A Diagonalization-Based Approach to Lower Bounds in Proof Complexity

Rahul Santhanam* Iddo Tzameret†

Abstract

We propose a diagonalization-based approach to several important questions in proof complexity. We illustrate this approach in the context of the algebraic proof system IPS and in the context of propositional proof systems more generally.

We use the approach to give an explicit sequence of CNF formulas $\{\phi_n\}$ such that $\mathsf{VNP} \neq \mathsf{VP}$ iff there are no polynomial-size IPS proofs for the formulas ϕ_n. This is the first *equivalence* between proof complexity lower bounds and standard algebraic complexity lower bounds. Our proof of this fact uses the implication from IPS lower bounds to algebraic complexity lower bounds due to Grochow and Pitassi [GP18] together with a diagonalization argument: the formulas ϕ_n themselves assert the non-existence of short IPS proofs for formulas encoding $\mathsf{VNP} \neq \mathsf{VP}$ at a different input length. Our result also has meta-mathematical implications: it gives evidence for the difficulty of proving strong lower bounds for IPS within IPS.

For any strong enough propositional proof system R, we define the *iterated R-lower bound formulas*, and propose them as explicit hard candidates for the proof system R. We observe that this conjecture holds for Resolution, and give evidence in favour of it for other proof systems.

*Department of Computer Science, Oxford University. rahul.santhanam@cs.ox.ac.uk. Supported in part by ERC Consolidator Grant Agreement No. 615075.

†Department of Computer Science, Royal Holloway, University of London. iddo.tzameret@gmail.com. Part of this work was done while on sabbatical visit to Oxford University, supported in part by ERC Consolidator Grant Agreement No. 615075.

1 Introduction

Diagonalization has been used to show many of the foundational results in logic, including Cantor's theorem about the uncountability of the reals, Gödel's Incompleteness Theorems, and Turing's proof of the undecidability of the Halting Problem. It has also found extensive application in complexity theory, where many unconditional lower bounds use diagonalization directly or indirectly. However, diagonalization has had very little impact in *proof complexity* so far.

Proof complexity studies the question of whether tautologies have short proofs in a given proof system. Cook and Reckhow [CR79] showed that $\mathsf{NP} \neq \mathsf{coNP}$ if and only if there are hard instances for every propositional proof system R, i.e., a sequence of instances that does not have short R-proofs. It is known that there are hard instances for relatively weak proof systems such as Resolution [Hak85] and constant-depth Frege [KPW95, PBI93], but thus far it remains completely open whether the same is the case for the proof systems Frege and Extended Frege. There does not seem to be any promising approach to lower bounds for these stronger proof systems, and we even lack good candidates for hard instances.

In this paper, we propose that a diagonalization-based approach to make progress on several important directions in proof complexity, including *connections between proof complexity and circuit complexity*, *meta-mathematics of proof complexity lower bounds* and *explicit hard instances for propositional proof systems*. We first give some background on each of these directions.

Connections Between Proof Complexity and Circuit Complexity: It is an intriguing fact that many of the known proof complexity lower bounds are shown using techniques that were originally developed in the context of circuit complexity, eg., the technique of random restrictions. Intuitively, it seems as though progress on lower bounds in both areas is stalled for similar reasons, but there are few formal connections known between the two areas. For weak proof systems, the notion of *feasible interpolation* [Kra97] provides such a connection, enabling us to derive small circuits for certain computational problems from short proofs of related formulas. In the converse direction, recent lifting results [GKRS19] give a way to derive circuit complexity lower bounds for weak circuit classes from proof complexity lower bounds for related weak proof systems.

For strong proof systems, there are some intriguing connections known to algebraic complexity (cf. [PT16] for a survey). First, the strong algebraic proof system IPS (Ideal Proof System) introduced by Grochow and Pitassi in [GP18] was introduced in part to relate proof complexity lower bounds to complexity class separations such as VP vs. VNP. Furthermore, we know that some natural conjectures from algebraic circuit complexity, such as the Shub and Smale conjecture about hardness of expressing factorial numbers with $0, 1, -1$ constants and $\times, +$ gates, imply lower bounds on the IPS as shown recently in [AGHT20]. Moreover, unconditional lower bounds on some restricted versions of IPS have been shown to follow from certain algebraic circuits lower bounds in [FSTW16], while [LTW18] established connections between Frege lower bounds and lower bounds on the weak model of noncommutative formulas. Finding further connections is an important problem, so that progress in either area can be transferred to the other.

Meta-mathematics of Proof Complexity Lower Bounds: Proof complexity lower bounds seem to be difficult to show, but there seems to be little formal justification for this. In contrast, the difficulty of showing computational complexity lower bounds is evidenced by barriers such as the relativization barrier and the natural proofs barrier. Understanding the barriers to lower bounds better in the context of proof complexity might help us to make progress. One approach is via connections to circuit complexity. Grochow and Pitassi [GP18] show that super-polynomial

lower bounds for CNFs in the algebraic proof system IPS imply $\mathsf{VNP} \neq \mathsf{VP}$; thus if we believe $\mathsf{VNP} \neq \mathsf{VP}$ is hard to show, then we must believe the same for IPS lower bounds. In a recent work, [PS19] take a different approach, formulating an analogue of the natural proofs barrier for proof complexity. They show unconditionally that for some (non-uniform) propositional proof system R, super-polynomial lower bounds on R-proofs for random truth table formulas cannot be shown efficiently in any non-uniform propositional proof system. However, their proof is non-constructive, and does not seem to yield any new information for commonly studied proof systems such as Frege and Extended Frege.

Explicit Hard Instances for Propositional Proof Systems: Known proof complexity lower bounds for proof systems such as Resolution and constant-depth Frege hold for explicit formulas, indeed the Pigeonhole Principle is hard in both cases. However, for Frege and above, there are few explicit candidates for hard instances, as discussed in [Raz15]. Random CNFs of constant density and random truth table formulas are plausible candidates, but are not explicit. The only plausible explicit candidates of which we are aware are explicit circuit lower bound tautologies and the more general proof complexity generators [ABSRW00, Kra01]. However, even explicit circuit lower bound tautologies are no longer hard candidates for strong propositional proof systems that *can* prove circuit lower bounds, perhaps for the trivial reason that they are axiomatized with circuit lower bounds. In general, there is a need for more plausible candidates for hardness, prior to any approach to showing lower bounds for strong proof systems.

We make progress along the three directions above in the context of the algebraic proof system IPS and for propositional proof systems in general. We next explain our results.

1.1 Our Results

Our main technical result is an *equivalence* between super-polynomial algebraic circuit lower bounds for the Permanent and IPS lower bounds for a certain sequence of explicit CNFs. In the following informal statement of the result, we use "$\mathsf{VNP} \neq \mathsf{VP}$" to denote appropriate CNF encodings of $\mathsf{VNP} \neq \mathsf{VP}$ (itself a sequence of statements asserting that the Permanent, as given by its list of monomials at a given input length, lacks small algebraic circuits) over an implicit finite field $\mathbb{F}$, and $\mathsf{lb}_{IPS}(\text{“}\phi_n\text{”}, s)$ to denote appropriate CNF encoding of the statements that there are no IPS-proof of ϕ_n of size $s(|\phi|)$, where $s : \mathbb{N} \to \mathbb{N}$ is a function.

Theorem 1.1 (Equivalence between Algebraic Circuit Lower Bounds and IPS Lower Bounds; informal)**.** $\mathsf{VNP} \neq \mathsf{VP}$ *iff there is a constant c such that there are no polynomial-size IPS proofs of* $\mathsf{lb}_{IPS}(\text{“}\mathsf{VNP} \neq \mathsf{VP}\text{”}, m^c)$

Note that the explicit formulas $\{\phi_n\}$ which lack efficient IPS proofs iff $\mathsf{VNP} \neq \mathsf{VP}$ are *themselves* encodings of IPS lower bounds for the statement $\mathsf{VNP} \neq \mathsf{VP}$ - this is how we use diagonalization. Our argument combines diagonalization ideas with the result of Grochow and Pitassi [GP18] that IPS lower bounds for CNFs imply circuit lower bounds. We give more details on the proof ideas in the next section. Our proof ideas generalize to give an analogue of Thm. 1.1 for *every* algebraic proof system that efficiently simulates IPS.

Thm. 1.1 is relevant to two of the three directions we cited as motivation earlier. It gives a new connection between algebraic complexity and proof complexity, which could be useful in either direction. It is also relevant to the meta-mathematics of IPS lower bounds. For a certain natural explicit family of formulas expressing algebraic circuit lower bounds for the Permanent, super-polynomial IPS lower bounds are *themselves* hard to show in IPS, if we believe that $\mathsf{VNP} \neq \mathsf{VP}$. Thus, under a standard complexity conjecture, IPS finds it hard to reason about itself.

One possibility, of course, is that $\mathsf{VNP} \neq \mathsf{VP}$ actually has *polynomial-size* IPS proofs, in which case $\mathsf{lb}_{IPS}(\text{“}\mathsf{VNP} \neq \mathsf{VP}\text{”}, m^{\omega(1)})$ is not a tautology, and hence trivially lacks small IPS proofs. This, however, cannot happen if we assume an algebraic analogue of Razborov's Conjecture [Raz15]. Razborov's Conjecture states that Extended Frege cannot efficiently prove super-polynomial circuit lower bounds for any Boolean function (as expressed in the so-called truth table formulas). A reasonable algebraic analogue of Razborov's Conjecture is that IPS cannot efficiently prove super-polynomial algebraic circuit lower bounds for any polynomial, which rules out the possibility above.

Thm. 1.1 gives evidence that IPS finds it hard to reason efficiently about itself - could this hold for proof systems more generally[1]? A heuristic way to think of Thm. 1.1 is as a *fixed point* theorem in the following sense: consider $\mathsf{lb}_R(\cdot, m^{\omega(1)})$ for a proof system R as an operator mapping formulas to formulas. Let lb_R^2 be the composition of this operator with itself. Then the sequence of formulas expressing that $\mathsf{VNP} \neq \mathsf{VP}$ is a fixed point for lb_R^2 in the sense that it preserves truth when R is IPS. Indeed, our diagonalization approach is inspired partly by [AM19, Gar19] who showed implicitly that *every* sequence of formulas is a fixed point for lb_R^2 when R is Resolution, and by [PS19] who showed implicitly that for every strong enough (nonuniform) propositional proof system R (simulating Extended Frege), the *distribution* of random truth table formulas is a fixed point for lb_R^2.

Here we explore the idea that if R finds it hard to reason about itself, then iterating lb_R provides an explicit hard sequence of formulas for R. Assume that R is not polynomially bounded, and let ϕ be a fixed formula that does not have $|\phi|^c$ size R-proofs, for some constant c. We define the iterated lower bound formulas $\mathsf{lb}_R^k(\phi, n^c)$ inductively as follows:

$$\mathsf{lb}_R^0(\phi, n^c) = \phi$$

$$\mathsf{lb}_R^{k+1}(\phi, n^c) = \mathsf{lb}_R(\mathsf{lb}_R^k(\phi, n^c), n^c)$$

We propose the Iterated Lower Bound Conjecture: for every reasonably strong[2] propositional proof system R that is not polynomially bounded, there is a ϕ such that the sequence $\{\mathsf{lb}_R^k(\phi)\}$ is a sequence of hard instances for R.

This gives a candidate family of hard instances for *every* strong enough proof system. We believe that there is a win-win aspect to studying this Conjecture - even it fails, this gives us information about propositional proof systems can reason about lower bounds for themselves. We are not currently aware of any *natural* propositional proof system R that is capable of this.

We provide some evidence in favour of the conjecture. First, as a corollary to [AM19, Gar19], it follows that the Conjecture holds for Resolution.

Theorem 1.2. *The Iterated Lower Bounds Conjecture holds for Resolution.*

Second, we show that under a conjecture of Rudich [Rud97] about non-existence of short propositional proofs for random truth table formulas, any finite number of iterations preserves hardness for random truth table formulas, for some strong propositional proof system R that efficiently simulates Extended Frege.

Theorem 1.3. *There is a propositional proof system R efficiently simulating Extended Frege such that under Rudich's Conjecture, for every large enough constant c and every positive integer k,*

[1] Independently of our work, [Pud19] poses the same question.

[2] We formally define "reasonably strong" later.

$\mathsf{lb}_R^k(\phi, n^c)$ *does not have R-proofs of size* $|\mathsf{lb}_R^k(\phi, n^c)|^c$ *with high probability over* ϕ *a random truth table formula.*

1.2 Overview of Techniques

We give a high-level overview of the ideas required to show Thm. 1.1. Informally, we would like to show that $\mathsf{VNP} \neq \mathsf{VP}$ iff IPS cannot efficiently prove IPS lower bounds for "$\mathsf{VNP} \neq \mathsf{VP}$".

The proof builds on several technical constructions. The gist of the argument is a form of diagonalisation: the existence of a short proof of a proof complexity lower bound for the statement $\mathsf{VNP} \neq \mathsf{VP}$ implies in fact that there *is* a short proof of $\mathsf{VNP} \neq \mathsf{VP}$ (at a smaller input length) due to the reduction of Grochow and Pitassi [GP18], that we show is efficiently formalizable already inside IPS.

For this purpose we make the following assumptions, which we will justify later: There is a reasonable CNF encoding of "$\mathsf{VNP} \neq \mathsf{VP}$"; there is a reasonable CNF encoding of the statement that there are no IPS lower bounds of size s for a CNF ϕ; If ϕ is a tautology and there are short IPS proofs of "IPS does not efficiently prove ϕ", then there are short IPS proofs of "$\mathsf{VNP} \neq \mathsf{VP}$". Where the final assumption can be thought of as the formalization of the Grochow-Pitassi implication from IPS lower bounds for CNFs to $\mathsf{VNP} \neq \mathsf{VP}$ within IPS. A priori, it is hard to see how to establish it since we only know that ϕ is a tautology and do not have proofs of this fact. It will turn out that the parameters can be set so that truth-table proofs of ϕ suffice.

