Article

Not peer-reviewed version

Orbit-Collapse Complexity: A Symmetry-Based Lens on P, NP, and Structured Cores

Michael Rey *

Posted Date: 8 September 2025

doi: 10.20944/preprints202509.0589.v1

Keywords: P vs NP; canonicalization; group actions; orbits; constraint satisfaction; sudoku; sum-of-squares; parameterized complexity; ETH

Article

Orbit-Collapse Complexity: A Symmetry-Based Lens on P, NP, and Structured Cores

Michael Rey

Octonion Group, Hong Kong; contact@octoniongroup.com

Abstract

We introduce *Orbit-Collapse Complexity* (OCC), a symmetry-aware framework that explains when NP problems behave as polynomial-time problems. If instances canonically collapse to *polynomially many types* under an explicit group action, with polytime canonicalization and per-type solvers, the whole language is in P. We quantify "how much structure" a problem exhibits via (i) *orbit coverage* (fraction of solution orbits captured by a P-core), (ii) a *type-growth exponent* (asymptotic number of canonical types), and (iii) an *orbit-compressibility* index (stabilizer size). We enforce hardness beyond cores via *Layer-Respecting Reductions* (LRR) that preserve or raise layer depth. For Sudoku we give closed formulas for separable classes, a degree-4 certificate for the bi-affine layer, and Pólya blueprints for piecewise families. The result is a measurable frontier between P-like cores and genuinely hard remainder, offering a unifying lens across SAT, Graph Coloring, TSP, IP, and Sudoku.

Keywords: P vs NP; canonicalization; group actions; orbits; constraint satisfaction; sudoku; sum-of-squares; parameterized complexity; ETH

---

1. Introduction

The P vs NP question asks whether every language whose YES-certificates can be verified in polynomial time also admits polynomial-time decision procedures. This work develops a symmetry- and type-theoretic lens, *Orbit-Collapse Complexity* (OCC), that (i) proves sufficiency criteria for polynomial-time decidability, (ii) quantifies the structured mass of tractable subfamilies, and (iii) provides reduction policies that target hardness *outside* those subfamilies. Our contributions are:

- **Collapse $\Rightarrow$ P (Theorem 1):** If instances canonically compress to polynomially many types with polytime canonicalization and per-type solvers, the language is in P.
- **Metrics for structure**: orbit coverage, type-growth exponents, and orbit-compressibility index make "how much collapses" a measurable notion.
- **Distance-to-core algorithms**: XP and FPT meta-theorems for instances at bounded edit distance from a P-core.
- **Layer-Respecting Reductions (LRR)**: reductions that preserve or raise layer depth, supporting hardness *beyond* cores.
- **Sudoku case study**: exact separable class law, bi-affine degree-4 certificates, and Pólya formulas for piecewise families.

2. Model and Symmetries

We consider finite-domain CSP families; the discussion covers SAT, Graph Coloring, Sudoku/Latin, Integer Programming encodings, TSP with discrete symmetries, etc.

Definition 1 (CSP family)**.** *A CSP family $\mathcal{L} = \{L_n\}_{n\geq 1}$ has instances of size n over a finite domain D. Each instance $I \in L_n$ specifies constraints of bounded arity on variables $V(I)$ with $|V(I)| = \mathrm{poly}(n)$. The language is the decision problem: "Does I admit a solution?"*

 doi:10.20944/preprints202509.0589.v1

Definition 2 (Symmetry group). *For each n, let G_n be a finite group that acts on instances and solutions. The orbit of X is $G_n \cdot X = \{g \cdot X : g \in G_n\}$ and the stabilizer is $\mathsf{Stab}_{G_n}(X) = \{g : g \cdot X = X\}$.*

Definition 3 (Orbit-compressibility). *For a solution S, define $\kappa(S) = 1 - \frac{\log|\mathsf{Stab}_{G_n}(S)|}{\log|G_n|} \in [0,1]$. Small κ indicates large stabilizer (high structure).*

