Iterated Lower Bound Formulas: A Diagonalization-Based Approach to Proof Complexity*

Rahul Santhanam† Iddo Tzameret‡

December 3, 2024

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 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 provides a natural equivalence between proof size 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 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.

More generally, for any strong enough propositional proof system R we propose a new explicit hard candidate, the *iterated R-lower bound formulas*, which inductively asserts the non-existence of short R proofs for formulas encoding this same statement at a different input length. We show that these formulas are unconditionally hard for Resolution following recent results of Atserias and Müller and of Garlik. We further give evidence in favour of this hypothesis for other proof systems.

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, surprisingly in the area of resource bounded proofs, namely proof complexity, diagonalization has had very little impact so far.

Propositional proof complexity is concerned with 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

*A preliminary version of this work appears in *53rd Ann. Symp. Theory of Comput., STOC 2021.*

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

‡Department of Computing, Imperial College London. iddo.tzameret@gmail.com. Part of this project has received funding from the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant agreement No. 101002742). Part of this work was done while on sabbatical visit to Oxford University, supported in part by ERC Consolidator Grant Agreement No. 615075.

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 [Ajt88, KPW95, PBI93], but thus far it remains completely open whether the same is the case for strong proof systems like Frege and Extended Frege. Lower bounds on these systems seem extremely hard to attain, and there is even a shortage of good explicit candidates for hard instances.

In this paper, we propose 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, e.g., 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 [SS95] about hardness of expressing factorial numbers with $0, 1, -1$ constants and $\times, +$ gates, imply lower bounds on the IPS as shown recently in [AGHT24]. Moreover, unconditional lower bounds on some restricted versions of IPS have been shown to follow from certain algebraic circuits lower bounds in [FSTW21], 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 clause-to-variable density and random truth table formulas[1] are plausible candidates, but are not explicit. The only plausible explicit candidates of which we are aware, apart from canonical examples such as reflection principles for strong proof systems[2], are explicit circuit lower bound tautologies and the more general proof complexity generators [ABSRW04, Kra01]. However, even explicit circuit lower bound tautologies are no longer hard candidates for strong propositional proof systems that *can* prove circuit lower bounds - such a strong proof system can be defined by adding a circuit lower bound tautology as an axiom to Frege, if circuit lower bounds do indeed hold for functions in exponential time. 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

In the first part of this work we show how our diagonalisation applies to strong algebraic proof systems with relation to algebraic circuit lower bounds. In the second part we introduce our iterated lower bounds formulas as a means to attain proof complexity lower bounds via diagonalisation for general propositional proof systems.

1.1.1 Proof complexity characterisation of $\mathsf{VP} \neq \mathsf{VNP}$

Our main technical result is an *equivalence* between super-polynomial algebraic circuit lower bounds for the Permanent and non-existence of IPS short refutations 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 a finite field $\mathbb{F}$, and $\mathsf{lb}_{IPS}(\text{“}\phi_n\text{”}, s)$ to denote appropriate CNF encodings of the statements that there are no IPS-proofs of ϕ_n of size $s(|\phi|)$, where $s : \mathbb{N} \to \mathbb{N}$ is a function (below, m refers to the size of "$\mathsf{VNP} \neq \mathsf{VP}$").[3]

Theorem 1.1 (Equivalence between Algebraic Circuit Lower Bounds and IPS Non-Short Provability; informal)**.** *Let p be any prime different than 2, and $\mathbb{F}_p$ be the field of size p. $\mathsf{VNP}_{\mathbb{F}_p} \neq \mathsf{VP}_{\mathbb{F}_p}$ iff there is a constant c such that (infinitely often) there are no polynomial-size IPS proofs over $\mathbb{F}_p$ of the CNF formula $\mathsf{lb}_{IPS}(\text{“}\mathsf{VNP}_{\mathbb{F}_p} \neq \mathsf{VP}_{\mathbb{F}_p}\text{”}, m^c)$.*

Note that the explicit CNF formulas 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 Theorem 1.1 for *every* algebraic proof system that efficiently simulates IPS.

It is important to clarify that the formulas $\mathsf{lb}_{IPS}(\text{“}\mathsf{VNP} \neq \mathsf{VP}\text{”}, m^c)$ are not necessarily tautologies: we do not know if there are IPS lower bounds against proving $\mathsf{VNP} \neq \mathsf{VP}$ (let alone whether $\mathsf{VNP} \neq \mathsf{VP}$ holds). We work with formulas that are *conjectured* to be tautologies, and show that

[1] Random truth table formulas are propositional formulas expressing that the truth table of a random Boolean function does not have small circuits.

[2] That is, principles expressing the soundness of proof systems that are conjectured to be stronger than the proof system we wish to lower bound.

[3] Here and elsewhere we abuse the terminology by writing "IPS proof of a CNF formula expressing a statement" to formally mean an IPS *refutation* refuting the *unsatisfiable* CNF expressing the *negation* of the statement.

for these conjectured tautologies, their proof complexity hardness is equivalent to $\mathsf{VNP} \neq \mathsf{VP}$. It is possible that $\mathsf{VNP} \neq \mathsf{VP}$ actually *has* polynomial-size IPS proofs, in which case Theorem 1.1 holds because $\mathsf{lb}_{IPS}(\text{“}\mathsf{VNP} \neq \mathsf{VP}\text{”}, m^c)$ is not a tautology and hence trivially lacks small IPS proofs. This, however, cannot happen if we assume a natural algebraic analogue of Razborov's Conjecture [Raz15, Kra04b] as we now explain.

Razborov conjectured that under standard circuit complexity assumptions, Frege cannot efficiently prove super-polynomial circuit lower bounds for any Boolean function (as expressed in the so-called truth table formulas). Let us formulate a reasonable algebraic analogue of Razborov's Conjecture: IPS cannot efficiently prove super-polynomial algebraic circuit lower bounds for any polynomial. Note that this algebraic variant of Razborov's conjecture, ruling out short IPS proofs of $\mathsf{VNP} \neq \mathsf{VP}$, does not rule out short proofs of the statement expressing that "*there are no short* IPS *proofs of* $\mathsf{VNP} \neq \mathsf{VP}$". Theorem 1.1 shows that assuming this variant of Razborov's conjecture, $\mathsf{VNP} \neq \mathsf{VP}$ is equivalent to a lower bound on IPS proofs of this latter statement.[4]

Theorem 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. The proof of Theorem 1.1 also provides an argument that can be applied more generally: it shows a way to *flip* the direction of any reduction that reduces proof hardness to circuit hardness, to obtain a reduction that reduces circuit hardness to proof hardness (or more precisely to non-efficient provability), as explained in Section 1.2. Theorem 1.1 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 reasonable complexity conjectures, namely that $\mathsf{VNP} \neq \mathsf{VP}$ and that the algebraic analogue of Razborov's conjecture holds, IPS finds it hard to reason about itself.

Comparison with Grochow-Pitassi [GP18] result. Our result is incomparable to that in [GP18] because it is an equivalence for a *specific* family of formulas that is conjectured to be hard but not known to be hard. Note that for such a family, neither implication (to circuit lower bounds or from circuit lower bounds) follows directly from Grochow-Pitassi. In other words, [GP18] showed that if an unsatisfiable family of CNF formulas does not have short IPS refutations, then $\mathsf{VNP} \neq \mathsf{VP}$. On the other hand, we show that if the family of formulas $\mathsf{lb}_{IPS}(\text{“}\mathsf{VNP}_{\mathbb{F}_p} \neq \mathsf{VP}_{\mathbb{F}_p}\text{”}, m^c)$ in Theorem 1.1 do not have short IPS refutations (regardless if they are unsatisfiable), then $\mathsf{VNP} \neq \mathsf{VP}$.

1.1.2 Iterated lower bounds formulas

Theorem 1.1 gives evidence that IPS finds it hard to reason efficiently about itself - could this hold for proof systems more generally[5]? A heuristic way to think of Theorem 1.1 is as a *fixed point* result in the following sense: consider $\mathsf{lb}_R(\cdot, m^{\omega(1)})$ for a proof system R as an operator mapping formulas to formulas, namely $\varphi \mapsto \mathsf{lb}_R(\varphi, m^{\omega(1)})$ (in applications we take instead of $m^{\omega(1)}$ a concrete polynomial lower bound m^c, for a constant c). Let lb_R^2 be the composition of this operator with itself, namely $\varphi \mapsto \mathsf{lb}_R(\mathsf{lb}_R(\varphi, m^{\omega(1)}), m^{\omega(1)})$. 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: if $\mathsf{VNP} \neq \mathsf{VP}$ holds then $\mathsf{lb}_R(\mathsf{lb}_R(\mathsf{VNP} \neq \mathsf{VP}, m^{\omega(1)}), m^{\omega(1)})$ holds. Indeed, our diagonalisation approach is inspired partly

[4] Razborov conjectured in [Raz15] that Frege cannot efficiently prove super-polynomial circuit lower bounds for any Boolean function. More specifically, [Raz15, Conjecture 1] with suitable parameters for the underlying combinatorial designs implies under some hardness assumptions that Frege cannot efficiently prove that $\mathrm{SAT} \not\subseteq \mathsf{P/poly}$. Further conjectures about the impossibility of *Extended* Frege to efficiently prove circuit lower bounds have been circulated in the proof complexity literature and discussions (cf. [Raz16, Raz21, Kra11]).

[5] Independently of our work, Pudlák [Pud20] poses the same question.

by Atserias and Müller, and Garlik [AM20, Gar19], who showed implicitly that *every* sequence of formulas is such 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 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:

1. $\mathsf{lb}_R^0(\phi, n^c) = \phi$;
2. $\mathsf{lb}_R^{k+1}(\phi, n^c) = \mathsf{lb}_R(\mathsf{lb}_R^k(\phi, n^c), n^c)$.

We propose the **Iterated Lower Bound Hypothesis**: for every reasonably strong[6] propositional proof system R that is not polynomially bounded, there is a ϕ such that the sequence $\{\mathsf{lb}_R^k(\phi, n^c)\}_{k=0}^{\infty}$ is a sequence of hard instances for R.

This generically 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 hypothesis - even if it fails, this gives us information about whether 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. Though there have been discussions on whether constant-depth Frege for example can efficiently prove its known lower bounds, even if this is the case our hypothesis would make sense for strong proof systems, as well as for weak proof systems.

We provide evidence in favour of the hypothesis. First, as a corollary to [AM20, Gar19], it follows that the hypothesis holds for Resolution.

Theorem 1.2. *The Iterated Lower Bounds Hypothesis 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. *Assuming Rudich's Conjecture, there is a propositional proof system R efficiently simulating Extended Frege such that for every large enough constant c and every fixed positive integer k, $\mathsf{lb}_R^k(\phi, |\phi|^c)$ does not have R-proofs of size $|\mathsf{lb}_R^k(\phi, |\phi|^c)|^c$ with high probability over the truth table formula ϕ expressing that a random Boolean function on n variables does not have circuits of size n^c.*

1.2 Overview of Techniques and Arguments

We give a high-level overview of the ideas required to show Theorem 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 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. Note that as we mentioned above in Section 1.1.1, this argument can be applied more generally: for any proof system P and any circuit

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

lower bound statement denoted *Circuit-Hard*, the existence of a short P proof of the statement that P does not admit short proofs of *Circuit-Hard* implies in fact that there *is* a short P proof of *Circuit-Hard* (at a smaller input length), assuming a proof complexity lower bound on P proofs implies *Circuit-Hard* (and as long as we can formalise this implication efficiently in P).

Applying the diagonalisation argument. For the purpose of our argument we make the following assumptions, which we will justify later:

(1) There is a reasonable CNF encoding of "$\mathsf{VNP} \neq \mathsf{VP}$".

(2) There is a reasonable CNF encoding of the statement that there are no IPS lower bounds of size s for a CNF ϕ.

(3) 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 "$\mathsf{VNP} \neq \mathsf{VP}$".

Assumption 3 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 Assumption 3 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 backward direction. Either $\phi =$ "$\mathsf{lb}_{IPS}(\text{"}\mathsf{VNP} \neq \mathsf{VP}\text{"}, n^{\omega(1)})$" 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 $\mathsf{VNP} \neq \mathsf{VP}$.

For the forward direction we proceed as follows. 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 $\mathsf{lb}_{IPS}(\text{"}\mathsf{VNP} \neq \mathsf{VP}\text{"}, n^{\omega(1)})$ 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.

To clarify further the forward direction which is the challenging direction, we now describe a slightly more detailed overview, highlighting the logic behind the argument. To match the proofs in Section 3.2, we shall switch to using *refutations* instead of proofs. Namely, we show that assuming $\mathsf{VNP} \neq \mathsf{VP}$, infinitely often there are no polynomial-size IPS refutations of "$\mathsf{ub}_{IPS}(\text{"}\mathsf{VNP} = \mathsf{VP}\text{"}, n^c)$", expressing that there exists a constant c such that "$\mathsf{VP} = \mathsf{VNP}$" has IPS refutations of size bounded from above by the polynomial n^c.

Proof sketch for the forward direction of Theorem 1.1. Let i.o. abbreviate *infinitely often*, and let a.e. denote its converse *almost everywhere*, that is, "always except for finite many cases". Let $G\vdash^{f(n)}_{\text{IPS}} 1 = 0$ stands for an IPS refutation of G of refutation-size bounded from above $f(n)$. Our goal is to prove:

$$\exists c' \forall c, \text{i.o. } \mathsf{ub}_{\text{IPS}}(\text{"}\mathsf{VNP} = \mathsf{VP}\text{"}, n^{c'}) \not\vdash^{n^c}_{\text{IPS}} 1 = 0, \tag{1}$$

meaning that there is a polynomial $n^{c'}$, such that no polynomial-size n^c bounds from above the refutation size of $\mathsf{ub}_{\text{IPS}}(\text{"}\mathsf{VNP} = \mathsf{VP}\text{"}, n^{c'})$.