Given these assumptions, we proceed as follows. First, we show the forward direction. Assume $\mathsf{VNP} \neq \mathsf{VP}$. Then "$\mathsf{VNP} \neq \mathsf{VP}$" is a CNF tautology, using Assumption 1. Assume for the sake of contradiction that lb_{IPS}("$\mathsf{VNP} \neq \mathsf{VP}$", $n^{\omega(1)}$) has polynomial-size proofs in IPS. Then by Assumption 3, "$\mathsf{VNP} \neq \mathsf{VP}$" has polynomial-size proofs in IPS. But this contradicts the soundness of IPS, since IPS has polynomial-size proofs of the statement that "$\mathsf{VNP} \neq \mathsf{VP}$" requires superpolynomial-size IPS-proofs.

For the backward direction, either $\phi =$ "lb_{IPS}("$\mathsf{VNP} \neq \mathsf{VP}$", $n^{\omega(1)}$)" is true or it is not. If it is true, then $\mathsf{lb}_{IPS}(\phi, n^{\omega(1)})$ implies $\mathsf{VNP} \neq \mathsf{VP}$ by [GP18], using Assumption 2. If it is false, then "$\mathsf{VNP} \neq \mathsf{VP}$" has poly-size proofs in IPS. By the soundness of IPS, this implies $\mathsf{VNP} \neq \mathsf{VP}$.

The above is only a template and hides many technical issues and details. For example, in practice, we work with the statement $\mathsf{VNP} \neq \mathsf{VP}$ at a given input length, and we need to clarify how the input lengths of different occurrences of the statement relate to each other.

Technically, for the sake of the formalisation in IPS we need to be able to speak at the IPS proof level about circuit lower bounds, proof complexity lower bounds, and the reduction between them as in [GP18]. To formalise statements about IPS proofs and circuit class separations we express polynomials as vectors of coefficients. To express the existence of small circuits we use universal circuits with t edges as defined in Raz [Raz10]. To express that a polynomial is computable by a small circuit we use a set of equations, each stating the appropriate value of the coefficient of a given monomial in a polynomial computed by a universal circuit with t edges.

Using this formalisation of polynomials computed by small circuits we can encode $\mathsf{VP} \neq \mathsf{VNP}$ as the statement expressing that the coefficients vector of the permanent polynomial does not correspond to the monomial coefficients of any small universal circuit. Similarly, the IPS proof predicate is expressed by stating the existence of a small universal circuit that computes (similarly, based on its monomial coefficients) the IPS certificate of a given CNF. For the purpose of expressing statements about algebraic circuits such as $\mathsf{VP} \neq \mathsf{VNP}$ as CNF formulas we first need to work over finite fields, and second need to devise ways to move from CNF formulas encoding circuits to the circuit they express.

2 Preliminaries

2.1 Basic Algebraic Complexity

For a very good treatise on algebraic circuits and their complexity see Shpilka and Yehudayoff [SY10]. Let $\mathbb{G}$ be a ring. Denote by $\mathbb{G}[X]$ the ring of (commutative) polynomials with coefficients from $\mathbb{G}$ and variables $X := \{x_1, x_2, \dots\}$. A *polynomial* is a formal linear combination of monomials, where a *monomial* is a product of variables. Two polynomials are *identical* if all their monomials have the same coefficients. The *degree* of a polynomial is the maximal total degree of a monomial in it.

Algebraic circuits and formulas over the ring $\mathbb{G}$ compute polynomials in $\mathbb{G}[X]$ via addition and multiplication gates, starting from the input variables and constants from the ring. More precisely, an *algebraic circuit* C is a finite directed acyclic graph (DAG) with *input nodes* (i.e., nodes of in-degree zero) and a single *output node* (i.e., a node of out-degree zero). Input nodes are labeled with either a variable or a ring element in $\mathbb{G}$. All the other nodes have *fan-in* (that is, in-degree) *two* and are labeled by either an addition gate $+$ or a product gate $\times$. Every node in an algebraic circuit C *computes* a polynomial as follows: an input node computes the variable or scalar that labels it. A $+$ (or $\times$) gate is said to compute the addition (product, resp.) of the (commutative) polynomials computed by its incoming nodes. The polynomial computed by a node u in an algebraic circuit C is denoted $\widehat{u}$. Given a circuit C, we denote by $\widehat{C}$ the polynomial computed by C, that is, the polynomial computed by the output node of C. The ***size*** of a circuit C is the number of nodes in it, denoted $|C|$, and the *depth* of a circuit is the length of the longest directed path in it. For an algebraic circuit C we write $C(a/x)$ to denote the *substitution instance* of C in which every occurrence of the node x is replaced by the sub-circuit a; in case $C(x)$ is written with its displayed variable(s) x we can write $C(x)(a/x)$ for this substitution instance. We say that a polynomial is *homogeneous* whenever every monomial in it has the same (total) degree.

Definition 2.1 (Syntactic-degree $\mathrm{sdeg}(\cdot)$)**.** *Let C be a circuit (without division) and v a node in C. The* syntactic-degree $\mathrm{sdeg(v)}$ *of v is defined as follows:*

1. *If v is a field element or a variable, then* $\mathrm{sdeg(v)} := 0$ *and* $\mathrm{sdeg(v)} := 1$*, respectively;*
2. *If $v = u + w$ then* $\mathrm{sdeg(v)} := \max\{\mathrm{sdeg(u)}, \mathrm{sdeg(w)}\}$*;*
3. *If $v = u \cdot w$ then* $\mathrm{sdeg(v)} := \mathrm{sdeg(u)} + \mathrm{sdeg(w)}$*.*

An algebraic circuit is said to be *syntactic-homogeneous* if for every plus gate $u + v$, $\deg(u) = \deg(v)$.

Algebraic Complexity Classes. We now recall some basic notions from algebraic complexity (for more details see [SY10, Sec. 1.2]). Over a ring R, VP_R (for "Valiant's P") is the class of families $f = (f_n)_{n=1}^{\infty}$ of formal polynomials f_n such that f_n has $\mathrm{poly}(n)$ input variables, is of $\mathrm{poly}(n)$ degree, and can be computed by algebraic circuits over R of $\mathrm{poly}(n)$ size. VNP_R (for "Valiant's NP") is the class of families g of polynomials $(g_n)_{n=1}^{\infty}$ such that g_n has $\mathrm{poly}(n)$ input variables and is of $\mathrm{poly}(n)$ degree, and can be written as

$$g_n(x_1, \dots, x_{\mathrm{poly}(n)}) = \sum_{\overline{e} \in \{0,1\}^{\mathrm{poly}(n)}} f_n(\overline{e}, \overline{x})$$

for some family $(f_n)_{n=1}^{\infty} \in \mathsf{VP}_R$.

A polynomial $f(\overline{x})$ is a *projection* of a polynomial $g(\overline{y})$ if $f(\overline{x}) = g(L(\overline{x}))$ identically as polynomials in $\overline{x}$, for some map L that assigns to each y_i either a variable or a constant. In other words, a projection of $g(\overline{y})$ is a substitution instance of $g(\overline{y})$ in which $\overline{y}$ variables are substituted by $\overline{x}$ variables or field elements. A family of polynomials (f_n) is a polynomial projection or *p-projection* oft another family (g_n) if there is a function $t(n) = n^{\Theta(1)}$ such that f_n is a projection of $g_{t(n)}$ for all (sufficiently large) n. The *permanent* polynomial $\sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i,\sigma(i)}$ (for S_n the permutation group on n elements) is complete under p-projections for VNP. The *determinant* polynomial on the other hand is known to be in VP but is not known to be complete for VP under p-projections.

Two central questions in algebraic complexity theory are whether the permanent is a p-projection of the determinant (a stronger variant speaks about quasi-polynomial projections); and whether VP equals VNP [Val79a, Val79b, Val82]. Since the permanent is complete for VNP (under p-projections), showing VP≠VNP amounts to proving that the permanent cannot be computed by polynomial-size algebraic circuits.

2.2 Algebraic Proof Systems

Grochow and Pitassi [GP18] suggested the following algebraic proof system which is essentially a Nullstellensatz proof system ([BIK+96]) written as an algebraic circuit. A proof in the Ideal Proof System is given as a *single* polynomial. We provide below the *boolean* version of IPS (which includes the boolean axioms), namely the version that establishes the unsatisfiability over 0-1 of a set of polynomial equations. In what follows we follow the notation in [FSTW16]:

Definition 2.2 ((boolean) Ideal Proof System (IPS), Grochow-Pitassi [GP18])**.** *Let $f_1(\overline{x}), \ldots, f_m(\overline{x}), p(\overline{x})$ be a collection of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ over the field $\mathbb{F}$. An **IPS proof of** $p(\overline{x}) = 0$ **from** $\{f_j(\overline{x}) = 0\}_{j=1}^{m}$, showing that $p(\overline{x}) = 0$ is semantically implied from the assumptions $\{f_j(\overline{x}) = 0\}_{j=1}^{m}$ over 0-1 assignments, is an algebraic circuit $C(\overline{x}, \overline{y}, \overline{z}) \in \mathbb{F}[\overline{x}, y_1, \ldots, y_m, z_1, \ldots, z_n]$ such that (the equalities in what follows stand for formal polynomial identities[3]):*

1. *$C(\overline{x}, \overline{0}, \overline{0}) = 0$; and*
2. *$C(\overline{x}, f_1(\overline{x}), \ldots, f_m(\overline{x}), x_1^2 - x_1, \ldots, x_n^2 - x_n) = p(\overline{x})$.*

*The **size of the IPS proof** is the size of the circuit C. The variables $\overline{y}, \overline{z}$ are called the* placeholder variables *since they are used as placeholders for the axioms. An IPS proof $C(\overline{x}, \overline{y}, \overline{z})$ of $1 = 0$ from $\{f_j(\overline{x}) = 0\}_{j \in [m]}$ is called an **IPS refutation** of $\{f_j(\overline{x}) = 0\}_{j \in [m]}$ (note that in this case it must hold that $\{f_j(\overline{x}) = 0\}_{j=1}^{m}$ have no common solutions in $\{0,1\}^n$).*

Notice that the definition above adds the equations $\{x_i^2 - x_i = 0\}_{i=1}^{n}$, called the set of ***boolean axioms*** denoted $\overline{x}^2 - \overline{x}$, to the system $\{f_j(\overline{x}) = 0\}_{j=1}^{m}$. This allows to refute over $\{0,1\}^n$ unsatisfiable systems of equations. Also, note that the first equality in the definition of IPS means that the polynomial computed by C is in the ideal generated by $\overline{y}, \overline{z}$, which in turn, following the second equality, means that C witnesses the fact that 1 is in the ideal generated by $f_1(\overline{x}), \ldots, f_m(\overline{x}), x_1^2 - x_1, \ldots, x_n^2 - x_n$ (the existence of this witness, for unsatisfiable set of polynomials, stems from the Nullstellensatz theorem [BIK+96]).

In order to use IPS as a propositional proof system (namely, a proof system for propositional tautologies), we need to fix the encoding of clauses as algebraic circuits.

[3] That is, $C(\overline{x}, \overline{0}, \overline{0})$ computes the zero polynomial and $C(\overline{x}, f_1(\overline{x}), \ldots, f_m(\overline{x}), x_1^2 - x_1, \ldots, x_n^2 - x_n)$ computes the polynomial $p(\overline{x})$.

Definition 2.3 (algebraic translation of CNF formulas)**.** *Given a CNF formula in the variables $\overline{x}$, every clause $\bigvee_{i\in P} x_i \vee \bigvee_{j\in N} \neg x_j$ is translated into $\prod_{i\in P}(1-x_i)\cdot\prod_{j\in N} x_j = 0$. (Note that these terms are written as algebraic circuits as displayed, where products are not multiplied out.)*

Notice that in this way a 0-1 assignment to a CNF is satisfying iff the assignment is satisfying all the equations in the algebraic translation of the CNF.

Therefore, using Definition 2.3 to encode CNF formulas, boolean IPS is considered as a propositional proof system for the language of unsatisfiable CNF formulas, sometimes called *propositional IPS*. We say that an IPS proof is an ***algebraic IPS*** proof, if we do not use the boolean axioms $\overline{x}^2 - \overline{x}$ in the proof. In our applications we are going to use algebraic IPS refutations, while sometimes explicitly adding the boolean axioms for some variables (while leaving them out for some other variables). *As a default when referring to IPS we mean the boolean* IPS *version. When we use algebraic IPS we will say that explicitly.*

The following is the main structural-complexity result for IPS. Notice that it already works for *algebraic* IPS and this will be important for us.

Theorem 2.1 (Grochow-Pitassi [GP18])**.** *For any ring R, a super-polynomial lower bound on algebraic* IPS *refutations (and hence also on* IPS *refutations) over R for any family of CNF formulas implies $\mathsf{VNP}_R \neq \mathsf{VP}_R$. The same result hold if we assume that the IPS refutation size lower bound holds* only infinitely often*.*

The following lemma is the key to the proof of the Theorem, and is used in our application:

Lemma 2.2. *Every family of unsatisfiable CNF formulas (φ_n) has a family of algebraic* IPS *(and hence also of* IPS*) certificates (C_n) in VNP_R.*

Proof of Thm. 2.1, assuming Lemma 2.2. For a given set $\mathcal{F}$ of unsatisfiable polynomial equations $F_1 = \cdots = F_m = 0$, a lower bound on algebraic IPS refutations of $\mathcal{F}$ is equivalent to giving the same circuit lower bound on *all* IPS certificates for $\mathcal{F}$. A super-polynomial lower bound on IPS implies that some function in VNP—namely, the VNP-IPS certificate guaranteed by Lemma 2.2—cannot be computed by polynomial-size algebraic circuits, and hence that $\mathsf{VNP} \neq \mathsf{VP}$. □

2.2.1 Conventions and Notations for IPS Proofs

An IPS (algebraic or not) proof over a specific field or ring is sometimes denoted $\mathrm{IPS}_{\mathbb{F}}$ noting it is over $\mathbb{F}$. For two algebraic circuits F, G, we define the *size of the equation* $F = G$ to be the total circuit size of F and G, namely, $|F| + |G|$. For a set $\overline{\mathcal{F}}$ of equations between circuits we denote by $|\overline{\mathcal{F}}|$ to be the total size of the equations in the set.