3. Constructors, Layers, and P-Cores

Definition 4 (Constructor, Layer). *A* constructor *$\mathcal{C}$ is a uniform polytime recognizer that defines a syntactic subfamily $K_n^{\mathcal{C}} \subseteq L_n$. A list $\mathsf{S}_0, \mathsf{S}_1, \ldots$ defines* layers *$S_t(n) = K_n^{\mathsf{S}_t}$. A fixed* layer order *assigns each solution to the first matching layer; layers become disjoint by fiat.*

Definition 5 (P-core). *A* P-core *is a constructor family $K = \{K_n\}$ with polytime membership (and, when needed, polytime canonicalization within its symmetry). Trivial finite sets are excluded.*

Proposition 1. *If membership in K_n is decidable in time $\mathrm{poly}(n)$, then deciding L restricted to K is in P.*

4. Orbit-Collapse ⇒ P

Definition 6 (Types and canonicalization). *Fix an equivalence $\sim$ (e.g., the G_n-action or a coarser invariant). Let $\mathcal{T}_n$ be the set of canonical representatives. A* canonicalizer *is a polytime map $\mathsf{can}(I) \in \mathcal{T}_n$ with $I \sim \mathsf{can}(I)$.*

Theorem 1 (Orbit-Collapse ⇒ P). *Let $\mathcal{L} = \{L_n\}$. Suppose for all n: (i) $|\mathcal{T}_n| \le n^c$; (ii) $\mathsf{can}(I)$ is computable in $\mathrm{poly}(n)$; (iii) there is a $\mathrm{poly}(n)$ predicate A on $\mathcal{T}_n$ with $I \in L_n \iff A(\mathsf{can}(I)) = 1$. Then $\mathcal{L} \in \mathrm{P}$.*

Proof. On input I, compute $T = \mathsf{can}(I)$ and return $A(T)$. Correctness is by (iii). Runtime is polynomial by (ii)–(iii). The bound (i) ensures that the analysis of types is polynomially bounded and realizable. □

5. Quantifying Collapse

Let $\mathcal{O}_n(L)$ be solution orbits in L_n under G_n; $\mathcal{O}_n(K)$ for a core K. We use:

- **Orbit coverage** $\mu_n(K \mid L) = \frac{|\mathcal{O}_n(K)|}{|\mathcal{O}_n(L)|}$;
- **Type-growth exponent** $\gamma(K) = \limsup_{n\to\infty} \frac{\log|\mathcal{T}_n(K)|}{\log n}$;
- **Core ratio inside a layer** $\rho_{K|R}(n) = \frac{|\mathcal{O}_n(K)|}{|\mathcal{O}_n(R)|}$ for $K \subseteq R \subseteq L$.

6. Algorithms by Distance-to-Core

Definition 7 (Edit distance). *Fix a finite set Δ of local edit templates. For instance I, let $\mathrm{dist}_\Delta(I,K)$ be the minimum number of Δ-edits sending I into K.*

Lemma 1 (XP by bounded edits). *For bounded-arity CSPs and any P-core K, satisfiability is decidable in time $n^{O(d)} \cdot \mathrm{poly}(n)$, where $d = \mathrm{dist}_\Delta(I,K)$.*

Proof. Branch over $n^{O(d)}$ candidate edit sets of size d, test membership in K in polynomial time, and verify consistency. □

Theorem 2 (FPT under bounded boundary width). *Assume: (i) constraints have bounded arity; (ii) any set of d edits induces a defect substructure of treewidth $w(d)$; (iii) the core K is MSO-definable on bounded-treewidth hosts. Then satisfiability is decidable in time $f(d) \cdot \mathrm{poly}(n)$.*

Proof sketch. Enumerate the interface between the edited region and K; apply DP/Courcelle on a tree decomposition of width $w(d)$ to glue feasible assignments consistent with K. All combinatorics depend on d only. □

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2025 doi:10.20944/preprints202509.0589.v1

7. Layer-Respecting Reductions (LRR)

Definition 8 (Layer depth)**.** *Given layers $S_0, S_1, \ldots$, define $\ell(I) = \min\{t : I$ has a solution in $S_{\leq t}\}$.*