Assume by way of contradiction that the converse holds, namely:

$$\forall c' \exists c, \text{a.e. } \mathsf{ub}_{\text{IPS}}(\text{"}\mathsf{VNP} = \mathsf{VP}\text{"}, n^{c'}) \vdash^{n^c}_{\text{IPS}} 1 = 0. \tag{2}$$

Using the argument of [GP18], we get that if VNP has polynomial-size circuits, then IPS is polynomially bounded, namely:

$$\forall c' \exists c, \text{a.e. } \mathsf{VNP} = \mathsf{VP} \vdash^{n^{c''}}_{\text{IPS}} \mathsf{ub}_{\text{IPS}}(\text{"}\mathsf{VNP} = \mathsf{VP}\text{"}, n^{c'}) \vdash^{n^c}_{\text{IPS}} 1 = 0.$$

Hence, in particular, by combining the two proofs into one, we get

$$\exists c''', \text{a.e. } \mathsf{VNP} = \mathsf{VP} \left|\frac{n^{c'''}}{\text{IPS}}\right. 1 = 0.$$

Thus,

$$\exists c''', \text{a.e. } \mathsf{ub}_{\text{IPS}}(\text{“}\mathsf{VNP} = \mathsf{VP}\text{”}, n^{c'''})$$

is a *satisfiable* CNF formula, which means that

$$\exists c'''\forall c, \text{a.e. } \mathsf{ub}_{\text{IPS}}(\text{“}\mathsf{VNP} = \mathsf{VP}\text{”}, n^{c'''}) \left|\frac{n^{c}}{\text{IPS}}\right. 1 = 0,$$

which is a contradiction to Equation (2).

Some details are hidden in the above sketch. For example, 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 as defined in Raz [Raz10]. To express that a given polynomial is computable by a small circuit we use a set of equations. Each equation states that the coefficient of a monomial computed by the universal circuit has the appropriate value.

Using this formalisation of polynomials computed by small circuits we can encode $\mathsf{VP} \neq \mathsf{VNP}$ as the statement expressing that the coefficient vector of the permanent polynomial is not equal to the coefficient vector 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.

1.3 Related Work

Following on unpublished work of Friedman [Fri79], Pudlak [Pud86, Pud87] showed finitistic versions of Gödel's Theorem: for any strong enough first-order theory T of arithmetic, there is a constant $\epsilon > 0$ such that the finitistic consistency principle $Con_T(n)$ stating that there are no T-proofs of $1 = 0$ of size at most n requires T-proofs of size at least n^ϵ. For first-order theories satisfying additional properties, stronger lower bounds approaching n can be obtained [Pud87]. By the standard translation between first-order theories and propositional proof systems, this yields non-trivial lower bounds on the Q-proof size of the reflection principle for Q when Q is a strong enough propositional proof system. However, the lower bounds obtained in this way are *sub-linear* in the formula size, while we are interested in the question of *super-polynomial* size lower bounds. Note that the reflection principle in fact has proofs of polynomial size [Pud86].

Diagonalization techniques have also been explored in work of Krajicek [Kra04c, Kra04a]. In [Kra04c], the notion of *implicit proofs* is defined and studied. This concept is used in [Kra04a] to show a conditional result: if E requires exponential-size circuits and $\mathsf{NP} = \mathsf{coNP}$, then there is no p-optimal propositional proof system. Our techniques here are very different, and our results seem unrelated to those in [Kra04a].

On the other hand, the results of Grochow and Pitassi [GP18] are crucial to our work. Grochow and Pitassi defined the Ideal Proof System, an algebraic proof system for which proofs are verifiable by polynomial size circuits (and in polynomial time under a standard derandomization assumption). They showed that super-polynomial IPS lower bounds for CNFs would imply $\mathsf{VNP} \neq \mathsf{VP}$. This connection between proof complexity lower bounds and circuit complexity lower bounds has been further developed in [FSTW21, LTW18, AGHT24]. However, none of these previous works establish an *equivalence* between proof complexity lower bounds and standard circuit lower bounds, as in Theorem Theorem 1.1.

We are also inspired by recent work of [AM20, PS19]. Atserias and Muller [AM20] settle the long-standing open problem of whether Resolution is automatable (assuming $\mathsf{P} \neq \mathsf{NP}$) by giving a reduction from SAT to proof complexity lower bounds for Resolution via variants of the proof complexity lower bound formulas. Their reduction relies on the hardness of the Pigeonhole Principle, which only holds for weak proof systems such as Resolution, while we are interested here in strong proof systems such as IPS. Pich and Santhanam [PS19] unconditionally establish a version of the Natural Proofs barrier [RR97] in proof complexity, but they do so for a proof system that is defined *non-constructively* and moreover for instances which are *randomly generated*. In contrast, we are interested here in the meta-mathematics of proof complexity for well-studied and concrete proof systems such as IPS and for *explicit* instances. We do use ideas from [AM20, PS19] to give evidence for the Iterated Lower Bounds Hypothesis.

Finally, the idea of iterating proof complexity lower bounds is novel to the best of our knowledge. Nevertheless, as discussed above, considering the proof complexity of proof complexity lower bounds has been investigated to different degrees of explicitness in the literature. And we refer the reader to the recent excellent survey by Pudlak [Pud20] that explores this and other related questions.

2 Preliminaries

2.1 Basic Algebraic Complexity

For an excellent 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. 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 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 for a pair $s, t : \mathbb{N} \to \mathbb{N}$ of polynomially bounded functions, g_n has $t(n)$ input variables and can be written as

$$g_n\left(x_1, \ldots, x_{t(n)}\right) = \sum_{\overline{e} \in \{0,1\}^{s(n)}} f_{s(n)}(e_1, \ldots, e_{s(n)}, x_1, \ldots, x_{t(n)})$$

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* of another family of polynomials (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 when the ring R is a field of characteristic different from 2. Namely, every polynomial family $(f_n)_n^{\infty} \in \mathsf{VNP}$ is a p-projection of the permanent polynomial: for every n, there exists a projection $L(\overline{x}) : y_1 \ldots, y_{t(n)} \to \overline{x} \cup \mathbb{F}$, such that $perm_{t(n)}(L(\overline{x})) = f_n(\overline{x})$, where $t(n) : \mathbb{N} \to \mathbb{N}$ is a polynomial bounded function, and $\mathrm{perm}_{t(n)}$ is the permanent function in $t(n)$ $\overline{y}$-variables [Val79a]. 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 [Val79b, Val79a, Val82]. Since the permanent is complete for VNP (under p-projections), showing $\mathsf{VP} \neq \mathsf{VNP}$ 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 the Nullstellensatz proof system ([BIK+96a]) with size of refutations measured by algebraic circuit size (instead of the usual number of monomials). 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 [FSTW21]:

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*[7]*):*

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+96a]).

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.

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, denoted IPS^{alg}, 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 2.4 (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 2.5 ([GP18])**.** *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 Theorem 2.4, assuming Lemma 2.5. 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 2.5—cannot be computed by polynomial-size algebraic circuits, and hence that $\mathsf{VNP} \neq \mathsf{VP}$. □

[7] 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 $p(\overline{x})$.

2.2.1 Conventions and Notations for IPS Proofs

An IPS (algebraic or not) proof over a specific field or ring is sometimes denoted $\mathrm{IPS}_{\mathbb{F}}$ specifying explicitly it is over $\mathbb{F}$. For two algebraic circuits F, G, we define the *size of the circuit 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).

The following fact shows that polynomial identities are proved for free in IPS:

Fact 2.6 ([AGHT24])**.** *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). □

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 by Fact 2.6).

We denote by $C : \overline{\mathcal{F}} \big|_{\overline{\mathrm{IPS}}}^{s} p = 0$ (resp. $C : \overline{\mathcal{F}} \big|_{\overline{\mathrm{IPS}}}^{s} p = g$) the fact that $p = 0$ (resp. $p = g$) has an IPS proof $C(\overline{x}, \overline{y}, \overline{z})$ of size s from assumptions $\overline{\mathcal{F}}$, and we use $\big|_{\overline{\mathrm{IPS}}}^{s,\ d}$ when we want to specify that the degree of the IPS proof is upper bounded by d. 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 $C : \overline{\mathcal{F}} \big|_{\overline{\mathrm{IPS}}}^{s} p$ for $C : \overline{\mathcal{F}} \big|_{\overline{\mathrm{IPS}}}^{s} p = 0$. 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 $C : \overline{\mathcal{F}} \big|_{\overline{\mathrm{IPS}}} F$, 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 $\overline{\mathcal{F}}$. If $\{p_i = 0\}_{i=1}^{\infty}$ and $\{\overline{\mathcal{F}}_i\}_{i=0}^{\infty}$ are sequences of circuit equations and sets of circuit equations, respectively, then we write $\overline{\mathcal{F}}_i \big|_{\overline{\mathrm{IPS}}}^{*} p_i = 0$ to denote that there is an IPS proof of $p_i = 0$ from the assumptions $\overline{\mathcal{F}}_i$ of size polynomial in $|\overline{\mathcal{F}}_i| + |p_i|$. We write $\big|_{\overline{\mathrm{IPS}}}^{d,s}$ to denote the *negation* of $\big|_{\overline{\mathrm{IPS}}}^{d,s}$, and similarly with all other possible super and subscripts.

When we deal with *algebraic* IPS proofs we will use the same notation as above, only using $\mathrm{IPS}^{\mathrm{alg}}$ instead of IPS.

3 Proof Complexity Characterization of VP ≠ VNP

We show that IPS cannot efficiently prove that the algebraic circuit class separation VP ≠ 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 VP = VNP. On the other hand, if VP ≠ VNP is in fact easy to prove in IPS or if VP = VNP, then the result is less interesting, since then the result stems simply from soundness of IPS. Similarly, if the IPS refutations of VP = VNP are of exponential degree, then the result becomes trivially true again,

since there *is* a polynomial-size refutation of our formulation of $\mathsf{VP} = \mathsf{VNP}$ (since the formulation is exponential in itself). The more interesting scenario is under the reasonable assumption that indeed algebraic circuit class separations are hard to establish in polynomial-degree IPS, and in this case we show that this fact would not have efficient proofs in IPS.

3.1 Formalisations

3.1.1 CNF Encoding of Algebraic Circuit Equations

We are going to work at times with IPS that does *not* have the boolean axioms for all variables (but only for some variables that will be explicitly specified), namely algebraic IPS, denoted $\mathrm{IPS}^{\mathrm{alg}}$. 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 the boolean axioms that we add to the formulas are.

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. *When we work with CNF formulas in IPS we assume that the CNF formulas are translated according to Definition 2.3.* The size of objects like circuits, circuit equations, sets of circuit equations and formulas in CNF are denoted by $|\cdot|$ (where "$\cdot$" is replaced by the respective object).

Definition 3.1 (Algebraic extension axioms and unary bits)**.** *Given a circuit C 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 each variable x_{gi} 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 $x_{gi} = 1$. Note also that the unary-bits are disjoint from the (algebraic input) variables of the circuit C.

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

1. *If x_i is an input gate in C, the plain CNF encoding of C uses the variables $x_{x_i 0}, \ldots, x_{x_i (q-1)}$ that are the unary-bits of x_i, and contains the clauses that express that precisely one unary-bit is 1 and all other unary-bits are 0:* [8]

$$\bigvee_{j=0}^{q-1} x_{x_i j} \wedge \bigwedge_{j \neq \ell \in \{0,\ldots,q-1\}} (\neg x_{x_i j} \vee \neg x_{x_i \ell}). \tag{3}$$

[8] 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 takes on the value one.

2. *If $\alpha \in \mathbb{F}_q$ is a scalar input gate in C, the plain CNF encoding of C contains the* $\{0,1\}$ constants *corresponding to the unary-bits of α. These constants are used when they are fed to (translations of) gates according to the wiring of C in Item 3.*

3. *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 $\overline{\alpha}$; "$*$" here means the Kleene star). 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 of C), 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 values 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).*

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* $\mathrm{IPS}^{\mathrm{alg}}$*):*

$$x_{gi} \vee \neg x_{g_i} .$$

Note that the size of the encoding in Item 3 is exponential in q, but since q is constant this will be sufficient for our purposes. We now define the same encoding, only for a circuit *equation* instead of merely a circuit, namely the CNF is satisfied by an assignment to the unary-bits precisely when the gates encode correctly the computation of the circuit and additionally the output gate evaluates to zero.

Definition 3.3 (Plain CNF encoding of algebraic circuit equations; $\mathsf{cnf}\,(C(\overline{x})=0)$)**.** *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 plain CNF encoding from Definition 3.2 in the unary-bits variables of all the gates in C (and only in the unary-bit variables), together with the equations*

$$x_{g_{out}0} = 1 \quad \text{and} \quad x_{g_{out}i} = 0, \quad \text{for all } i = 1,\ldots,q-1,$$

which express that $g_{out} = 0$, where g_{out} is the output gate of C.

Definition 3.4 (Extended CNF encoding of a circuit equation (circuit, resp.); $\mathsf{ecnf}\,(C(\overline{x})=0)$ ($\mathsf{ecnf}\,(C(\overline{x}))$, resp.))**.** *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 $C(\overline{x}) = 0$ (circuit $C(\overline{x})$, resp.), in symbols $\mathsf{ecnf}\,(C(\overline{x})=0)$ ($\mathsf{ecnf}\,(C(\overline{x}))$, resp.), 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-bit variables corresponding to every node g in C, that consists of:*

1. *the plain CNF encoding of the circuit equation $C(\overline{x}) = 0$ (circuit $C(\overline{x})$, resp.), namely, $\mathsf{cnf}\,(C(\overline{x})=0)$ ($\mathsf{cnf}\,(C(\overline{x}))$, resp.) translated to polynomials via Definition 2.3; and*

2. *the algebraic extension axiom of g, for 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 these variables are meant to range over all $\mathbb{F}_q$ and not merely $\{0,1\}$ (while it does contain the boolean axioms for the unary-bit variables).

