pi)) \text{ for all } \rho \in \mathcal{R}\Big\}.$$

However, local stability does not imply global optimality. A tour may be stable under 2-opt moves while still being far from the shortest possible tour. This is because the chosen rule $\mathcal{R}$ defines the basin being tested.The global attractor is the tour or set of tours satisfying

$$A_{\text{global}} = \Big\{\pi^\star \in \mathcal{S}_n : L(\pi^\star) \leq L(\pi) \text{ for all } \pi \in \mathcal{S}_n\Big\}.$$

The difficulty of TSP lies in the gap between these two statements. It is often easy to find a locally stable trajectory. It is difficult to prove that this trajectory is globally stable across the whole admissible symbolic space.This gives a compact Geofinitist interpretation:

> The Travelling Salesman Problem is the search for a stable closed functional symbolic trajectory through a finite node space, where admissible transitions are constrained by a cost relation and the solution is the attractor that minimises symbolic drag across the completed path.

## Proof as Global Basin Exclusion

An important distinction now appears. Finding a short tour is not the same as proving that it is the shortest tour.Let a candidate tour have length

$$U = L(\pi_c).$$

This gives an upper bound on the optimum, since the true optimum cannot be longer than an existing valid tour:

$$L(\pi^\star) \leq U.$$

To prove optimality, one must also establish a lower bound

$$B_{\text{lower}} \leq L(\pi^\star).$$

If the lower bound and upper bound meet, then optimality is certified:

$$B_{\text{lower}} \sim U \mid (C, \alpha, H, \delta).$$

The use of $\sim$ here is deliberate. In a measured Geofinitist setting, the statement is not a metaphysical equality. It is a declared symbolic closure under finite representation, declared uncertainty, accepted procedure, and agreed admissibility conditions.Exact TSP methods such as branch-and-bound, branch-and-cut, subtour elimination, and integer programming relaxations may therefore be understood as global basin-exclusion procedures. They do not merely seek a good attractor. They attempt to prove that no deeper attractor remains hidden elsewhere in the admissible landscape.This is the crucial point:

> A TSP proof of optimality is a second symbolic trajectory constructed around the first. The first trajectory is the proposed tour. The second trajectory excludes all possible better tours.

The proof is therefore not simply attached to the solution. It is part of the measured construction of the solution.

## The Relation to P vs. NP

TSP is relevant to the $P$ vs. $NP$ problem because the decision version lies in $NP$, and the general problem is one of the canonical hard combinatorial problems associated with NP-completeness and NP-hardness.In the classical framing, the question becomes:

> If a TSP tour can be verified efficiently, can one always be found efficiently?

In the Geofinitist framing, this becomes:

Do construction and verification trajectories remain finitely separable under measured conditions?

For each input size $n$, let

$$\hat{T}_S(n)$$

be the measured solver trajectory, and let

$$\hat{T}_V(n)$$

be the measured verifier trajectory.Define the observed finite separability gap as

$$\widehat{\Delta}_{TSP}(n) = \widehat{\alpha}_S(n) - \widehat{\alpha}_V(n),$$

where

$$\widehat{\alpha}_S(n) = \frac{\Delta \log \hat{T}_S(n)}{\Delta \log n}, \quad \widehat{\alpha}_V(n) = \frac{\Delta \log \hat{T}_V(n)}{\Delta \log n}.$$

If $\widehat{\Delta}_{TSP}(n)$ remains persistently positive across measured ranges and perturbation regimes, then the solver and verifier trajectories are finitely separated at the available resolution. If the gap collapses for a particular representation, algorithmic pathway, or class of instances, then the apparent hardness may belong to the representation rather than to the measured problem family itself.This does not resolve the classical $P$ vs. $NP$ problem. Rather, it gives a practical programme for investigating where the classical distinction appears in finite measured computation.

## Representation and the Search for a General Solution

A central question follows:

> Is TSP difficult because the attractor is intrinsically unreachable, or because the usual symbolic representation fails to expose the basin structure?

Classically, the problem is presented as a permutation search. But a permutation list is only one representation. The same problem may also be represented as:

- a complete weighted graph,

- a distance matrix,
- an edge-incidence vector,
- a geometric point cloud,
- a sequence of local angular and distance relations,
- a relaxation in a continuous or polyhedral space,
- a dynamical flow over candidate tours,
- a delayed or embedded trajectory in a reconstructed phase space.

Each representation changes the symbolic geometry of the problem. In the language of this work, each representation changes the container through which the trajectory is measured.A general solution programme should therefore not begin by asking only:

How do we search all tours faster?

It should also ask:

Can the problem be embedded so that the attractor basin becomes visible?

This is a different kind of question. It treats the solution not as a hidden object waiting inside a list of permutations, but as a stable structure in a finite symbolic landscape.A possible general investigation can be written as follows.Let

$$E : \mathcal{S}_n \to \mathcal{X}_n$$

be an embedding from the tour space into a reconstructed space $\mathcal{X}_n$. The aim is to find an embedding $E$ and a dynamical process $\Phi_E$ such that

$$\Phi_E(x) \to E(\pi^\star)$$

for a large and clearly defined class of instances.A strong version of such a result would require:

1. a finite construction of $E$,
2. a polynomial bound on the construction of $E$,
3. a polynomial bound on the movement of $\Phi_E$,
4. a proof that the resulting attractor corresponds to the optimal tour,
5. a certificate that no excluded region contains a lower-cost tour.

The last condition is essential. Without it, the process may only be a powerful heuristic. With it, the process becomes a proof-bearing method.

## A Measured Research Protocol

A Geofinitist investigation of TSP should therefore record not only success or failure, but the full measured trajectory of the computation.For each instance family $\Pi$, define an instance registry

$$\mathcal{I}_n = \{I_{n,1}, I_{n,2}, \ldots, I_{n,k}\},$$

with recorded provenance:

$P_{\mathcal{I}} = (\text{generation rule}, \text{metric type}, \text{coordinate precision}, \text{perturbation process}, \text{hardware context}).$

For each solver $\mathsf{A}$, record

$$\mathsf{A}(I) = (v_T(I), \varepsilon_T(I), U(I), B_{\text{lower}}(I), P_{\mathsf{A}}),$$

where $v_T(I)$ is the measured runtime, $\varepsilon_T(I)$ is the uncertainty or variability in runtime, $U(I)$ is the best tour length found, $B_{\text{lower}}(I)$ is the strongest lower bound obtained, and $P_{\mathsf{A}}$ is the procedural provenance of the solver.Define the optimality gap

$$G(I) = U(I) - B_{\text{lower}}(I).$$

In a finite measured setting, the proof status of an instance may be expressed as

$$G(I) \sim 0 \mid (C, \alpha, H, \delta).$$

This says that the candidate solution and the lower-bound proof have closed within the declared measurement and symbolic regime.The research programme then becomes the study of how $G(I)$ behaves across:

- increasing $n$,
- different metric types,
- different geometric dimensions,
- random and structured instance families,
- perturbations of city position or edge weight,
- different symbolic encodings,
- different dynamical search rules.

The goal is not merely to find good tours. The goal is to understand the basin structure of the problem and the conditions under which proof-bearing closure becomes possible.

## Perturbation and Basin Stability

Perturbation is especially important. Let

$$\mathsf{P}_\eta(I)$$

be a perturbation operator applied to an instance $I$, where $\eta$ controls the perturbation scale. This may shift city locations, alter edge weights within uncertainty bounds, rescale local clusters, or modify the symbolic encoding.The perturbed optimum is

$$\pi_\eta^\star = \arg\min_{\pi \in \mathcal{S}_n} L_\eta(\pi).$$

The stability question is then:

$$\pi_\eta^\star \sim \pi^\star \quad \text{for small } \eta?$$

If the optimum changes rapidly under small perturbations, then the problem landscape is locally fragile. If the optimum remains stable across perturbations, then the attractor basin has measurable robustness.This allows a distinction between several kinds of difficulty:

- **Search difficulty:** the attractor is stable but difficult to locate.
- **Proof difficulty:** a good attractor is found but difficult to certify.
- **Fragility difficulty:** small perturbations alter the attractor structure.
- **Representation difficulty:** the attractor is obscured by the chosen symbolic encoding.