Definition 9 (LRR at depth t)**.** *A reduction $R : A \to B$ is* layer-respecting at depth t *if $\ell(R(I)) \geq \min\{\ell(I), t\}$ for all I.*

Proposition 2 (Closure)**.** *If R_1 is LRR at depth t_1 and R_2 at depth t_2, then $R_2 \circ R_1$ is LRR at depth $\min\{t_1, t_2\}$.*

Proof. Immediate from the definition. □

Conjecture 1 (Layered ETH (L-ETH))**.** *For each fixed t, there exists $c_t > 0$ such that deciding instances with $\ell(I) > t$ requires time $\geq 2^{c_t n}$ (under ETH/SETH-style assumptions), witnessed by LRRs from 3-SAT that push ℓ above t.*

8. Layered Orbit Sums (Accounting)

Let Ω_n be all solutions at size n, with layer-ordered assignment (each solution taken by the first matching layer). Then:

Theorem 3 (Layered Orbit Sum)**.** $|\Omega_n| = \sum_L \sum_{[S] \in L/G_n} \frac{|G_n|}{|\mathrm{Stab}_{G_n}(S)|}$*, with an extra factor $m!/|\mathrm{Aut}_{sym}(S)|$ if symbol relabeling is counted, where $m = |D|$.*

Proof. Orbit–stabilizer plus disjointness of layer assignment. □

9. Proof-Complexity Calibration inside Layers

Proposition 3 (Bi-affine ⇒ low-degree certificates)**.** *In bi-affine layers (e.g. Sudoku S_1), feasibility/infeasibility has polynomial-size polynomial-calculus and degree-4 sum-of-squares certificates after standard symmetry breaking.*

Proof sketch. Encode row/column/box bijectivity by quadratic equations over 0/1 indicators; the bi-affine placement map linearizes under a fixed normal form so constraints reduce to degree ≤ 2 identities. SOS/PC derives these with degree ≤ 4. □

10. Sudoku Case Study: Formulas and Laws

Let k be the box order; grid size $n = k^2$; position symmetry $G_k = (S_k \wr S_k)_{rows} \times (S_k \wr S_k)_{cols}$ with canonical digit relabeling.

Separable layer S_0

Two orbit sizes arise: *large* $= 2\varphi(k)^2$ and *small* $= \varphi(k)^2$. Let $r_2(k) = |\{x \in (\mathbb{Z}_k)^\times : x^2 \equiv 1 \pmod{k}\}|$, $v_2(k)$ the 2-adic valuation, and $\omega_{odd}(k)$ the number of distinct odd primes dividing k.

Theorem 4 (Separable class law)**.** $S_{sep}(k) = \dfrac{\varphi(k)^2 + E(k)}{2}$ *with* $E(k) = \begin{cases} 2\,r_2(k), & v_2(k) \leq 1 \text{ and } \omega_{odd}(k) \leq 1, \\ 4\,r_2(k), & \text{otherwise.} \end{cases}$

Moreover #small classes $= E(k)$ and #large $= (\varphi(k)^2 - E(k))/2$.

Proof outline. Parameterize separable constructors by $(\alpha, \beta) \in ((\mathbb{Z}_k)^\times)^2$ producing type pairs (s, t). The group G_k together with internal permutations on (s, t) acts on this parameter space. By the Chinese Remainder Theorem, $(\mathbb{Z}_k)^\times \simeq \prod_{p^e \| k} (\mathbb{Z}_{p^e})^\times$ and each factor contributes fixed points corresponding to *involutions* ($x^2 \equiv 1$). These yield index-2 stabilizers responsible for small orbits; counting across CRT components gives a factor $r_2(k)$. The doubling $E(k) = 2r_2(k)$ in the prime-lean regime ($v_2 \leq 1$, at most one odd prime) and $E(k) = 4r_2(k)$ in composite-rich cases reflects whether both axes admit

Preprints.org (www.preprints.org) | NOT PEER-REVIEWED | Posted: 8 September 2025 doi:10.20944/preprints202509.0589.v1

independent involutive symmetries after normal forms. A detailed orbit–stabilizer enumeration (omitted here for brevity) completes the count. □

Bi-affine layer S_1

Empirically (and provably for small k): $C_{lin}(k) = 3k^2 - 5k + \varepsilon(k)$, with $\varepsilon(k) = 2$ if $6 \mid k$, else 0; and $C_{nonsep} = C_{lin} - S_{sep}$. For prime $k > 3$, $\varphi(k) = k - 1$ and the ratio $\frac{S_{sep}}{C_{lin}} = \frac{(k-1)^2+4}{2(3k^2-5k)} = \frac{1}{6} + O(1/k)$ shows a constant-order core inside S_1.