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 of $C(\overline{x})$ (which is an equivalent in strength formulation to algebraic circuits):

Definition 3.5 (Straight line program (SLP))**.** *An **SLP** of a circuit $C(\overline{x})$ is a sequence of equations between variables such that the extension variable for the output gate computes the value of the circuit assuming all equations hold. Formally, we choose any topological order $g_1, g_2, \ldots, g_{|C|}$ 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 set of equations to be the SLP of $C(\overline{x})$:*

$$g_i = g_j \circ g_k \text{ for } \circ \in \{+, \times\} \text{ iff } g_i \text{ is a } \circ\text{-gate in } C \text{ with two incoming edges from } g_j \text{ and } g_k.$$

An SLP representation of a circuit equation $C(\overline{x}) = 0$ *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 show how to efficiently go, within 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 3.6 (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 (recall that $\vdash^*_{\text{IPS}^{\text{alg}}}$ means polynomial-size proofs):*

$$\mathsf{ecnf}\left(C(\overline{x}) = 0\right) \vdash^*_{\text{IPS}^{\text{alg}}} C(\overline{x}) = 0 \tag{4}$$

and

$$\left\{x_g = \sum_{i=0}^{q-1} i \cdot x_{gi} : g \text{ a node in } C\right\},$$
$$\left\{x_{gi}^2 - x_{gi} = 0 \ : \ g \text{ is a node in } C,\ 0 \le i < q\right\}, \quad \vdash^*_{\text{IPS}^{\text{alg}}} \mathsf{ecnf}\left(C(\overline{x}) = 0\right). \tag{5}$$
$$\left\{\sum_{i=0}^{q-1} x_{gi} = 1 \ : \ g \text{ is a node in } C\right\}, C(\overline{x}) = 0$$

Note that in Equation (5) the first set of axioms from the left are the algebraic extension axioms, the second are the boolean axioms for the unary bits, and the bottom left set of axioms expresses that each gate has only one unary-bit equal to 1. The bottom right circuit equation $C(\overline{x}) = 0$ *must* be written as an SLP (otherwise, the lemma does not immediately hold because there is no way to refer to a specific gate in the circuit).

The proof uses the fact that over finite fields the truth table of each algebraic gate is easy to describe and can be reasoned about efficiently.

Proof: The proof of Equation (4) 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 the unary-bits of u, v

as inputs, 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 $\mathsf{times}(\overline{x_{gi}}, \overline{x_{ui}}, \overline{x_{vi}})$, for $\circ = +$, $\circ = \times$, 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

$$\mathsf{plus}(\overline{x_{gi}}, \overline{x_{ui}}, \overline{x_{vi}}), \text{ and} \tag{6}$$

$$x_g = \sum_{i=0}^{q-1} i \cdot x_{gi}, \quad x_u = \sum_{i=0}^{q-1} i \cdot x_{ui}, \quad x_v = \sum_{i=0}^{q-1} i \cdot x_{vi} \tag{7}$$

we can derive with a constant-size IPS derivation

$$x_g = x_u + x_v\,, \tag{8}$$

and similarly for the product gate. This suffices to conclude the proof because the collection of all Equation (8) 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 for the unary-bits and Equation (6) semantically imply (over $\mathbb{F}_q$)

$$\sum_{i=0}^{q-1} i \cdot x_{gi} = \sum_{i=0}^{q-1} i \cdot x_{ui} + \sum_{i=0}^{q-1} i \cdot x_{vi}\,. \tag{9}$$

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; see [BIK+96b, Theorem 5.2] for a proof of this for Nullstellensatz which is simulated by IPS [FSTW21]) we get that from the boolean axioms and Equation (6) we have a constant-size derivation of Equation (9) (since the number of variables involved is constant and all variables involved have their corresponding boolean axioms). By substituting Equation (7) in Equation (9) we finish.

The proof of (5) is identical, only that we go in the other direction: we show that from Equation (8) and Equation (7) we can derive efficiently Equation (6). For that purpose, replace in Equation (8) the right hand sides of Equation (7). This substitution, together with the boolean axioms for the unary-bits that are given as assumptions, semantically imply Equation (6), and by implicational completeness of IPS over 0-1 assignments (since this derivation involves only unary-bit variables, hence all variables have their own boolean axioms) we conclude the case for the plus gate. The case of the product gate is similar. □

We wish to use at times CNF formulas, rather than extended CNF formulas. This is needed 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 CNF formulas.

We have the following simple proposition:

Proposition 3.7. *Let $C(\overline{x}) = 0$ be a circuit equation over $\mathbb{F}_q$. Then, $C(\overline{x}) = 0$ is unsatisfiable over $\mathbb{F}_q$ iff $\mathsf{cnf}\,(C(\overline{x}) = 0)$ is an unsatisfiable formula in CNF iff $\mathsf{ecnf}\,(C(\overline{x}) = 0)$ is an unsatisfiable set of equations 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 0-1 satisfying assignment to $\mathsf{cnf}\,(C(\overline{x}) = 0)$ (recall that $\mathsf{cnf}\,(C(\overline{x}) = 0)$ includes the boolean axioms for all of its variables hence a satisfying assignment to it must be a 0-1 assignment) can be extended to a satisfying assignment of $\mathsf{ecnf}\,(C(\overline{x}) = 0)$, and every satisfying assignment of $\mathsf{ecnf}\,(C(\overline{x}) = 0)$ is also a satisfying assignment to $\mathsf{cnf}\,(C(\overline{x}) = 0)$ (when restricted to the variables in $\mathsf{cnf}\,(C(\overline{x}) = 0)$).

The fact that $C(\overline{x}) = 0$ is unsatisfiable over $\mathbb{F}_q$ iff $\mathsf{cnf}\,(C(\overline{x}) = 0)$ is an unsatisfiable CNF follows from the construction of $\mathsf{cnf}\,(C(\overline{x}) = 0)$ (Definition 3.3). □

Corollary 3.8. *If* $\mathsf{ecnf}\,(C(\overline{x})=0)$ *is unsatisfiable over* $\mathbb{F}_q$ *then it has an* $\mathrm{IPS}^{\mathrm{alg}}$ *refutation in* VNP.

Proof: By Proposition 3.7 $\mathsf{cnf}\,(C(\overline{x})=0)$ is an unsatisfiable CNF. By Lemma 2.5 every unsatisfiable CNF has an $\mathrm{IPS}^{\mathrm{alg}}$ refutation computable in VNP, namely a refutation that does not use the boolean axioms. Since by definition $\mathsf{cnf}\,(C(\overline{x})=0) \subseteq \mathsf{ecnf}\,(C(\overline{x})=0)$, the $\mathrm{IPS}^{\mathrm{alg}}$ refutation of $\mathsf{cnf}\,(C(\overline{x})=0)$ is also a refutation of $\mathsf{ecnf}\,(C(\overline{x})=0)$. (Note that in this $\mathrm{IPS}^{\mathrm{alg}}$ refutation we can assume we do not use the boolean axioms of the unary-bit variables from Item 3 in Definition 3.2, even though we added these axioms explicitly to the system of equations.) □

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 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 encoded by the $\overline{w}$ variables. We shall call these $\overline{w}$-variables *edge-variables* since they should be thought of variables whose values determine which field element labels an edge in a circuit (i.e, an edge between two circuit nodes): different values to the edge-variables determine different circuits.

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

Theorem 3.9 (Existence of universal circuits for homogeneous polynomials; Raz [Raz10])**.** *Let* $\mathbb{F}$ *be a field and* $\overline{x}$ *be* n *variables, and let* $\mathcal{C}^{\mathrm{hom}}_{t,d}$ *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* $O(d^2t^8)$ *and syntactic-degree* $5d-3$ *such that* $\overline{w}$ *are* $K_{t,d}=O(dt^4)$ *variables which are disjoint from* $\overline{x}$*, that is* universal *for* $\mathcal{C}^{\mathrm{hom}}_{t,d}$ *in the following sense:* $f(\overline{x})\in\mathcal{C}^{\mathrm{hom}}_{t,d}$ *iff there exists* $\overline{\alpha}\in\mathbb{F}^{K_{t,d}}$ *such that* $U(\overline{x},\overline{\alpha})=f(\overline{x})$*.*

The idea behind the proof of Theorem 3.9 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 necessarily work with homogeneous circuits and polynomials we will assume that the universal circuit $U(\overline{x},\overline{w})$ is in fact universal for *general* (non-homogeneous) circuits of a given degree d and size t. We can assume this by defining a universal circuit as a sum of the universal circuits for each homogeneous degree as follows (this does not constitute a restriction when considering polynomial degrees, since the classical result of Strassen [Str73] shows that every algebraic circuit computing a degree at most d polynomial can be written as a sum of homogeneous circuits for each of the homogeneous components of the polynomial; where in addition each homogeneous circuit has size polynomially bounded in the original circuit size).

Definition 3.10. *The* ***universal circuit for degree*** d ***and size*** t ***circuits*** *is defined as:*

$$U(\overline{x},\overline{w})=\sum_{i=0}^{d}U_i(\overline{x},\overline{w}), \tag{10}$$

where $U_i(\overline{x},\overline{w})$ *is the universal circuit for homogeneous* $\overline{x}$*-polynomials of* i *degree* $\mathcal{C}^{\mathrm{hom}}_{t,i}$ *and where the* $\overline{w}$*-variables in each distinct* $U_i(\overline{x},\overline{w})$ *are pairwise disjoint (namely, no variable* w_l *appears in both* $U_i(\overline{x},\overline{w})$ *and* $U_j(\overline{x},\overline{w})$ *for* $i\neq j$*).*

By Theorem 3.9 the size of $U(\overline{x},\overline{w})$ is $\sum_{i=0}^{d} O(i^2t^8) = O(d^3t^8)$.

Let N be the total number of possible $\overline{x}$-monomials in a degree at most d polynomial with a given number of variables, usually n. In our formalisation, we are going to express that a circuit computes a polynomial in $\mathbb{F}[\overline{x}]$ by stating that the coefficient vector of a universal circuit described by edge variables $\overline{w}$ equals some fixed vector in $\mathbb{F}^N$. Hence, we represent a circuit of size t using $K_{t,d}$ variables $\overline{w}$ for the t edges in the circuit $U(\overline{x},\overline{w})$ (where $K_{t,d} = O(dt^4)$). Note that $U(\overline{x},\overline{w})$ is a circuit in both $\overline{x},\overline{w}$, *while our formalisation will* not *use in the end the $\overline{x}$-variables*: each $\overline{x}$-monomial will be encoded as a polynomial in the $\overline{w}$-variables representing the coefficient of this $\overline{x}$-monomial. 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, where N is the total number of $\overline{x}$-monomials.

We need to show how to compute given $U(\overline{x},\overline{w})$ the coefficient of an $\overline{x}$-monomial M as a polynomial in the edge variables $\overline{w}$. Such a polynomial is denoted $\mathrm{Coeff}_M(U(\overline{x},\overline{w}))$.

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 $f(\overline{x},\overline{w})$, for some $\alpha_i,\beta_i \in \mathbb{N}$ (where $0 \in \mathbb{N}$). Then, we call $\sum_{i\in I}\alpha_i$ the *$\overline{x}$-degree* of M.

Definition 3.11 ($\mathrm{Coeff}_M(\cdot)$)**.** *Let $f(\overline{x},\overline{w})$ be a polynomial in $\mathbb{F}[\overline{x},\overline{w}]$ in the disjoint sets of variables $\overline{x},\overline{w}$. Let M be an $\overline{x}$-monomial of degree j. Then, $\mathrm{Coeff}_M(f(\overline{x},\overline{w}))$ is the (polynomial) coefficient in $\mathbb{F}[\overline{w}]$ (that is, in the $\overline{w}$-variables only) of M in $f(\overline{x},\overline{w})$.*[9]

Note that $f(\overline{x},\overline{w}) = \sum_{M_i} M_i \cdot \mathrm{Coeff}_{M_i}(f(\overline{x},\overline{w}))$, where the M_i's are all possible $\overline{x}$-monomials of degree at most d, for d the maximal $\overline{x}$-degree of a monomial in $f(\overline{x},\overline{w})$.

Proposition 3.12 (Computation of coefficients)**.** *Let $f(\overline{x},\overline{w}) \in \mathbb{F}[\overline{x},\overline{w}]$ be a polynomial in $\mathbb{F}[\overline{x},\overline{w}]$ in the disjoint sets of variables $\overline{x},\overline{w}$. Suppose that M is an $\overline{x}$-monomial of degree d, and assume that there is an algebraic circuit computing $f(\overline{x},\overline{w})$ of size s and syntactic-degree ℓ. Then, there is a circuit of size $O(7^d \cdot s)$ computing $\mathrm{Coeff}_M(f(\overline{x},\overline{w}))$ of syntactic-degree $O(\ell)$.*

Proof: For a variable x_i we show how to construct circuits that compute the polynomials g,h, respectively, such that $x_i \cdot g + h = f$, with h having no occurrences of x_i (i.e., x_i does not appear with a positive power in any monomial in h). Thus, g is the polynomial coefficient of x_i in f (the variables x_i can appear in g). We then continue in this manner to extract the polynomial coefficient of M in f. Each such circuit-construction increases the size of f by a constant factor of 7 according to the claim below. Hence, using d such iterations for each of the d variables in M we shall get a $O(7^d \cdot s)$-size circuit D computing the polynomial coefficient of M in $f(\overline{x},\overline{w})$. However, this polynomial coefficient may contain also monomials in *both* the $\overline{x}$ and $\overline{w}$ variables (while by Definition 3.11 it should not), and *to eliminate these monomials we simply assign zeroes to all $\overline{x}$-variables in D.*

It remains to prove the following claim (recall that $\widehat{C}$ is the polynomial computed by a circuit C and that $|C|$ is the size of C; we denote by $\widehat{C}(\overline{x}) \restriction_{x_i=0}$ the polynomial $\widehat{C}(\overline{x})$ where x_i is assigned 0).

