# The Unfeasible General Formula for P vs NP

A Conceptual Framework on Conditional Subsets, Infinite Formulas, and Non-Reversibility

**This is purely theoretical and conceptual; no formal proof is claimed.**

Yuhenba Rizwan Shah
Age 14

October 15, 2025

**Abstract**

This work presents a conceptual exploration of a general formula for P vs NP, based on the following assumptions:

- A general formula for P vs NP must, *in principle*, contain *every possible algorithm and method* to solve every problem instance.
- Since each problem can have infinitely many solutions or approaches, the full set of solution methodologies, $\mathcal{S}$, is hypothesized to have cardinality $\geq 2^{\aleph_0}$, making the General Formula $G$ **too immense to be fully represented or computed by macro-scale physics**.
- Only **conditional or selective subsets** ($\mathcal{C}_P \subset G$) are activated for specific problem instances $P$; inactive variables are effectively set to zero.
- Finite problems can yield finite results because only the relevant subset is activated.
- **Full utilization of** $G$ requires an infinitely large set of problems—including repeated instances—since each problem can be solved in infinitely many ways.
- A new conceptual class, NNP (Non-reversible NP), is introduced, characterized by **non-injective** (many-to-one) mappings, where reconstructing the input from the output is computationally infeasible.

This framework provides a conceptual bridge connecting computational complexity, fractal geometry, and theoretical physics.

*Purpose: To provide a theoretical and quantitative exploration of P vs NP across mathematics, science, and physics, aiding future research with reference.*

# Contents


**1 P vs NP: Assumptions for a General Formula** **3**

1.1 Conditional Activation of Subsets . . . 3

1.2 Fractals and Complexity . . . 3

1.3 Full Utilization . . . 3

**2 Formalizing the General Formula** **3**

2.1 Definitions . . . 3

2.2 Representation Examples . . . 3

2.2.1 Algebraic . . . 3

2.2.2 Numerical . . . 4

2.2.3 Geometric . . . 4

2.2.4 Set-theoretic . . . 4

2.2.5 Computational / NP-type . . . 4

2.2.6 Combinatorial / Coordinate . . . 4

**3 A New Proposal: P vs NP vs NNP** **5**

3.1 Conceptual Hierarchy . . . 5

3.2 Binary Example (NNP) . . . 5

3.3 Defining NNP Problems . . . 5

**4 Discussion** **6**

**5 Conclusion** **6**

# 1 P vs NP: Assumptions for a General Formula

**General Formula Assumption:** We hypothesize a General Formula $G$, a conceptual operator capable of solving *every* computational problem:

$$G : \mathcal{P} \to \mathcal{S},$$

where $\mathcal{P}$ is the set of all possible problems and $\mathcal{S}$ contains every conceivable algorithm, solution method, or formula. $G$ is **infinitely large**, containing all possible solution branches and approaches.

## 1.1 Conditional Activation of Subsets

For a finite problem instance $P \in \mathcal{P}$, only a **conditional subset** $\mathcal{C}_P \subset G$ is activated. All other variables remain zero, ensuring finite outputs. For example, in $x+y+z+z^2+z^3+\dots$, if the problem involves only $x$ and $y$, then $z$ and its powers do not contribute.

## 1.2 Fractals and Complexity

Conditional subsets $\mathcal{C}_P$ can be visualized as fractal-like structures embedded in the manifold of $G$. Each activated subset corresponds to the solution with minimal Big-O complexity, while inactive variables remain zero. This illustrates how infinite structures yield finite, computable results.

## 1.3 Full Utilization

The entire structure of $G$ is activated only over an infinite sequence of problem instances, revealing all solution branches. Finite problems activate only relevant paths, maintaining computational feasibility.

# 2 Formalizing the General Formula

## 2.1 Definitions

- **Set of All Problems** ($\mathcal{P}$): Includes all P, NP, and hypothetical higher complexity classes.
- **Set of Solution Methodologies** ($\mathcal{S}$): All conceivable algorithms, formulas, and methods. $|\mathcal{S}| \geq 2^{\aleph_0}$.
- **General Formula** ($G$): Maps problems to all possible solutions, i.e., $G : \mathcal{P} \to \mathcal{S}$.
- **Conditional Subset** ($\mathcal{C}_P$): For a problem $P$, $\mathcal{C}_P \subset G$ contains only the methods relevant to $P$. All elements in $G \setminus \mathcal{C}_P$ are zero.