Let $\overline{\mathcal{F}}$ denote a set of polynomial equations $\{f_i(\overline{x}) = 0\}_{i=1}^m$, and let $C(\overline{x}, \overline{y}, \overline{z}) \in \mathbb{F}[\overline{x}, \overline{y}, \overline{z}]$ be an IPS proof of $f(\overline{x})$ from $\overline{\mathcal{F}}$ as in Definition 2.2. Then we write $C(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x})$ to denote the circuit C in which y_i is substituted by $f_i(\overline{x})$ and z_i is substituted by the boolean axiom $x_i^2 - x_i$. By a slight abuse of notation we also call $C(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x}) = f(\overline{x})$ an IPS proof of $f(\overline{x})$ from $\overline{\mathcal{F}}$ and $\overline{x}^2 - \overline{x}$ (that is, displaying $C(\overline{x}, \overline{y}, \overline{z})$ *after* the substitution of the placeholder variables $\overline{y}, \overline{z}$ by the axioms in $\overline{\mathcal{F}}$ and $\overline{x}^2 - \overline{x}$, respectively).

For two polynomials $f(\overline{x}), g(\overline{x})$, *an IPS proof of* $f(\overline{x}) = g(\overline{x})$ from the assumptions $\overline{\mathcal{F}}$ is an IPS proof of $f(\overline{x}) - g(\overline{x}) = 0$ (note that in case $f(\overline{x})$ and $g(\overline{x})$ are identical as polynomials this is trivial to prove; see Fact A.1).

We denote by $C : \overline{\mathcal{F}} \vdash^s_{\mathrm{IPS}} p = 0$ (resp. $C : \overline{\mathcal{F}} \vdash^s_{\mathrm{IPS}} p = g$) the fact that $p = 0$ (resp. $p = g$) has an IPS proof $C(\overline{x}, \overline{y}, \overline{z})$ of size s from assumptions $\overline{\mathcal{F}}$. Assumptions $\overline{\mathcal{F}}$ can be written either as a set of equations or as a sequence of equations or sets thereof separated by commas. We may also

suppress "$= 0$" and write simply $C : \overline{\mathcal{F}} \vdash^{s}_{\text{IPS}} p$ for $C : \overline{\mathcal{F}} \vdash^{s}_{\text{IPS}} p = 0$. Whenever we are only interested in claiming the existence of an IPS proof of size s of $p = 0$ from $\overline{\mathcal{F}}$ we suppress the C from the notation. Similarly, we can suppress the size parameter s from the notation. If F is a circuit computing a polynomial $\widehat{F} \in \mathbb{F}[\overline{x}]$, then we can talk about *an IPS proof C of F from assumptions $\overline{\mathcal{F}}$*, in symbols $C : \overline{\mathcal{F}} \vdash_{\text{IPS}} F$, meaning an IPS proof of $\widehat{F}$. Accordingly, for two circuits F, F' such that $\widehat{F} = \widehat{F}'$, we may speak about an *an IPS proof C of F from assumptions $\overline{\mathcal{F}}$* to refer to an IPS proof of F' from assumptions $\overline{\mathcal{F}}$.

When we deal with *algebraic* IPS proofs we will use the same notation as above, only using IPS^{nb} instead of IPS, to denote "no boolean" IPS.

3 Proof Complexity Characterization of VP $\neq$ VNP

3.1 Formalisations

3.1.1 CNF Encoding of Algebraic Circuit Equations

We are going to work at times with IPS *without* the boolean axioms present for all variables (but only for some variables), namely algebraic IPS, denoted IPS^{nb} (for "IPS with no boolean axioms"). If we show that it is hard to refute that IPS without the boolean axioms has small refutations of some statements, then we also show that it is hard to refute that an IPS *with* the boolean axioms has small refutations of this statement (otherwise, a short refutation of the existence of IPS refutation with boolean axioms would imply a short refutation of the existence of a refutation in a weaker system). Hence in what follows even when we discuss CNF formulas translated to the algebraic setting we shall write precisely what are the boolean axioms that we add to the formulas.

For an algebraic circuit C and b a field element, we call $C = b$ a *circuit equation* (we sometimes use the same notion for equations between two circuits). We work over a *finite* field $\mathbb{F}_q$. This is necessary in our argument to be able to switch between formulas in CNF and algebraic circuit equations. Recall that a formula in CNF ("a CNF" for short) is a conjunction of clauses, where a clause is a disjunction of literals, and literals are variables or their negation. The algebraic translation of a formula in CNF is defined according to Definition 2.3. The size of objects like circuits, circuit equations, sets of circuit equations and formulas in CNF is denoted by $|\cdot|$ (where "$\cdot$" is replaced by the object).

Definition 3.1 (Algebraic extension axioms and unary bits)**.** *Given a circuit C in the $\overline{x}$ variables and a gate g in C, we call the equation*

$$x_g = \sum_{i=0}^{q-1} i \cdot x_{gi}$$

*the **algebraic extension axiom of** g, with the variable x_{g_i} being the ith **unary-bit of** g.*

Note that if the unary-bits of g are taken over $\{0, 1\}$ and assuming that $x_{gi} = 1$ for precisely one $0 \leq i \leq q-1$, then $x_g = i$ iff $x_{gi} = 1$.

Definition 3.2 (Plain CNF encoding of algebraic circuits)**.** *Let $C(\overline{x}) = 0$ be a circuit equation in the variables $\overline{x}$. The **plain CNF encoding** of the circuit equation $C(\overline{x}) = 0$ denoted $\mathsf{cnf}(C(\overline{x}) = 0)$ consists of the following CNFs in the unary-bits variables of all the gates in C:*

1. *If x_i is an input gate in C, the plain CNF encoding of C contains the variables $x_{x_i 0}, \ldots, x_{x_i (q-1)}$ that are the unary-bits of x_i, as well as the following clauses that express*

that precisely one unary-bit is 1 and all other unary-bits are 0: [4]

$$\bigvee_{j=0}^{q-1} x_{x_i j} \wedge \bigwedge_{j \neq \ell \in \{0,\ldots,1\}} (\neg x_{x_i j} \vee \neg x_{x_i \ell}). \tag{1}$$

2. *For every gate g in $C(\overline{x})$ and every satisfying assignment $\overline{\alpha}$ to the plain CNF encoding, the corresponding unary-bit x_{gi} evaluates to 1 iff the value of g is $i \in \{0,\ldots,q-1\}$ (when the algebraic inputs $\overline{x} \in \mathbb{F}_q^*$ to $C(\overline{x})$ take on the values corresponding to the boolean assignment $\overline{\alpha}$). This is ensured with the following equations: if $g = u \circ v$ is an internal gate in C (including the output gate, but excluding the input gates), for $\circ \in \{+,\times\}$, we have a CNF φ_g in the unary-bits variables of g, u, v that is satisfied by an assignment precisely when the output unary-bits of g get their correct value based on the (constant-size) truth table of $\circ$ over $\mathbb{F}_q$ and the input unary-bits of u, v (we ensure that if more than one unary-bit is assigned 1 in any of the unary-bits of g, u, v then the CNF is unsatisfiable).*

3. *For the output gate g_{out} of C we add the equations: $x_{g_{out}0} = 1$ and $x_{g_{out}i} = 0$ for all $i = 1,\ldots,q-1$, which express that $g_{out} = 0$.*

4. *For every unary-bit variable x_{gi} we have the boolean axiom (recall we write these boolean axioms explicitly since we are going to work with* $\mathrm{IPS}^{\mathrm{nb}}$*):*

$$x_{gi}^2 - x_{gi} = 0\,.$$

Definition 3.3 (Extended CNF encoding of a circuit equation; $\mathsf{ecnf}(\cdot)$)**.** *Let $C(\overline{x})$ be a circuit in the $\overline{x}$ variables over the finite field $\mathbb{F}_q$. Then the* ***extended CNF encoding*** *of the circuit equation $C(\overline{x}) = 0$, in symbols $\mathsf{ecnf}(C(\overline{x}) = 0)$, is defined to be a set of algebraic equations over $\mathbb{F}_q$ in the variables x_g and $x_{g0},\ldots,x_{gq-1}$ which are the unary-bits variables corresponding to node g in C, that consists of:*

1. *the plain CNF encoding of $C(\overline{x}) = 0$; and*
2. *the algebraic extension axiom of g, for every gate g in C; and*
3. *the boolean axioms for the unary-bits variables: $x_{gj}^2 - x_{gj} = 0$, for every $j \in \{0,\ldots,q-1\}$ and every gate g in C.*

Notice that the extended CNF encoding is not formally a CNF since it uses the algebraic extension axioms which are not clauses. Also, note that the extended CNF encoding does not contain the boolean axioms for the algebraic extension variables x_g, for g a gate in C, as this variable is meant to range over $\mathbb{F}_q$.

Since we work with extension variables for each gate in a given circuit equation $C(\overline{x}) = 0$, it is more convenient to express circuit equations as a *set* of equations that correspond to the *straight line program* (**SLP**) of $C(\overline{x})$ (which is an equivalent formulation to algebraic circuits). In a SLP we have a sequence of equations between variable such that the extension variable for the output gate computes the value of the circuit assuming all equations hold as follows: we choose any topological order $g_1, g_2, \ldots, g_i, \ldots, g_{|C|}$ on the gates of the circuit C (that is, if g_j has a directed path to g_k in C then $j < k$) and define the following equations: $g_i = g_j \circ g_k$ for $\circ \in \{+,\times\}$ iff g_i is a $\circ$ gate in C with two incoming edges from g_j and g_k. And SLP representation of a circuit equation $C(\overline{x}) = 0$

[4] This conditions is needed only for inputs. For internal gates the CNFs expressing the truth table for the gate will make sure that only one output unary-bit is one.

means that we add to the SLP above the equation $g_{|C|} = 0$, where $g_{|C|}$ is the output gate of the circuit.

Using the concept of extended CNF encoding we can now show how to efficiently go in IPS from a circuit equation written as a set of equations for the corresponding SLP to a CNF, and vice versa. The idea is to augment the SLP of $C(\overline{x}) = 0$ with $x_g = \sum_{i=0}^{q-1} i \cdot x_{gi}$ which is the algebraic extension axiom of g, for every gate g in C. We show that, efficiently in IPS, we can go from this representation of $C(\overline{x}) = 0$ to its extended CNF encoding, and vice versa.

Proposition 3.1 (Translating between extended CNFs and circuit equations)**.** *Let $\mathbb{F}$ be a finite field, and let $C(\overline{x})$ be a circuit in the $\overline{x}$ variables over $\mathbb{F}$ that is* written as a set of equations *corresponding to the SLP of $C(\overline{x})$. Then, the following both hold:*

$$\mathsf{ecnf}(C(\overline{x}) = 0) \vdash^*_{\text{IPS}} C(\overline{x}) = 0 \tag{2}$$

$$\left\{ x_g = \sum_{i=0}^{q-1} i \cdot x_{gi} : g \text{ a node in } C \right\}, C(\overline{x}) = 0 \vdash^*_{\text{IPS}} \mathsf{ecnf}(C(\overline{x}) = 0). \tag{3}$$

Proof: The proof of eq. 2 is as follows. For every gate $g = u \circ v$ (for $\circ \in \{+, \times\}$) we have the corresponding truth table CNF for $\circ$ over $\mathbb{F}_q$, that is satisfied only when the unary-bits of gate g, denoted collectively as $\overline{x_{gi}}$ correspond to the correct output of g given as input the unary-bits of u, v, denoted accordingly by $\overline{x_{ui}}, \overline{x_{vi}}$, respectively. Denote this CNF by $\mathsf{plus}(\overline{x_{gi}}, \overline{x_{ui}}, \overline{x_{vi}})$ or $\mathsf{times}(\overline{x_{gi}}, \overline{x_{ui}}, \overline{x_{vi}})$, according to $\circ = +$, $\circ = \times$, respectively. In $\mathsf{ecnf}(C(\overline{x}) = 0)$ we also have the algebraic extension axioms for g, v, u. It remains to prove that from

$$\mathsf{plus}(\overline{x_{gi}}, \overline{x_{ui}}, \overline{x_{vi}}), \text{and} \tag{4}$$

$$x_g = \sum_{i=0}^{q-1} i \cdot x_{gi}, \quad x_u = \sum_{i=0}^{q-1} i \cdot x_{ui}, \quad x_v = \sum_{i=0}^{q-1} i \cdot x_{vi}\,. \tag{5}$$

we can derive in constant-size IPS derivation

$$x_g = x_u + x_v\,, \tag{6}$$

and similarly for the product gate. This suffices to conclude the proof because the collection of eq. 6 equations for all gates g in C are precisely the SLP of $C(\overline{x})$.