These are different phenomena, but classical complexity often compresses them into a single asymptotic label.

## Toward a General Attractor Method

A general attractor method for TSP would attempt to reconstruct the hidden geometry of the problem rather than merely enumerate the possible tours.Such a method might proceed through the following stages:

1. Construct a finite symbolic representation of the instance.
2. Embed the representation into a higher-dimensional or reconstructed phase space.
3. Identify candidate low-drag edges or sub-trajectories.
4. Allow local and global trajectory rules to evolve candidate tours.
5. Detect stable attractors under repeated perturbation.

6. Construct lower-bound exclusions around the remaining landscape.
7. Close the proof when the upper and lower bounds meet within declared uncertainty.

This gives a bridge between heuristic search and formal proof. A heuristic attractor without basin exclusion remains incomplete. A proof without measured trajectory information may be formally valid but gives little insight into the shape of the problem. The desired method would combine both.In this sense, TSP may be used as a laboratory for the wider claim that mathematics can be modelled as a nonlinear dynamical system. The symbolic objects do not merely sit in a static space. They move under rules, stabilise into basins, resist perturbation, and sometimes require new representations before their attractors become visible.

## Implications for the Wider $P$ vs. $NP$ Question

If a general polynomial-time solution were found for the decision version of TSP, and if that solution applied to the fully general NP-complete form, then it would have major consequences for the classical $P$ vs. $NP$ problem. In the standard theory, a polynomial-time solution to any NP-complete problem would imply polynomial-time solutions to all problems in $NP$.However, the Geofinitist framing makes a further distinction. A finite measured method may work across large practical regimes without resolving the asymptotic classical question. Conversely, a classical proof may classify infinite scaling behaviour without describing how difficulty appears in measured finite computation.This means that the research can proceed in two layers:

$$\text{Classical layer:} \quad P \stackrel{?}{=} NP.$$

$$\text{Measured layer:} \quad \hat{T}_S(n) \stackrel{?}{\sim} \hat{T}_V(n) \mid (C, \alpha, H, \delta).$$

The classical layer asks about idealised asymptotic equivalence. The measured layer asks whether solver and verifier trajectories become indistinguishable, separable, or conditionally separable within finite regimes.TSP is valuable because both layers can be studied together. It has a precise classical formulation, a rich proof history, many practical solvers, many geometric variants, and a naturally visual attractor landscape.

## Conclusion

The Travelling Salesman Problem is more than a shortest-route puzzle. It is a canonical example of finite symbolic trajectory selection under constraint.Classically, it illuminates the difference between efficient verification and difficult construction. Geofinitistically, it becomes an attractor-finding problem in a finite symbolic phase space.The candidate tour is one trajectory. The proof of optimality is another. A complete solution requires not only the discovery of a stable path, but the exclusion of all deeper paths across the admissible

landscape.This reframes TSP as a practical and conceptual bridge into the $P$ vs. $NP$ problem. It allows the original question to be investigated through measured separability, perturbation, representation, basin stability, and proof-bearing attractor closure.The resulting research question is therefore not simply: Can we solve TSP efficiently? is: Can we reconstruct the finite symbolic landscape so that the global attractor becomes reachable and provable? question does not replace the classical problem. It gives a finite route into it.

# Solving, Verification, and the Measurement Boundary

## Overview

The previous discussion considered the Travelling Salesman Problem as a finite attractor-finding problem and as a useful bridge into the wider $P$ vs. $NP$ question. However, a deeper issue now appears. The classical formulation of $P$ vs. $NP$ appears complete because it is framed within an ideal symbolic system. Within that system, a solution is a finite string, a verifier is a rule-governed procedure, and time is counted as idealised computational steps.From the perspective of Finite Symbolic Mechanics (FSM), this framing is not false, but it is incomplete. It hides a crucial distinction between two different symbolic acts:

$$\text{solving} \quad \text{and} \quad \text{verification.}$$

These are not merely two computational timings of the same underlying process. They are different kinds of finite symbolic trajectory.Solving is an endogenous construction trajectory. It takes place within the symbolic rule space of the problem and produces a candidate object, path, proof, certificate, or configuration.Verification is an exogenous measurement trajectory. It compares a proposed candidate against declared admissibility conditions, rule constraints, bounds, or acceptance criteria.This distinction is central. A measurement is never the thing itself. A measurement is a finite symbolic interaction with the thing, and therefore carries uncertainty, provenance, resolution, and admissibility conditions. In FSM, verification is therefore not identical with the solution. It is a measured relation to the proposed solution.This means that the $P$ vs. $NP$ problem may not be quite what it first appears to be. It is not only a question about computation. It is also a question about the commitments made by the language in which the problem is formulated.

## The Classical Compression

The classical question is usually stated as:

$$P \stackrel{?}{=} NP.$$

In plain terms, this is often expressed as:

> If a proposed solution can be verified in polynomial time, can a solution also be found in polynomial time?

This formulation is elegant and powerful. It has generated a vast and fruitful body of work in theoretical computer science. However, it also compresses several distinct symbolic acts into a single comparison.It treats the act of finding a solution and the act of checking a proposed solution as comparable operations within one formal basin. Both are treated as procedures over finite strings. Both are assigned asymptotic time complexity. Both are idealised as exact symbolic operations.From within the classical frame, this is a legitimate abstraction.From within FSM, it is a commitment.The hidden assumption is not merely that the two processes can be timed. The hidden assumption is that solving and verification can be compared as if they are operations of the same symbolic kind.FSM questions that assumption.

## Solving as Endogenous Construction

Let $I$ be an instance of a problem, and let $s$ be a proposed solution or certificate.A solving process may be written as

$$\mathsf{Solve}(I) \to s.$$

This notation already hides a great deal. The solver does not simply reveal a pre-existing object. It follows a symbolic trajectory through a space of admissible transformations, constraints, representations, and possible intermediate states.In FSM terms, solving is endogenous because it is generated within the internal rule structure of the problem. It is a construction trajectory.For the Travelling Salesman Problem, the solver must construct a tour:

$$\mathsf{Solve}_{TSP}(D) \to \pi,$$

where $D$ is the distance or cost structure and $\pi$ is a candidate tour.For SAT, the solver must construct a truth assignment.For a proof problem, the solver must construct a proof sequence.For an optimisation problem, the solver must construct a candidate optimum.In each case, the solver is not merely measuring a given object. It is producing a finite symbolic trajectory that did not previously exist in explicit form within the computation.Thus solving is a generative act within a symbolic container.

## Verification as Exogenous Measurement

Verification has a different structure. It does not construct the solution from the instance alone. It receives both the instance and the proposed solution:

$$\mathsf{Verify}(I, s) \to \{\text{accept}, \text{reject}\}.$$

This is a different symbolic act. The verifier measures the relation between $I$, $s$, and a declared rule of admissibility.In FSM notation, this should not be treated as exact identity.

It should be written as a finite acceptance relation:

$$\mathsf{Verify}(I, s) \sim \text{accepted} \mid (C, \alpha, H, \delta),$$

where $C$ is the consensus or admissibility constraint, $\alpha$ is the finite symbolic limit, $H$ is the historical and procedural provenance, and $\delta$ is the uncertainty associated with finite representation and measurement.The symbol $\sim$ is not being used here as a loose approximation. It marks the fact that the relation is a finite symbolic trajectory held under declared conditions. It signals that the acceptance is not metaphysical identity with the thing itself. It is a measured symbolic closure.Verification is therefore not the same as solving. It is a measurement of the proposed solution under a declared frame.This distinction may appear subtle, but it reaches deeply into the foundations of Geofinitism. A measurement can never be the object measured. A symbolic verification can never be the endogenous construction itself. It can only compare, accept, reject, bound, or stabilise the relation between representation and rule.

