Scheme-theoretic Approach to Computational Complexity I. The Separation of P and NP

Ali Çivril*

September 7, 2022

Abstract

We lay the foundations of a new theory for algorithms and computational complexity by parameterizing the instances of a computational problem as a moduli scheme. Considering the geometry of the scheme associated to 3-SAT, we separate P and NP.

1 Introduction

This paper introduces the rudiments of a new theory for algorithms and computational complexity via schemes. One of the most important consequences of the theory is the resolution of the conjecture $\mathsf{P} \neq \mathsf{NP}$.

An easily understood reason for the difficulty of the problem we consider is the superficial similarity between the problems in P and NP-complete problems. More concretely, one has not been able to find a metric somehow measuring the time complexity of a problem so that the difference between the values for 3-SAT and 2-SAT is large enough. Extracting this intrinsic property from a problem seems out of reach when it is treated by only combinatorial means.

From an elementary point of view, a computational problem is considered to be a *language* recognized by a *Turing machine*. Through a slightly refined lens, it is a *Boolean function* computed by a *circuit*. We recognize the existence of a much deeper perspective: A computational problem is a (moduli) *scheme* formed by its instances, and an algorithm is a *morphism* contracting it to a single point. This opens the possibility of understanding computational complexity using the language of category theory. In particular, we define a functor from the category of computational problems to the category of schemes parameterizing the instances of a computational problem, albeit currently restricted to k-SAT.

For concreteness, consider a satisfiable instance of 3-SAT represented by the formula ϕ with variables $x_1, \ldots, x_n$. We associate with this instance all the solutions that make ϕ satisfiable, which can be expressed as the zeros of a polynomial $\phi(x_1, \ldots, x_n)$ over $\mathbb{F}_2$. We then identify this information by considering the closed subscheme $\operatorname{Proj} \overline{\mathbb{F}}_2[x_0, x_1, \ldots, x_n]/(\phi(x_0, x_1, \ldots, x_n))$. The global scheme corresponding to the computational problem 3-SAT is the Hilbert scheme parameterizing these closed subschemes together with a set of others to ensure connectedness.

The next step is to unify the notion of a reduction and an algorithm in the new setting. Consider 1-SAT $\in$ P. In order to separate P and NP, one needs to rule out a polynomial-time reduction f satisfying $x \in$ 3-SAT $\Leftrightarrow f(x) \in$ 1-SAT. We extend this line of thinking by introducing the trivial

*Atlas University, Computer Engineering Department, Kagithane, Istanbul Turkey, e-mail: ali.civril@atlas.edu.tr, website: www.alicivril.com

object in the category of computational problems: the trivial problem defined via a single instance with an empty set of variables. The associted scheme of this problem consists of a single point. In our new language, solving a problem is nothing but reducing it to the trivial problem. One then needs to show that, in geometric terms we will later formalize, it is impossible to reduce the scheme of 3-SAT to a single point with polynomial number of unit contractions.

2 Computational Problems and the Amplifying Functor

A computational problem consists of a set of *instances*. In this paper we impose that each instance consists of a finite set of polynomial equations over $\mathbb{F}_2$. We thus use a *polynomial system* as a synonym for an instance. The synonym for a single polynomial equation is a *clause*. One seeks, given an instance, an assignment to the variables in $\mathbb{F}_2$ satisfying all the equations of the instance. Throughout the paper an instance is one which has such a solution. We give below examples by listing the possible set of polynomials that might be considered for an equation. The simplest problem is what we call TRIVIAL or T for short, defined via a single instance with an empty set of variables. The simplest problem after T is UNIT or U for short, a special case of 1-SAT and 3-SAT.

Problem: TRIVIAL or T
Polynomials: $p(x) \in \{0\}$.

Problem: UNIT or U
Polynomials: $p(x) \in \{x, 1-x\}$.

Problem: 1-SAT
Polynomials: All $\{p(x_1, \ldots, x_n)\}$ with $p(x_1, \ldots, x_n) = t$, where $t = x_\ell$ or $t = 1 - x_\ell$ for some $\ell \in \{1, \ldots, n\}$.

Problem: 3-SAT
Polynomials: All $\{p(x_1, \ldots, x_n)\}$ with $p(x_1, \ldots, x_n) = t_1 t_2 t_3$, where $t_j = x_\ell$ or $t_j = 1 - x_\ell$ for some $\ell \in \{1, \ldots, n\}$, for $j \in \{1, 2, 3\}$.

For the sake of explicitness, we give the following examples regarding instances.

Problem: UNIT or U
Logical form: $\{x\}, \{\overline{x}\}$.
Algebraic form: $\{1 - x = 0\}, \{x = 0\}$.

Problem: 1-SAT
Logical form: $\{x_1 \wedge \overline{x_2} \wedge x_3\}$.
Algebraic form: $\{1 - x_1 = 0, x_2 = 0, 1 - x_3 = 0\}$.

Problem: 3-SAT
Logical form: $\{(x_1 \vee \overline{x_3} \vee x_4) \wedge (\overline{x_2} \vee x_3 \vee \overline{x_5})\}$.
Algebraic form: $\{(1 - x_1)x_3(1 - x_4) = 0, x_2(1 - x_3)x_5 = 0\}$.