Claim 3.13. *Let $C(\overline{x})$ be a circuit of syntactic-degree ℓ in the $\overline{x}$ variables over the field $\mathbb{F}$. Then, for every variable x_i there is a circuit of size $7|C|$ and syntactic-degree at most ℓ that computes the polynomial $g(\overline{x})$, such that $\widehat{C}(\overline{x}) = x_i \cdot g(\overline{x}) + \widehat{C}(\overline{x}) \restriction_{x_i=0}$.*

Proof of claim: The proof is by 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.

[9] Note that there can be polynomial coefficients in $\mathbb{F}[\overline{x},\overline{w}]$ of M that involve also the $\overline{x}$-variables. But we wish to consider the polynomial coefficients in the $\overline{w}$-variables alone. For this reason, in Proposition 3.12 we will assign zero values to the $\overline{x}$-variables after taking the polynomial coefficient of M in the $\overline{w}$-variables.

Denote by $\mathbf{P}_{x_i}(p_v)$ the unique polynomial such that $p_v = x_i \cdot \mathbf{P}_{x_i}(p_v) + p_v \restriction_{x_i=0}$ (note that $\mathbf{P}_{x_i}(p_v)$ can contain x_i because p_v is not necessarily linear in x_i). For every gate v in C we add at most 7 new gates. The gate v itself is duplicated twice so that the first duplicate computes $\mathbf{P}_{x_i}(p_v)$ and the second duplicate computes $p_v \restriction_{x_i=0}$.

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

Induction step:
Case 1: $C(\overline{x}) = w + u$. Then, $\mathbf{P}_{x_i}(p) := \mathbf{P}_{x_i}(p_w) + \mathbf{P}_{x_i}(p_u)$ and $p \restriction_{x_i=0} := p_w \restriction_{x_i=0} + p_u \restriction_{x_i=0}$.
Case 2: $C(\overline{x}) = w \cdot u$. Then,

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

Note that the construction results in a *circuit* and 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-usage of gates in the original circuit C). This brings us to a circuit of size at most $7|C|$, and by construction syntactic-degree ℓ (note that the syntactic-degree of $\mathbf{P}_{x_i}(p_v)$ is bounded from above by the syntactic-degree of p_v). ■Claim □

Remark 3.14. *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 hypothesized hard candidates will use a circuit computing $U(\overline{x}, \overline{w})$. The diagonalisation argument works regardless of the size of the instance, only that if the size of the formulas proved is too big, and specifically exponential in the number of variables, we land in the uninteresting case in which there will be no short refutations of the statement Φ expressing short* IPS *refutations of* $\mathsf{VNP} = \mathsf{VP}$*, simply because* Φ is *satisfiable, namely there are polynomial-size (in the unsatisfiable input CNF) refutations of* $\mathsf{VP} = \mathsf{VNP}$.

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 [Val79a], over fields of characteristic different from 2.

Let $\mathrm{perm}(\overline{x})$ be the permanent polynomial on the variables $\overline{x}$. We are going to 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 3.15 (Formalisation of $\mathsf{VP} = \mathsf{VNP}$)**.** *The formalisation of* $\mathsf{VNP} = \mathsf{VP}(t, n, d)$ *denoted "*$\mathsf{VNP} = \mathsf{VP}(t, n, d)$*", expressing that there is a universal circuit for degree $d \geq n$ and size t that computes the permanent polynomial of dimension n (with $\overline{x}$ being the n^2 variables of the permanent), is the following set of polynomial equations (in the $\overline{w}$-variables only*[10]*):*

$$\{\mathrm{Coeff}_{M_i}(U(\overline{x}, \overline{w})) = b_i \ : \ 1 \leq i \leq N\} \ , \tag{11}$$

where $\overline{b} = \mathrm{coeffs}(\mathrm{perm}(\overline{x})) \in \mathbb{F}^N$ is the coefficient vector of the permanent polynomial of dimension n, $\overline{w}$ are the t edge variables, $\{M_i\}_{i=1}^N$ is the set of all possible $\overline{x}$-monomials of degree at most d,

[10]Recall that $\mathrm{Coeff}_M(U(\overline{x}, \overline{w}))$ is a polynomial in the $\overline{w}$-variables only, for M an $\overline{x}$-monomial.

and $N = \sum_{j=0}^{d} \binom{n^2+j-1}{j} = 2^{O(n^2+d)}$ *is the number of monomials of total degree at most* d *over* n^2 *variables.*

By Proposition 3.12 the size of each circuit equation in Equation (11) is $O(7^j \cdot |U(\overline{x},\overline{w})|) = O(7^n \cdot d^3 t^8)$ (with $j \leq d$ the degree of the $\overline{x}$-monomial M_i), meaning that

$$\text{the } \textit{size of } \text{“}\mathsf{VNP} = \mathsf{VP}(t,n,d)\text{”} \text{ is } O(7^n \cdot d^3 t^8 \cdot N) = t^8 \cdot 2^{O(n^2+d)}, \tag{12}$$

and the syntactic-degree of each circuit equation in (11) is $d^{O(1)}$.

Remark 3.16.

1. *Note that* “$\mathsf{VNP} = \mathsf{VP}(t,n,d)$” *(which we denote by* “$\mathsf{VNP} = \mathsf{VP}$” *when we do not care for the parameters* t,n,d*) does not contain the* $\overline{x}$ *variables, rather only the* $\overline{w}$ *variables.*

2. *The degree* d *parameter is displayed in* “$\mathsf{VNP} = \mathsf{VP}(t,n,d)$” *due to technical reasons: we would want the universal circuit that purportedly computes the permanent to have unbounded syntactic-degree* d *(and not merely syntactic-degree* n*, which is sufficient for computing th permanent of dimension* n*), because in our diagonalization argument later we will turn this universal circuit and the statement* “$\mathsf{VNP} = \mathsf{VP}(t,n,d)$” *into the statement asserting that there is a small IPS refutation of* degree d *(computed by the universal circuit) and not merely an IPS refutation of smaller degree* n*. (It is worth mentioning again that any circuit computing a polynomial of degree* n *can be converted with only a polynomial increase in size to a circuit of syntactic-degree* n*, following the standard homogenization argument [Str73].)*

3. *Note that assuming* $\mathsf{VNP} \neq \mathsf{VP}$*,* “$\mathsf{VNP} = \mathsf{VP}(t,n,d)$” *is indeed an unsatisfiable set of polynomial equations over* $\mathbb{F}_q$ *for* n *big enough and* t *polynomially bounded in* n*: any assignment to the* $\overline{w}$ *variables that satisfies* “$\mathsf{VNP} = \mathsf{VP}(t,n,d)$” *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).*

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 3.17 (IPS refutation predicate $\mathrm{IPS}_{ref}(t,d,\overline{\mathcal{F}})$)**.** *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 for degree* d *and size* t *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 label variables* $\overline{w}$*), and the* t *edge label variables* $\overline{w}$*. We formalise the existence of a size* t *and degree* d *circuit that computes the* IPS *refutation of* $\overline{\mathcal{F}}$*, denoted* $\mathrm{IPS}_{ref}(t,d,\overline{\mathcal{F}})$*, with the following set of circuit equations (in the* $\overline{w}$*-variables only):*

$$\mathrm{Coeff}_{M_i}(U(\overline{x},\mathbf{0},\overline{w})) = 0\,, \tag{13}$$

$$\mathrm{Coeff}_{M_i}\left(U(\overline{x},\overline{\mathcal{F}},\overline{w})\right) = \begin{cases} 1, & M_i = 1 \quad \textit{(i.e., the constant 1 monomial);} \\ 0, & \textit{otherwise} \quad \textit{(i.e., for all other monomials);} \end{cases} \tag{14}$$

where 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* d*,* $N = \sum_{j=0}^{d} \binom{n^2+j-1}{j} = 2^{O(n^2+d)}$ *is the number of monomials of total degree at most* d *over* n^2 *variables, and* $\mathbf{0}$ *is the all-zero vector of length* m *(replacing the* $\overline{y}$*-variables in* $U(\overline{x},\overline{y},\overline{w})$*).*

By Proposition 3.12 the size of each circuit equation in Equation (13) and Equation (14) is $O(7^j \cdot |U(\overline{x}, \overline{y}, \overline{w})| \cdot |\overline{\mathcal{F}}|) = O(7^d \cdot d^3 t^8 \cdot |\overline{\mathcal{F}}|)$ (with $j \leq d$ the degree of the $\overline{x}$-monomial M_i), meaning that

$$\text{the } \textit{size of } \mathrm{IPS}_{\textit{ref}}(t, d, \overline{\mathcal{F}}) \text{ is } \; O(7^d \cdot d^3 t^8 \cdot |\overline{\mathcal{F}}| \cdot N) = t^8 \cdot |\overline{\mathcal{F}}| \cdot 2^{O(n^2+d)}, \tag{15}$$

and the syntactic-degree of each circuit equation in Equation (13) and Equation (14) is $d^{O(1)}$.

Encoding an IPS *lower bound* is simply providing an ostensible *unsatisfiable* set of polynomials that express the existence of size at most t and degree at most d IPS refutation of a CNF $\overline{\mathcal{F}}$. Thus refuting this set of polynomials amounts to proving an 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 that IPS cannot prove certain IPS lower bounds assuming $\mathsf{VNP} \neq \mathsf{VP}$. The gist of the argument, as we explain below, is showing how from a short IPS proof of an IPS lower bound on an unsatisfiable CNF formula 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 some CNF formula, then there is an efficient IPS refutation of $\mathsf{VP} = \mathsf{VNP}$.

We shall start from the CNF "$\mathsf{VP} = \mathsf{VNP}$" and reach a contradiction (in IPS). The argument is this: if we have the CNF formula "$\mathsf{VNP} = \mathsf{VP}(t, n, d)$", meaning that $\mathrm{perm}(\overline{x})$ of dimension n is computable by size t circuits (of syntactic-degree d), 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 (Theorem 2.4), 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 formula "$\mathsf{VP} = \mathsf{VNP}$" 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 by soundness of IPS we arrive at a contradiction.

Adding helper axioms to the "$\mathsf{VP} = \mathsf{VNP}$" instance. We shall need to expand 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 $C_i(\overline{x})$. These extension axioms *do not* express though that any of the $C_i(\overline{x})$ equals any specific value. In particular this means that these additional extension axioms do not meaningfully add any information to the statement. That is, they do not actually make the statement "more unsatisfiable", and hence easier to refute. More precisely, any assignment to the original formulas—denote the set of original formulas by F, for now—can be extended to the full new set of formulas—denoted by $F \cup G$, for G the new set of additional formulas—so that the newly added formulas G are all satisfied by the assignment (even though the full set of formulas $F \cup G$ are unsatisfiable). In this sense, the additional extension axioms do not make the statement stronger (that is, "more unsatisfiable" and hence easier to refute).

In the proof complexity context, adding extension axioms for the new circuits $C_i(\overline{x})$ written as SLP does not make the instance easier to prove in Extended Frege (and similarly in IPS, which simulates Extended Frege [GP18]), since Extended Frege can introduce (sequentially) extension axioms $x_e = \psi(\overline{x})$ freely, as long as x_e is a new variable not appearing before in the proof and not appearing in the final formula proved (which is the case for refutations). Note that adding the extension axioms for the SLP of a circuit $C_i(\overline{x})$ is done in a similar fashion: we add a sequence of equations between a fresh gate in the circuit that only use previous gates or original variables. The

only difference is that in addition to innocuous extension axioms for circuits, we also add algebraic extension axioms (Definition 3.1) for the new gates: (i) $x_g = \sum_{j=0}^{q-1} j \cdot x_{gj}$, for every gate g in the new circuits $C_i(\overline{x})$ written as SLP we add to the instance (for q the characteristic of the field we work over); (ii) the Boolean axioms to the unary-bits x_{gj}; and (iii) the equation $\sum_{i=0}^{q-1} x_{gi} = 1$, stating that only one unary-bit is true. Note that here x_g was already used before when we added the SLP of the algebraic circuit $C_i(\overline{x})$. This means that we cannot use as before the (immediate) argument that the axioms do not make the instance stronger. Nevertheless, it is expected that these three set of axioms do not make the statement stronger and this could be proved formally by substituting every x_g with the original subcircuit rooted at gate g and computing each unary-bit as a polynomial over the original variables $\overline{x}$ of the circuit $C(\overline{x})$. This is expected to be efficiently provable because the field is finite. However, we shall simply add these extension axioms as part of the formalisation, without proving that they do not make the instance stronger.[11]

We shall work with polynomial degrees IPS refutations. Thus, in what follows, we let $d : \mathbb{N} \to \mathbb{N}$ be a (monotone) *size function* $d(t) = \text{poly}(t)$.

The following formulas are the ones we use:

- $\varphi^{\text{cnf}}_{r,m,d} := \mathsf{cnf}\,(\text{“}\mathsf{VNP} = \mathsf{VP}(r, m, d(r))\text{”})$.