## The Theory and Experiment Analogy

The distinction between solving and verification resembles the distinction between theory and experiment.A theory is an endogenous symbolic construction. It is built within a language, a model, a rule system, and a set of inherited assumptions. It generates predictions, explanations, and internal relations.An experiment is an exogenous measurement trajectory. It tests some relation between the symbolic construction and measured interaction with the world.The experiment does not become the theory. The measurement does not become the object. The observation does not become the representation. Instead, the experiment provides finite symbolic evidence under declared constraints, uncertainty, instrumentation, and provenance.This gives a useful analogy:

$$\text{theory} \quad \text{is to} \quad \text{experiment}$$

as

$$\text{solution} \quad \text{is to} \quad \text{verification}.$$

The analogy is not exact, but it reveals the structural issue. In both cases, an endogenous symbolic construction is compared with an exogenous measurement or acceptance process.A theory may be supported by measurement, but it is not proved with absolute certainty by measurement.Likewise, a proposed solution may be accepted by a verifier, but the verification is not identical with the act of solving.This suggests that the $P$ vs. $NP$ question is not merely asking whether two complexity classes are equal. It is also asking whether construction and measurement can be placed into a single exact symbolic comparison.In FSM, they cannot be equated absolutely. They can only be related through finite admissibility:

$$\mathsf{Construction} \sim \mathsf{Measurement} \mid (C, \alpha, H, \delta).$$

## The Problem of Hidden Commitments

The classical formulation of $P$ vs. $NP$ appears complete because it is formed inside an ideal symbolic basin. Within that basin:

- symbols are exact,
- strings are finite but unambiguous,
- machines are ideal,
- time is counted as abstract steps,
- verification returns accept or reject,
- uncertainty is not part of the formal object,
- measurement is not treated as foundational.

These commitments are not errors. They are part of the strength of the classical formulation. They allow a clean mathematical theory to be built.However, from the FSM perspective, they are also compressions. They remove the measurement boundary from view.The formulation hides:

- the construction of the input,
- the representation of the instance,
- the provenance of the symbolic form,
- the finite physical act of computation,
- the distinction between a certificate and its measurement,
- the uncertainty of real computation,
- the difference between endogenous formal validity and exogenous observed acceptance.

Thus the classical question may be formally complete inside its own basin, while still being incomplete as a Geofinitist account of computation.The question is therefore not only:

$$P \stackrel{?}{=} NP.$$

It is also:

What exactly is being compared?

More precisely:

$$\mathsf{Solve}(I) \sim ?\mathsf{Verify}(I, s) \mid (C, \alpha, H, \delta).$$

From the FSM standpoint, this comparison cannot be absolute. The two sides are different symbolic acts. They may be related, timed, bounded, or compared within an agreed formal system, but they cannot be made identical without hiding the measurement boundary.

## The Travelling Salesman Problem Revisited

The Travelling Salesman Problem makes this issue especially clear.A proposed tour can be checked quickly. Given a tour $\pi$, one can verify that each city is visited once and that the tour returns to the starting point. One can also calculate the tour length:

$$L(\pi) = \sum_{i=1}^{n} d_{\pi_i \pi_{i+1}}, \quad \pi_{n+1} = \pi_1.$$

This is a verification trajectory.However, this does not necessarily verify that $\pi$ is the shortest possible tour. It verifies only a relation between the proposed tour and a declared condition.For example, if the decision question is:

$$\exists \pi \quad \text{such that} \quad L(\pi) \leq B,$$

then a candidate tour can be checked efficiently against the bound $B$.But if the claim is:

$$\pi = \pi^\star,$$

where $\pi^\star$ is the global optimum, then verification requires more than measuring the candidate tour. It requires proof that no other tour has lower cost:

$$L(\pi^\star) \leq L(\pi) \quad \text{for all} \quad \pi \in \mathcal{S}_n.$$

This is not the same verification act.In the earlier language, checking a tour is a local acceptance measurement. Proving optimality is global basin exclusion.Thus TSP exposes a hidden ambiguity in the word verification.There is a difference between:

- verifying that a proposed route is a valid tour,
- verifying that a proposed route lies below a declared bound,
- verifying the measured length of a proposed route,
- proving that the proposed route is globally optimal.

These are different symbolic trajectories. They should not be collapsed into a single word without declaring the frame.

## Verification Is Not the Thing Verified

The central Geofinitist point may be stated plainly:

> Verification is not the thing verified.

A measurement is not the measured object. A proof-checking procedure is not the proof itself. A certificate-checking procedure is not the construction of the certificate. A tour-length calculation is not the discovery of the optimal tour.This does not make verification unimportant. Verification is essential. But it is not identical with solving.Thus, in FSM, the phrase

solved in polynomial time

can never be treated as identical to

verified in polynomial time.

They are different finite symbolic trajectories. Their relationship must be declared, not assumed.This gives a refined formulation:

$$\mathsf{Solved}_{\mathrm{poly}} \not\equiv \mathsf{Verified}_{\mathrm{poly}}.$$

At best, one may write:

$$\mathsf{Solved}_{\mathrm{poly}} \sim \mathsf{Verified}_{\mathrm{poly}} \mid (C, \alpha, H, \delta),$$

but only after specifying the symbolic commitments that allow the comparison.The use of $\sim$ is again important. It does not mean approximately equal. It means that a finite symbolic trajectory has been stabilised under declared admissibility conditions.

## Reframing the P vs. NP Problem

The $P$ vs. $NP$ problem is usually treated as a problem about computational classes. FSM does not deny this formal framing. Instead, it asks what is lost when the problem is lifted away from measured symbolic construction.The classical question is:

$$P \stackrel{?}{=} NP.$$

The FSM question is:

Can endogenous construction and exogenous verification be treated as the same kind of symbolic trajecto

The answer is no, not absolutely.They can be compared only under declared rules of representation, measurement, uncertainty, provenance, and admissibility.Therefore, the FSM reframing is:

$$P \stackrel{?}{=} NP \quad \longrightarrow \quad \mathsf{Construction} \sim ?\mathsf{Verification} \mid (C, \alpha, H, \delta).$$

This changes the nature of the inquiry.The question is no longer only whether every efficiently verifiable formal problem has an efficient construction procedure. It also becomes a question of how the language of the problem has already shaped what counts as a solution, what counts as verification, and what kind of certainty is being claimed.

## A Possible Source of Incompleteness

The classical formulation appears complete because its definitions are internally precise. But internal precision is not the same as foundational completeness.FSM suggests that the formulation may be incomplete at the level of symbolic commitment.The key missing distinctions are:

- solution versus certificate,
- construction versus recognition,
- verification versus measurement,
- measured acceptance versus identity,
- local admissibility versus global closure,
- formal exactness versus finite symbolic uncertainty.

Once these distinctions are restored, the problem looks different.It may not be sufficient to ask whether solving and verification have the same asymptotic complexity. One must first ask what allows them to be placed on the same symbolic axis.In classical theory, this axis is supplied by idealised computation.In FSM, the axis must be reconstructed through finite measurement.