We prove the case for plus gates (the case of product gates is similar). Observe that the boolean axioms and eq. 4 semantically imply (over $\mathbb{F}_q$)

$$\sum_{i=0}^{q-1} i \cdot x_{gi} = \sum_{i=0}^{q-1} i \cdot x_{ui} + \sum_{i=0}^{q-1} i \cdot x_{vi}\,. \tag{7}$$

By implicational completeness of IPS over 0-1 assignments (this is proved by induction on the number of variables, essentially trying out all possible 0-1 assignments to the variables) we get that from the boolean axioms and eq. 4 we have a constant-size derivation of eq. 7 (since the number of variables involved is constant). By substituting eq. 5 in eq. 7 we finish. □

We wish to speak about CNF formulas and not extended CNFs. This is necessary since the work of [GP18], showing that an IPS lower bound implies $\mathsf{VNP} \neq \mathsf{VP}$, is known to hold only for lower bounds against CNFs.

We have the following simple proposition:

Proposition 3.2. *Let $C(\overline{x}) = 0$ be circuit equation over $\mathbb{F}_q$. Then, $\mathsf{cnf}(C(\overline{x}) = 0)$ is an unsatisfiable CNF iff $\mathsf{ecnf}(C(\overline{x}) = 0)$ is an unsatisfiable set of equations over $\mathbb{F}_q$ iff $C(\overline{x}) = 0$ is unsatisfiable over $\mathbb{F}_q$.*

Proof: $\mathsf{ecnf}(C(\overline{x}) = 0)$ only adds to $\mathsf{cnf}(C(\overline{x}) = 0)$ the algebraic extension axioms in the *new* variables x_g, for every gate g in C, where x_g does not occur anywhere else in $\mathsf{ecnf}(C(\overline{x}) = 0)$ (recall there are no boolean axioms for the variables x_g). Hence, every satisfying assignment to $\mathsf{cnf}(C(\overline{x}) = 0)$ can be extended to a satisfying assignment of $\mathsf{ecnf}(C(\overline{x}) = 0)$. □

Since by [GP18] every unsatisfiable CNF has an IPS refutation computable in VNP, by the proposition above we get:

Corollary 3.3. *If $\mathsf{ecnf}(C(\overline{x}) = 0)$ is unsatisfiable then it has an* IPS *refutation in* VNP.

3.1.2 Encoding Universal Circuits

To express in the theory that a circuit computes a certain polynomial we will use the concept of a universal circuit as introduced by Raz [Raz10]. A universal (algebraic) circuit is an algebraic circuit that loosely speaking embeds all possible circuits of a certain size. More precisely, a universal circuit for the class of polynomials in $\mathbb{F}[\overline{x}]$ that have algebraic circuits of size at most t is a circuit $U(\overline{x}, \overline{w})$ with two sets of variables $\overline{x}$ and $\overline{w}$, such that $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ has circuit of size at most t iff there is a fixed choice of values $\overline{\alpha}$ to $\overline{w}$ for which $U(\overline{x}, \overline{\alpha}) = f(\overline{x})$ (as a polynomial identity). Intuitively one can think of the $\overline{w}$ variables as the *circuit variables*, while the $\overline{x}$ variables are the algebraic variables of the circuit that is computed by the $\overline{w}$ variables. Formally, $\overline{w}$ describe the edge labels put on edges of a universal circuit.

Raz [Raz10] showed the existence of small algebraic universal circuits for homogeneous polynomials (see also an intuitive description in [SY10]):

Theorem 3.4 (Existence of universal circuits; Raz [Raz10])**.** *Let $\mathbb{F}$ be a field and $\overline{x}$ be n variables, and let $\mathcal{C}^{\text{hom}}_{t,d}$ denote the class of all homogeneous polynomials of total degree exactly d in $\mathbb{F}[\overline{x}]$ that have algebraic circuits of size at most t. Then there is a circuit $U(\overline{x}, \overline{w}) \in \mathbb{F}[\overline{x}, \overline{w}]$ of size $O(dt^4)$ and syntactic-degree d such that $\overline{w}$ are t variables which are disjoint from $\overline{x}$, that is* universal *for $\mathcal{C}^{\text{hom}}_{t,d}$ in the following sense: $f(\overline{x}) \in \mathcal{C}^{\text{hom}}_{t,d}$ iff there exists $\overline{\alpha} \in \mathbb{F}^t$ such that $U(\overline{x}, \overline{\alpha}) = f(\overline{x})$.*

The idea behind the proof of Thm. 3.4 is to provide a normal form for circuits: every syntactic homogeneous circuit of degree d is reduced with a small increase in size to a normal form in which different choices of edge labels determine the polynomial the circuit computes.

Since we do not work necessarily with homogeneous circuits and polynomials we will assume that the universal circuit $U(\overline{x}, \overline{w})$ is in fact universal for general homogeneous circuits. We can assume this by defining a universal circuit as a sum of the universal circuits for each homogeneous degree:

$$U(\overline{x}, \overline{w}) = \sum_{i=0}^{d} U_i(\overline{x}, \overline{w}),$$

where $U_i(\overline{x}, \overline{w})$ is the universal circuit for $\mathcal{C}^{\text{hom}}_{t,d}$.

In our formalisation we are going to express that a circuit computes a polynomial in $\mathbb{F}[\overline{x}]$ by stating that the coefficients vector of a universal circuit described by edge variables $\overline{w}$ equals some fixed vector in $\mathbb{F}^N$, with N being the total number of possible $\overline{x}$-monomials in a degree d polynomial with n variables. Hence, we represent a circuit of size s using s variables $\overline{w}$ for the s edges in the circuit $U(\overline{x}, \overline{w})$. Note that $U(\overline{x}, \overline{w})$ is a circuit in both $\overline{x}, \overline{w}$, *while our formalisation will* not *use at*

the end the $\overline{x}$-variables: each $\overline{x}$-monomial is a polynomial in the $\overline{w}$-variables only. In other words, the coefficient vector of the polynomial in the $\overline{x}$ variables computed by $U(\overline{x},\overline{w})$ is a vector of N polynomials in the $\overline{w}$ variables. We need to show how to compute the coefficient of an $\overline{x}$-monomial M as a polynomial in the edge variables $\overline{w}$. Such a polynomial is denoted $\text{Coeff}_M(U(\overline{x},\overline{w}))$.

Let $f(\overline{x},\overline{w}) \in \mathbb{F}[\overline{x},\overline{w}]$ be a polynomial, and let $M = \prod_{i\in I} x_i^{\alpha_i} \cdot \prod_{j\in J} w_j^{\beta_j}$ be a monomial in $f(\overline{x},\overline{w})$. We call $\sum_{i\in I}\alpha_i$ the *$\overline{x}$-degree* of M.

Definition 3.4 ($\text{Coeff}_M(\cdot)$)**.** *Let $f(\overline{x},\overline{w})$ be a polynomial in $\mathbb{F}[\overline{x},\overline{w}]$ in the disjoint sets of variables $\overline{x},\overline{w}$. Assume that the $\overline{x}$-degree of every polynomial in $f(\overline{x},\overline{w})$ is precisely d (that is, $f(\overline{x},\overline{w})$ is homogeneous with respect to the $\overline{x}$ variables). Let M be an $\overline{x}$-monomial of degree d. Then, $\text{Coeff}_M(f(\overline{x},\overline{w}))$ is the polynomial coefficient in $\mathbb{F}[\overline{w}]$ of M.*

Note that by Definition 3.4 we have $f(\overline{x},\overline{w}) = \sum_{M_i} \text{Coeff}_{M_i}(f(\overline{x},\overline{w})) \cdot g_{M_i}(\overline{w})$, where the $M_i's$ are all the degree d $\overline{x}$-monomials and where $g_{M_i}(\overline{w})$ is the polynomial coefficient in $\overline{w}$ of M_i.

Proposition 3.5 (Computation of coefficients)**.** *Let $f(\overline{x},\overline{w}) \in \mathbb{F}[\overline{x},\overline{w}]$ and M be as in Definition 3.4, and assume that there is an algebraic circuit computing $f(\overline{x},\overline{w})$ of size s. Then there is a circuit of size $O(6^d \cdot s)$ computing $\text{Coeff}_M(f(\overline{x},\overline{w}))$.*

Proof: For each variable x_i in M we construct a circuit that computes the polynomial $x_i \cdot g + h = f$, with h having no occurrence of x_i. Each such circuit construction increases the size by a constant factor of 6 according to the claim below. Hence, after d iteration of this construction we get $O(6^d \cdot s)$ size. More formally we proceed as follows (recall that $\widehat{C}$ is the polynomial computed by the circuit C and that $|C|$ is the size of C; we denote by $g(\overline{x}) \restriction_{x_i=0}$ the polynomial $g(\overline{x})$ where x_i is assigned 0).

Claim 3.6. *Let $C(\overline{x})$ be a circuit in the $\overline{x}$ variables over the field $\mathbb{F}$. Then, for every variable x_i there is a circuit of size $6|C|$ that computes the polynomial $g(\overline{x})$, such that $\widehat{C}(\overline{x}) = x_i \cdot g(\overline{x}) + g(\overline{x}) \restriction_{x_i=0}$.*

Proof of claim: The proof is a standard induction on the circuit size. Denote by p the polynomial computed by C and for every gate v in C denote by p_v the polynomial computed at gate v. For every gate v in C we add at most 6 new gates. The gate v itself is duplicated twice so that the first duplicate computes $\partial_{x_i}(p_v)$, which is the partial derivative of p_v by the variable x_i, and the second duplicate computes $p_v \restriction_{x_i=0}$, hence it is a polynomial that does not use the variable x_i.

Base case:
Case 1: $C(\overline{x}) = x_i$. Then, $\partial_{x_i}(p) := 1$ and $p \restriction_{x_i=0} := 0$.
Case 2: $C(\overline{x}) = x_j$, for $j \neq i$. Then, $\partial_{x_i}(p) := 0$ and $p \restriction_{x_i=0} := x_j$.
Case 3: $C(\overline{x}) = \alpha$, for $\alpha \in \mathbb{F}$. Then, $\partial_{x_i}(p) := 0$ and $p \restriction_{x_i=0} := \alpha$.

Induction step:
Case 1: $C(\overline{x}) = w + u$. Then, $\partial_{x_i}(p) := \partial_{x_i}(p_w) + \partial_{x_i}(p_u)$.
Case 2: $C(\overline{x}) = w \cdot u$. Then,

$$\partial_{x_i}(p) := \partial_{x_i}(p_w) \cdot x_i \cdot \partial_{x_i}(p_u) + \partial_{x_i}(p_w) \cdot (p_u \restriction_{x_i=0}) + \partial_{x_i}(p_u) \cdot (p_w \restriction_{x_i=0}), \text{ and}$$
$$p \restriction_{x_i=0} := (p_u \restriction_{x_i=0}) \cdot (p_w \restriction_{x_i=0}).$$

Note that the construction results in a circuit (not a formula), that is, in the induction step above we *re-use* gates that where already used if necessary (in correspondence with the wiring and re-use of gates in the original circuit C). This brings us to a circuit of size at most $6|C|$. ■Claim □

Remark 3.7. *It will be important for us that there are* small *universal circuits, namely that the size of $U(\overline{x},\overline{w})$ is small, since our conjectural hard candidates will use a circuit computing $U(\overline{x},\overline{w})$. Our hard candidates include a computation of the coefficient of each $\overline{x}$-monomial in $U(\overline{x},\overline{w}) \in \mathbb{F}[\overline{x},\overline{w}]$ as a polynomial in $\overline{w}$, and by Prop. 3.5 the size of this computation depends on the size of $U(\overline{x},\overline{w})$ (where an* $\overline{x}$-monomial *is a product of $\overline{x}$ variables, namely a monomial in the $\overline{x}$ variables).*

More precisely, we will have

- *$|\overline{x}| = n^k$, that is, the number of algebraic variables in the polynomials we talk about in our formalisation is n^k for some constant k. These will serve as the variables in the formulas encoding IPS lower bounds.*
- *$|\overline{w}| = n^c$ is the number of "edge" variable, where n^c is the size of circuits we wish to encode.*

The size of our hard candidates will be which is exponentially smaller than 2^{n^c} whenever $c > k$. *Thus, we distinguish between these two types of exponential functions:* good exponentials $2^{O(n^k)}$ *and* bad exponentials $2^{O(n^c)}$. *Since the bad exponential is exponentially bigger than the good one, for us a good exponential will be considered still a* feasible *size.*

3.1.3 Formalising VNP≠VP

The class VP is expressed using a universal circuit while the class VNP is expressed explicitly as the coefficient vector of the permanent polynomial, where permanent is known to be complete for VNP.

Let $\mathrm{perm}(\overline{x})$ be the permanent polynomial on the variables $\overline{x}$. We are encode the negation of $\mathsf{VP} \neq \mathsf{VNP}$ (since we work with the refutation system IPS we prove statements by refuting their negations):

Definition 3.5 (Formalisation of $\mathsf{VP} = \mathsf{VNP}$)**.** *The formalisation of $\mathsf{VNP} = \mathsf{VP}(n,t)$ denoted "VNP=VP"(n,t), expressing that there is a universal circuit for circuits with t edges that computes (for some fixed assignment to the t edge labels $\overline{w}$) the permanent polynomial of dimension n (with $\overline{x}$ being the n^2 variables of the permanent), is the following set of polynomial equations:*

$$\mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w})) = b_i \tag{8}$$

where $\overline{b} = \mathrm{coeffs}(\mathrm{perm}(\overline{x})) \in \mathbb{F}^N$, is the coefficient vector of the permanent polynomial of dimension n, i ranges over $i \in [N]$ so that $\{M_i\}_{i=1}^N$ are the set of all possible $\overline{x}$-monomials of degree at most n, and $N = \sum_{d=0}^{n} \binom{n^2+d-1}{d}$ is the number of monomials of total degree at most n over n^2 variables.

Remark 3.8.