  That is, the plain CNF encoding (Definition 3.3) of the circuit equation “$\mathsf{VNP} = \mathsf{VP}(r, m, d(r))$” (Definition 3.15) over the field $\mathbb{F}_q$ expressing that there are size r and syntactic-degree $d(r)$ circuits for the permanent of dimension m over $\mathbb{F}_q$.

- $\varphi^{\text{ecnf}}_{r,m,d} := \mathsf{ecnf}\,(\text{“}\mathsf{VNP} = \mathsf{VP}(r, m, d(r))\text{”})$.

  That is, $\varphi^{\text{cnf}}_{r,m,d}$ together with all the algebraic extension axioms (Definition 3.1) for each gate g in “$\mathsf{VNP} = \mathsf{VP}(r, m, d(r))$”.

- $\varphi^{\star}_{r,m,d} := \varphi^{\text{ecnf}}_{r,m,d} \cup E$, for E a set of SLP equations and algebraic extension axioms for some circuits (defined precisely in Definition 3.20 below).

- $\Phi_{t,r,m,d} := \mathsf{cnf}\left(\text{IPS}_{ref}\left(t, d, \varphi^{\star}_{r,m,d}\right)\right)$.

  That is, the CNF formula expressing that IPS refutes $\varphi^{\star}_{r,m,d}$ in size t and degree $d(t)$ over $\mathbb{F}_q$, according to Definition 3.17.

Theorem 3.18 (Main)**.** $\mathsf{VP} \neq \mathsf{VNP}$ *over* $\mathbb{F}_q$ *iff the CNF family of formulas* $\{\Phi_{t,r,m,d}\}$ *do not have polynomial-size* IPS *refutations infinitely often in the following sense: there exists a constant* c_1*, such that for every sufficiently large constant* c_0*, and every constant* $c_2 > c_1$*, for infinitely many* $t, r, m \in \mathbb{N}$*, if* $t > |\varphi^{\star}_{r,m,d}|^{c_1}$ *and* $m^{c_1} < r < m^{c_2}$*,* $\Phi_{t,r,m,d}$ *has no* IPS *refutations of size at most* $|\Phi_{t,r,m,d}|^{c_0}$*.*

[11] We explain here the argument that is expected to be applicable to the three helper axioms (i)–(iii) above in order to show the axioms do not add strength to the statement “VP=VNP”. First, using the Boolean axioms, we can derive the field axioms $\prod_{j \in \mathbb{F}_q}(x - j)$, for every variable x. Using this axiom, instead of (i), we derive efficiently the equation $x_g = \sum_{j=0}^{q-1} j \cdot \text{UBit}_j(x_g)$, where $\text{UBit}_j(x_g)$ is the polynomial that equals 1 iff $x_g = j$, for $j \in \{0, \ldots, q-1\}$. This polynomial equation has a constant size IPS derivation because it has a constant number of variables and it is implied (over $\mathbb{F}_q$) from the field axioms. Similarly, instead of (ii) we use the Boolean statement $\text{UBit}_j(x_g)^2 - \text{UBit}_j(x_g) = 0$, for every x_g and i, which also has a small derivation. Instead of (iii) we can similarly efficiently derive $\sum_{i=0}^{q-1} \text{UBit}_j(x_g) = 1$ for every x_g. This sketch should then show that over finite fields IPS can efficiently derive $\mathsf{cnf}\,(C(\overline{x}) = 0)$ from $C(\overline{x}) = 0$, without the helper axioms we supply. However, we have not verified all the details.

Notice that every infinite set of $(t,r,m) \in \mathbb{N}^3$, for which $t > |\varphi^\star_{r,m,d}|^{c_1}$ and $m^{c_1} < r < m^{c_2}$, must contain an infinite set $I \subseteq \mathbb{N}$ of increasing distinct triples $\{(t_i, r_i, m_i)\}_{i\in I}$, in the sense that for all $i \in I$, $t_{i+1} > t_i, r_{i+1} > r_i$ and $m_{i+1} > m_i$. Accordingly, (the $\exists\forall$-statement of) Theorem 3.18 establishes that there exists a polynomial $p'(t) = t^{c_1}$ (standing for the IPS lower bound stated in $\Phi_{t,r,m,d}$), and a polynomial $p''(m) = m^{c_2}$ (standing for the upper bound on the size of the permanent for matrices of dimension m), such that for every polynomial $p(N) = N^{c_0}$, infinitely often there is no IPS refutation of $\Phi_{t,r,m,d}$ of size bounded by $p(|\Phi_{t,r,m,d}|)$.

Proof of direction ($\Leftarrow$) of Theorem 3.18. This is the easier direction. Let $c_1 < c_2$ be constants and denote by $\mathscr{F}_{c_1,c_2}$ the CNF formulas $\{\Phi_{t,r,m,d} : t,r,m,d \in \mathbb{N}$ such that $t > |\varphi^\star_{r,m,d}|^{c_1}$ and $m^{c_1} < r < m^{c_2}\}$.
Case 1: There exists a pair of constants $c_1 < c_2$ such that $\mathscr{F}_{c_1,c_2}$ is an *unsatisfiable* CNF family almost everywhere. Thus, by assumption, $\mathscr{F}_{c_1,c_2}$ has no polynomial-size IPS refutations of the unsatisfiable instances $\Phi_{t,r,m,d}$, infinitely often. By Theorem 2.4, $\mathsf{VNP} \neq \mathsf{VP}$ over $\mathbb{F}_q$ (note that the result of [GP18] holds also for lower bounds on IPS *without* the boolean axioms).
Case 2: Otherwise, for every pair of constants $c_1 < c_2$, $\mathscr{F}_{c_1,c_2}$ is a family of CNF formulas that is infinitely often *satisfiable*. Thus, by Proposition 3.7 the corresponding set of *polynomial equations* (in contrast to merely the CNF formulas) $\mathrm{IPS}_{ref}(t,d,\varphi_{r,m,d})$ are *satisfiable over* $\mathbb{F}_q$ infinitely often, for every pair of constants $c_1 < c_2$. Hence, for every constant c_1 there is an IPS refutation of $\varphi_{r,m,d}$ infinitely often. By soundness of IPS we get that infinitely often there is no circuit of size at most m^{c_1} for the Permanent of dimension m, and hence $\mathsf{VP} \neq \mathsf{VNP}$. □

3.2.1 Proof of direction ($\Rightarrow$) of Theorem 3.18

We work over $\mathbb{F}_q$.

We assume that:

Assumption 1. $\mathsf{VP} \neq \mathsf{VNP}$ *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 prove that:

Conclusion 1. *There exists a constant c_1 such that for every constant c_0 and every constant $c_2 > c_1$, for infinitely many $t,r,m \in \mathbb{N}$ with $t > |\varphi^\star_{r,m,d}|^{c_1}$ and $m^{c_1} < r < m^{c_2}$, the following CNF*

$$\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}\left(t,d,\varphi^\star_{r,m,d}\right)\right) \tag{16}$$

has no IPS refutations of size at most $\left|\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}\left(t,d,\varphi^\star_{r,m,d}\right)\right)\right|^{c_0}$.

We are going to describe the general argument first, and then prove separately Lemma 3.21 which is a sort of formalisation of Grochow-Pitassi result within IPS, in the sequel.

The main diagonalisation argument. Suppose that $\mathsf{VP} \neq \mathsf{VNP}$, namely Assumption 1 above holds. Let c_3 be some fixed constant (that is derived from the polynomial-size proof in Proposition 3.22). Recall that "$\left|\frac{s,\ d}{\mathrm{IPS}}\right.$" means size s and degree d IPS proofs. Let *a.e.* stand for "almost everywhere", namely, everywhere except for a finite many cases.

Suppose by way of contradiction the converse of Conclusion 1, that is: for every constant c_1, there exist a constant c_0 and a constant $c_2 > c_1$, such that a.e. for $t, r, m \in \mathbb{N}$ where $m^{c_1} < r < m^{c_2}$ and $t > |\varphi^{\star}_{r,m,d}|^{c_1}$, the following holds:

$$\underbrace{\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \overbrace{\varphi^{\star}_{r,m,d}}^{\sigma})\right)}_{\lambda} \vdash^{|\lambda|^{c_0},\ d(|\lambda|)}_{\mathrm{IPS}} 1 = 0, \tag{17}$$

and $\varphi^{\star}_{r,m,d}$ is unsatisfiable a.e. over $\mathbb{F}_q$ by Assumption 1 (since $r < m^{c_2}$).

We use the following notation

$$\gamma = \varphi^{\star}_{t,k,d} \quad \text{and} \quad k = 6(\ell + N'), \tag{18}$$

where ℓ, N' are the number of variables and clauses in $\varphi^{\star}_{r,m,d}$, respectively, and assume that

$$t > k^{c''}$$

for a constant c'' that we pick big enough so that

$$|\gamma|^{c_3} + |\lambda|^{c_0} = |\gamma|^{c_3} + |\gamma|^{c' \cdot c_0} < |\gamma|^{c''} \tag{19}$$

(we use here that $|\lambda| = |\gamma|^{c'}$ for some constant c', by Equation (12) and Equation (15)); and moreover, as before, $\varphi^{\star}_{t,k,d}$ is $\varphi^{\mathrm{ecnf}}_{t,k,d}$ augmented 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 in Definition 3.20.

In Lemma 3.21 in the sequel we construct an $\mathrm{IPS}^{\mathrm{alg}}$ derivation of $\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \varphi^{\star}_{r,m,d})\right)$ from $\varphi^{\star}_{t,k,d}$ of size at most $|\gamma|^{c_3}$ and degree at most $d(|\lambda|)$, assuming that $\varphi^{\mathrm{ecnf}}_{r,m,d}$ is unsatisfiable.

Therefore, by (17) we conclude that for every constant c_1, there exist a constant c_0 and a constant $c_2 > c_1$, such that a.e. for $t, r, m \in \mathbb{N}$ and where $m^{c_1} < r < m^{c_2}$ and $t > |\varphi^{\mathrm{ecnf}}_{r,m,d}|^{c_1}$, the following holds:

$$\overbrace{\varphi^{\star}_{t,k,d}}^{\gamma} \vdash^{|\gamma|^{c_3},\ d(|\gamma|)}_{\mathrm{IPS}^{\mathrm{alg}}} \underbrace{\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \overbrace{\varphi^{\star}_{r,m,d}}^{\sigma})\right)}_{\lambda} \vdash^{|\lambda|^{c_0},\ d(|\lambda|)}_{\mathrm{IPS}} 1 = 0. \tag{20}$$

We now combine the two proofs into one. Namely, by Equation (20), for every constant c_1, there exist constants c_0, c_3 and $c_2 > c_1$, such that a.e. for $t, r, m \in \mathbb{N}$ and where $m^{c_1} < r < m^{c_2}$ and $t > |\varphi^{\star}_{r,m,d}|^{c_1}$, the following holds:

$$\overbrace{\varphi^{\star}_{t,k,d}}^{\gamma} \vdash^{|\gamma|^{c_3} + |\lambda|^{c_0},\ d(|\lambda|)}_{\mathrm{IPS}} 1 = 0. \tag{21}$$

Using (19), for every constant c_1, there exist a constant c'' and a constant $c_2 > c_1$, such that a.e. for $t, r, m \in \mathbb{N}$ and where $m^{c_1} < r < m^{c_2}$ and $t > |\varphi^{\star}_{r,m,d}|^{c_1}$, the following holds:

$$\overbrace{\varphi^{\star}_{t,k,d}}^{\gamma} \vdash^{|\gamma|^{c''},\ d(|\lambda|)}_{\mathrm{IPS}} 1 = 0. \tag{22}$$

We now obtained a contradiction, because this is a short refutation of an upper bound for the Permanent polynomial. More precisely, fix a pair of constants $c_1 < c''$, for which (22) holds. Thus, there exists a constant c'' such that for every constant $c_2' > c''$, a.e. for $w, t, k \in \mathbb{N}$, where $w > |\varphi^{\star}_{t,k,d}|^{c''}$ and $k^{c_1} < k^{c''} < t$ (and in particular, for $k^{c''} < t < k^{c_2'}$), the following holds:

$$\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(w, d, \overbrace{\varphi^{\star}_{t,k,d}}^{\gamma})\right) \not\vdash_{\mathrm{IPS}} 1 = 0\,. \tag{23}$$

In other words, there exists a constant c'' such that for every constant c_0' and every constant $c_2' > c''$, a.e. for $w, t, k \in \mathbb{N}$, where $w > |\varphi^{\star}_{t,k,d}|^{c''}$ and $k^{c_1} < k^{c''} < t$ (and in particular, for $k^{c''} < t < k^{c_2'}$), the following holds:

$$\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(w, d, \overbrace{\varphi^{\star}_{t,k,d}}^{\gamma})\right) \not\vdash_{\mathrm{IPS}} 1 = 0 \tag{24}$$

namely, $\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(w, d, \overbrace{\varphi^{\star}_{t,k,d}}^{\gamma})\right) \not\vdash^{c_0'}_{\mathrm{IPS}} 1 = 0\,.$

But this contradicts (17), concluding the main argument. $\square_{\text{main argument}}$

Remark 3.19. *1. Our lower bound in Conclusion 1 holds for all $t > k^{c_1}$. Note that as $t > k^{c_1}$ becomes bigger refuting $\mathrm{IPS}_{ref}(t, d, \varphi^{\star}_{r,m,d})$ becomes even* harder*, hence if there are no* IPS *refutations of $\mathrm{IPS}_{ref}(t, d, \varphi^{\star}_{r,m,d})$ of size $|\lambda|^{c_0}$ then there are no such refutations also for bigger t's.*