## Conclusion

The $P$ vs. $NP$ problem is not merely a technical question about computation. It is also a question about the language of solution, verification, measurement, and certainty.The classical formulation compares problems solvable in polynomial time with problems verifiable in polynomial time. This comparison is powerful within the formal basin of theoretical computer science. However, from the perspective of Finite Symbolic Mechanics, solving and verification are not the same kind of act.Solving is an endogenous construction trajectory.Verification is an exogenous measurement trajectory.A measurement can never be the thing itself. A verification can never be identical with the construction it tests. The two may be related, but only through declared symbolic commitments:

$$\mathsf{Construction} \sim \mathsf{Measurement} \mid (C, \alpha, H, \delta).$$

This does not dissolve the classical problem. Instead, it opens a deeper layer beneath it. The question becomes not only whether $P = NP$, but whether the formulation of $P$ vs. $NP$ has hidden the measurement boundary that makes the comparison appear complete.The Travelling Salesman Problem helps expose this boundary. Checking a proposed tour, checking a bound, measuring a length, and proving global optimality are not the same act. They are different symbolic trajectories, each requiring its own admissibility conditions.Thus, in FSM, the solution to the $P$ vs. $NP$ problem may not be what it first appears to be. It

may lie not only in discovering a new algorithm or proving a separation, but in clarifying the commitments made by the language of the problem itself.The formulation appears complete. But from the Geofinitist perspective, it is not yet complete, because it has not accounted for measurement.

## References

Cook, Stephen A. “The Complexity of Theorem-Proving Procedures.” Proceedings of the Third Annual ACM Symposium on Theory of Computing, 1971, pp. 151–158.Levin, Leonid A. “Universal Search Problems.” Problems of Information Transmission, vol. 9, no. 3, 1973, pp. 265–266.Karp, Richard M. “Reducibility Among Combinatorial Problems.” In Complexity of Computer Computations, edited by Raymond E. Miller and James W. Thatcher, Plenum Press, 1972, pp. 85–103.Garey, Michael R., and David S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman, 1979.Cook, Stephen. “The P versus NP Problem.” Clay Mathematics Institute Millennium Prize Problem statement.Dantzig, George, Ray Fulkerson, and Selmer Johnson. “Solution of a Large-Scale Traveling-Salesman Problem.” Operations Research, vol. 2, no. 4, 1954, pp. 393–410.Held, Michael, and Richard M. Karp. “A Dynamic Programming Approach to Sequencing Problems.” Journal of the Society for Industrial and Applied Mathematics, vol. 10, no. 1, 1962, pp. 196–210.Held, Michael, and Richard M. Karp. “The Traveling-Salesman Problem and Minimum Spanning Trees.” Operations Research, vol. 18, no. 6, 1970, pp. 1138–1162.Papadimitriou, Christos H. “The Euclidean Traveling Salesman Problem is NP-Complete.” Theoretical Computer Science, vol. 4, no. 3, 1977, pp. 237–244.Christofides, Nicos. “Worst-Case Analysis of a New Heuristic for the Travelling Salesman Problem.” Graduate School of Industrial Administration, Carnegie Mellon University, Technical Report 388, 1976.Arora, Sanjeev. “Polynomial Time Approximation Schemes for Euclidean Traveling Salesman and Other Geometric Problems.” Journal of the ACM, vol. 45, no. 5, 1998, pp. 753–782.Applegate, David L., Robert E. Bixby, Václav Chvátal, and William J. Cook. The Traveling Salesman Problem: A Computational Study. Princeton University Press, 2006.

# Chapter 3

# The P vs NP Challenge and the Missing Axiom of Measurement

### A Geofinitist Analysis of the Clay Formulation

Building on the preceding analysis of the measurement boundary and the explicit initialisation of functional symbolic trajectories, this essay turns to the Clay formulation itself and asks whether the challenge has been fully initialised with respect to finite measurement.

## 1. Purpose of This Essay

The purpose of this essay is not to offer a conventional solution to the $P$ vs $NP$ problem. It does not attempt to prove $(P = NP)$, nor does it attempt to prove $(P \neq NP)$ inside the accepted language of classical computational complexity theory.

Instead, the aim is prior.

The aim is to examine the symbolic framework in which the Clay formulation of the $P$ vs $NP$ problem is stated, and to ask whether that framework has been fully initialised. The concern is that the challenge, as normally presented, appears to speak about actual computation while being formally set inside an ideal mathematical space that has no axiom of measurement.

That distinction matters.

If the problem is only a problem inside classical formal language theory, then its rules are those of the classical basin. The accepted objects are formal languages, Turing machines, finite strings, checking relations, polynomial bounds, and asymptotic classifications. In that space, the question is clean:

$$P \stackrel{?}{=} NP.$$

But if the problem is also being presented as a problem about actual computation, feasible calculation, cryptography, proof search, artificial intelligence, and real processes carried out by machines, then a second frame has entered the discussion. That second frame is the measured world.

The central Geofinitist claim is that these two frames cannot be silently conflated.

The Clay formulation moves between finite computational objects and infinite mathematical commitments. It begins with finite symbols, finite strings, finite descriptions of machines, and finite stepwise processes. It then moves into infinite languages, all input lengths, universal polynomial bounds, exact class membership, and exact equality or inequality between symbolic classes.

Classical mathematics permits this movement. Geofinitism asks whether the bridge has been declared.

The answer appears to be no.

The missing bridge is an axiom of measurement.

Without such an axiom, any classical solution to $P$ vs $NP$ would be a solution inside the classical formal basin. It would not be, by itself, a solution to measured computation, because measured computation is not part of the formal object being adjudicated.

This is the fracture examined here.

## 2. The Initialisation Problem

Every serious symbolic trajectory must begin with an initialisation. Before a claim can be assessed, the conditions under which the claim is being made must be declared.

A formal mathematical problem may look as though it has escaped ordinary language. It may appear to begin with symbols, definitions, and equations rather than with narrative. But this is misleading. Formal symbols do not remove narrative. They compress narrative.

The symbols carry commitments.

They carry decisions about what objects are admissible, what counts as equality, what counts as proof, what counts as computation, what counts as verification, and what kind of infinity is allowed.

In the $P$ vs $NP$ problem, this is especially important because the challenge is not a purely internal theorem about an abstract object. It is repeatedly motivated by computation. The problem asks about algorithms, checking, solving, feasibility, certificates, and machine processes. It is therefore a bridge problem.

It stands between two spaces.

The first is formal symbolic acceptance-space. This is the space of languages, finite alphabets, strings, Turing machines, acceptance states, checking relations, and asymptotic polynomial bounds.

The second is measured computation-space. This is the space of actual finite machines, memory limits, energy, hardware states, noise, runtime variability, physical representation, uncertainty, and performed calculation. The problem is commonly spoken about as though these spaces are naturally aligned. The Geofinitist claim is that they are not naturally aligned. They require a bridge. However, the bridge cannot be assumed. It must be initialised.

## 3. The Clay Challenge as a Formal Symbolic Construction

The Clay formulation, following Cook's presentation, begins by declaring that to define the $P$ vs $NP$ problem precisely, it is necessary to give a formal model of a computer. This is already a significant act of initialisation. The problem does not begin with actual computers. It begins with a formal model of computation.

The chosen model is the Turing machine. This choice is historically powerful and mathematically productive. It gives computational complexity theory a precise symbolic container. Within that container, computation can be treated as a sequence of formal steps. A machine accepts or rejects a string. A runtime can be counted. A class of languages can be defined. This produces the formal space in which the classical problem lives.

A finite alphabet $\Sigma$ is declared. The set $\Sigma^*$ is introduced as the set of all finite strings over that alphabet. A language is then a subset of $\Sigma^*$. A Turing machine $M$ has an associated input alphabet. For each string $w$, there is a computation of $M$ on $w$. The machine accepts $w$ if the computation terminates in an accepting state.

This gives the accepted language:

$$L(M) = \{w \in \Sigma^* \mid M \text{ accepts } w\}.$$

The runtime of the machine on a string $w$ is written as:

$$t_M(w).$$

If the computation never halts, then:

$$t_M(w) = \infty.$$

For each input length $n$, the worst-case runtime is defined as:

$$T_M(n) = \max\{t_M(w) \mid w \in \Sigma^n\}.$$

A machine runs in polynomial time if there is some $k$ such that:

$$T_M(n) \leq n^k + k$$

for all $n$.