We now recall the definitions regarding the Hilbert functor. Let S be a scheme, and let $X \subseteq \mathbb{P}^n_S$

be a closed subscheme. Define

$$H(X/S) := \{Z \subseteq X \text{ is a closed subscheme}, Z \to S \text{ is flat}\}.$$

The Hilbert functor $\mathcal{H}_{X/S}$ is the functor $T \mapsto H(X \times_S T/T)$ for any S-scheme T. We set $S = \operatorname{Spec} \overline{\mathbb{F}}_2$, and denote $\mathcal{H}_{X/\overline{\mathbb{F}}_2}$ briefly as $\mathcal{H}_X$.

Let X be a projective scheme over $\overline{\mathbb{F}}_2$, and let $Z \subseteq X$ be a closed subscheme. Let $\mathscr{F}$ be a coherent sheaf on Z. The *Hilbert polynomial of Z with respect to $\mathscr{F}$* is $P(Z, \mathscr{F})(m) := \chi(Z, \mathscr{F}(m))$, where $\mathscr{F}(m)$ is the twisting of $\mathscr{F}$ by m, and $\chi(Z, \mathscr{F})$ denotes the *Euler characteristic* of $\mathscr{F}$ given by

$$\chi(Z, \mathscr{F}) := \sum_{i=0}^{\dim Z} (-1)^i \dim_{\overline{\mathbb{F}}_2} H^i(Z, \mathscr{F}). \tag{1}$$

The *Hilbert polynomial of Z* is

$$P(Z)(m) := \chi(Z, \mathscr{O}_Z(m)) \tag{2}$$

where $\mathscr{O}_Z$ is the structure sheaf of Z. Let $\mathcal{H}_X^P$ denote the subfunctor of $\mathcal{H}_X$ induced by the closed subschemes of X with a fixed Hilbert polynomial $P \in \mathbb{Q}[x]$. By the following result stated in our context, the Hilbert functor is representable by a projective scheme over $\overline{\mathbb{F}}_2$.

Theorem 2.1 ([1])**.** *Let X be a projective scheme over $\overline{\mathbb{F}}_2$. Then for every polynomial $P \in \mathbb{Q}[x]$, there exists a projective scheme* $\operatorname{Hilb}^P(X)$ *over $\overline{\mathbb{F}}_2$, which represents the functor $\mathcal{H}_X^P$. Furthermore, the Hilbert functor $\mathcal{H}_X$ is represented by the* Hilbert scheme

$$\operatorname{Hilb}(X) := \coprod_{P \in \mathbb{Q}[x]} \operatorname{Hilb}^P(X).$$

We consider the computational problem $\mathsf{\Pi} := k\text{-}\mathsf{SAT}$ defined via the variable set $\{x_1, \dots, x_n\}$. Note first that given a homogenized polynomial ϕ, one might consider the closed subscheme

$$\operatorname{Proj} \overline{\mathbb{F}}_2[x_0, x_1, \dots, x_n]/(\phi(x_0, x_1, \dots, x_n)),$$

so that each polynomial equation and hence a polynomial system of $\mathsf{\Pi}$ identifies a closed subscheme of $\mathbb{P}^n_{\overline{\mathbb{F}}_2}$ via the corresponding ideal. In particular, we refer to the Hilbert polynomial of an instance.

Definition 2.2. Given an instance I of $\mathsf{\Pi}$ and $S \subseteq \{x_1, \dots, x_n\}$, the *sub-instance* of I induced by S is the instance consisting of the set of all clauses of I in which an element of S appears.

Definition 2.3. Given a subset S of the instances of $\mathsf{\Pi}$, a computational problem whose instances form a set of sub-instances of the instances in S induced by the same set $S \subseteq \{x_1, \dots, x_n\}$, is called a *sub-problem* of $\mathsf{\Pi}$. By definition, T is a sub-problem of $\mathsf{\Pi}$.

Definition 2.4. A sub-problem $\mathsf{\Lambda}$ of $\mathsf{\Pi}$ is called a *simple sub-problem* if the instances of $\mathsf{\Lambda}$ have the same Hilbert polynomial. By definition, T is not a simple sub-problem of $\mathsf{\Pi}$.

As noted, we associate with each instance the ideal defined by its polynomials. The ideals of two instances having the same solution set might be identical. We will naturally consider the moduli of ideals that are guaranteed to be distinct in this sense.

Definition 2.5. Two sub-instances of $\mathsf{\Pi}$ with distinct solution sets are said to be *distinct*.

Definition 2.6. A sub-problem $\mathsf{\Lambda}$ of $\mathsf{\Pi}$ is said to be *homogeneous* if the instances of $\mathsf{\Lambda}$ are pair-wise distinct.

Definition 2.7. Given two distinct instances I_1 and I_2 of a sub-problem Λ of Π, a computational procedure transforming I_1 to I_2 is called a *unit operation.*

An example of a unit operation is as follows. Suppose I_1 is $\{x_1 = 0, 1-x_2 = 0\}$. Then replacing x_1 with $1 - x_1$ and x_2 with $1 - x_2$, we get another instance I_2, which is $\{1 - x_1 = 0, x_2 = 0\}$.

Definition 2.8. Two unit operations are said to be *distinct* if the instances they result in are distinct when applied on the same instance.

Consider the example given above with $I_1 : \{x_1 = 0, 1-x_2 = 0\}$. The unit operation permuting the variables x_1 and x_2 is not distinct from the aforementioned unit operation, as it results in the same instance I_2.

Definition 2.9. Two unit operations are said to be *disparate* if they are distinct and one is not a subset of another. In this case we also say that one operation is *disparate from* the other.