1. *Note that "VNP=VP" does not contain the $\overline{x}$ variables, only the $\overline{w}$ variables. (Since we defined universal circuits as a sum of universal circuits each for a different degree* homogeneous *circuits, each monomial in $\overline{x}$ will be computed as a polynomial in the $\overline{w}$ variables.)*
2. *Note that assuming $\mathsf{VNP} \neq \mathsf{VP}$, "VNP=VP" is indeed unsatisfiable set of polynomial equations over $\mathbb{F}_q$. Any assignment that satisfies "VNP=VP" for the $\overline{w}$ variables means that there is a t-size circuit (induced by the edge variables $\overline{w}$ assignment) that computes the permanent polynomial (in the $\overline{x}$ variables).*

The size of "VNP=VP"(n,t) is $O(t^c \cdot N)$ with t the number of variables in XXX and c a constant independent of t, and $N = \binom{n^c+d+1}{d}$ [inaccurate] .

3.1.4 Formalising IPS Refutations

To express an IPS lower bound as a set of polynomial equations we will formalise the negation of this statement, namely the existence of a small IPS refutation for a specific CNF formula.

Definition 3.6 (IPS refutation predicate $\mathrm{IPS}_{\mathrm{ref}}(t, \overline{\mathcal{F}})$)**.** *Let $\overline{\mathcal{F}}$ be a CNF formula with m clauses and (for simplicity, since we deal with matrix inputs) n^2 variables $\overline{x}$ written as a set of polynomial equations according to Definition 2.3. Let $U(\overline{x}, \overline{y}, \overline{w})$ be a universal circuit in the $\overline{x}$ variables and the m placeholder variables $\overline{y}$ (*both *of these sets of variables are the algebraic variables in a polynomial computed by the universal circuit when assigned field values to the edge labels $\overline{w}$), and the t edge label variables $\overline{w}$. We formalise the existence of a size t circuit that computes the* IPS *refutation of $\overline{\mathcal{F}}$ as follows:*

$$\mathrm{Coeff}_{M_i \restriction \overline{y}=\mathbf{0}}(U(\overline{x}, \mathbf{0}, \overline{w})) = 0\,, \tag{9}$$

$$\mathrm{Coeff}_{M_i \restriction \overline{y}=\overline{\mathcal{F}}}(U(\overline{x}, \overline{\mathcal{F}}, \overline{w})) = \begin{cases} 1, & M_i \restriction \overline{y} = \overline{\mathcal{F}} \text{ is the constant 1 monomial;} \\ 0, & \textit{otherwise}\,, \end{cases} \tag{10}$$

where i ranges over $i \in [N]$ so that $\{M_i\}_{i=1}^{N}$ are the set of all possible $\overline{x}$,$\overline{y}$-monomials (monomials in both $\overline{x}$, $\overline{y}$ variables) of degree at most n, and $N = \sum_{d=0}^{n} \binom{n^2+m+d-1}{d}$ is the number of monomials of total degree at most n over $n^2 + m$ variables, $\mathbf{0}$ is the all zero vector of length m and $M_i \restriction \overline{y} = \overline{a}$ denotes the monomial M_i in which the $\overline{y}$-variables are substituted according to the assignment $\overline{a}$ of polynomials in the $\overline{x}$-variables to the $\overline{y}$-variables.

Encoding an IPS lower bound is simply providing an ostensible *unsatisfiable* set of polynomials that express the existence of size at most t IPS refutation of a CNF $\overline{\mathcal{F}}$. Thus refuting this set of polynomials amounts to proving the IPS lower bound.

3.2 Characterizing $\mathsf{VNP} \neq \mathsf{VP}$ as a Proof Complexity Lower Bound

We are now ready to prove our main theorem. We show that IPS cannot efficiently prove that the algebraic circuit class separation $\mathsf{VP} \neq \mathsf{VNP}$ is hard to prove in IPS. Note that this result is *unconditional*: IPS unconditionally does not have polynomial-size refutations of the statement asserting the existence of short IPS refutations for $\mathsf{VP} = \mathsf{VNP}$. On the other hand, if $\mathsf{VP} \neq \mathsf{VNP}$ is in fact easy to prove in IPS or if $\mathsf{VP} = \mathsf{VNP}$, then the result is less interesting, since then the result stems simply from soundness of IPS. The more interesting scenario is under the reasonable assumption that indeed algebraic circuit class separations are hard to establish in IPS, and in this case we show that this fact would not have efficient proofs in IPS. We start by describing the main idea behind the argument.

The gist of the argument is showing how from a short IPS proof of an IPS lower bound on an unsatisfiable CNF formulas one gets a short IPS proof that $\mathsf{VP} \neq \mathsf{VNP}$—this can be considered a formalisation in IPS of the Grochow and Pitassi argument [GP18]. Since we work with a refutation system, we will show equivalently that if IPS efficiently refutes the existence of small IPS refutations of F, we get an efficient IPS refutation of $\mathsf{VP} = \mathsf{VNP}$. Hence, it suffices to start from the CNF "$\mathsf{VP} = \mathsf{VNP}$" and reach a contradiction (in IPS). The idea now is quite simple: if we have the CNF "$\mathsf{VP} = \mathsf{VNP}$"$(t, n)$, meaning that $\mathrm{perm}(\overline{x})$ of dimension n is computable by size t circuits, then we know in particular, by the completeness of the permanent for VNP and the fact that every unsatisfiable CNF has an IPS refutation computable in VNP (Thm. 2.1), that there is a "projection", namely, an assignment $\overline{a}$ of variables and field elements to $\overline{x}$, such that $\mathrm{perm}(\overline{x} \restriction \overline{a})$ is the IPS refutation of the CNF "$\mathsf{VP} = \mathsf{VNP}$" of some lower dimension. Since the class of size t

circuits is closed under assignments we conclude that $\mathrm{perm}(\overline{x} \restriction \overline{a})$ has a size t circuit, that is, there is a small IPS refutation of "$\mathsf{VP} = \mathsf{VNP}$" of some lower dimension. But we assumed that IPS can (efficiently) *refute* the existence of such small IPS refutations for "$\mathsf{VP} = \mathsf{VNP}$", and we finish by soundness of IPS.

We are going to use the following lemma that will allow us to express a proof complexity lower bound on aa seemingly stronger statement (and hence, stating such a lower bound *weakens* the lower bound statement). In particular we will extend the statement of $\mathsf{VP} = \mathsf{VNP}$ with additional extension axioms for certain new circuits $C_i(\overline{x})$. These extension axioms include the equations for gates in $C_i(\overline{x})$ written as SLPs and the algebraic extension axioms for each gate g in $C_i(\overline{x})$. On the other hand, these extension axioms *do not* express that any of the $C_i(\overline{x})$ equals any value. In particular this means that these additional extension axioms do not add any information to the statement (that is, they do not make the statement "less satisfiable"). In return we will be able to state that such additional extension axioms do not make a lower bound against the statement weaker, in the following sense:

Lemma 3.9 (Disjoint extension axioms do not make refutations easier, provably in IPS)**.** *Let $\overline{F}$ be a set of circuit equations written as SLPs of total size s, $\overline{E}$ be a set of circuits (*not *circuit equations) each written as a SLP with extension variables (not appearing in $\overline{F}$) for every node gate g in the circuits in $\overline{E}$, together with the algebraic extension axioms for each such gate g. Then,*

$$\mathsf{cnf}(\mathrm{IPS}_{ref}(t, \mathsf{ecnf}(\overline{F} \cup \overline{E}))) \vdash^{s^{O(1)}}_{\mathrm{IPS}} \mathsf{cnf}(\mathrm{IPS}_{ref}(t, \mathsf{ecnf}(\overline{F}))) \,. \tag{11}$$

Proof: The argument is a straightforward substitution to the extension axioms in $\overline{E}$ by constants that satisfy the extension axioms (every assignment to the original variables in $\overline{F}$ can be extended in such way because the variables in $\overline{E}$ do not occur in $\overline{F}$). We omit the details of this formalisation. □

Lemma 3.10 (Algebraic extension axioms do not make refutations easier, provably in IPS)**.** *For every set of circuit equations $\overline{F}$ written as SLPs of total size s the following holds:*

$$\mathsf{cnf}(\mathrm{IPS}_{ref}(t, \mathsf{ecnf}(\overline{F}))) \vdash^{s^{O(1)}}_{\mathrm{IPS}} \mathsf{cnf}(\mathrm{IPS}_{ref}(t, \mathsf{cnf}(\overline{F}))) \,. \tag{12}$$

Proof: As before, the argument is a straightforward substitution to the algebraic extension axioms (Definition 3.1) in $\mathsf{ecnf}(\overline{F})$ by constants that satisfy the extension axioms (every assignment to the original variables in $\mathsf{cnf}(\overline{F})$ can be extended in such way because the unary-bits variables in the extension axioms in $\mathsf{ecnf}(\overline{F})$ do not occur in $\mathsf{cnf}(\overline{F})$). Note that unlike the case in Lemma 3.9, here we do *not* add equations between different extension variables for different gates in the circuits in $\overline{\mathcal{F}}$ (this may insert new constraints between already existing extension variables and hence may cause the resulting system of equations to become easier to refute). We omit the details of this formalisation. □

Theorem 3.11 (main)**.** *Let $\varphi_{t,m}$ denote the extended CNF encoding of the circuit equation "VNP=VP"(t, m) over the field $\mathbb{F}_q$ expressing that there are t size circuits for the permanent over $\mathbb{F}_q$ of dimension m. Let $\Phi_{m,s,t}$ denote the CNF formula $\mathsf{cnf}(\mathrm{IPS}_{ref}(s, \varphi_{t,m}))$ expressing that* IPS *refutes $\varphi_{t,m}$ in size s over $\mathbb{F}_q$. Then, $\mathsf{VP} \neq \mathsf{VNP}$ over $\mathbb{F}_q$ iff the CNF family $\{\Phi_{m,s,t}\}$ does not have short* IPS *refutations, in the following sense: for any sufficiently big constant c_1 infinitely often for $m \in \mathbb{N}$, there exist constants c_0, c_2 such that if $t < m^{c_0}$ and $s < |\varphi_{t,m}|^{c_1}$, then $\mathsf{cnf}(\mathrm{IPS}_{ref}(s, \varphi_{t,m}))$ has no IPS refutation of size at most $|\mathsf{cnf}(\mathrm{IPS}_{ref}(s, \varphi_{t,m}))|^{c_2}$.*

Proof: ($\Leftarrow$) This is the easier direction. Assume that there exist sufficiently big constants c_0, c_1, c_2 such that for infinitely many $m \in \mathbb{N}$ the family of CNF formulas $\{\mathsf{cnf}(\mathrm{IPS}_{\mathrm{ref}}(s, \varphi_{t,m}))\}_{n=1}^{\infty}$ does not have polynomial-size IPS refutations where $t < m^{c_0}$ (that is, the purported circuit for the permanent of dimension m is polynomially bounded in m) and $s < |\varphi_{t,m}|^{c_1}$ (that is, the purported size of an IPS refutation of the formula $\varphi_{t,m}$ is polynomially bounded in the size of the formula).

If $\{\mathsf{cnf}(\mathrm{IPS}_{\mathrm{ref}}(s, \varphi_{t,m}))\}_{n=1}^{\infty}$ is an *unsatisfiable* CNF family for infinitely many $m \in \mathbb{N}$, then by Thm. 2.1 [GP18] $\mathsf{VNP} \neq \mathsf{VP}$ over $\mathbb{F}_q$ (note that the result of [GP18] holds also for lower bounds on IPS *without* the boolean axioms).

Otherwise, $\{\mathsf{cnf}(\mathrm{IPS}_{\mathrm{ref}}(s, \varphi_{t,m}))\}_{n=1}^{\infty}$ is a satisfiable CNF family for infinitely many $m \in \mathbb{N}$. Thus, by Prop. 3.2 the set of polynomial equations $\mathrm{IPS}_{\mathrm{ref}}(s, \varphi_{t,m})$ is satisfiable over $\mathbb{F}_q$ for infinitely many $m \in \mathbb{N}$ and hence there is an IPS refutation of $\varphi_{t,m}$ for infinitely many $m \in \mathbb{N}$ and by soundness of IPS we get that infinitely often there is no size at most m^{c_0} for the permanent of dimension m, and we are done.

($\Rightarrow$) We work over $\mathbb{F}_q$. First, consider $\varphi_{t,m} = \mathsf{cnf}(\text{“}\mathsf{VNP}{=}\mathsf{VP}\text{”}(t, m))$, and let $\varphi^{\star}_{t,m}$ be $\mathsf{ecnf}(\text{“}\mathsf{VNP}{=}\mathsf{VP}\text{”}(t, m))$ together with additional extension axioms $\overline{E}$ for some new set of circuits written as SLPs and a new set of algebraic extension axioms for all the gates in these new circuits that we will specify later (all the variables in these new equations will be disjoint from the original variables in $\varphi_{t,m}$).

We assume that:

Assumption: $\mathsf{VP} \neq \mathsf{VNP}$ over $\mathbb{F}_q$. Namely, there is no constant c such that for all but constant many $n \in \mathbb{N}$, the permanent polynomial $\mathrm{perm}(\overline{x})$ of dimension n has an algebraic circuit of size at most n^c, over $\mathbb{F}_q$.

And we will prove that:

Conclusion: Let c_1 be a sufficiently big constant. Infinitely often (i.o. for short) for $m \in \mathbb{N}$, there exist constants c_0, c_2 such that if $t < m^{c_0}$ and $s < |\varphi^{\star}_{t,m}|^{c_1}$, then

$$\mathsf{cnf}(\mathrm{IPS}_{\mathrm{ref}}(s, \varphi^{\star}_{t,m})) \tag{13}$$

has no IPS refutations of size at most $|\mathsf{cnf}(\mathrm{IPS}_{\mathrm{ref}}(s, \varphi^{\star}_{t,m}))|^{c_2}$.