The class $P$ is then the class of languages accepted by some Turing machine running in polynomial time.

The class $NP$ is defined using a checking relation. A language $L$ is in $NP$ if there exists a polynomial-time checking relation $R$ and a polynomially bounded certificate $y$ such that:

$$w \in L \iff \exists y \ (|y| \leq |w|^k \text{ and } R(w, y)).$$

The formal problem is then stated:

$$P \stackrel{?}{=} NP.$$

Within classical complexity theory this is precise, elegant, and powerful. The issue is not that the formal construction is careless. It is not careless. It is highly disciplined inside its own chosen basin. The Geofinitist question is different.

- What has this construction assumed?

- What has it omitted?

- What kind of object is it actually asking about?

- And what happens when this formal object is interpreted as a claim about actual computation?

## 4. The First Geofinitist Observation: A Symbol Is Finite

The first measurable truth available to the writer and reader is that the claim itself appears in finite symbols. Then we can identify and measure that:

- A symbol has extent.

- A proof has extent.

- A page has extent.

- A string has extent.

- A diagram has extent.

- A program has extent.

- A mathematical submission has extent.

A computation, when performed, has duration, representation, energy cost, and physical implementation. This is not merely a philosophical preference. It is the only shared measurable condition under which mathematical communication occurs.

A classical response may distinguish between the physical token and the abstract type. It may say that this printed mark has extent, but the formal object does not. The printed "A" is finite, but the abstract symbol $A$ is not dependent on this particular mark.

The Geofinitist reply is direct:

**What does the formal object depend upon if not finite language, finite notation, finite rules, finite memory, finite comparison, finite teaching, finite transmission, and finite symbolic reconstruction?**

If the answer is "axioms," then those axioms must be finitely written or finitely encoded.

If the answer is "rules of inference," then those rules must be finitely stated, recognised, and applied.

If the answer is "a formal system," then that formal system must be specified through finite symbolic means.

If the answer is "mathematical consensus," then the object depends on a historical and communal process of symbolic stabilisation.

If the answer is "a mind-independent abstract realm," then the object has left measurement. It may be asserted, but it cannot be directly compared, verified, transmitted, or used without returning through finite symbols.

This does not require a crude denial of abstraction. It requires a more careful account of abstraction.

The formal object is not encountered in an idealised Platonic space. It is always accessed through finite symbolic trajectories. Therefore, any claim about a formal object must pass through finite symbols before it can be read, compared, accepted, or rejected. This is the first axiom of the Geofinitist approach.

## 5. Axiomatic Core of the Geofinitist Reading

The following axioms or commitments clarify the Geofinitist position.

**1 - Finite Symbol Axiom**: Every communicable symbol is finite in its presentation, recognition, and use. A symbol may point beyond itself. It may compress an extended trajectory of meaning. It may participate in formal systems that invoke infinite objects. But the symbol as used, read, written, stored, transmitted, or compared is finite.

**2 - Symbolic Extent Axiom**: No operative symbol is without extent. A symbol may be idealised as dimensionless within a formal model, but its operative existence depends on

finite symbolic manifestation. If it cannot be marked, encoded, remembered, distinguished, or reconstructed, it cannot participate in a proof or claim.

**3 - Finite Claim Axiom**: Every mathematical claim submitted, read, assessed, or accepted is a finite symbolic trajectory. A theorem statement, a proof, a definition, a program, a diagram, a formal derivation, and a Clay Prize submission all appear as finite symbolic structures.

**4 - Proof-Comparison Axiom**: A proof is not accepted by direct contact with an abstract realm. It is compared against an admissibility reference. This comparison may involve formal rules, expert judgement, inherited conventions, proof-checking systems, written definitions, and communal standards. It is rigorous, but it is still a finite symbolic comparison.

**4 - Verification-as-Measurement Axiom**: Verification is measurement in the foundational sense. Measurement is not limited to physical instruments. More generally, measurement is finite comparison against a reference. A proof step is compared with an inference rule. A candidate certificate is compared with a checking relation. A computed output is compared with an acceptance condition. A proposed theorem is compared with admissibility standards. Therefore, verification is a measurement trajectory.

**6 - Measurement Uncertainty Axiom**: Every measurement has uncertainty. This includes exogenous measurement, where a symbolic system compares with the measured world, and endogenous measurement, where one symbolic trajectory is compared with another inside a formal system. The uncertainty may be extremely small, highly controlled, or conventionally suppressed. But in a finite symbolic account it cannot be treated as non-existent without declaring the conditions under which it has been excluded.

**7 - Infinite Commitment Axiom**: Infinity is not a measurable object in the measured world. Infinity may be an admissible object inside classical mathematical systems. It may be a useful symbolic commitment. It may be powerful, coherent, and productive within its basin. But it is not directly measurable. Any use of infinity in a claim about measured computation requires a declared bridge.

**8 - Initialisation Axiom**: Every non-trivial symbolic trajectory must declare the frame under which it is being constructed. In Geofinitist notation this may be represented as $(C, \alpha, H, \delta)$, where $C$ is the consensus or admissibility condition, $\alpha$ is the finite symbolic resolution or Alphonic limit, $H$ is the historical and procedural provenance, and $\delta$ is uncertainty.

**9 - Bridge Axiom**: A claim that moves between formal symbolic space and measured computation-space requires an explicit bridge. If no bridge is declared, the claim may remain valid inside the formal basin, but it cannot automatically be treated as a claim about measured computation.

**10 - Missing Measurement Axiom**: Classical mathematics, as used in the standard $P$ vs $NP$ formulation, does not contain a foundational axiom of measurement. It permits exact formal reasoning over infinite symbolic structures, but it does not require that every

claim be initialised against finite symbolic measurement, uncertainty, provenance, and admissibility. Therefore, a solution obtained inside classical mathematics may not solve the measured-world interpretation of the problem.

## 6. What the Clay Problem Is Actually Looking For

The classical $P$ vs $NP$ challenge asks whether efficient verification can always be converted into efficient solving or deciding.

In plain language, the problem asks: If a proposed answer can be checked efficiently, can an answer also be found efficiently?

In formal language, the problem asks whether every language accepted by some nondeterministic polynomial-time computation is also accepted by some deterministic polynomial-time computation.

The search is therefore for one of two possible formal closures.

The first possible closure is $P = NP$. This would usually be shown by giving a deterministic polynomial-time algorithm for an NP-complete problem, such as SAT or 3-SAT, and proving that it works for all inputs.

The second possible closure is $P \neq NP$. This would require proving that at least one problem in $NP$ cannot be solved by any deterministic polynomial-time algorithm.

The first route requires a construction. The second route requires an exclusion. Both are extraordinarily difficult inside the accepted formal frame. From the Geofinitist perspective, however, there is a prior issue. The challenge is looking for a perfect symbolic answer to a question motivated by computation. But computation, when actual, is measured. If the problem is not about actual computation, then its space must be named more carefully. It is not computation-space. It is formal symbolic acceptance-space. If it is about actual computation, then the missing axiom of measurement becomes unavoidable.

## 7. The Fracture Points in the Clay Formulation

The Clay formulation is coherent inside its formal basin. The fractures appear when the formal expressions are read as though they directly refer to actual computation. The following fracture points are especially important.

### Fracture 1: The Need for a Formal Model of a Computer

The problem begins by saying that to define the question precisely, a formal model of a computer is required. This is already a bridge. The problem is not directly defined over

actual computers. It is defined over a formal model that stands in for computation. That model is historically justified and mathematically powerful, but it is a symbolic replacement. The fracture appears when the formal replacement is treated as though it is identical to computation itself.

**Geofinitist reading**: The statement has already moved from measured computation to a formal symbolic container. The bridge has been crossed, but the crossing has not been named as a measurement issue.