Definition 2.10. A sub-problem Λ of Π defined via the set of instances $\{I_1, \ldots, I_p\}$ is said to be *prime* if there exists an ordering π of its instances such that there are unit operations from $I_{\pi(i)}$ to $I_{\pi(i+1)}$ for $i = 1, \ldots, p-1$ that are pair-wise disparate.

Consider the following as an example. Let Λ be defined via the instances

$$\begin{aligned} &I_1 : \{x_1 = 0, x_2 = 0\}, \\ &I_2 : \{x_1 = 0, 1 - x_2 = 0\}, \\ &I_3 : \{1 - x_1 = 0, 1 - x_2 = 0\}. \end{aligned}$$

Then Λ is prime since the unit operation from I_1 to I_2, replacing x_2 with $1 - x_2$ and the unit operation from I_2 to I_3, replacing x_1 with $1 - x_1$ are disparate. Notice that both of these unit operations are subsets of the unit operation from I_1 to I_3, which performs the union of them, and if one adds the instance $I_4 : \{1 - x_1 = 0, x_2 = 0\}$ to Λ, it is not prime anymore.

Let Λ be a prime homogeneous simple sub-problem of Π consisting of a set of polynomial systems $\{P_i\}_{i=1}^{\ell}$ defined via the variables $x_1, \ldots, x_n$. We assume without loss of generality that all the variables appear in each polynomial system of Λ. Let ϕ_{ij} be the homogenized j-th polynomial in the polynomial system P_i:

$$\phi_{ij} := \phi_{ij}(x_0, x_1, \ldots, x_n),$$

for $j = 1, \ldots, |P_i|$. Define

$$X_i := \operatorname{Proj} \overline{\mathbb{F}}_2[x_0, x_1, \ldots, x_n]/(\phi_{i1}, \ldots, \phi_{i|P_i|}), \tag{3}$$

for $i = 1, \ldots, \ell$. Let $X_\Lambda := \bigcup_{i=1}^{\ell} X_i$. In words, X_Λ contains all the closed subschemes identified by the instances of Λ. Define the *amplifying functor* $\mathcal{A}_\Lambda$ on Λ as

$$T \mapsto \{Y \times_{\overline{\mathbb{F}}_2} T | Y \in X_\Lambda, Y \times_{\overline{\mathbb{F}}_2} T \to T \text{ is flat}\},$$

for any scheme T over $\overline{\mathbb{F}}_2$. It is clear that $\mathcal{A}_\Lambda$ is a subfunctor of the Hilbert functor. Let $\operatorname{Amp}(\Lambda)$ be the scheme representing $\mathcal{A}_\Lambda$.

Define $\operatorname{Hilb}(\Lambda) := \operatorname{Hilb}^{P(\Lambda)}(\mathbb{P}^n_{\overline{\mathbb{F}}_2})$, where $P(\Lambda)$ is the Hilbert polynomial associated to Λ. For a fixed Hilbert polynomial P, $\operatorname{Hilb}^P(\mathbb{P}^n_{\overline{\mathbb{F}}_2})$ is connected by a result of Hartshorne [2]. Thus, $\operatorname{Hilb}(\Lambda)$ is connected. For convenience, in the rest of the paper we only speak of the underlying topological space of a scheme (and mappings between these spaces), disregarding the extra structure imposed by the definition of a scheme.

Given two sub-problems Γ and Γ' of Π, a reduction is a set-theoretic map $f : \Gamma \to \Gamma'$, where a sub-problem is considered as a set consisting of its instances. $\tau(\Gamma, \Gamma')$ denotes the minimum number of *deterministic* unit operations required by such a reduction, possibly with an advice string, thus simulating circuits. It is called the *complexity* of f. We set $\tau(\Gamma) := \tau(\Gamma, \mathsf{T})$, and refer this as the complexity of *solving* Γ.

The essence of our strategy, which we explain in the next section, is via an implicitly defined functor from the category of computational problems to the category of schemes. In particular, we map Λ to a geometric object $B(\Lambda)$ whose connectivity is crucial, and is provided by the connectivity of $\mathrm{Hilb}(\Lambda)$, as noted above. $B(\Lambda)$ are the objects of the target category. We next map the reduction $\Lambda \to \mathsf{T}$ to what we call a *contraction procedure* on $B(\Lambda)$, and relate this to the complexity of the reduction. Contraction procedures as we define them, are distinct from the usual concept of an algebro-geometric morphism, and essentially form the category-theoretical morphisms of the target category.

3 Lower Bounds via Prime Homogeneous Simple Sub-problems

Let Λ be a prime homogeneous simple sub-problem of Π. Over all such sub-problems Λ of Π, let $\kappa(\Pi)$ denote the maximum value of $b(\Lambda)$, where $b(\Lambda)$ denotes the number of instances of Λ.

Lemma 3.1 (Fundamental Lemma)**.**

$$\tau(\Pi) \geq \kappa(\Pi).$$

Proof. Let Λ be a sub-problem of Π attaining $\kappa(\Pi)$. Since $\tau(\Pi) \geq \tau(\Lambda)$, it suffices to show $\tau(\Lambda) \geq b(\Lambda)$. We argue by induction on $p := b(\Lambda)$.

Let $\Lambda := \{I_1, \ldots, I_p\}$, where the instances are ordered accordingly as in the definition of a prime sub-problem. Note first that by definition, $\mathrm{Amp}(\Lambda)$ is a subspace of $\mathrm{Hilb}(\Lambda)$. Given this, let $B(\Lambda)$ be a space satisfying the following:

- $B(\Lambda)$ contains the points of $\mathrm{Amp}(\Lambda)$.
- $B(\Lambda)$ is a connected subspace of $\mathrm{Hilb}(\Lambda)$.
- $B(\Lambda)$ has minimal number of points satisfying the first two properties.