2. Our lower bound in Conclusion 1 holds for all $m^{c_1} < r < m^{c_2}$. Note that as $r > m^{c_1}$ becomes bigger it becomes harder *to refute $\varphi^{\star}_{r,m,d}$ (as this would establish a* stronger *lower bound). Hence, the lower bound against $\mathrm{IPS}_{ref}(t, d, \varphi^{\star}_{r,m,d})$ becomes* easier *to prove. On the other hand, once $r > m^{c_1}$ becomes too big, namely exceeds m^{c_2}, $\varphi^{\star}_{r,m,d}$ may become* satisfiable*, hence, refuting $\mathrm{IPS}_{ref}(t, d, \varphi^{\star}_{r,m,d})$ becomes easier: since $\varphi^{\star}_{r,m,d}$ has a satisfying assignment, we can use this assignment to refute the existence of a refutation of $\varphi^{\star}_{r,m,d}$.*

It remains to construct an $\mathrm{IPS}^{\mathrm{alg}}$ derivation of $\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \varphi^{\star}_{r,m,d})$ from $\varphi^{\star}_{t,k,d}$ of size at most $|\varphi^{\star}_{t,k,d}|^{c_3}$. We first define precisely $\varphi^{\star}_{t,k,d}$. For this, we consider a fixed constant c_1, and the corresponding fixed constants $c_0, c_2 > c_1$, for which (17) holds, namely, a.e. there is an IPS refutation of $\Phi_{t,r,m,d}$ of size bounded by $|\Phi_{t,r,m,d}|^{c_0}$. We can fix for simplicity all sizes to depend on t, that is, $k(t), r(t), m(t) : \mathbb{N} \to \mathbb{N}$ are functions such that $m(t)^{c_1} < r(t) < m(t)^{c_2}$ and $k(t) = 6(\ell + N')$, where ℓ, N' are the number of variables and clauses in $\varphi^{\star}_{r(t),m(t),d(r)}$, respectively (see Equation (18)). Note that *for $t > |\varphi^{\star}_{r(t),m(t),d(t)}|^{c_1}$ to hold we must have $r = O(\log(t))$* because the size of $\varphi^{\star}_{r(t),m(t),d(t)}$ is greater than that of $\varphi^{\mathrm{ecnf}}_{r(t),m(t),d(t)}$, which is exponential in $m(t)$, and $m(t)$ is polynomial in $r(t)$ (see (12)).

Define the sequence t_i as follows:

$$t_i := \begin{cases} t & , \ i = 1; \\ r(t_{i-1}) & , \ \text{otherwise}, \end{cases}$$

and let $\xi = \min\{i \;:\; t_i < 1000\}$ (where 1000 was picked as an arbitrary big constant). Then, $\log^{(\xi)}(t) < 1000$ (where $\log^{(\xi)}(t)$, means the ξth logarithmic of t). Consider the family $R_t := \{\mathrm{IPS}^{\mathrm{alg}}_{ref}(t_i, d(t_i), \varphi^{\star}_{r(t_i),m(t_i),d(r(t_i))})\}_{t_i=1}^{\xi}$. Note that the total size of all circuits in this family is polynomial in $|\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d(t), \varphi^{\star}_{r(t),m(t),d(r)})|$ because we added only logarithmic in t many new smaller than $|\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d(t), \varphi^{\star}_{r(t),m(t),d(r)})|$ instances. When writing $\varphi^{\star}_{t,k,d}$ we shall omit the dependence on t of k and d, from now on.

Definition 3.20 ($\varphi^{\star}_{t,k,d}$)**.** *Define $\varphi^{\star}_{t,k,d}$ to be $\varphi^{\mathrm{ecnf}}_{t,k,d}$ (which is the extended CNF of "$\mathsf{VNP} = \mathsf{VP}(t, k, d(t))$") together with the following additional extension axioms in new variables (that are needed to invoke Proposition 3.6 in what follows):*

extension axioms: *for every circuit (*not *circuit equation) of the form* $\mathrm{Coeff}_{M_i}(U(\overline{x}, \overline{w}))$*, abbreviated D, from Equation (11) in Definition 3.17 corresponding to a circuit in R_t defined above, we have the following:*

1. *the corresponding SLPs equations (Definition 3.5) between extension variables for the gates in the circuit D;*
2. *the algebraic extension axioms (Definition 3.1) for every gate g in the circuit D;*
3. *the Boolean axioms $x_{gi}^2 - x_{gi} = 0$, for $i \in \{0, \ldots, q-1\}$, for every gate g in the circuit D;*
4. *the axiom stating only one unary bit is true for every gate g in D: $\sum_{i=0}^{q-1} x_{gi} = 1$.*

Note that the extension axioms in $\varphi^{\star}_{t,k,d}$ do not add the full statement $\mathrm{IPS}^{\mathrm{alg}}_{ref}(t_i, d(t_i), \varphi^{\mathrm{ecnf}}_{r(t_i),m(t_i),d(r(t_i))})$, because for every circuit $\mathrm{Coeff}_{M_i}(U(\overline{x}, \overline{w}))$, we only add extension axioms that refer to the gates of this circuit, but we do not state that the circuit is equal to any specific value. The size of $\varphi^{\star}_{t,k,d}$ is $\log t \cdot t^4 \cdot |\overline{\mathcal{F}}| \cdot 2^{O(k^2+d)}$ by Equation (15). It is worth recalling that all our proofs use only the $\overline{w}$-variables because every polynomial in the $\overline{x}$-variables will eventually be fed into a $\mathrm{Coeff}_M(\cdot)$ operator for M an $\overline{x}$-monomial, and this operator outputs polynomials in the $\overline{w}$-variables.

Lemma 3.21. *There is a constant c_3, such that if $\varphi_{r,m,d}$ is unsatisfiable, then under the above notation and parameters:*

$$\overbrace{\varphi^{\star}_{t,k,d}}^{\gamma} \;\Big|^{\frac{|\gamma|^{c_3},\ d(|\gamma|)}{\mathrm{IPS}^{\mathrm{alg}}}}\; \mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, \overbrace{\varphi^{\star}_{r,m,d}}^{\sigma})\right).$$

Proof: We start with $\gamma = \varphi^{\star}_{t,k,d}$ 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 Proposition 3.22) of the equations "$\mathsf{VNP} = \mathsf{VP}(t, k, d(t))$" expressing that there exists a circuit of size t and syntactic-degree $d(t)$ computing the permanent of dimension k over $\mathbb{F}_q$.

The following is the crux of the argument: since the permanent polynomial $\mathrm{perm}(\overline{x})$ of dimension k is complete for VNP with $k/6 = \ell + N'$ variables [Val79b] (when the characteristic of the field is different from 2), and since every unsatisfiable CNF has an IPS refutation whose certificate is computable in VNP by [GP18], and similarly every *extended* CNF formula has an $\mathrm{IPS}^{\mathrm{alg}}$ refutation that is computable in VNP by Corollary 3.8, there exists an assignment $\overline{a}$ of field $\mathbb{F}_q$ elements and $\overline{x}$ variables to $\overline{x}$, with $\overline{x}$ having k number of variables, such that $\mathrm{perm}(\overline{x} \restriction \overline{a})$ is the $\mathrm{IPS}^{\mathrm{alg}}$ refutation of the extended CNF $\varphi^{\star}_{r,m,d}$. Formally we have the following (note that any $\mathrm{IPS}^{\mathrm{alg}}$ derivation is also an IPS derivation, hence the formalisation is doable also in IPS):

Proposition 3.22 (Grochow-Pitassi formalization in $\mathrm{IPS}^{\mathrm{alg}}$)**.** *There is an* $\mathrm{IPS}^{\mathrm{alg}}$ *derivation from* $\varphi^{\star}_{t,k,d}$*, denoted* γ*, of* $\mathrm{IPS}^{\mathrm{alg}}_{ref}(t,d,\varphi^{\star}_{r,m,d})$*, with size at most* $|\gamma|^b$*, for some constant* b*.*

Proof of Proposition 3.22. It is sufficient to derive $\mathrm{IPS}^{\mathrm{alg}}_{ref}(t,d,\varphi^{\mathrm{ecnf}}_{r,m,d})$ instead of $\mathrm{IPS}^{\mathrm{alg}}_{ref}(t,d,\varphi^{\star}_{r,m,d})$. This is because $\varphi^{\mathrm{ecnf}}_{r,m,d} \subseteq \varphi^{\star}_{r,m,d}$, so every refutation of $\varphi^{\mathrm{ecnf}}_{r,m,d}$ is also a refutation $\varphi^{\star}_{r,m,d}$ (hence the the same argument, shown below, that applies to the former applies to the latter as well).

First, use Proposition 3.6 to derive from the extended CNF $\varphi^{\star}_{t,k,d}$ the corresponding circuit equations "$\mathsf{VNP} = \mathsf{VP}(t,k,d(t))$". The latter is formulated as a set of circuit equations expressing that the coefficients of the monomials of a universal circuit of size t are the coefficients of $\mathrm{perm}(\overline{x})$ of dimension k (see Definition 3.15).

Consider $\overline{a} : \overline{x} \to \overline{x} \cup \mathbb{F}_q$ to be the assignment of variables and field elements to the k $\overline{x}$-variables that results in $\mathrm{perm}(\overline{x} \restriction \overline{a})$ being a (correct) $\mathrm{IPS}^{\mathrm{alg}}$ refutation of $\varphi^{\mathrm{ecnf}}_{r,m,d}$ (such an assignment $\overline{a}$ exists as discussed above). It is left to efficiently derive in $\mathrm{IPS}^{\mathrm{alg}}$ from "$\mathsf{VNP} = \mathsf{VP}(t,k,d(t))$" the equations of $\mathrm{IPS}^{\mathrm{alg}}_{ref}(t,d,\varphi^{\mathrm{ecnf}}_{r,m,d})$. The latter is formulated by Definition 3.17 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 $\varphi^{\mathrm{ecnf}}_{r,m,d}$.

The idea is to use the equations of the form $\mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w})) = b_i$ from Equation (11) in "$\mathsf{VNP} = \mathsf{VP}(t,k,d(t))$", and derive the equations that express that the coefficients of $\mathrm{perm}(\overline{x} \restriction \overline{a})$ are precisely those of a polynomial that constitutes an $\mathrm{IPS}^{\mathrm{alg}}$ refutation of the extended CNF $\varphi^{\mathrm{ecnf}}_{r,m,d}$. Since we know that $\mathrm{perm}(\overline{x} \restriction \overline{a})$ is a correct $\mathrm{IPS}^{\mathrm{alg}}$ refutation of $\varphi^{\mathrm{ecnf}}_{r,m,d}$, the equations for all coefficients will satisfy the conditions of Definition 3.17. Note that we will use three different assignments: first, the assignment $\overline{a}$ that will make $\mathrm{perm}(\overline{x} \restriction \overline{a})$ an $\mathrm{IPS}^{\mathrm{alg}}$ refutation in the variables $\overline{x}$. We then compose separately $\overline{a}$ with two different assignments: the assignment of all zeros to those $\overline{x}$-variables that are identified as the IPS "placeholder" variables (denoted $\overline{y}$), and the assignment of the clauses of $\varphi^{\mathrm{ecnf}}_{r,m,d}$ to these placeholder variables. The size bound on the $\mathrm{IPS}^{\mathrm{alg}}$ proofs follows from the construction, which concludes Proposition 3.22 (for some constant ℓ).

We first assert some polynomial identities. The following are polynomial identities, where 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 d (as in Definition 3.15):

$$U(\overline{x} \restriction \overline{a}, \overline{w}) = \sum_{i \in [N]} \mathrm{Coeff}_{M_i}(U(\overline{x} \restriction \overline{a}, \overline{w})) \cdot M_i \tag{25}$$

$$= \left(\sum_{i \in [N]} \mathrm{Coeff}_{M_i}(U(\overline{x}, \overline{w})) \cdot M_i \right) \restriction \overline{a} \tag{26}$$

$$= \sum_{i \in [N]} \mathrm{Coeff}_{M_i}(U(\overline{x}, \overline{w})) \cdot (M_i \restriction \overline{a}) \,. \tag{27}$$

Equation (25) holds by Definition 3.11 and Proposition 3.12, and Equation (26) holds because if we assign to U and represent it as a sum of monomials or else we represent U as a sum of monomials and then assign to it, we get the same polynomial. And Equation (27) is immediate because by construction $\mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w}))$ (for some $\overline{x}$-monomial M_i) is a circuit that contains only the $\overline{w}$-variables. By Equation (27) we have the following polynomial identity for every $\overline{x}$-monomial M:

$$\mathrm{Coeff}_{M}(U(\overline{x} \restriction \overline{a}, \overline{w})) = \mathrm{Coeff}_{M} \left(\sum_{i \in [N]} \mathrm{Coeff}_{M_i}(U(\overline{x}, \overline{w})) \cdot (M_i \restriction \overline{a}) \right) . \tag{28}$$

Since $\overline{b}$ is the coefficient vector of the permanent polynomial, we have also the following polynomial identity:

$$\sum_{i\in[N]} b_i \cdot (M_i \restriction \overline{a}) = \mathrm{perm}(\overline{x} \restriction \overline{a}). \tag{29}$$

Recall that $\mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w}))$ (for some $\overline{x}$-monomial M_i) contains only the $\overline{w}$-variables. Let M be an $\overline{x}$-monomial. Thus, by inspection of the construction of the circuit $\mathrm{Coeff}_M(\cdot)$ in Proposition 3.12, $\mathrm{Coeff}_M(\mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w}) \cdot (M_i \restriction \overline{a}))) = \mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w})) \cdot \mathrm{Coeff}_M(M_i \restriction \overline{a})$. This implies the following polynomial identity:

$$\mathrm{Coeff}_M\left(\sum_{i\in[N]} \mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w})) \cdot (M_i \restriction \overline{a})\right) = \sum_{i\in[N]} \mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w})) \cdot \mathrm{Coeff}_M(M_i \restriction \overline{a})). \tag{30}$$

By Fact 2.6 polynomial identities are proved in IPS (and similarly in $\mathrm{IPS}^{\mathrm{alg}}$) with a size that is equivalent to the size of the circuits appearing in the equations. Therefore, in $\mathrm{IPS}^{\mathrm{alg}}$ we can prove Equation (30) and then use substitutions of $\mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w}))$ for all i, by b_i, using the axioms $\mathrm{Coeff}_{M_i}(U(\overline{x},\overline{w})) = b_i$. Thus, from Equation (28), Equation (29) and Equation (30) we get in $\mathrm{IPS}^{\mathrm{alg}}$

$$\mathrm{Coeff}_M(U(\overline{x} \restriction \overline{a},\overline{w})) = \mathrm{Coeff}_M\left(\sum_{i\in[N]} b_i \cdot (M_i \restriction \overline{a})\right) = \mathrm{Coeff}_M(\mathrm{perm}(\overline{x} \restriction \overline{a})), \tag{31}$$

where the left equation is by substitutions in IPS, and the right equation is a polynomial identity.