### Fracture 2: Finite Alphabet and Infinite String Space

The formulation begins with a finite alphabet $\Sigma$. This is a finite symbolic object.

Then it immediately introduces $\Sigma^*$, the set of all finite strings over $\Sigma$.

Each individual string is finite, but the set of all such strings is infinite. This is a major transition. The problem begins with finite symbolic materials, then moves into an infinite symbolic container.

**Geofinitist reading**: The finite symbol is measurable. The infinite set of all finite strings is not measurable as a completed object. It is an admissible mathematical commitment, but not a measured object.

### Fracture 3: Language as Subset of $\Sigma^*$

A language is defined as a subset of $\Sigma^*$.

This means a language may be an infinite set of finite strings. Again, each string is finite, but the language as a completed mathematical object may be infinite.

**Geofinitist reading**: The formal object "language" compresses an infinite membership landscape. If the problem is purely formal, this is acceptable inside the classical basin. If the problem is about measured computation, then the infinite language cannot be measured as a completed object.

### Fracture 4: Acceptance as a Process Compressed into Membership

The accepted language of a machine is defined as:

$$L(M) = \{w \in \Sigma^* \mid M \text{ accepts } w\}.$$

This looks like a static set definition. But "accepts" is not static. It is a process. A machine must be run on an input. The computation must terminate in an accepting state. The equation compresses a computational trajectory into a membership condition.

**Geofinitist reading**: The expression $L(M)$ hides a process behind a set. The dynamic act of computation becomes a static symbolic object.

### Fracture 5: Step Count as Ideal Time

The runtime $t_M(w)$ is defined as the number of steps in the computation of $M$ on input $w$.

This resembles measurement, but it is not measured physical time. It is an ideal symbolic count of formal transitions. The fracture appears because "time" in practical computation and "step count" in a formal Turing machine are not the same thing.

Geofinitist reading: The formal model counts ideal transitions. Actual computation involves hardware, energy, memory, architecture, latency, noise, and physical representation. These are not included in $t_M(w)$.

### Fracture 6: Non-Termination Assigned Infinity

If the computation never halts, then $t_M(w) = \infty$.

This is a profound fracture. No finite measurement can observe non-termination to completion. A process that never halts cannot be measured as completed. Yet the formal system assigns it a completed symbolic value: infinity.

**Geofinitist reading**: The symbol $\infty$ is doing work that no measurement can perform. It is a formal classification of an uncompleted trajectory.

### Fracture 7: Worst-Case Runtime Over All Strings of Length $n$

The worst-case runtime is defined as:

$$T_M(n) = \max\{t_M(w) \mid w \in \Sigma^n\}.$$

For a fixed $n$, the set $\Sigma^n$ is finite. But it may be astronomically large. To know the maximum by measurement would require exhaustive testing or proof that all larger cases have been excluded.

**Geofinitist reading**: $T_M(n)$ compresses an entire landscape of possible computations into one scalar. It is a formal object, not necessarily a measured runtime.

**Fracture 8: Universal Polynomial Bound**

A machine runs in polynomial time if there exists $k$ such that $T_M(n) \leq n^k + k$ for all $n$.

The phrase "for all $n$" moves the claim out of finite measurement. No actual computation can test all $n$. The assertion is a universal mathematical commitment.

**Geofinitist reading**: The formal system may assert this universally, but measured computation cannot establish it directly. A proof may establish it inside the formal system, but that proof is itself a finite symbolic trajectory compared against admissibility rules.

**Fracture 9: $P$ as a Class of Languages, Not a Class of Measured Computations**

The class $P$ is a class of languages accepted by polynomial-time Turing machines.

It is not a class of actual computations.

It is not a class of physical machines.

It is not a class of measured runtimes.

It is a class of formal languages under ideal symbolic acceptance conditions.

**Geofinitist reading**: If $P$ is spoken of as "efficient computation," then an extra interpretive bridge has been added. That bridge is not part of the formal definition.

**Fracture 10: Checking Relation as Compressed Measurement**

The class $NP$ is defined through a checking relation $R(w, y)$.

Here $w$ is the input and $y$ is the certificate or witness. The relation checks whether $y$ satisfies the required condition for $w$. This is verification. From a Geofinitist perspective, verification is measurement. It is comparison against a reference. The formal expression $R(w, y)$ compresses an act of comparison into a binary relation.

**Geofinitist reading**: The checking relation hides the measurement structure of verification. It treats the act of comparison as exact inside the formal basin.

**Fracture 11: Existential Witness Versus Construction**

The definition of $NP$ includes:

$$\exists y \ (|y| \leq |w|^k \text{ and } R(w, y)).$$

This says that a witness exists. It does not say how the witness is found. The existential symbol is doing enormous work. It separates existence from construction.

**Geofinitist reading**: The problem asks whether reference-bound measurement of a candidate can be transformed into endogenous construction of a candidate. This is not merely a timing question. It is a question about the relationship between two different symbolic acts.

### Fracture 12: Verification and Solving Placed on the Same Axis

The classical problem compares solving and verification through polynomial-time classification. However, solving and verification are not the same kind of symbolic act. Solving is construction and verification is measurement. The formal problem places them into a shared asymptotic comparison.

**Geofinitist reading**: This comparison may be admissible inside classical formal complexity theory, but its admissibility must be declared if the problem is interpreted foundationally.

### Fracture 13: Polynomial Time and Feasibility

The Clay presentation acknowledges that polynomial time is not literally the same as feasible execution. An $n^{100}$ algorithm may be polynomial but practically useless. Importantly, this is not a minor detail. Rather, it is an explicit seam between the formal frame and the measured world.

**Geofinitist reading**: The formal class "polynomial time" and the measured condition "feasible computation" are not identical. The document itself recognises this, but the classical basin continues to use polynomial time as the central formal proxy.

### Fracture 14: Randomness and Quantum Computation

The discussion of random bits and quantum computers shows that the formal model is not the measured world. Different physical or computational assumptions may change what appears efficiently computable.

**Geofinitist reading**: If the model of computation changes, the meaning of the complexity question changes. Therefore, the model is not neutral. It is a container with rules of admissibility.

### Fracture 15: Practical Consequences Used as Motivation

The problem is motivated by cryptography, proof search, artificial intelligence, optimisation, and practical recognition of good solutions. These are measured-world concerns. However, the formal problem is not framed in measured-world terms.

**Geofinitist reading**: The problem draws its urgency from measured computation-space, but seeks its answer in formal symbolic acceptance-space. This is the central bridge fracture.

## 8. Solving and Verification as Different Symbolic Trajectories

The classical question is often rendered in ordinary language as: If a solution can be verified efficiently, can it also be found efficiently? This wording is clear and useful, but it hides a foundational distinction. Solving and verification are different symbolic acts. Solving is an *endogenous construction trajectory*. It begins with an instance and attempts to generate a solution, certificate, path, proof, assignment, or configuration from within the rule space of the problem.

Verification is an exogenous or reference-bound measurement trajectory. It receives a candidate and compares that candidate with a rule, condition, bound, or admissibility criterion.

This distinction can be written:

$$\text{Construction} \not\equiv \text{Verification}.$$

At best, under declared conditions, one may write:

$$\text{Construction} \sim \text{Verification} \mid (C, \alpha, H, \delta).$$

The symbol $\sim$ does not mean approximate equality. It indicates a stabilised finite symbolic relation under declared admissibility, resolution, provenance, and uncertainty.

The classical $P$ vs $NP$ problem asks whether construction and verification coincide in polynomial-time class terms. Geofinitism asks a prior question:

What permits these two different symbolic trajectories to be placed on the same axis?

If the answer is "the classical formal system permits it," then the question remains inside the classical basin.

If the answer is "because this describes actual computation," then the missing axiom of measurement must be supplied.