For $p \geq 2$, let $\Lambda' := \Lambda \setminus \{I_1\}$, and let $B(\Lambda')$ be a space satisfying the following:

- $B(\Lambda')$ contains the points of $\mathrm{Amp}(\Lambda')$.
- $B(\Lambda')$ is a connected subspace of $B(\Lambda)$.
- $B(\Lambda')$ has minimal number of points satisfying the first two properties.

By excluding instances in order, we can define (connected) subspaces of $B(\Lambda')$ corresponding to the subsets of Λ'. For $p = 1$, we set $\Lambda' = \mathsf{T}$, and $B(\Lambda') = \mathrm{Spec}\ \overline{\mathbb{F}}_2$.

As noted in the previous section, $B(\Lambda)$ is the geometric object associated to Λ. We next map the reduction $\Lambda \to \mathsf{T}$ to what we call a *contraction procedure*. For $p \geq 2$, given a reduction $\Lambda \to \Lambda'$, which reduces to identity on the instances of Λ' and maps I_1 to I_2, we consider a surjective mapping $B(\Lambda) \to B(\Lambda')$, which agrees with the reduction on the points representing the instances. For $p = 1$, the reduction and the (unique) mapping we consider are $\Lambda \to \mathsf{T}$ and $B(\Lambda) \to \mathrm{Spec}\ \overline{\mathbb{F}}_2$, respectively. In both cases, we call the reduction and the mapping a *unit reduction* and a *unit contraction*, respectively. These definitions can be extended to the subsets of Λ obtained by excluding the instances in order. In this case a unit reduction is identified by a unit operation between two consecutive instances. A chain of unit contractions is called a *contraction procedure*. The complexity of a contraction procedure is defined to be the complexity of the corresponding reduction.

For the base case $p = 1$ of the induction, we clearly have $\tau(\Lambda) \geq 1$, since the complexity of solving a problem other than T is non-zero. For $p > 1$, consider the following diagrams.

$$\Lambda \xrightarrow{\alpha} \Lambda' \xrightarrow{\beta} \mathsf{T} \qquad\qquad B(\Lambda) \longrightarrow B(\Lambda') \longrightarrow \text{Spec } \overline{\mathbb{F}}_2$$

The arrows on the left are reductions between problems, satisfying the aforementioned properties. In particular, the unit reductions in β follow the instances in order. The arrows on the right are contraction procedures. Since Λ is prime, α is disparate from any of the unit reductions in β, i.e., in reducing Λ to T one does not repeat α. Considering then the contraction procedure $B(\Lambda) \to \text{Spec } \overline{\mathbb{F}}_2$ on the right, we have that its complexity satisfies $\tau(\Lambda) = \tau(\alpha) + \tau(\beta)$ with $\tau(\alpha) \geq 1$. Given these, we obtain $\tau(\Lambda) \geq \tau(\Lambda') + 1 \geq (p-1) + 1 = p$, where the second inequality follows from the induction hypothesis. This completes the induction and the proof. □

4 3-SAT: The Separation of P and NP

Denote by k-$\mathsf{SAT}(n, m)$ the problem k-SAT with n variables and m clauses.

Theorem 4.1. *There exist infinitely many $n \in \mathbb{Z}^+$ such that for any constant $\epsilon > 0$, we have*

$$\kappa(3\text{-}\mathsf{SAT}(n, 2n)) \geq 2^{\left(\frac{3}{8}-\epsilon\right)n}.$$

Proof. We construct a prime homogeneous simple sub-problem of 3-SAT with $\binom{r}{r/2} \cdot 2^{r/2}$ instances, each having $4r$ variables and $8r$ clauses, for $r \geq 1$.

The Initial Construction: A Homogeneous Simple Sub-problem Each instance consists of r *blocks*. For $r = 1$, a block of an instance is initially defined via 4 variables x_1, x_2, x_3, x_4, and 8 clauses. We first construct 3 instances with the solution sets over $\mathbb{F}_2$ consisting of the following points, listed for each instance in a separate column:

Instance 1	Instance 2	Instance 3
$(0,0,1,0)$	$(0,0,1,0)$	$(0,1,0,0)$
$(1,0,0,0)$	$(0,1,0,0)$	$(0,1,1,0)$
$(1,1,0,0)$	$(1,0,1,0)$	$(1,0,0,0)$

A single block for each instance can be described by a procedure using the truth table of the variables. Each of the 8 clauses is introduced one by one to rule out certain assignments over $\mathbb{F}_2$ in the tables. We enumerate the rows of the tables for each instance by an indexing of these clauses in Table 2, Table 3, and Table 4. The solution sets over $\mathbb{F}_2$ are the entries left out by the introduced clauses. The corresponding schemes over $\overline{\mathbb{F}}_2$ have isomorphic cohomology groups with respect to any coherent sheaf, so that by (1) and (2) the Hilbert polynomials of the instances are the same. In particular, they are the disjoint union of a closed point and an affine line as shown below.