By Lemma 3.9 and Lemma 3.10 as long as the set of new axioms $\overline{E}$ has total algebraic circuit-size polynomial in $|\varphi_{t,m}|$, if Conclusion above holds then also the same conclusion holds when the CNF

$$\mathsf{cnf}(\mathrm{IPS}_{\mathrm{ref}}(s, \varphi_{t,m})) \tag{14}$$

replaces eq. 13 in Conclusion. This will then conclude the theorem.

Under Assumption above, let c_0, c_1, c_2 be constants such that: for infinitely many $m \in \mathbb{N}$, $\mathrm{perm}(\overline{x})$ of dimension m has no circuit of size m^{c_0} (since $\mathsf{VP} \neq \mathsf{VNP}$ this is true almost everywhere), and assume that the following holds:

$$\mathsf{cnf}(\mathrm{IPS}_{\mathrm{ref}}(s, \varphi^{\star}_{t,m})) \vdash^{\lambda^{c_2}}_{\mathrm{IPS}} 1 = 0\,, \tag{15}$$

for all s, t, m such that $t < m^{c_0}$ and $s < |\varphi_{t,m}|^{c_1}$ and $\lambda = |\mathsf{cnf}(\mathrm{IPS}_{\mathrm{ref}}(s, \varphi^{\star}_{t,m}))|$. Because $\mathrm{IPS}^{\mathrm{nb}}$ is a *subsystem* (i.e., it is simulated by) IPS, we have also (because a lower bound on a subsystem follows from a lower bound on its super-system):

$$\mathsf{cnf}(\mathrm{IPS}^{\mathrm{nb}}{}_{\mathrm{ref}}(s, \varphi^{\star}_{t,m})) \vdash^{\lambda^{c_2}}_{\mathrm{IPS}} 1 = 0\,. \tag{16}$$

Using eq. 16, we are going to construct the following short IPS refutation:

$$\varphi^{\star}_{s,r} \vdash^{\gamma^{c''}}_{\mathrm{IPS}} 1 = 0\,, \tag{17}$$

with s as in eq. 16, r being the number of variables in $\varphi_{t,m}$ plus the number of equations in $\varphi_{t,m}$ (this corresponds to the number of variables in an IPS refutation of $\varphi_{t,m}$ which includes the number of algebraic variables in $\varphi_{t,m}$ and the placeholder variables for each equation in $\varphi_{t,m}$), $\gamma = |\varphi^{\star}_{s,r}|$, and c'' is some constant. Putting c_0 such that

$$s < r^{c_0}$$

and assuming we choose c_0, c_1 small enough such that

$$\gamma^{c''} = |\varphi^{\star}_{s,r}|^{c''} < |\varphi^{\star}_{s,r}|^{c_1},$$

eq. 17 will conclude the theorem because this shows that under the assumption that eq. 15 holds, eq. 17, which is expressed as a CNF formula by $\mathsf{cnf}(\mathrm{IPS}_{\mathrm{ref}}(\gamma^{c''}, \varphi^{\star}_{s,r}))$, cannot have an IPS *refutation* (by soundness of IPS) which is sufficient to satisfy the conditions in Conclusion. On the other hand, if eq. 15 does not hold, we similarly finish the proof of the theorem.

We thus start with $\varphi^{\star}_{s,r}$ and recall that it is the extended CNF encoding (with some small size of added extension axioms that we will specify later in the proof of Lemma 3.12) of the equations "VNP=VP"(s, r) expressing that there exists a circuit of size s for the permanent of dimension r over $\mathbb{F}_q$.

The following is the crux of the argument: since the permanent polynomial $\mathrm{perm}(\overline{x})$ of dimension r is complete for VNP with $\lfloor r/6 \rfloor$ variables [Val79b], and since every unsatisfiable CNF has an IPS refutation whose certificate is computable in VNP by [GP18], and similarly every *extended* CNF formula is computable in VNP by Corollary 3.3, there exists an assignment $\overline{a}$ of field $\mathbb{F}_q$ elements and $\overline{x}$ variables to $\overline{x}$, with $\overline{x}$ having r number of variables, such that $\mathrm{perm}(\overline{x} \restriction \overline{a})$ is the IPS refutation of the extended CNF $\varphi_{t,m}$ (here, it will be sufficient to consider a refutation of $\varphi_{t,m}$ instead of $\varphi^{\star}_{t,m}$). Formally we have the following:

Lemma 3.12 (main Grochow-Pitassi simulation lemma)**.** *There is an* $\mathrm{IPS}^{\mathrm{nb}}$ *derivation from* $\varphi^{\star}_{s,r}$ *of* $\mathrm{IPS}^{\mathrm{nb}}{}_{\mathrm{ref}}(s, \varphi_{t,m})$*, with size at most* $\gamma^{c'}$*, for some constant* c'*.*

Proof of Lemma 3.12. Consider $\varphi^{\star}_{s,r}$ which is defined to be the extended CNF of "VNP=VP"(s, r) (written as a set of corresponding SLP equations over $\mathbb{F}_q$) together with the following additional extension axioms (that are needed to invoke Prop. 3.1 in what follows):

for every circuit equation in "VNP=VP"(t, m) (written as equations corresponding to the SLPs of this statement) we have the algebraic extension axioms (Definition 3.1) for every gate in a circuit equation, together with the corresponding SLPs equations between extension variables for the gates in the circuits.

First use Prop. 3.1 to derive from the extended CNF $\varphi^{\star}_{s,r}$ the corresponding circuit equations "VNP=VP"(s, r). The latter is formulated (Definition 3.5) as a set of circuit equations expressing that the coefficients of the monomials of a universal circuit of size s are the coefficients of $\mathrm{perm}(\overline{x})$ of dimension r.

Consider $\overline{a}$ as an assignment of variables and field elements to the $\overline{x}$ variables that results in $\mathrm{perm}(\overline{x} \restriction \overline{a})$ being a (correct) $\mathrm{IPS}^{\mathrm{nb}}$ refutation of $\varphi_{t,m}$ (such an assignment $\overline{a}$ exists as discussed above). It is left to efficiently derive in $\mathrm{IPS}^{\mathrm{nb}}$ from "VNP=VP"(s, r) the equations for

$\mathrm{IPS}^{\mathrm{nb}}{}_{\mathrm{ref}}(s, \varphi_{t,m})$. The latter is again formulated (Definition 3.6) as a set of circuit equations expressing that the coefficients of the monomials of a universal circuit of size s are precisely the coefficients of a polynomial that constitutes an IPS refutation of the extended CNF $\varphi_{t,m}$. To derive in $\mathrm{IPS}^{\mathrm{nb}}$ from "VNP=VP"(s, r) the equations for $\mathrm{IPS}^{\mathrm{nb}}{}_{\mathrm{ref}}(s, \varphi_{t,m})$ we use the fact that IPS linearly proves every polynomial identity by Fact A.1, and the following claim on the structure of monomials as constructed in Prop. 3.5:

Claim 3.13 (Derivation of coefficients of monomials after assignment)**.** *Let $U(\overline{x}, \overline{y}, \overline{w})$ be a universal circuit in the algebraic variables $\overline{x}$, $\overline{y}$ and the edge variables $\overline{w}$.*[5] *Let M_i for $1 \leq i \leq N$ denote the monomials in both the $\overline{x}$ and $\overline{y}$ variables, and let $b_{M_i} \in \mathbb{F}_q$ denote a possible coefficient of the monomial M_i, and $U((\overline{x}, \overline{y}) \restriction \overline{a}, \overline{w})$ denote $U(\overline{x}, \overline{y}, \overline{w})$ in which we substitute the $\overline{x}$ and $\overline{y}$ variables by an assignment of $\overline{x}$,$\overline{y}$-polynomials $\overline{a} : (\overline{x} \cup \overline{y}) \to \mathbb{F}_q[\overline{x}, \overline{y}]$. Then for every M_i there exists a circuit $C_i(\overline{z})$ of size $N^{O(1)}$ with $\overline{z}$ being N variables, such that the following is a* polynomial identity*:*

$$\mathrm{Coeff}_{M_i}(U((\overline{x}, \overline{y}) \restriction \overline{a}, \overline{w})) = C_i\left(\mathrm{Coeff}_{M_1}(U(\overline{x}, \overline{y}, \overline{w})), \ldots, \mathrm{Coeff}_{M_N}(U(\overline{x}, \overline{y}, \overline{w}))\right). \tag{18}$$

This claim simply says that knowing the values of all the coefficients of monomials M_i's in U we can plug these values in the polynomial-size circuit C_i to get the coefficient of M_i in a substitution instance of U. The proof of the claim proceeds by inspecting the structure of the circuit using the structure of the circuit Coeff_{M_i} as devised in Prop. 3.5. We omit the details.

The claim allows us to complete the proof of the lemma: starting from the equations that give the values of all the coefficients of "VNP=VP"(s, r) we derive the equations that express that the coefficients of $\mathrm{perm}((\overline{x}, \overline{y}) \restriction \overline{a})$ are precisely those of a polynomial that constitutes an $\mathrm{IPS}^{\mathrm{nb}}$ refutation of the extended CNF $\varphi_{t,m}$ (here we switched to using both $\overline{x}$ and $\overline{y}$ in $\mathrm{perm}(\overline{x}, \overline{y})$ for clarity of exposition). Since we know that $\mathrm{perm}((\overline{x}, \overline{y}) \restriction \overline{a})$ is a correct $\mathrm{IPS}^{\mathrm{nb}}$ refutation of $\varphi_{t,m}$, the equations for all coefficients will satisfy the conditions of Definition 3.6 (note that we will use three different assignments: first the assignment $\overline{a}$ that will make $\mathrm{perm}((\overline{x}, \overline{y}) \restriction \overline{a})$ an IPS refutation in the variables $\overline{x}$,$\overline{y}$ (where $\overline{y}$ are the placeholder variables), and then we need to compose this assignment with either the assignment of all zeros to $\overline{y}$, or the assignment of the clauses of $\varphi_{t,m}$ to the $\overline{y}$-variables).

The size bound on the IPS proofs follows from the construction. □

We are now in a position to complete the proof of the theorem:

$$\varphi^{\star}_{s,r} \vdash^{\gamma^{c'}}_{\mathrm{IPS}} \mathrm{IPS}^{\mathrm{nb}}{}_{\mathrm{ref}}(s, \varphi_{t,m}) \qquad \text{(by Lemma 3.12)}, \tag{19}$$

and by using again Prop. 3.1 (in the other direction to that in Lemma 3.12) and the fact that we have extension axioms for all the subcircuits in $\mathrm{IPS}^{\mathrm{nb}}{}_{\mathrm{ref}}(s, \varphi_{t,m})$ by definition of $\varphi^{\star}_{s,r}$ (this is where we use the fact that we started with $\varphi^{\star}_{s,r}$ and not merely $\varphi_{s,r}$), to get

$$\vdash^{\gamma^{\ell}}_{\mathrm{IPS}} \ \mathsf{ecnf}(\mathrm{IPS}^{\mathrm{nb}}{}_{\mathrm{ref}}(s, \varphi_{t,m})), \tag{20}$$

for some constant ℓ.

By eq. 16 (and the fact that $\mathsf{cnf}(C = 0) \subset \mathsf{ecnf}(C = 0)$, for every (set of) circuits C, as well as the fact that by Lemma 3.9 we do not need to use the additional extension axioms that are added to $\varphi^{\star}_{t,m}$ on top of $\varphi_{t,m}$) we get an IPS refutation from eq. 20 of size at most $\gamma^{\ell'}$ for some constant ℓ' that we can assume is smaller than c''. This concludes the theorem. □

[5] We define universal circuits to have algebraic variables not only $\overline{x}$ variables but also $\overline{y}$ variables. This is done for the sake of clarity, since we need to define universal circuits that compute IPS refutations in both the $\overline{x}$ variables and the $\overline{y}$ placeholder variables.

We also have the following immediate corollary:

Corollary 3.14. *Thm. 3.11 holds for any proof system that simulates IPS.*

4 Candidate Hard Formulas for Every Propositional Proof System

4.1 Iterated Lower Bound Formulas

We use *pps* to refer to a propositional proof system.

Definition 4.1. *Given pps R, propositional formula ϕ and size function $s : \mathbb{N} \to \mathbb{N}$, $\mathsf{lb}_R(\phi, s)$ is a propositional DNF formula of size poly$(|\phi| + s(|\phi|))$ over poly$(|\phi| + s(|\phi|))$ variables expressing that there is no R-proof of ϕ having size $s(|\phi|)$.*

More explicitly, the formula $\mathsf{lb}_R(\phi, s)$ contains s variables $y_1, \ldots, y_s$ encoding R-proofs of length s and poly$(|\phi| + s)$ auxiliary variables encoding the computation of the relation R, to verify that $y_1, \ldots, y_s$ does not constitute an R-proof of ϕ.

Definition 4.2 (Reasonably Strong Proof System)**.** *We say that a pps R is reasonably strong if it satisfies the following conditions:*

1. *R p-simulates* Res *(Resolution).*
2. *There are polynomial-size R-proofs of $\neg\phi \vee \phi(\overline{a})$ for every DNF ϕ and assignment $\overline{a}$ to the variables of ϕ.*
3. *R has poly-size proofs of its own reflection principle $\mathsf{lb}_R(\phi, s) \vee \phi$.*
4. *R is closed under modus ponents, i.e., if there are polynomial-size R-proofs of $\tau \vee \neg\phi$ and of $\phi \vee \psi$ for DNFs τ, ϕ, ψ, then there are polynomial-size R-proofs of $\tau \vee \psi$.*

The conditions above are satisfied for any standard strong enough pps, eg., Frege and Extended Frege.

We now formally define iterated lower bound formulas relative to a pps R.