By assumption, $\overline{a}$ was chosen such that $\mathrm{perm}(\overline{x} \restriction \overline{a})$ is an $\mathrm{IPS}^{\mathrm{alg}}$ refutation of $\varphi^{\mathrm{ecnf}}_{r,m,d}$. This in particular means that for the sake of clarity we can distinguish in $\overline{x} \restriction \overline{a}$ two sets of variables: the "placeholder" variables (denoted $\overline{y}$ in Definition 2.2) and the "original" variables (denoted $\overline{x}$ in Definition 2.2). We do not need to use the placeholder variables $\overline{z}$ as in Definition 2.2 because we use $\mathrm{IPS}^{\mathrm{alg}}$ which does not use the boolean axioms (though some boolean axioms appear independently in $\varphi^{\mathrm{ecnf}}_{r,m,d}$). Let us partition $\overline{x}$ into $\overline{x}' \uplus \overline{y}$ whereas we denote those $\overline{x}$-variables in $\mathrm{perm}(\overline{x} \restriction \overline{a})$ that correspond to the placeholder variables by $\overline{y}$, and all the rest of the $\overline{x}$-variables in $\mathrm{perm}(\overline{x} \restriction \overline{a})$ (and similarly in $\mathrm{perm}(\overline{x})$ and in $U(\overline{x},\overline{w})$) by $\overline{x}'$. Note that as an $\mathrm{IPS}^{\mathrm{alg}}$ refutation, $\mathrm{perm}(\overline{x} \restriction \overline{a})$, or $\mathrm{perm}(\overline{x}' \restriction \overline{a}, \overline{y} \restriction \overline{a})$ in the new notation, must have all placeholder variables $\overline{y}$ *unassigned*, hence $\overline{a}$ leaves $\overline{y}$ intact, meaning that $\mathrm{perm}(\overline{x}' \restriction \overline{a}, \overline{y} \restriction \overline{a}) = \mathrm{perm}(\overline{x}' \restriction \overline{a}, \overline{y})$ is the $\mathrm{IPS}^{\mathrm{alg}}$ refutation of $\varphi^{\mathrm{ecnf}}_{r,m,d}$.

Denote by $\restriction \overline{y} = \mathbf{0}$ the restriction that assigns 0 to all $\overline{y}$ variables, and by $\restriction \overline{y} = \varphi^{\mathrm{ecnf}}_{r,m,d}$ the restriction that assigns to each variable y_i the corresponding ith equation from $\varphi^{\mathrm{ecnf}}_{r,m,d}$ (equation $f = g$ here means the term $f - g$). Denote by $\restriction \overline{a} \cup \overline{y} = \mathbf{0}$ the restriction that first assigns $\overline{a}$ to the $\overline{x}'$-variables and then assigns zeros to the $\overline{y}$-variables (since $\overline{x}'$ and $\overline{y}$ are disjoint the order of assignments is unimportant). Similarly, denote by $\restriction \overline{a} \cup \overline{y} = \varphi^{\mathrm{ecnf}}_{r,m,d}$ the restriction that first assigns $\overline{a}$ to the $\overline{x}'$-variables and then assigns to each variable y_i the corresponding ith equation from $\varphi^{\mathrm{ecnf}}_{r,m,d}$.

In the same way we proved Equation (31) in $\mathrm{IPS}^{\mathrm{alg}}$ using the assignment $\overline{a}$ we can prove with a linear size proof the following, for every $\overline{x}$-monomial M and for the assignment $\overline{a} \cup \overline{y} = \mathbf{0}$:

$$\begin{aligned}
\mathrm{Coeff}_M(U(\overline{x} \restriction \overline{a} \cup \overline{y} = \mathbf{0}, \overline{w})) &= \mathrm{Coeff}_M(U(\overline{x}' \restriction \overline{a}, \mathbf{0}, \overline{w})) && (\text{by } \overline{x} = \overline{x}' \uplus \overline{y}) \\
&= \mathrm{Coeff}_M\left(\sum_{i\in[N]} b_i \cdot (M_i \restriction \overline{a} \cup \overline{y} = \mathbf{0})\right) && (\text{similar to Equation (31)}) \\
&= \mathrm{Coeff}_M(\mathrm{perm}(\overline{x} \restriction \overline{a} \cup \overline{y} = \mathbf{0})) \\
&= \mathrm{Coeff}_M(\mathrm{perm}(\overline{x}' \restriction \overline{a}, \overline{y} \restriction \mathbf{0})) = 0\,,
\end{aligned} \tag{32}$$

where Equation (32) holds because by assumption $\mathrm{perm}(\overline{x} \restriction \overline{a})$ is a correct $\mathrm{IPS}^{\mathrm{alg}}$ refutation of $\varphi^{\mathrm{ecnf}}_{r,m,d}$ which means that every nonzero monomial in it is a product of some placeholder $\overline{y}$-variables, and we assign zeros to the $\overline{y}$-variables. In particular Equation (32) holds for every M monomial in the $\overline{x}'$-variables, which are precisely the desired first set of equations in $\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \varphi^{\mathrm{ecnf}}_{r,m,d})$ (corresponding to Equation (13) in Definition 3.17).

Similarly, we can consider the assignment $\overline{a} \cup \overline{y} = \varphi^{\mathrm{ecnf}}_{r,m,d}$. Here we use the fact that $\varphi^{\star}_{t,k,d}$ provides all the extension axioms for gates in the circuits of $\varphi^{\mathrm{ecnf}}_{r,m,d}$, so that we can write $\varphi^{\mathrm{ecnf}}_{r,m,d}$ with SLPs (note that $\varphi^{\mathrm{ecnf}}_{r,m,d}$ is a set of polynomials in the $\overline{x}'$-variables). Hence, similarly as above, we get a proof in $\mathrm{IPS}^{\mathrm{alg}}$ of:

$$\begin{aligned}\mathrm{Coeff}_M(U(\overline{x} \restriction \overline{a} \cup \overline{y} = \varphi^{\mathrm{ecnf}}_{r,m,d}, \overline{w})) &= \mathrm{Coeff}_M(\mathrm{perm}(\overline{x}' \restriction \overline{a}, \overline{y} \restriction \varphi^{\mathrm{ecnf}}_{r,m,d})) \\ &= \begin{cases} 1, & M = 1 \quad \text{(i.e., the constant 1 monomial);} \\ 0, & \text{all other } \overline{x}'\text{-monomials } M\,. \end{cases}\end{aligned} \tag{33}$$

It remains to inspect that the size of all the circuits in the $\mathrm{IPS}^{\mathrm{alg}}$ proofs above are bounded by a polynomial in $\gamma = |\varphi^{\star}_{t,k,d}| = t^4 \cdot 2^{O(k^2+d)}$. For this notice that when we apply $\mathrm{Coeff}_M(\cdot)$ with M of degree d on a polynomial of size $t^4 \cdot 2^{O(k^2+d)}$ as in Equation (30) we get by Proposition 3.12 a size of $7^d \cdot t^4 \cdot 2^{O(k^2+d)} = t^4 \cdot 2^{O(k^2+d)}$. This concludes the proof of Proposition 3.22. □

We are now in a position to complete the proof of Lemma 3.21, by showing there is a constant c_3 such that:

$$\varphi^{\star}_{t,k,d} \left|\frac{|\gamma|^{c_3},\ d(|\gamma|)}{\mathrm{IPS}^{\mathrm{alg}}}\right. \mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \varphi^{\star}_{r,m,d})\right). \tag{34}$$

By Proposition 3.22, there is a constant ℓ such that:

$$\varphi^{\star}_{t,k,d} \left|\frac{|\gamma|^{\ell},\ d(|\gamma|)}{\mathrm{IPS}^{\mathrm{alg}}}\right. \mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \varphi^{\mathrm{ecnf}}_{r,m,d}). \tag{35}$$

By using Proposition 3.6 (in the other direction to that in the beginning of Proposition 3.22, namely, deriving an extended CNF formula from circuit equations) and the fact that we have extension axioms for all the subcircuits in $\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \varphi^{\mathrm{ecnf}}_{r,m,d})$ by definition of $\varphi^{\star}_{t,k,d}$ (this is where we use the helper extension axioms in $\varphi^{\star}_{t,k,d}$), we get

$$\varphi^{\star}_{t,k,d} \left|\frac{|\gamma|^{\ell'},\ d(|\gamma|)}{\mathrm{IPS}^{\mathrm{alg}}}\right. \mathsf{ecnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \varphi^{\star}_{r,m,d})\right), \tag{36}$$

for some constant ℓ'.

Since $\mathsf{cnf}\,(C = 0) \subset \mathsf{ecnf}\,(C = 0)$ for every (set of) circuits C, we get an $\mathrm{IPS}^{\mathrm{alg}}$ derivation of $\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \varphi^{\star}_{r,m,d})\right)$ from the right hand side of (36). Overall, we got a derivation of $\mathsf{cnf}\left(\mathrm{IPS}^{\mathrm{alg}}_{ref}(t, d, \varphi^{\star}_{r,m,d})\right)$ from $\varphi^{\star}_{t,k,d}$ with size at most $|\gamma^{c_3}|$, for some constant $c_3 \geq \ell + \ell'$. This concludes Lemma 3.21. □

This concludes the proof of the ($\Rightarrow$) direction of Theorem 3.18. □

Remark 3.23. *Note that the number of variables in* $\lambda = \mathrm{IPS}_{ref}(t, d, \overline{\mathcal{F}})$ *is* $K_{t,d} = O(dt^4)$ *(see Theorem 3.9), while* $|\lambda| = t^8 \cdot |\overline{\mathcal{F}}| \cdot 2^{O(n^2+d)}$ *(see (15)). Therefore, the trivial (exponential-size) IPS proof of* λ *has size* $2^{O(dt^4)}$*, which is exponentially bigger than* $|\lambda|$ *for any degree* d *we choose (for large enough* t*). Hence, our formulation* λ *is reasonable in the sense that it is not trivially an easy instance (and similarly for* $\mathsf{cnf}(\lambda)$*).*

We also have the following immediate corollary:

Corollary 3.24. *Theorem 3.18 holds for any proof system that simulates* IPS*.*

4 Candidate Hard Formulas for Every Propositional Proof System

While the previous section dealt with algebraic proof systems and used the specific properties of the IPS refutation systems, in this section we consider similar notions of diagonalisations for general propositional proof systems.

4.1 Iterated Lower Bound Formulas

For basic definitions on propositional proof complexity, please refer to the comprehensive overview of Krajíček [Kra19].

Definition 4.1. *Given propositional proof system* R*, propositional formula* ϕ *and size function* $s : \mathbb{N} \to \mathbb{N}$*,* $\mathsf{lb}_R(\lceil\phi\rceil, s)$ *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 $\mathrm{poly}(|\phi| + s)$ auxiliary variables encoding the computation of the relation R,[12] to verify that $y_1, \ldots, y_s$ does not constitute an R proof of ϕ. For a detailed description of how this encoding works, see [Pud20].

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

1. R *p-simulates* Res *(Resolution).*
2. R *is closed under partial assignments, i.e., there is a constant* k *such that for any integer* t *and formula* ϕ*, if there is a* R*-proof of* ϕ *of size* t*, and* $\overline{a}$ *is a partial truth assignment to the variables of* ϕ*, then there is a* R*-proof of* $\phi(\overline{a})$ *of size at most* t^k*.*
3. *There is a constant* k *such that for all formulas* ϕ *and integers* s*, the formula* $\psi = \mathsf{lb}_R(\phi, s) \vee \phi$ *has* R*-proofs of size at most* $|\psi|^k$*.*
4. R *is closed under modus ponens, i.e., there is a constant* k *such that for all integers* t, t' *and formulas* τ, ϕ, ψ*, if there is an* R*-proof of* $\tau \vee \neg\phi$ *of size* t *and an* R*-proof of* $\phi \vee \psi$ *of size* t'*, then there is an* R*-proof of* $\tau \vee \psi$ *of size at most* $(t + t')^k$*.*

The conditions above are satisfied for any standard strong enough propositional proof system, eg., Frege and Extended Frege [Kra19].

We now formally define iterated lower bound formulas relative to a propositional proof system R.

[12] A propositional proof system P can be interpreted as relation R where $R(x, y)$ holds iff y is a P-proof of the tautology encoded by x.

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

1. $\mathsf{lb}_R^0(\phi, s) = \phi$
2. $\mathsf{lb}_R^{k+1}(\phi, s) = \mathsf{lb}_R(\mathsf{lb}_R^k(\phi, s), s)$.

The meaning of the iterated lower bound formulas is thus that we consider a single fixed base hard formula ϕ, and we iterate the statement that expresses the lower bound against this base hard formula.