The first 5 clauses of the instances are common. Clause 1 forces at least one of x_1, x_2 and x_3 to be 1, as it corresponds to

$$(1 - x_1)(1 - x_2)(1 - x_3) = 0.$$

Clause	Instance 1	Instance 2	Instance 3
1	$x_1 \vee x_2 \vee x_3$	$x_1 \vee x_2 \vee x_3$	$x_1 \vee x_2 \vee x_3$
2	$x_2 \vee \overline{x_3} \vee \overline{x_4}$	$x_2 \vee \overline{x_3} \vee \overline{x_4}$	$x_2 \vee \overline{x_3} \vee \overline{x_4}$
3	$\overline{x_2} \vee x_3 \vee \overline{x_4}$	$\overline{x_2} \vee x_3 \vee \overline{x_4}$	$\overline{x_2} \vee x_3 \vee \overline{x_4}$
4	$\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$	$\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$	$\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$
5	$\overline{x_1} \vee x_2 \vee \overline{x_4}$	$\overline{x_1} \vee x_2 \vee \overline{x_4}$	$\overline{x_1} \vee x_2 \vee \overline{x_4}$
6	$x_1 \vee x_3 \vee x_4$	$\overline{x_1} \vee x_3 \vee x_4$	$\overline{x_1} \vee \overline{x_2} \vee x_4$
7	$\overline{x_1} \vee \overline{x_3} \vee x_4$	$x_2 \vee x_3 \vee x_4$	$\overline{x_1} \vee \overline{x_3} \vee x_4$
8	$\overline{x_2} \vee \overline{x_3} \vee x_4$	$\overline{x_2} \vee \overline{x_3} \vee x_4$	$x_2 \vee \overline{x_3} \vee x_4$

Table 1: The clauses of the 3 instances satisfying Table 2, Table 3, and Table 4

Clause	x_1	x_2	x_3	x_4	Clause	x_1	x_2	x_3	x_4
1	0	0	0	0		1	0	0	0
1	0	0	0	1	5	1	0	0	1
	0	0	1	0	7	1	0	1	0
2	0	0	1	1	2	1	0	1	1
6	0	1	0	0		1	1	0	0
3	0	1	0	1	3	1	1	0	1
8	0	1	1	0	8	1	1	1	0
4	0	1	1	1	4	1	1	1	1

Table 2: The truth table of a block of Instance 1 with clause-indexing

Clause	x_1	x_2	x_3	x_4	Clause	x_1	x_2	x_3	x_4
1	0	0	0	0	7	1	0	0	0
1	0	0	0	1	5	1	0	0	1
	0	0	1	0		1	0	1	0
2	0	0	1	1	2	1	0	1	1
	0	1	0	0	6	1	1	0	0
3	0	1	0	1	3	1	1	0	1
8	0	1	1	0	8	1	1	1	0
4	0	1	1	1	4	1	1	1	1

Table 3: The truth table of a block of Instance 2 with clause-indexing

Given this, the following 4 clauses make $x_4 = 0$, since $x_4 \neq 0$ implies $x_1 = x_2 = x_3 = 0$ by these clauses. In other words, $x_i = 1$ for any $i \in \{1, 2, 3\}$ implies a contradiction in the following system:

$$(1 - x_2)x_3 = 0.$$
$$x_2(1 - x_3) = 0.$$
$$x_2x_3 = 0.$$
$$x_1(1 - x_2) = 0.$$

Given that $x_4 = 0$ (or more generally $x_4 \neq 1$), we now examine the last 3 clauses of the instances.

Clause	x_1	x_2	x_3	x_4	Clause	x_1	x_2	x_3	x_4
1	0	0	0	0		1	0	0	0
1	0	0	0	1	5	1	0	0	1
8	0	0	1	0	8	1	0	1	0
2	0	0	1	1	2	1	0	1	1
	0	1	0	0	6	1	1	0	0
3	0	1	0	1	3	1	1	0	1
	0	1	1	0	7	1	1	1	0
4	0	1	1	1	4	1	1	1	1

Table 4: The truth table of a block of Instance 3 with clause-indexing

1. Instance 1:
$$(1-x_1)(1-x_3)=0.$$
$$x_1x_3=0.$$
$$x_2x_3=0.$$
$x_1=1 \Rightarrow x_3=0, x_2 \in \overline{\mathbb{F}}_2$.
$x_2=1 \Rightarrow x_1=1, x_3=0$.
$x_3=1 \Rightarrow x_1=0, x_2=0$.
Thus, the solution set is $\{(0,0,1)\} \cup \{(1,\alpha,0)\}$, where $\alpha \in \overline{\mathbb{F}}_2$.
2. Instance 2:
$$x_1(1-x_3)=0.$$
$$(1-x_2)(1-x_3)=0.$$
$$x_2x_3=0.$$
$x_1=1 \Rightarrow x_2=0, x_3=1$.
$x_2=1 \Rightarrow x_1=0, x_3=0$.
$x_3=1 \Rightarrow x_2=0, x_1 \in \overline{\mathbb{F}}_2$.
Thus, the solution set is $\{(0,1,0)\} \cup \{(\alpha,0,1)\}$, where $\alpha \in \overline{\mathbb{F}}_2$.
3. Instance 3:
$$x_1x_2=0.$$
$$x_1x_3=0.$$
$$(1-x_2)x_3=0.$$
$x_1=1 \Rightarrow x_2=0, x_3=0$.
$x_2=1 \Rightarrow x_1=0, x_3 \in \overline{\mathbb{F}}_2$.
$x_3=1 \Rightarrow x_1=0, x_2=1$.
Thus, the solution set is $\{(1,0,0)\} \cup \{(0,1,\alpha)\}$, where $\alpha \in \overline{\mathbb{F}}_2$.