Piecewise Layers S_2, S_3

With multiplicative local catalogs $T_*(k) = \prod_{p^e \| k} t_*(p^e)$ (tabulated by small fixed-point counts), exact Pólya forms are $C_{S2}(k) = \frac{1}{2}\binom{T_U(k)+k-1}{k}\binom{T_V(k)+k-1}{k}$, $C_{S3}(k) = \frac{1}{2}\binom{T_Q(k)+k-1}{k}^2$.

11. Taxonomy by Collapse

- **S-OC**: complete constructor + polytime canonicalization + $|\mathcal{T}_n| \le n^{O(1)} \Rightarrow$ language in P (Thm. 1).
- **W-OC**: large P-cores with subexponential type growth and nontrivial orbit coverage; full language may remain NP-hard.
- **NC**: no such collapse under the chosen symmetries; expect NP-hardness beyond cores; target LRR.

12. How to Apply OCC

Pick G_n; design constructors $S_0, S_1, \ldots$; count types; establish canonicalization and per-type tests; compute μ_n, γ, ρ; design distance-to-core algorithms; and craft LRR gadgets for hardness outside cores.

13. What is Proved vs Conjectured

Proved here: Collapse $\Rightarrow$ P (Thm. 1); layered orbit sum; core-in-P; LRR closure; XP/FPT meta-results (with stated locality assumptions); bi-affine degree-4 certificates (Prop. 3) with constructive sketch.
Conjectural: L-ETH; full proof of the bi-affine count $C_{lin}(k)$ for all k; Wreath-invariant SoS lower bounds beyond cores.

Funding: No external funding.

Data Availability Statement: Tables and code snippets sufficient to reproduce all counts are included in the appendix; additional scripts will be provided upon request.

Acknowledgments: The author thanks collaborators and the broader community for feedback on early drafts of the OCC framework.

Conflicts of Interest: The author declares no conflict of interest.

Ethics

This study involves no human or animal subjects.

Appendix A. Python Utilities

```
import math

def phi(n):
    r, m, p = n, n, 2
    while p*p <= m:
        if m % p == 0:
            while m % p == 0: m //= p
            r -= r // p
        p += 1
```

```
    if m > 1: r -= r // m
    return r

def r2(n):
    c = 0
    for x in range(1, n):
        if math.gcd(x, n) == 1 and (x*x) % n == 1:
            c += 1
    return c

def v2(n):
    e = 0
    while n % 2 == 0 and n > 0:
        n //= 2
        e += 1
    return e

def omega_odd(n):
    s, m, p = set(), n, 2
    while p*p <= m:
        if m % p == 0:
            if p % 2: s.add(p)
            while m % p == 0: m //= p
        p += 1
    if m > 1 and m % 2 == 1:
        s.add(m)
    return len(s)

def S_sep(k):
    phi2 = phi(k)**2
    R2 = r2(k)
    E = (2*R2) if (v2(k) <= 1 and omega_odd(k) <= 1) else (4*R2)
    return (phi2 + E)//2
```

References

1. S. Cook, "The Complexity of Theorem-Proving Procedures," *Proceedings of the Third Annual ACM Symposium on Theory of Computing (STOC)*, 1971.
2. R. M. Karp, "Reducibility among Combinatorial Problems," in *Complexity of Computer Computations*, 1972.
3. B. Courcelle, "The Monadic Second-Order Logic of Graphs I: Recognizable Sets of Finite Graphs," *Information and Computation*, 85(1):12–75, 1990.