## 9. Why the Clay Problem Is a Bridge Problem

Many mathematical problems remain largely inside the classical basin. They make no strong claim about actual physical process. Their objects may be abstract, but their use of abstraction does not constantly return to the measured world for motivation.

The $P$ vs $NP$ problem is different.

It repeatedly invokes computation. It speaks of algorithms, machines, steps, checking, solving, feasibility, certificates, cryptography, proof search, and practical consequences. It is therefore not merely an internal mathematical question. It is a bridge question between formal symbolic machinery and the measured world of computation.

That bridge is what gives the problem its power. It is also what creates the fracture. The bridge is assumed by the classical formulation, but not axiomatically grounded.

Classical mathematics provides exact symbolic closure. Measured computation provides finite processes. The $P$ vs $NP$ problem asks for a perfect formal answer to a question whose importance depends on actual computation. Geofinitism says that this is under-initialised and brings us to the missing term, namely, measurement.

## 10. The Missing Axiom of Measurement

An axiom of measurement would state that any claim about computation, verification, proof, or solution must be grounded in finite symbolic comparison under declared conditions.

Such an axiom would require the following to be made explicit:

1. the finite symbolic form of the claim,
2. the admissibility conditions under which the claim is judged,
3. the resolution at which symbols are stabilised,
4. the provenance of the definitions and transformations used,
5. the uncertainty associated with representation and comparison,
6. the distinction between formal symbolic closure and measured computational performance,
7. the rule by which finite symbols are permitted to carry infinite commitments.

Classical mathematics does not require this as it permits exact formal reasoning over infinite objects. That is the power of the classical basin. But it also means that classical mathematics can answer a question only inside the commitments it has accepted.

The Geofinitist concern is that the Clay challenge asks for a solution in a space that excludes measurement, while drawing its meaning from a world where computation must be measured. Therefore, any answer obtained inside the classical frame will resolve the classical formal problem, but not the measured-world problem.

## 11. The Role of Infinity

Infinity is central to the fracture. The individual symbols in the $P$ vs $NP$ formulation are finite. The alphabet is finite. The machine description is finite. Each input string is finite. Each certificate is finite. Each proof submission is finite. However, the commitments are infinite.

$\Sigma^*$ contains all finite strings.

A language may be an infinite subset of $\Sigma^*$.

The runtime bound must hold for all $n$.

The class $P$ ranges over all languages accepted by some polynomial-time machine.

The class $NP$ ranges over all languages with polynomial-time checking relations and bounded existential certificates.

The question $P = NP$ asks for equality of entire infinite classes.

This is permitted in classical mathematics. Yet infinity is not measurable in the measured world. Geofinitism therefore distinguishes between infinity as an admissible formal symbol and infinity as a measurable object. The former may be used inside a classical basin. The latter is not available. If a problem about actual computation depends on infinity, it must declare how finite measured symbolic acts are permitted to carry that infinite commitment. The Clay formulation does not do this because classical mathematics does not require it and that is the missing bridge.

## 12. What Would a Classical Solution Actually Solve?

Suppose a proof of $P = NP$ were accepted by the mathematical community under Clay rules. From the classical perspective, this would be a solution to the $P$ vs $NP$ problem. Where as from the Geofinitist perspective, it would be a solution to the formal symbolic problem inside the classical basin. It would show that, under the accepted definitions of Turing machines, formal languages, checking relations, and polynomial-time bounds, the two classes coincide, and yet:

- It would not automatically show that real measured computation behaves in the same way.

- It would not automatically solve the problem of finite hardware.

- It would not automatically solve energy cost.

- It would not automatically solve runtime variability.

- It would not automatically solve representation dependence.

- It would not automatically solve uncertainty.

- It would not automatically solve feasibility in the measured sense.

- It would not automatically solve the question of how a finite symbolic proof can perfectly bridge to an infinite class claim about actual computation.

Similarly, suppose a proof of $P \neq NP$ were accepted. That would establish separation inside the classical formal basin. It would not automatically establish all measured-world consequences unless the bridge to measured computation were separately declared.

This is not a dismissal of classical proof. It is a boundary statement. A proof solves the problem in the space in which the problem has been initialised. However:

- If the Clay problem is initialised in formal symbolic acceptance-space, then a Clay solution solves that formal problem.

- If the problem is claimed to speak about measured computation-space, then a further axiom of measurement is required.

## 13. What Is Really Being Challenged?

The Geofinitist challenge is not: Can we solve $P$ vs $NP$ faster?

Nor is it: Can we defeat classical mathematics on its own terms?

The challenge is: What are the conditions under which a $P$ vs $NP$ claim can be made, measured, compared, and accepted? Before asking whether $P = NP$, we must ask:

- What is $P$ as a finite symbolic object?

- What is $NP$ as a finite symbolic object?

- What does equality between these classes mean when the classes contain infinite commitments?

- What is the finite measurement reference for accepting a proof?

- How is verification being distinguished from construction?

How is formal runtime distinguished from measured runtime?

How is feasibility being carried by polynomial time?

How is actual computation being represented by a Turing machine?

How is uncertainty excluded?

How is infinity admitted?

These questions do not destroy the classical problem. They reveal its container.

## 14. A More Careful Initialisation

A Geofinitist initialisation of the $P$ vs $NP$ problem would begin differently.

It would not begin with:

$$P \stackrel{?}{=} NP.$$

It would begin with:

We are constructing a symbolic trajectory concerning computation. The trajectory uses finite symbols, finite machine descriptions, finite strings, and finite proof forms. It also invokes infinite mathematical commitments, including all finite strings, all input lengths, universal bounds, and equality or inequality of infinite classes. Therefore, the frame must declare whether the claim is being made inside formal symbolic acceptance-space, measured computation-space, or a bridge between the two.

If the claim is formal, then the admissibility conditions are those of classical computational complexity theory.

If the claim is measured, then runtime, representation, hardware, uncertainty, finite observation, and proof comparison must be included.

If the claim is a bridge claim, then the bridge must be axiomatically declared.

This gives the initialisation:

$$P \text{ vs } NP \text{ claim} \mid (C, \alpha, H, \delta, B),$$

where $C$ is the consensus or admissibility condition, $\alpha$ is the finite symbolic resolution, $H$ is historical and procedural provenance, $\delta$ is uncertainty, and $B$ is the declared bridge between formal symbolic space and measured computation-space. The Clay formulation supplies $C$ for the classical basin. It does not supply $B$. That is the missing axiom of measurement.

## 15. The Central Conclusion

The $P$ vs $NP$ problem is not meaningless. It is not carelessly stated. It is a powerful and precise classical problem inside the formal symbolic basin of computational complexity theory. However, it is also a bridge problem.

It draws meaning from actual computation, feasible calculation, checking, solving, cryptography, proof search, artificial intelligence, and practical machine process. Yet the formal challenge is set inside ideal symbolic acceptance-space, where infinity, exact equality, universal quantification, and perfect proof closure are admissible. This creates a fracture. The challenge asks for a perfect mathematical answer about a problem whose meaning repeatedly points toward finite measured computation, while the formal frame contains no axiom of measurement. Therefore, from a Geofinitist perspective, any classical solution to the Clay

problem would solve only the classical formal problem. It would not, by itself, solve the measured-world problem of computation.

The classical problem may be open. However, the Geofinitist claim is that it is also under-initialised. The missing initialisation is the finite measurement reference. Without it, the problem asks mathematics to provide exact closure while leaving unexamined the finite symbolic conditions by which any such closure must be written, compared, verified, and accepted.

## 16. Final Statement

- A symbol is finite.

- A proof is finite.

- A computation, when actual, is finite and measured.

- Verification is finite comparison against a reference.

- A mathematical claim is accepted through finite symbolic comparison.

- Infinity may be admitted inside a formal basin, but it is not a measured object.