Since these instances are distinct, they form a homogeneous simple sub-problem. Assume now the induction hypothesis that there exists a homogeneous simple sub-problem of size 3^r, for some $r \geq 1$. In the inductive step, we introduce 4 new variables $x_{4r+1}, x_{4r+2}, x_{4r+3}, x_{4r+4}$, and 3 new blocks on these variables each consisting of 8 clauses with the exact form as in Table 1. Appending these blocks to each of the 3^r instances of the induction hypothesis, we obtain 3^{r+1} instances. The constructed sub-problem is a homogeneous simple sub-problem. We now describe a procedure to make it into a prime homogeneous simple sub-problem.

Mixing the Blocks For simplicity and the purpose of providing examples, we describe the procedure for $r = 2$. The construction is easily extended to the general case. Suppose that the first block is defined via Instance 1. We perform the following operation: Replace the literals of Clause 4 except $\overline{x_4}$ with appropriate literals of variables belonging to the second block, depending on which instance it is defined via. If the second block is defined via Instance 1, then Clause 4 becomes $(x_5 \vee x_7 \vee \overline{x_4})$. If it is defined via Instance 2, it becomes $(\overline{x_6} \vee \overline{x_7} \vee \overline{x_4})$. If it is defined via Instance 3, it becomes $(\overline{x_5} \vee \overline{x_6} \vee \overline{x_4})$. In extending this to the general case, the second block is generalized as the next block to the current one, and the variables used for replacement are the ones with the first three indices of the next block in increasing order, respectively corresponding to x_5, x_6 and x_7.

If the first block is defined via Instance 2 or Instance 3, we disregard blocks defined via Instance 1 in defining the second block, i.e., Instance 2 and Instance 3 blocks are not mixed with Instance 1 blocks. We then perform the same operations above, but this time considering Clause 5 of the first block for the operation on Instance 2. If the first block is defined via Instance 3, we consider Clause 2. All possible cases are illustrated in Table 5-Table 8, where the interchanged literals are shown in bold. In the general case, the described operation is also performed for the last block indexed r for which the next block is defined as the first block (or the first block that is not an Instance 1 block for Instance 2 and Instance 3), so that the operations complete a cycle over the blocks.

The following, which we call the *mixing property* for future reference, is an important characteristic of this construction, as it will help us argue that we have a prime sub-problem. In mixing the

Clause	Instance 1	Instance 1
1	$x_1 \vee x_2 \vee x_3$	$x_5 \vee x_6 \vee x_7$
2	$x_2 \vee \overline{x_3} \vee \overline{x_4}$	$x_6 \vee \overline{x_7} \vee \overline{x_8}$
3	$\overline{x_2} \vee x_3 \vee \overline{x_4}$	$\overline{x_6} \vee x_7 \vee \overline{x_8}$
4	$\mathbf{x_5} \vee \mathbf{x_7} \vee \overline{x_4}$	$\mathbf{x_1} \vee \mathbf{x_3} \vee \overline{x_8}$
5	$\overline{x_1} \vee x_2 \vee \overline{x_4}$	$\overline{x_5} \vee x_6 \vee \overline{x_8}$
6	$x_1 \vee x_3 \vee x_4$	$x_5 \vee x_7 \vee x_8$
7	$\overline{x_1} \vee \overline{x_3} \vee x_4$	$\overline{x_5} \vee \overline{x_7} \vee x_8$
8	$\overline{x_2} \vee \overline{x_3} \vee x_4$	$\overline{x_6} \vee \overline{x_7} \vee x_8$

Table 5: Modification to form a prime sub-problem on Instance 1 and Instance 1 blocks

Clause	Instance 1	Instance 2
1	$x_1 \vee x_2 \vee x_3$	$x_5 \vee x_6 \vee x_7$
2	$x_2 \vee \overline{x_3} \vee \overline{x_4}$	$x_6 \vee \overline{x_7} \vee \overline{x_8}$
3	$\overline{x_2} \vee x_3 \vee \overline{x_4}$	$\overline{x_6} \vee x_7 \vee \overline{x_8}$
4	$\overline{\mathbf{x_6}} \vee \overline{\mathbf{x_7}} \vee \overline{x_4}$	$\overline{x_6} \vee \overline{x_7} \vee \overline{x_8}$
5	$\overline{x_1} \vee x_2 \vee \overline{x_4}$	$\overline{x_5} \vee x_6 \vee \overline{x_8}$
6	$x_1 \vee x_3 \vee x_4$	$\overline{x_5} \vee x_7 \vee x_8$
7	$\overline{x_1} \vee \overline{x_3} \vee x_4$	$x_6 \vee x_7 \vee x_8$
8	$\overline{x_2} \vee \overline{x_3} \vee x_4$	$\overline{x_6} \vee \overline{x_7} \vee x_8$

Table 6: Modification to form a prime sub-problem on Instance 1 and Instance 2 blocks

Clause	Instance 2	Instance 3
1	$x_1 \vee x_2 \vee x_3$	$x_5 \vee x_6 \vee x_7$
2	$x_2 \vee \overline{x_3} \vee \overline{x_4}$	$\overline{\mathbf{x_2}} \vee \overline{\mathbf{x_3}} \vee \overline{x_8}$
3	$\overline{x_2} \vee x_3 \vee \overline{x_4}$	$\overline{x_6} \vee x_7 \vee \overline{x_8}$
4	$\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$	$\overline{x_6} \vee \overline{x_7} \vee \overline{x_8}$
5	$\overline{\mathbf{x_5}} \vee \overline{\mathbf{x_6}} \vee \overline{x_4}$	$\overline{x_5} \vee x_6 \vee \overline{x_8}$
6	$\overline{x_1} \vee x_3 \vee x_4$	$\overline{x_5} \vee \overline{x_6} \vee x_8$
7	$x_2 \vee x_3 \vee x_4$	$\overline{x_5} \vee \overline{x_7} \vee x_8$
8	$\overline{x_2} \vee \overline{x_3} \vee x_4$	$x_6 \vee \overline{x_7} \vee x_8$