Lemma 4.4. *Let R be any reasonably strong propositional proof system, 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 $\neg\tau(\overline{a})$ by substituting the assignment $\overline{a}$ into τ Since R is reasonably strong, R can prove its own reflection principle efficiently, and since R is closed under partial assignments, it can also prove $\mathsf{lb}_R(\tau, s) \vee \tau(\overline{a})$, using the fact that the variables of τ do not occur in $\mathsf{lb}_R(\tau, s)$. 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. This dichotomy shows there are only two extreme cases—the highly interesting and the trivial one, with nothing in between: either the iterated lower bound formulas are *all* hard proof complexity instances, or *none* of them is (since when they do not have short proofs it follows from them have no proof at all).

Theorem 4.5 (Dichotomy for Iterated Lower Bound Formulas)**.** *Let R be a reasonably strong propositional proof system 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 tautology 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 4.4 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 Theorem 4.5 the *useful* case of the dichotomy, as this is the case that gives us hard tautologies for R.

Iterated Lower Bound Hypothesis: Let R be a reasonably strong propositional proof system 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 $s(|\phi_k|)$.

We observe that the Iterated Lower Bounds Hypothesis fails if *optimal proof systems* exist. Here an optimal proof system is a propositional proof system R such that for each propositional proof system Q there is a polynomial f such that for any tautology ϕ, if there are size t Q-proofs of ϕ, then there are size $f(t)$ R-proofs of ϕ. In other words, the R-proof size of any tautology ϕ is bounded by a polynomial in the Q-proof size of ϕ.

Proposition 4.6. *Suppose there exists an optimal proof system R that is reasonably strong. Then for any time-constructible super-polynomial function $s : \mathbb{N} \to \mathbb{N}$ and any formula ϕ, there is a positive integer k such that $\phi_k = \mathsf{lb}_R^k(\phi, s)$ is not a tautology.*

Proof: Suppose there exists an optimal proof system R that is reasonably strong. Assume, for the sake of contradiction, that there is a formula ϕ and a time-constructible super-polynomial function s such that $\phi_k = \mathsf{lb}_R^k(\phi, s)$ is a tautology for every positive integer k. Define the propositional proof system R' which is R plus the formulas ϕ_k added as axioms. Thus R' has polynomial-size (indeed size zero) proofs of ϕ_k for each k. R' is indeed a propositional proof system: it is complete since R is a propositional proof system, it is sound since the formulas ϕ_k are tautologies by assumption, and it is polynomially verifiable since R is polynomially verifiable and the sequence ϕ_k is polynomial-time computable by time-constructibility of s.

Since R is optimal, we have that R polynomially simulates R'. Since R' has constant-size proofs of each ϕ_k, this means that there are R-proofs of ϕ_k of size at most $|\phi_k|^c$ for each k, where c is a constant. Choose k large enough so that $s(|\phi_k|) > |\phi_k|^c$. Since ϕ_{k+1} is a tautology, we have that ϕ_k does not have proofs of size $s(|\phi_k|)$ and hence does not have proofs of size $|\phi_k|^c$, which is a contradiction. □

Recently, it was shown that Resolution is NP-hard to automate, by using a reduction based on proof complexity lower bound formulas [AM20, Gar19]. The result below follows from Theorem 2 in [Gar19].

Theorem 4.7 (Atserias-Muller [AM20], Garlik [Gar19]). *Let $t : \mathbb{N} \to \mathbb{N}$ be any super-polynomial function such that $t(n) = 2^{n^{o(1)}}$. There is a constant k such that for any large enough formula ϕ, if ϕ is a tautology, then $\tau = \mathsf{lb}_{\mathsf{Res}}(\phi, |\phi|^k)$ does not have* Res*-proofs of size $t(|\tau|)$.*

It follows by induction, since the base formula ϕ is hard, that the Iterated Lower Bounds Hypothesis holds at least for the relatively weak proof system Resolution:

Corollary 4.8. *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 $s(|\phi|)$ (e.g., 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* Res*-proofs of size $s(|\phi_k|)$.*

4.2 Iteration Preserves Hardness for Random Truth Table Formulas

The Iterated Lower Bounds hypothesis is about the hardness of a *fixed* propositional formula to which the proof complexity lower bound operator is applied an *unbounded* number of times. In this section, we give partial evidence in favour of the Hypothesis by considering a candidate hard distribution on formulas, and showing that under a reasonable hypothesis, applying the proof complexity lower bound operator a *fixed* number of times to a formula sampled from the distribution does not decrease hardness. The interesting aspect of this result is that the hardness holds for a *sufficiently strong* propositional proof system R rather than a relatively weak one such as Resolution.

Definition 4.9 (Truth Table Formulas [PS19]). *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 4.10 (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 4.11 (Distributional Iterated Lower Bound Formulas). *Let $\mathfrak{D} = \{D_N\}$ be a sequence of distributions, where D_N is supported on propositional formulas of size N. Given propositional proof system R, non-negative integer k and size function $s : \mathbb{N} \to \mathbb{N}$, the sequence of distributions $\mathsf{lb}^k_R(\mathfrak{D}, s)$ is defined via induction on k as follows:*

1. *$\mathsf{lb}^0_R(\mathfrak{D}, s) = \mathfrak{D}$;*
2. *$\mathsf{lb}^{k+1}_R(\mathfrak{D}, s)$ is the sequence of distributions on formulas $\mathsf{lb}_R(\phi, s)$ where ϕ is sampled from $\mathsf{lb}^k_R(\mathfrak{D}, s)$*

The following theorem strengthens Lemma 3 in [PS19], which corresponds to the case $k = 2$.

Theorem 4.12. *If Rudich's Conjecture holds, then there exist a propositional proof system 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 $|\mathsf{lb}^k_R(D_N, N^c)|^c$ 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 4.9).*

Proof: The proof is by induction on k. Let $\ell > 0$ be the constant in Conjecture 4.10 and let R be a propositional proof system and c be a large enough constant to be specified later.

We will establish the case $k = 0$ for *every* propositional proof system R, under Rudich's Conjecture. We have that $\mathsf{lb}_R^k(D_N, N^c) = D_N$. 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 $\overline{\mathsf{MCSP}}[n^\ell]$ asking whether a given truth table y of a boolean function f on n bits requires circuits of size greater than 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 4.10. If A_R accepted τ for even a $1/N$ fraction of truth-table tautologies τ, this would contradict the second item of Conjecture 4.10.

We define R to be the propositional proof system 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 not hard 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.

Case 1: 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 4.4, where c is greater than the exponent of the polynomial in Lemma 4.4. 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 an 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 paragraph. 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.

Case 2: 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 string in 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, as mentioned before, 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 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 $\mathsf{ttable}(f^{\mathsf{SAT}}_n, 2n^\ell)$ as an axiom.

These facts can be used to show by the inductive argument in the proof of Theorem 4.5 that the second item of Theorem 4.5 holds for the formulas $\psi_k = \mathsf{lb}^k_R(\psi, N^c)$ 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, $\mathsf{lb}^k_R(\psi, N^c)$ 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 $\mathsf{lb}^k_R(\mathsf{ttable}(f_n, n^\ell), N^c)$ 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. □

Acknowledgements

We wish to thank Jan Pich for very helpful discussions during the work on this paper. We also thank Jan Krajíček for helpful comments on an earlier draft of this work and anonymous reviewers of this work for very useful comments that improved the exposition. We further are greatly indebted to Jiaqi Lu for corrections of the manuscript.

References

[ABSRW04] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. Pseudorandom generators in propositional proof complexity. *SIAM J. Comput.*, 34(1):67–88 (electronic), 2004. 1

[AGHT24] Yaroslav Alekseev, Dima Grigoriev, Edward A. Hirsch, and Iddo Tzameret. Semialgebraic proofs, IPS lower bounds, and the τ-conjecture: Can a natural number be negative? *SIAM J. Comput.*, 53(3):648–700, 2024. 1, 1.3, 2.6

[Ajt88] Miklós Ajtai. The complexity of the pigeonhole principle. In *Proceedings of the IEEE 29th Annual Symposium on Foundations of Computer Science*, pages 346–355, 1988. 1

[AM20] Albert Atserias and Moritz Müller. Automating resolution is NP-hard. *J. ACM*, 67(5):31:1–31:17, 2020. 1.1.2, 1.1.2, 1.3, 4.1, 4.7

[BIK+96a] 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. 2.2, 2.2

[BIK+96b] Samuel R. Buss, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák, Alexander A. Razborov, and Jiří Sgall. Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Computational Complexity*, 6(3):256–298, 1996. 3.1.1

[CR79] Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979. 1

[Fri79] Harvey Friedman. On the consistency, completeness and correctness problems. Unpublished, 1979. 1.3

[FSTW21] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. Proof complexity lower bounds from algebraic circuit complexity. *Theory Comput.*, 17:1–88, 2021. 1, 1.3, 2.2, 3.1.1

[Gar19] Michal Garlík. Resolution lower bounds for refutation statements. In *44th Int. Symp. Math. Found. CS MFCS 2019*, volume 138 of *LIPIcs*, pages 37:1–37:13, 2019. 1.1.2, 1.1.2, 4.1, 4.7

[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. 1

[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. 1, 1.1.1, 1.2, 1.2, 1.2, 1.3, 2.2, 2.2, 2.4, 2.5, 3.1.1, 3.2, 3.2, 3.2.1

[Hak85] Armin Haken. The intractability of resolution. *Theoret. Comput. Sci.*, 39(2-3):297–308, 1985. 1

[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. 1

[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. 1

[Kra01] Jan Krajíček. Tautologies from pseudo-random generators. *Bull. Symbolic Logic*, 7(2):197–212, 2001. 1

[Kra04a] Jan Krajíček. Diagonalization in proof complexity. *Fundamenta Mathematicae*, 182:181–192, 2004. 1.3

[Kra04b] Jan Krajíček. Dual weak pigeonhole principles, pseudo-surjective functions, and provability of circuit lower bounds. *Journal of Symbolic Logic*, 69(1):265–286, 2004. 1.1.1

[Kra04c] Jan Krajíček. Implicit proofs. *Journal of Symbolic Logic*, 69(2):387–397, 2004. 1.3

[Kra11] Jan Krajíček. On the proof complexity of the Nisan-Wigderson generator based on a hard NP ∩ coNP function. *J. Math. Log.*, 11(1):11–27, 2011. 4

[Kra19] Jan Krajíček. *Proof complexity*, volume 170. Cambridge University Press, 2019. 4.1, 4.1

[LTW18] Fu Li, Iddo Tzameret, and Zhengyu Wang. Characterizing propositional proofs as noncommutative formulas. *SIAM Journal on Computing*, 47(4):1424–1462, 2018. 1, 1.3

[PBI93] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. Exponential lower bounds for the pigeonhole principle. *computational complexity*, 3(2):97–140, 1993. 1

[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. 1, 1.1.2, 1.3, 4.9, 4.2, 4.2

[PT16] Toniann Pitassi and Iddo Tzameret. Algebraic proof complexity: progress, frontiers and challenges. *ACM SIGLOG News*, 3(3):2143, aug 2016. 1

[Pud86] Pavel Pudlak. On the length of proofs of finitistic consistency statements in first order theories. *Studies in Logic and the Foundations of Mathematics*, 120:165–196, 1986. 1.3

[Pud87] Pavel Pudlak. Improved bounds to the length of proofs of finite consistency statements. *Contemporary Mathematics*, 65:309–331, 1987. 1.3

[Pud20] Pavel Pudlak. Reflection principles, propositional proof systems, and theories. *ArXiV*, July 2020. 5, 1.3, 4.1

[Raz10] Ran Raz. Elusive functions and lower bounds for arithmetic circuits. *Theory of Computing*, 6(1):135–177, 2010. 1.2, 3.1.2, 3.9

[Raz15] Alexander A. Razborov. Pseudorandom generators hard for k-DNF resolution and polynomial calculus resolution. *Annals of Mathematics*, 181:415–472, 2015. 1, 1.1.1, 4

[Raz16] Alexander Razborov. Propositional proof complexity: Fifteen (or so) years after. Talk at "A Celebration of Mathematics and Computer Science. Celebrating Avi Wigderson's 60th Birthday October 5 - 8, 2016". https://youtu.be/7LfW6VTW8zo?t=2722, 2016. 4

[Raz21] Alexander Razborov. P, NP and proof complexity. Talk at "SAT and Foundations of Mathematics", Simons Institute. https://youtu.be/xx4mxcqAl5A?t=2333, 2021. 4

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. System Sci.*, 55(1, part 1):24–35, 1997. Proceedings of the 26th Annual ACM Symposium on the Theory of Computing (STOC '94) (Montreal, PQ, 1994). 1.3

[Rud97] Steven Rudich. Super-bits, demi-bits, and NP/qpoly-natural proofs. In *RANDOM'97, Proceedings*, volume 1269, pages 85–93, 1997. 1.1.2, 4.10

[SS95] Michael Shub and Steve Smale. On the intractability of Hilbert's Nullstellensatz and an algebraic version of "NP≠P?". *Duke Math. J.*, 81:47–54, 1995. 1

[Str73] Volker Strassen. Vermeidung von divisionen. *J. Reine Angew. Math.*, 264:182–202, 1973. (in German). 3.1.2, 2

[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. 2.1, 2.1, 3.1.2

[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. 2.1, 3.1.3

[Val79b] Leslie G. Valiant. The complexity of computing the permanent. *Theor. Comput. Sci.*, 8:189–201, 1979. 2.1, 3.2.1

[Val82] Leslie G. Valiant. Reducibility by algebraic projections. *Logic and Algorithmic: International Symposium in honour of Ernst Specker*, 30:365–380, 1982. 2.1