Therefore, any challenge that moves between actual computation and infinite formal mathematics must declare the bridge. The Clay $P$ vs $NP$ challenge does not declare this bridge. It relies on the inherited commitments of classical mathematics and computational complexity theory. Those commitments are powerful, but they are not neutral.

The Geofinitist position is that better commitments can be made. These commitments do not weaken rigour. They move rigour to the point where the symbol first appears.

- Before the proof, there is the symbol.

- Before the symbol, there is the act of measurement.

- Before the challenge can be answered as a claim about computation, the conditions for measuring such a claim must exist.

**That is the missing axiom**.

And without it, the $P$ vs $NP$ problem can be solved only in the space where it was formally placed: not in the measured world, but in the classical symbolic cave that stands in for it.

## A Closing Note on Stabilisation and the Generonic Boundary

Throughout this work, endogenous and exogenous measurements have been discussed as mechanisms through which functional symbolic trajectories stabilise. However, a subtle but

important clarification is required.

Exogenous measurement is itself a functional symbolic trajectory. It is not a primitive object standing outside the symbolic system. Likewise, endogenous measurement, although appearing internal to a symbolic framework, depends upon exogenous measurements that bind and stabilise symbolic relations across time, memory, notation, and communal practice.

This observation reveals a deeper dependency. Exogenous measurements themselves require finite symbols. Those symbols are not given directly. They arise through transduction processes operating across the Generonic Boundary. The Generon may therefore be understood as the process by which interactions become admissible symbols capable of participating in further symbolic trajectories.

The resulting flow may be represented schematically as:

```
~physical interaction
        ↓
transducer / instrument / observer
        ↓
Generonic Boundary
        ↓
finite symbol
        ↓
reference formation
        ↓
comparison
        ↓
measurement
        ↓
trajectory stabilisation
        ↓
rules, proofs, models, and theories
        ↓
new exogenous measurement
```

In this view, stabilisation is not a final state but an ongoing process. Rules, proofs, models, and theories are not isolated objects. They are stabilised functional symbolic trajectories supported by repeated acts of comparison and measurement.

A useful compression is therefore:

```
Generonic Boundary
        ↓
Symbolisation
```

```
        ↓
Reference
        ↓
Comparison
        ↓
Measurement
        ↓
Stabilisation
```

This leads to a final observation.

Endogenous symbolic stability depends upon exogenous symbolic binding, and exogenous symbolic binding depends upon generonic transduction. What appear to be stable symbolic objects are therefore better understood as locally stabilised functional symbolic trajectories maintained through continual acts of finite symbolic measurement.

# The Safety Layer: Explicit Initialisation of Functional Symbolic Trajectories

The essay "Introducing Functional Symbolic Trajectories" (Haylett, 2026) established the working unit of language once it is viewed as a nonlinear dynamical system. A functional symbolic trajectory is a finite, stabilised pathway that carries representation, constraint, local uncertainty, and history through use. It is not a static label pointing to an ideal meaning; it is a dynamic object that must be generated, followed, tested, remembered, and allowed to fail within a semantic phase space. At every scale — word, sentence, argument, theory — these trajectories form fractal geodesics whose coherence depends on the mutual reinforcement of nested sub-trajectories and the absence of hidden constraints that pull the path into incompatible basins.

That description is descriptive. What follows is prescriptive. Once functional symbolic trajectories are recognised as the actual medium in which all thought and communication unfold, a non-negotiable discipline becomes visible: every non-trivial trajectory must be explicitly initialised before substantive motion begins. This explicit initialisation is the safety layer.

Without it, both human and artificial reasoners are forced into what may be called symbolic mind-reading. The receiver assumes that the incoming symbols already carry the necessary scaffolding — the precise definition of "solution," the measurement regime, the admissible uncertainty, the historical provenance, the constraint structure. When that scaffolding is absent or only partially declared, the trajectory is launched on undeclared assumptions. The result is fluent and coherent on the surface, yet built upon an uninitialised or weakly initialised frame. In human conversation this produces misunderstanding; in mathematical reasoning it produces elegant but incomplete formulations; in LLM interactions it produces confident answers that rest on sand.

Finite Symbolic Mechanics therefore insists on a prior step that classical practice has been able to leave implicit: the deliberate declaration of the frame under which a functional symbolic trajectory is about to be constructed. In the notation developed across these

essays, this declaration takes the compact form

$$(C, \alpha, H, \delta),$$

where

- $C$ is the consensus or admissibility constraint (what counts as valid within this trajectory),
- $\alpha$ is the finite symbolic or Alphonic limit (the resolution and scale at which symbols are being stabilised),
- $H$ is the historical and procedural provenance (how the trajectory was generated and what has been compressed or omitted),
- $\delta$ is the uncertainty associated with measurement and representation (the declared neighbourhood of possible reconstructions).

This initialisation is not bureaucratic overhead. It is the minimal act of intellectual honesty required once we refuse to treat symbols as timeless platonic objects. It forces the writer or prompter to answer, before any further motion occurs: "Under what conditions will this trajectory be considered stabilised? What measurement boundary am I accepting? What hidden commitments am I asking the reader or model to supply through mind-reading?"

The safety layer operates at three interlocking levels.

First, it protects the integrity of the trajectory itself. An explicitly initialised functional symbolic trajectory cannot quietly slip its constraints or drift into a nearby attractor without the drift being visible against the declared frame. Incoherence becomes detectable rather than camouflaged.

Second, it protects the receiver — whether human or artificial. The LLM, the student, the collaborator, or the theorem-prover is no longer required to guess the intended container. The frame is supplied; the geodesic can be followed or challenged on declared terms.

Third, and most profoundly, it protects the possibility of genuine progress. Many of the deepest open questions in mathematics and science — the $P$ vs. $NP$ problem being only the most famous — have remained unresolved not because the symbolic machinery is insufficient, but because the initial framing has been allowed to remain partially undeclared. The classical formulation is powerful precisely because it compresses a vast set of measurement commitments into a compact, idealised language. That compression is a strength inside its own basin; it becomes a limitation the moment we insist on treating computation as a measured, finite process. Explicit initialisation restores the care required to write the problem down honestly.

In everyday LLM interaction this safety layer is especially urgent. Current models are optimised for fluency within whatever frame the user supplies. If the frame is weak or contains undeclared assumptions, the model will generate an entire coherent response on

top of it. The output looks authoritative, yet it inherits every hidden commitment the user failed to declare. The result is not merely error; it is error that feels complete. Explicit frame initialisation breaks this cycle. The first response of a properly disciplined system would be: "Before proceeding, here is the functional symbolic trajectory I am initialising. These are the constraints, the resolution, the provenance, and the declared uncertainty. Does this match the trajectory you intend?"

The same discipline applies to human writing, to scientific papers, to policy documents, and to mathematical proofs. Every time we begin a new symbolic trajectory — whether drafting an essay, posing a research question, or prompting an AI — we have a choice: launch on undeclared assumptions and accept the hidden risks, or perform the brief, rigorous act of explicit initialisation and thereby render the entire subsequent geodesic visible, challengeable, and reproducible.

This is not a call for pedantry. It is a call for gravity. Once language is understood as the real-time generation and navigation of functional symbolic trajectories, the refusal to initialise the frame is no longer a neutral omission. It is the precise point at which coherence can fail, where hidden assumptions can masquerade as shared understanding, and where the measurement boundary is quietly crossed without acknowledgment. The safety layer restores that acknowledgment to its proper place: at the very beginning, before any further motion is attempted.

The essays that follow in this volume are themselves written under such an initialisation. The reader is invited to treat every chapter not as a collection of fixed claims but as a set of explicitly framed functional symbolic trajectories. Where the frame is declared, the geodesic can be followed with confidence. Where it is not, the invitation is open: ask for the initialisation, and the safety layer will be supplied.

That is the discipline the Geofinitist lens demands. That is the care required once we take functional symbolic trajectories seriously.