Table 7: Modification to form a prime sub-problem on Instance 2 and Instance 3 blocks

Clause	Instance 3	Instance 1
1	$x_1 \vee x_2 \vee x_3$	$x_5 \vee x_6 \vee x_7$
2	$x_2 \vee \overline{x_3} \vee \overline{x_4}$	$x_6 \vee \overline{x_7} \vee \overline{x_8}$
3	$\overline{x_2} \vee x_3 \vee \overline{x_4}$	$\overline{x_6} \vee x_7 \vee \overline{x_8}$
4	$\overline{x_2} \vee \overline{x_3} \vee \overline{x_4}$	$\overline{\mathbf{x_1}} \vee \overline{\mathbf{x_2}} \vee \overline{x_8}$
5	$\overline{x_1} \vee x_2 \vee \overline{x_4}$	$\overline{x_5} \vee x_6 \vee \overline{x_8}$
6	$\overline{x_1} \vee \overline{x_2} \vee x_4$	$x_5 \vee x_7 \vee x_8$
7	$\overline{x_1} \vee \overline{x_3} \vee x_4$	$\overline{x_5} \vee \overline{x_7} \vee x_8$
8	$x_2 \vee \overline{x_3} \vee x_4$	$\overline{x_6} \vee \overline{x_7} \vee x_8$

Table 8: Modification to form a prime sub-problem on Instance 3 and Instance 1 blocks

blocks, we force one specific clause of a block depending on its type to contain variables belonging to the next block in a way distinctive to the type of the next block. Suppose we represent an instance as a sequence of blocks numbered according to their types. Then any unit operation from the instance 22 to the instance 23 is disparate from a unit operation from the instance 32 to the instance 33. In fact by the construction, the first operation can be more appropriately labeled as one from $(2,2)(2,2)$ to $(2,3)(3,2)$, since a block is essentially distinguished by itself together with the next block. With this token, the second operation is from $(3,2)(2,3)$ to $(3,3)(3,3)$, which better indicates that it is disparate from the first operation.

We next establish facts about the solution sets. We observe the following for the first block, which also holds for all the other blocks by the construction. Assume $x_4 \neq 0$ and $x_4 \neq 1$. We will show that this leads to a contradiction, so that $x_4 \neq 0$ implies $x_4 = 1$. Consider the case in which the first block is defined via Instance 1. By the equations numbered 2, 3 and 5 of the first block, we then have

$$(1 - x_2)x_3 = 0.$$
$$x_2(1 - x_3) = 0.$$
$$x_1(1 - x_2) = 0.$$

Since at least one of x_1, x_2, and x_3 is 1 by Equation 1, by checking each case, we have that the solution set to these equations is $\{(\alpha, 1, 1)\}$. As computed previously, this contradicts the solution set implied by the last 3 equations of the first block for $x_4 \neq 1$: $\{(0,0,1)\} \cup \{(1,\alpha,0)\}$.

Suppose now that the first block is defined via Instance 2. By looking at the equations numbered

2, 3 and 4 of the first block, we get

$$(1-x_2)x_3 = 0.$$
$$x_2(1-x_3) = 0.$$
$$x_2x_3 = 0.$$

Since at least one of x_1, x_2, and x_3 is 1 as noted, the solution set to these equations is $\{(1,0,0)\}$. This contradicts the solution set implied by the last 3 equations of Instance 2 for $x_4 \neq 1$: $\{(0,1,0)\} \cup \{(\alpha,0,1)\}$.

Finally, suppose that the first block is defined via Instance 3. By looking at the equations numbered 3, 4 and 5 of the first block, we obtain

$$x_2(1-x_3) = 0.$$
$$x_2x_3 = 0.$$
$$x_1(1-x_2) = 0.$$

With the same premise on x_1, x_2, and x_3 is 1, the solution set to these equations is $\{(0,0,1)\}$. This contradicts the solution set implied by the last 3 equations of Instance 3 for $x_4 \neq 1$: $\{(1,0,0)\} \cup \{(0,1,\alpha)\}$. Thus, either $x_4 = 0$ or $x_4 = 1$.

Observe next that the replaced clauses in each block are satisfiable. Assume $x_4 \neq 0$. If the second block is defined via Instance 1, $x_5 \vee x_7$ does not contradict the solution set for Instance 1, which is $\{(0,0,1)\} \cup \{(1,\alpha,0)\} \cup \{(\alpha,1,1)\}$. Similarly, if the second block is defined via Instance 2, $\overline{x_6} \vee \overline{x_7}$ does not contradict the solution set for Instance 2, which is $\{(1,0,0)\} \cup \{(0,1,0)\} \cup \{(\alpha,0,1)\}$. If the second block is defined via Instance 3, $\overline{x_5} \vee \overline{x_6}$ does not contradict the solution set for Instance 3, which is $\{(0,0,1)\} \cup \{(1,0,0)\} \cup \{(0,1,\alpha)\}$.