Definition 4.3 (Iterated Lower Bound Formulas)**.** *Given pps R, propositional formula ϕ and size function $s : \mathbb{N} \to \mathbb{N}$, the sequence $\{\mathsf{lb}^k_R(\phi, s)\}, k = 0 \ldots \infty$ is defined inductively as follows:*

$$\mathsf{lb}^0_R(\phi, s) = \phi$$

$$\mathsf{lb}^{k+1}_R(\phi, s) = \mathsf{lb}_R(\mathsf{lb}^k_R(\phi, s), s)$$

Lemma 4.1. *Let R be any reasonably strong pps, and let τ be a non-tautology. Then there are R-proofs of $\mathsf{lb}_R(\tau, s)$ of size poly$(|\mathsf{lb}_R(\tau, s)|)$.*

Proof: Since τ is a non-tautology, there is an assignment $\overline{a}$ to the variables of τ such that $\tau(\overline{a})$ evaluates to false. Since R is reasonably strong, it simulates Res and hence can prove $\neq \tau(\overline{a})$ by substituting the assignment $\overline{a}$ into τ. Since R is reasonably strong, R can prove $\neg\phi \vee \phi(\overline{a})$ efficiently. It follows from the closure under modus ponens that R can prove $\neg\phi$ efficiently. Since R is reasonably strong, R can prove its own reflection principle efficiently. It follows from the closure under modus ponens that R can prove $\mathsf{lb}_R(\phi, s)$ efficiently. □

We show a dichotomy for iterated lower bound formulas: either they are all hard (and hence all tautologies), or they divide up according to the parity of k, with one parity corresponding to non-tautologies and the other parity corresponding to tautologies with short proofs.

Theorem 4.2 (Dichotomy for Iterated Lower Bound Formulas)**.** *Let R be a reasonably strong pps and $s : \mathbb{N} \to \mathbb{N}$. There is a constant c such that for every s with $s(n) > n^c$ for all $n \in \mathbb{N}$ and for every ϕ that does not have R-proofs of size $s(|\phi|)$, exactly one of the following holds:*

1. *For every integer $k \geq 0$, $\phi_k = \mathsf{lb}_R^k(\phi, s)$ is a tautology that does not have R-proofs of size $s(|\phi_k|)$*
2. *There is an integer $k \geq 0$ such that for every integer $i \geq 0$, ϕ_{k+i} is not a tautology (and hence does not have R-proofs of any size) if i is odd, and ϕ_{k+i} is a tautology with R-proofs of size at most poly$(s(|\phi_{k+i}|))$ if i is even.*

Proof: Suppose that for every integer $k \geq 0$, ϕ_k is a tautology. We show that this implies that ϕ_k is also hard in the sense that it does not have R-proofs of size $s(|\phi_k|)$. Indeed, since ϕ_j is a tautology for every j, it follows that ϕ_{k+1} is a tautology. Since ϕ_{k+1} asserts that ϕ_k does not have R-proofs of size $s(|\phi_k|)$, it follows that ϕ_k is indeed hard as claimed. Thus, in this case, the first item in the statement of the theorem holds.

Otherwise, since $\phi_0 = \phi$ is a tautology with no R-proofs of size $s(|\phi|)$, there is a least positive integer k such that ϕ_k is a tautology but ϕ_{k+1} is a non-tautology. Since ϕ_{k+1} asserts that ϕ_k does not have R-proofs of size $s(|\phi_k|)$, it follows that ϕ_k does indeed have R-proofs of size $s(|\phi_k|)$. We will show by induction in this case that for each integer $i \geq 0$ the following statement $S(i)$ holds: ϕ_{k+2i+1} is not a tautology (and hence does not have R-proofs of any size), and ϕ_{k+2i} is a tautology with R-proofs of size at most $s(|\phi_{k+2i}|)$.

We first establish the base case $S(0)$. When $i = 0$, by assumption on ϕ_k, we have that $\phi_{k+2i} = \phi_k$ is a tautology with R-proofs of size at most $s(|\phi_k|)$. Also, by assumption on k, ϕ_{k+1} is a non-tautology, and since R is sound, it follows that ϕ_{k+1} does not have R-proofs of any size.

For the inductive step, we assume that $S(i)$ has been shown and deduce $S(i+1)$. Since $S(i)$ is true, we have that ϕ_{k+2i} is a tautology with R-proofs of size at most $s(|\phi_{k+2i}|)$ and ϕ_{k+2i+1} is not a tautology (and hence does not have R-proofs of any size). We have that ϕ_{k+2i+2} asserts that ϕ_{k+2i+1} does not have R-proofs of size $s(|\phi_{k+2i+1}|)$, which is tautologous since ϕ_{k+2i+1} does not have R-proofs of any size. In order to show that ϕ_{k+2i+2} has R-proofs of size at most $s(|\phi_{k+2i+2}|)$, we simply apply Lemma 4.1 with $\tau = \phi_{k+2i+1}$, where c is a constant such that $\mathsf{lb}_R(\tau, s)$ has R-proofs of size at most $|\mathsf{lb}_R(\tau, s)|^c$. Since $s(n) > n^c$ for all $n \in \mathbb{N}$, we have that $\mathsf{lb}_R(\tau, s)$ has R-proofs of size at most $s(|\mathsf{lb}_R(\tau, s)|)$. It follows that ϕ_{k+2i+3} is not a tautology, since ϕ_{k+2i+3} asserts that ϕ_{k+2i+2} does not have R-proofs of size at most $s(|\phi_{k+2i+2}|)$, which is false by the previous line, and we are done. □

We call the first item of Thm. 4.2 the *useful* case of the dichotomy, as this is the case that gives us hard tautologies for R.

Conjecture 4.3 (Iterated Lower Bound Conjecture)**.** *Let R be a reasonably strong pps that is not polynomially bounded. Then there is a super-polynomial function $s : \mathbb{N} \to \mathbb{N}$ and a formula ϕ with no R-proofs of size $s(|\phi|)$ such that for all non-negative integers k, $\phi_k = \mathsf{lb}_R^k(\phi, s)$ is a tautology that does not have R-proofs of size $s(|\phi_k|)$.*

Theorem 4.4 (Atserias-Muller [AM19], Garlik [Gar19])**.** *Let $s : \mathbb{N} \to \mathbb{N}$ be any super-polynomial function such that $s(n) = 2^{n^{o(1)}}$. For any formula ϕ, if ϕ is a tautology, then $\tau = \mathsf{lb}_{\mathsf{Res}}(\phi)$ does not have R-proofs of size $s(|\tau|)$.*

It follows by induction that the conjecture holds at least for the relatively weak proof system Resolution.

Corollary 4.5. *Let $s : \mathbb{N} \to \mathbb{N}$ be any super-polynomial function such that $s(n) = 2^{n^{o(1)}}$ and let ϕ be a tautology with no Resolution proofs of size $s(|\phi|)$ (eg., the Pigeonhole Principle). Then for all non-negative integers k, $\phi_k = \mathsf{lb}^k_{\mathsf{Res}}(\phi, s)$ is a tautology that does not have R-proofs of size $s(|\phi_k|)$.*

4.2 Iteration Preserves Hardness for Random Truth Table Formulas

Definition 4.4 (Truth Table Formulas)**.** *Given a Boolean function f_n on n variables and a size parameter t, $\mathsf{ttable}(f, t)$ is a propositional DNF formula of size $N = \tilde{O}(2^n s^3)$ over $\tilde{O}(s)$ variables expressing that f does not have Boolean circuits of size s. The distribution $\mathsf{Randtt}(n, t)$ over $\mathsf{ttable}(f, t)$, where f is chosen uniformly at random from all Boolean functions on n variables, is called the distribution of random truth table formulas.*

Conjecture 4.6 (Rudich's Conjecture [Rud97])**.** *There is a constant ℓ for which there is no sequence of polynomial-size non-deterministic circuits $\{C_m\}$ such that for infinitely many m for which $m = 2^n$ for some non-negative integer n:*

1. *If C_m accepts a string y, then y is the truth table of a Boolean function f_n that does not have Boolean circuits of size n^ℓ.*

2. *C_m accepts at least an inverse polynomial fraction of all inputs.*

Definition 4.5 (Distributional Iterated Lower Bound Formulas)**.** *Let D_N be a distribution on propositional formulas of size N. Given pps R and size function $s : \mathbb{N} \to \mathbb{N}$, the sequence of distributions $\{\mathsf{lb}^k_R(D_N, s)\}, k = 0 \ldots \infty$ is defined inductively as follows:*

$\mathsf{lb}^0_R(D_N, s) = D_N$

$\mathsf{lb}^{k+1}_R(\phi, s)$ *is the distribution on formulas $\mathsf{lb}_R(\phi, s)$ where ϕ is sampled from $\mathsf{lb}_R(\mathsf{lb}^k_R(D_n, s), s)$.*

Theorem 4.7. *If Rudich's Conjecture holds, then there exist a pps R efficiently simulating Extended Frege and a constant $\ell > 0$ such that for every large enough $c > 0$ and every non-negative integer k, $\mathsf{lb}^k_R(D_N, N^c)$ is a tautology with no R-proofs of size $|lb^k_R(D_N, N^c)|^c$ with probability $1-o(1)$ for all large enough N, where $D_N = \mathsf{Randtt}(n, n^\ell$ (for N an appropriate function of n and ℓ as given by Definition 4.4).*

Proof: The proof is by induction on k. Let $\ell > 0$ be the constant in Conjecture 4.6 and let R be a pps and c be a large enough constant to be specified later.

We will establish the case $k = 0$ for *every* pps R, under Rudich's Conjecture. We have that $\mathsf{lb}^k_R(D_N, N^c) = D_N$. Since a random Boolean function on n variables has circuits of size n^ℓ with exponentially small probability, a formula τ sampled according to D_N is a tautology with high probability. Moreover, Rudich's Conjecture implies that τ does not have R-proofs (or indeed proofs in *any* propositional proof system) of size N^c with high probability. To see this, note that R with size bound s defines a non-deterministic algorithm A_R running in time $\mathrm{poly}(s)$ for the problem $\mathsf{MCSP}[n^\ell]$ asking whether a given truth table y of a Boolean function f on n bits has circuits of size n^ℓ: A_R checks if $\mathsf{ttable}(f, n^\ell)$ has R-proofs of size s. A_R only accepts on hard Boolean functions, by the soundness of R, satisfying the first item of Conjecture 4.6. If A_R accepted τ for even a $1/N$ fraction of truth-table tautologies τ, this would contradict the second item of Conjecture 4.6.

We define R to be the pps that is Extended Frege together with axioms stating that $\mathsf{ttable}f_n^{\mathsf{SAT}}, 2n^\ell$ holds, where f_n^{SAT} is the truth table of SAT on n variables. The axioms indeed hold under Rudich's Conjecture, as Rudich's Conjecture implies that NP does not have polynomial-size circuits. It is easy to verify that R is reasonably strong according to Definition 4.2.

Now suppose we have established the assertion for all non-negative integers smaller than k and would like to establish it for k. The inductive strategy builds partly on ideas in [PS19]. We have by the inductive assumption that with probability $1-o(1)$ for ψ sampled from D_N, $\mathsf{lb}_R^{k-1}(\psi, N^c)$ does not have R-proofs of size $|\mathsf{lb}_R^{k-1}(\psi, N^c)|^c$. It follows immediately that with probability $1-o(1)$, for ψ sampled from D_N, $\mathsf{lb}_R^k(\psi, N^c)$ is a tautology.

For the lower bound on $\mathsf{lb}_R^k(D_N, N^c)$, we will treat k differently depending on its parity. If k is even, we will show inductively that if $\psi_k = \mathsf{lb}_R^k(\psi, N^c)$ is a tautology, then so is ψ. Indeed, this is trivially true when $k = 0$, and we show that if ψ_k is a tautology, then so is ψ_{k-2}. Assume contrapositively that ψ_{k-2} is not a tautology. This means that ψ_{k-1} is a tautology with R-proofs of size $|\psi_{k-1}|^c$, since R is reasonably strong, using Lemma 4.1, where c is greater than the exponent of the polynomial in Lemma 4.1. But this implies ψ_k is not a tautology, contradicting our assumption on ψ.

Now we use the assumption of Rudich's Conjecture to complete the inductive step. It remains to prove that with probability $1-o(1)$, for ψ sampled from D_N, $\mathsf{lb}_R^k(\psi, N^c)$ does not have R-proofs of size $|\mathsf{lb}_R^k(\psi, N^c)|^c$. Suppose, for the sake of contradiction that with probability $\Omega(1)$, for infinitely many N, for ψ sampled from D_N, $\mathsf{lb}_R^k(\psi, N^c)$ has R-proofs of the desired size. We use this to define a non-deterministic polynomial-time algorithm that accepts a constant fraction of truth tables of hard Boolean functions on n bits and does not accept any easy Boolean functions on n bits, for infinitely many n, in contradiction to Rudich's Conjecture. Given a truth table of a Boolean function f_n, the algorithm checks if there is a R-proof of $\psi_k = \mathsf{lb}_R^k(\psi, N^c)$ of size at most $|\psi_k|^c$, where $\psi = \mathsf{ttable}(f_n, n^\ell)$, and accepts if yes. If the algorithm does accept on ψ_k, then by the soundness of R, ψ_k is a tautology, and since k is even, so is ψ by the inductive argument in the previous para. Hence f is indeed a hard Boolean function, as desired. Moreover, for f_n chosen uniformly at random, the algorithm accepts with probability $\Omega(1)$ by assumption, for infinitely many n, contradicting Rudich's Conjecture.