## 2.2 Representation Examples

### 2.2.1 Algebraic

Infinite series $x + y + z + z^2 + z^3 + \dots$ demonstrates how finite problems activate only specific variables.

#### 2.2.2 Numerical

$$1 = \frac{2}{2}, \quad 1 = \sqrt{1}, \quad 1 = \sin^2(\pi/2) + \cos^2(\pi/2), \quad 1 = \sum_{k=0}^{0} k^0, \ldots$$

Each representation belongs to $G$; finite problems select a relevant $\mathcal{C}_P$.

#### 2.2.3 Geometric

- 1D line segment: multiple representations.
- 2D shapes (triangle, square): edges, rotations, and transformations.
- 3D shapes (cube, tetrahedron): coordinates, volumes, combinatorial arrangements.

#### 2.2.4 Set-theoretic

For $S = \{a, b, c\}$, $G$ contains all subsets, unions, intersections, and power sets. Conditional activation selects relevant subsets.

#### 2.2.5 Computational / NP-type

- Base-3 counting: count numbers of length $n$ satisfying a property $P$; irrelevant numbers contribute zero.
- Graph coloring: 3 nodes, 3 colors; only valid colorings activated.

#### 2.2.6 Combinatorial / Coordinate

2D grids illustrate multiple paths. $G$ contains infinitely many formulas, only relevant ones are active.

# 3 A New Proposal: P vs NP vs NNP

**Speculative Idea:** NNP (Non-reversible NP) extends NP by emphasizing non-injective mappings, where input reconstruction is infeasible.

## 3.1 Conceptual Hierarchy

- **P problems:** Efficiently solvable. Mapping may be injective; minimal activation subset $\mathcal{C}_P$.
- **NP problems:** Solutions verifiable efficiently. Mapping reversible in principle but may require large $\mathcal{C}_P$.
- **NNP problems:** Highly non-injective ($f : I \to O$). Even with known output, reconstructing input is computationally infeasible; subset $\mathcal{C}_P$ cannot easily yield inverse mapping.

*This classification uses conditional subset activation from G to distinguish NNP problems from P and NP.*

## 3.2 Binary Example (NNP)

Problem: Count 0s and 1s in a binary string of length $n$.

- Forward: Output $(n_0, n_1)$ computed linearly; minimal $\mathcal{C}_P$ activated.
- Reverse: $\binom{n}{n_1}$ strings produce same output. Exponentially many possibilities; reconstructing input infeasible.
- Conclusion: Illustrates non-injective, many-to-one property defining NNP. $G$ contains all $\binom{n}{n_1}$ possibilities.

## 3.3 Defining NNP Problems

1. Identify the output: number, set, graph, string, etc.
2. Examine forward computability: feasible with activated subset $\mathcal{C}_P$.
3. Test non-reversibility: reconstructing input requires exponential resources; multiple inputs map to same output.
4. Map to $G$: conceptualize all possible solutions; activate $\mathcal{C}_P$ for instance.
5. Classify: Label as NNP if conditions met. Hierarchy:

$$P \subset NP \subset NNP$$

*Example: Counting 0s and 1s in a string illustrates NNP: easy forward computation, infeasible reverse reconstruction.*

*This classification ties back to the general formula, linking conditional subset activation to P, NP, and NNP.*

## 4 Discussion

- The general formula $G$ is an abstract, infinitely large structure, not meant to be physically realized.
- Conditional subsets $\mathcal{C}_P$ explain why finite problems are solvable within $G$.
- NNP highlights a conceptual layer beyond NP, emphasizing non-reversibility.
- Connections to fractals, dimensionality, and combinatorial explosion illustrate how infinite structures yield finite, computable outputs.
- Future work could formalize metrics for subset activation, map fractal dimensions to computational complexity, or explore implications for cryptography and irreversible processes.

## 5 Conclusion

This paper proposes a conceptual framework unifying:

- A general formula $G$ containing all possible solutions for all problems.
- Conditional subsets $\mathcal{C}_P$ explaining finite computation.
- A new class NNP, highlighting non-reversibility and many-to-one mappings.

The framework is theoretical, exploratory, and purely conceptual, offering a foundation for further investigation into P vs NP, non-reversibility, and computational complexity.

## Acknowledgment / Disclosure

The ideas, concepts, and theoretical framework in this paper are entirely the original work of the author. AI assistance was used solely for refining the language, improving clarity, and organizing the presentation. All conceptual content and reasoning remain 100% the author's own.