We have already shown that for $x_4 = 0$, the solution sets associated to three different types of blocks have the same cohomology. Notice that for $x_4 = 1$, the solution sets associated to these blocks are the ones computed in the discussion above. For Instance 1, it is $(\alpha,1,1,1)$. For Instance 2, it is $(1,0,0,1)$. For Instance 3, it is $(0,0,1,1)$. Thus, the Hilbert polynomials associated to Instance 2 and Instance 3 are the same, whereas Instance 1 differs from them. We consider the following set of instances with uniform Hilbert polynomial. Select out of all instances having $r/2$ blocks defined via Instance 1 and $r/2$ blocks defined via either Instance 2 or Instance 3, where we assume r is even. The number of such instances is $\binom{r}{r/2} \cdot 2^{r/2}$. Using the Stirling approximation, we obtain

$$\binom{r}{r/2} = \frac{r!}{(r/2)!(r/2)!} > \sqrt{\frac{2}{\pi r}} \cdot 2^r \cdot \exp\left(\frac{1}{12r+1} - \frac{1}{3r}\right),$$

so that for all $\epsilon > 0$, we have

$$\binom{r}{r/2} \cdot 2^{r/2} > 2^{\left(\frac{3}{2}-\epsilon\right)r},$$

as r tends to infinity. Noting that $r = n/4$, there remains only to show that we have a prime sub-problem to complete the proof.

Ordering of the Instances: A Prime Sub-problem Represent an instance as a sequence of blocks numbered according to their types. Let $a \in \{2,3\}$. Then each instance is a sequence of r elements with $r/2$ 1s and $r/2$ as. First, fix one such configuration, say $a \ldots a1 \ldots 1$, where the first half is all as and the second half is all 1s. We will describe in this case a *sub-walk* on all the instances defined via the first half, so that the unit operations between instances are all disparate. This defines a prime sub-problem. We will then switch to other configurations one by one in a certain order, each corresponding to a disparate unit operation, so that these operations together with the union of all the sub-walks span all the instances to define an ordering as in the definition of a prime sub-problem.

We now describe a sub-walk spanning all the sub-instances of length $r/2$ consisting of blocks numbered 2 or 3. The sub-walk is essentially the ordering implied by the binary Gray code. Specific to our case, let us remember the inductive construction of the Gray code. Given the base case

$$22 \quad 23 \quad 33 \quad 32,$$

we write this in reverse as

$$32 \quad 33 \quad 23 \quad 22.$$

We then put 2 in front of the original sequence, 3 in front of the reflected one, and combine them:

$$222 \quad 223 \quad 233 \quad 232 \quad 332 \quad 333 \quad 323 \quad 322.$$

Given this construction, the following easily follows by induction, which we shall not prove. Suppose that the element $x \in \{2,3\}$ at position k of a sequence S_i is modified to obtain the next sequence S_{i+1}, and the element at position $k-1$ of S_i is $y \in \{2,3\}$. Then there is no other index j such that S_{j+1} is obtained from S_j by modifying the element x at position k of S_j, and the element at position $k-1$ of S_j is also y. This implies by the mixing property of our construction that the unit operations corresponding to the transitions in the Gray code are all disparate. The ordering implied by the Gray code then defines a prime sub-problem.

Consider now the binary Gray code on the alphabet $\{1, a\}$, starting from all as. Take the set of sequences, which has $r/2$ 1s and $r/2$ as in the order of their appearance. Here is an example for $r = 4$, where the selected sequences are in bold:

$$aaaa, aaa1, \mathbf{aa11}, aa1a, \mathbf{a11a}, a111, \mathbf{a1a1}, a1aa, \mathbf{11aa}, \ldots$$

Consider the unit operations between the instances corresponding to the selected sequences. By the aforementioned property of the Gray code, they are also all disparate. Since the unit operations described for the sub-walks above do not involve blocks of Instance 1, we clearly have that these unit operations are also disparate from them. We thus have defined an ordering of all the instances such that the unit operations between consecutive ones are disparate. This defines a prime sub-problem and completes the proof. □

By Theorem 4.1, Lemma 3.1, and the NP-completeness of 3-SAT [5]:

Corollary 4.2. $\mathsf{P} \neq \mathsf{NP}$.

The definition of τ also implies

Corollary 4.3. $\mathsf{NP} \not\subseteq \mathsf{P/poly}$.

Furthermore, by the specific lower bound derived for 3-SAT:

Corollary 4.4. *The exponential time hypothesis* [3] *is true against deterministic algorithms.*

Finally, this exponential lower bound implies the following by [4].

Corollary 4.5. $\mathsf{BPP} = \mathsf{P}$.

References

[1] A. Grothendieck. *Fondements de la Géométrie Algébrique [Extraits du Séminaire Bourbaki 1957-1962]*, chapter Techniques de construction et théorèmes d'existence en géométrie algébrique. IV. Les schémas de Hilbert. Secr. Math., 1962.

[2] R. Hartshorne. Connectedness of the Hilbert scheme. *Publications Mathématiques de l'IHÉS*, 29:5–48, 1966.

[3] R. Impagliazzo and R. Paturi. On the complexity of k-SAT. *J. Comput. Syst. Sci.*, 62(2):367–375, 2001.

[4] R. Impagliazzo and A. Wigderson. $P = BPP$ if E requires exponential circuits: Derandomizing the XOR lemma. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing*, pages 220–229. ACM, 1997.

[5] R. Karp. Reducibility among combinatorial problems. In R. Miller and J. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.