If k is odd, we need a slightly more involved argument, which generalizes the argument used to show Lemma 2 in [PS19]. Since Rudich's Conjecture holds, it follows that there are *succinct hitting sets* against polynomial-size non-deterministic circuits, i.e., a sequence $\{H_m\}$ of sets of strings in $\{0,1\}^m$, where each H_m is a truth table of a Boolean function on $\log(m)$ inputs with circuits of size $\log(m)^\ell$ (we assume without loss of generality that m is a power of 2), such that for every sequence $\{C_m\}$ of non-deterministic circuits that accept a $\Omega(1)$ fraction of their inputs, at least one element of H_m is accepted by C_m for all large enough m. Also, since Rudich's Conjecture holds, it follows that SAT does not have polynomial-size circuits. Using a straightforward argument, the sequence $\{H'_m\}$ of sets of strings in $\{0,1\}^m$, where each H'_m is $f^{\mathsf{SAT}}_{\log(m)} \oplus y$ for $y \in H_m$, is also a succinct hitting set sequence against polynomial-size non-deterministic circuits, but in this case the sets consist of truth tables of *hard* Boolean functions on $\log(m)$ inputs. We have that $\mathsf{ttable}(z, n^\ell)$ is a tautology for $z \in H'_m$ when $m = 2^n$ is large enough. Moreover, by the same argument as in the proof of Lemma 2 in [PS19], for each $z \in H'_m$, $\mathsf{ttable}(z, n^\ell)$ has R-proofs of size at most $|\mathsf{ttable}(z, n^\ell)|^c$, using the fact that R p-simulates Extended Frege and has $\mathsf{ttable}(f_n^{\mathsf{SAT}}, 2n^\ell)$ as an axiom.

These facts can be used to show by the inductive argument in the proof of Thm. 4.2 that the second item of Thm. 4.2 holds for the formulas $\psi_k = \mathsf{lb}_R^k(\psi, N^c)$ where $\psi = \mathsf{ttable}(z, n^\ell)$ for $z \in H'_m$: they are tautologies with short R-proofs when k is even, and non-tautologies (and hence without any R-proofs at all) when k is odd. In the current case of our inductive step, k is odd, and hence

every such formula ψ_k is a non-tautology, and hence does not have short proofs. Now suppose for the sake of contradiction, that with probability $\Omega(1)$ over uniformly chosen Boolean function f_n on n variables, for infinitely many n, $\mathsf{lb}_R^k(\psi, N^c)$ has short R-proofs, where $\psi = \mathsf{ttable}(f_n, n^\ell)$. This means that the polynomial-time non-deterministic algorithm that on input f_n, checks if there is a short R-proof of $\mathsf{lb}_R^k(\mathsf{ttable}(f_n, n^\ell), N^c)$ accepts a constant fraction of functions f_n. However, none of the functions z for $z \in H'_m$ is accepted. This contradicts the assumption that $\{H'_m\}$ is a hitting set sequence for all m, and hence also contradicts Rudich's Conjecture. □

Appendix

A Basic Reasoning in IPS

Here we develop basic efficient reasoning in IPS. This is taken verbatim from [AGHT20].

First we show that polynomial identities are proved for free in IPS:

Fact A.1. *If $F(\overline{x})$ is a circuit in the variables $\overline{x}$ over the field $\mathbb{F}$ that computes the zero polynomial, then there is an IPS proof of $F(\overline{x}) = 0$ of size $|F|$.*

Proof of fact. The IPS proof of $F(\overline{x}) = 0$ is simply $C(\overline{x}, \overline{z}) := F(\overline{x})$ (note that we do not need to use the boolean axioms nor any other axioms in this case). Observe that both conditions 1 and 2 for IPS hold in this case (Definition 2.2). □

Fact A.2. *Let F, G, H be circuits and $\mathcal{F}$ be a collection of polynomial equations such that $C : \mathcal{F} \vdash^{s_0}_{\mathrm{IPS}} F = G$ and $C' : \mathcal{F} \vdash^{s_1}_{\mathrm{IPS}} G = H$. Then, $(C + C') : \mathcal{F} \vdash^{s_0+s_1+1}_{\mathrm{IPS}} F = H$.*

Proof of fact. $C(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x}) + C'(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x}) = F - G + G - H$. □

Fact A.3. *Let F, G be circuits and $\overline{\mathcal{F}}$ be a collection of polynomial equations such that $C : \overline{\mathcal{F}} \vdash^{s_0}_{\mathrm{IPS}} F = G$ and $C' : \overline{\mathcal{F}} \vdash^{s_1}_{\mathrm{IPS}} H = K$. Then, $(C + C') : \overline{\mathcal{F}} \vdash^{s_0+s_1+1}_{\mathrm{IPS}} F + H = G + K$.*

Proof of fact. $C(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x}) + C'(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x}) = F - G + H - K$. □

Fact A.4. *Let F, G be circuits and $\overline{\mathcal{F}}$ be a collection of polynomial equations such that $C : \overline{\mathcal{F}} \vdash^{s_0}_{\mathrm{IPS}} F = G$ and $C' : \overline{\mathcal{F}} \vdash^{s_1}_{\mathrm{IPS}} H = K$. Assume that there is a circuit with two output gates, of size s, with one output gate computing H and the other output gate computing G. Then, $\overline{\mathcal{F}} \vdash^{s_0+s_1+s+5}_{\mathrm{IPS}} F \cdot H = G \cdot K$.*

Proof of fact. Observe that $C(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x}) \cdot H + C'(\overline{x}, \overline{\mathcal{F}}, \overline{x}^2 - \overline{x}) \cdot G = F \cdot H - G \cdot H + H \cdot G - K \cdot G = F \cdot H - G \cdot K$. Hence, the desired proof is the *circuit* $C(\overline{x}, \overline{y}, \overline{z}) \cdot H(\overline{x}) + C'(\overline{x}, \overline{y}, \overline{z}) \cdot G(\overline{x})$, which by assumption that there is a circuit of size s computing both H, G, is at most $s_0 + s_1 + s + 5$ (here, H, G can have common nodes). □

We now wish to show that basic reasoning by *boolean* cases is efficiently attainable in IPS. Specifically, we are going to show that if for a given constant many variables (or even boolean valued polynomials) V, for every choice of a fixed (partial) boolean assignment to the variables V a polynomial equation is derivable, then it is derivable regardless (namely, derivable from the boolean axioms alone) in polynomial-size.

Proposition A.5 (proof by boolean cases in IPS [AGHT20])**.** *Let $\mathbb{F}$ be a field. Let $V = \{H_i(\overline{x})\}_{i \in I}$ be a set of circuits with $|V| = r$, and $\overline{\mathcal{F}}$ be a collection of polynomial equations such that $\{H_i^2(\overline{x}) - H_i(\overline{x}) = 0\}_{i \in I} \subseteq \overline{\mathcal{F}}$. Assume that for every fixed assignment $\overline{\alpha} \in \{0,1\}^r$ we have $\overline{\mathcal{F}}, \{H_i(\overline{x}) = \alpha_i\}_{i \in I} \vdash^{s}_{\mathrm{IPS}} f(\overline{x}) = 0$, then $\overline{\mathcal{F}} \vdash^{c^r \cdot s}_{\mathrm{IPS}} f(\overline{x}) = 0$, for some constant c independent of r.*

Prop. A.5 allows us to reason by cases in IPS. For example, assume that we know that either $H_i(\overline{x}) = 0$ or $H_i(\overline{x}) = 1$; namely that we have the assumption $H_i(\overline{x}) \cdot (H_i(\overline{x}) - 1) = 0$. Then, we can reason by cases as follows: if we can prove from $H_i(\overline{x}) = 0$ that A, with a polynomial-size proof, and from $H_i(\overline{x}) = 1$ that B, with a polynomial-size proof, then using Prop. A.5 we have a polynomial-size proof that $A \cdot B = 0$ from $H_i(\overline{x}) \cdot (H_i(\overline{x}) - 1) = 0$.

As an immediate corollary of Prop. A.5 we get the same proposition with $H_i(\overline{x})$'s substituted for variables:

Corollary A.6. *Let $\mathbb{F}$ be a field. Let $V = \{x_i\}_{i \in I}$ be a set of variables with $|V| = r$, and $\overline{\mathcal{F}}$ be a collection of polynomial equations. Assume that for every fixed assignment $\overline{\alpha} \in \{0,1\}^r$ to the variables in V we have $\overline{\mathcal{F}}, \{x_i = \alpha_i\}_{i \in I} \vdash^s_{\mathrm{IPS}} f(\overline{x}) = 0$, then $\overline{\mathcal{F}} \vdash^{c^r \cdot s}_{\mathrm{IPS}} f(\overline{x}) = 0$, for some constant c independent of r.*

Fact A.7 (IPS proofs are closed under substitutions)**.** *Let $C(\overline{x}, \overline{y}, \overline{z})$ be an IPS proof of $f(\overline{x})$ from the assumptions $\{F_i(\overline{x})\}_{i=1}^m$, and let $\overline{H} = \{H_i(\overline{x})\}_{i=1}^n$ be a set of algebraic circuits. Then, $C(\overline{H}/\overline{x}, \overline{y}, \overline{z})$ is an IPS proof of $f(\overline{H}/\overline{x})$ from $\{F_i(\overline{H}/\overline{x})\}_{i=1}^m$, where $\overline{H}/\overline{x}$ stands for the substitution of x_i by $H_i(\overline{x})$, for all $i \in [n]$.*

The proof of Fact A.7 is immediate.

Acknowledgements

We wish to thank Jan Pich for very helpful discussions during the work on this paper.

References

[ABSRW00] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Pseudorandom generators in propositional proof complexity. In *41st Annual Symposium on Foundations of Computer Science (Redondo Beach, CA, 2000)*, pages 43–53. IEEE Comput. Soc. Press, Los Alamitos, CA, 2000.

[AGHT20] Yaroslav Alekseev, Dima Grigoriev, Edward A. Hirsch, and Iddo Tzameret. Semi-algebraic proofs, IPS lower bounds and the τ-conjecture: Can a natural number be negative? *STOC 2020*, 2020.

[AM19] Albert Atserias and Moritz Müller. Automating resolution is np-hard. In *60th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2019, Baltimore, Maryland, USA, November 9-12, 2019*, pages 498–509, 2019.

[BIK+96] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. London Math. Soc. (3)*, 73(1):1–26, 1996.

[CR74a] Stephen A. Cook and Robert A. Reckhow. Corrections for "On the lengths of proofs in the propositional calculus (preliminary version)". *SIGACT News*, 6(3):15–22, July 1974.

[CR74b] Stephen A. Cook and Robert A. Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *STOC 1974*, pages 135–148, 1974. For corrections see Cook-Reckhow [CR74a].

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979. This is a journal-version of Cook-Reckhow [CR74b] and Reckhow [Rec76].

[FSTW16] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 32:1–32:17, 2016.

[Gar19] Michal Garlík. Resolution lower bounds for refutation statements. In Peter Rossmanith, Pinar Heggernes, and Joost-Pieter Katoen, editors, *44th International Symposium on Mathematical Foundations of Computer Science, MFCS 2019, August 26-30, 2019, Aachen, Germany*, volume 138 of *LIPIcs*, pages 37:1–37:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2019.

[GKRS19] Mika Göös, Pritish Kamath, Robert Robere, and Dmitry Sokolov. Adventures in monotone complexity and TFNP. In *10th Innovations in Theoretical Computer Science Conference, ITCS 2019, January 10-12, 2019, San Diego, California, USA*, pages 38:1–38:19, 2019.

[GP18] Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing: The ideal proof system. *J. ACM*, 65(6):37:1–37:59, 2018.

[Hak85] Armin Haken. The intractability of resolution. *Theoret. Comput. Sci.*, 39(2-3):297–308, 1985.

[KPW95] Jan Krajíček, Pavel Pudlák, and Alan Woods. An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Structures Algorithms*, 7(1):15–39, 1995.

[Kra97] Jan Krajíček. Interpolation theorems, lower bounds for proof systems, and independence results for bounded arithmetic. *The Journal of Symbolic Logic*, 62(2):457–486, 1997.

[Kra01] Jan Krajíček. Tautologies from pseudo-random generators. *Bull. Symbolic Logic*, 7(2):197–212, 2001.

[LTW18] Fu Li, Iddo Tzameret, and Zhengyu Wang. Characterizing propositional proofs as noncommutative formulas. In *SIAM Journal on Computing*, volume 47, pages 1424–1462, 2018. Full Version: http://arxiv.org/abs/1412.8746.

[PBI93] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *Comput. Complexity*, 3(2):97–140, 1993.

[PS19] Jan Pich and Rahul Santhanam. Why are proof complexity lower bounds hard? In *60th Annual IEEE Symposium on Foundations of Computer Science FOCS 2019, November 9-12, 2019, Baltimore, Maryland USA*, 2019.

[PT16] Tonnian Pitassi and Iddo Tzameret. Algebraic proof complexity: Progress, frontiers and challenges. *ACM SIGLOG News*, 3(3), 2016.

[Pud19] Pavel Pudlak. Reflection principles, propositional proof systems, and theories. *ArXiV*, Jul 2019.

[Raz10] Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010.

[Raz15] Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, 181:415–472, 2015.

[Rec76] Robert A. Reckhow. *On the lengths of proofs in the propositional calculus*. PhD thesis, University of Toronto, 1976.

[Rud97] Steven Rudich. Super-bits, demi-bits, and np/qpoly-natural proofs. In José D. P. Rolim, editor, *Randomization and Approximation Techniques in Computer Science, International Workshop, RANDOM'97, Bolognna, Italy, July 11-12. 1997, Proceedings*, volume 1269 of *Lecture Notes in Computer Science*, pages 85–93. Springer, 1997.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

[Val79a] Leslie G. Valiant. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on the Theory of Computing*, pages 249–261. ACM, 1979.

[Val79b] Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979.

[Val82] Leslie G. Valiant. Reducibility by algebraic projections. *Logic and Algorithmic: International Symposium in honour of Ernst Specker*, 30:365–380, 1982.