Circuit Complexity, Proof Complexity, and Polynomial Identity Testing: The Ideal Proof System

JOSHUA A. GROCHOW, University of Colorado at Boulder and Santa Fe Institute
TONIANN PITASSI, University of Toronto and Institute for Advanced Study

We introduce a new and natural algebraic proof system, whose complexity measure is essentially the algebraic circuit size of Nullstellensatz certificates. This enables us to exhibit close connections between effective Nullstellensatzë, proof complexity, and (algebraic) circuit complexity. In particular, we show that any super-polynomial lower bound on any Boolean tautology in our proof system implies that the permanent does not have polynomial-size algebraic circuits (VNP ≠ VP). We also show that super-polynomial lower bounds on the number of lines in Polynomial Calculus proofs imply the Permanent versus Determinant Conjecture. Note that there was no proof system prior to ours for which lower bounds on an arbitrary tautology implied *any* complexity class lower bound.

Our proof system helps clarify the relationships between previous algebraic proof systems. In doing so, we highlight the importance of polynomial identity testing (PIT) in proof complexity. In particular, we use PIT to illuminate $AC^0[p]$-Frege lower bounds, which have been open for nearly 30 years, with no satisfactory explanation as to their apparent difficulty.

Finally, we explain the obstacles that must be overcome in any attempt to extend techniques from algebraic circuit complexity to prove lower bounds in proof complexity. Using the algebraic structure of our proof system, we propose a novel route to such lower bounds. Although such lower bounds remain elusive, this proposal should be contrasted with the difficulty of extending $AC^0[p]$ circuit lower bounds to $AC^0[p]$-Frege lower bounds.

CCS Concepts: • **Theory of computation** → **Algebraic complexity theory**; **Proof complexity**; **Pseudo-randomness and derandomization**; *Complexity classes*; • **Mathematics of computing** → *Gröbner bases and other special bases*;

Additional Key Words and Phrases: $AC^0[p]$-Frege, VNP, permanent versus determinant, polynomial calculus, lower bounds, polynomial identity testing, syzygies

ACM Reference format:
Joshua A. Grochow and Toniann Pitassi. 2018. Circuit Complexity, Proof Complexity, and Polynomial Identity Testing: The Ideal Proof System. *J. ACM* 65, 6, Article 37 (November 2018), 59 pages.
https://doi.org/10.1145/3230742

We gratefully acknowledge financial support from NSERC; in particular, J.A.G. was supported by A. Borodin's NSERC_Grant # 482671. During the preparation of this article, J.A.G. was also supported by a Santa Fe Institute Omidyar Fellowship and NSF_grant DMS-1620484.
Author's address: J. A. Grochow, 1111 Engineering Dr., ECOT 717, 430 UCB Boulder, CO, 80309, USA; email: jgrochow@colorado.edu; T. Pitassi, 10 Kings College Road, University of Toronto, Department of Computer Science, Toronto ON M5S 3G4, Canada; email: toni@cs.toronto.edu.

0004-5411/2018/11-ART37 $15.00
https://doi.org/10.1145/3230742

1 INTRODUCTION

NP versus coNP is the very natural question of whether, for every graph without a Hamiltonian path, there is a short proof of this fact. One of the arguments for the utility of proof complexity is that proving lower bounds for standard proof systems is a necessary step towards proving NP ≠ coNP. Moreover, standard proof systems correspond to standard circuit classes; for example, Frege corresponds to NC^1, and Extended Frege corresponds to P/poly; since the corresponding proof system can reason over the corresponding circuit class, it is speculated that a proof system lower bound would imply the corresponding circuit class lower bound, e.g., that lower bounds on Frege would imply NP ≠ NC^1. However, until now these arguments have been more the expression of a philosophy or hope, as there is no known proof system for which lower bounds imply computational complexity lower bounds of any kind, let alone NP ≠ coNP.

We remedy this situation by introducing a very natural algebraic proof system, which has tight connections to (algebraic) circuit complexity (albeit a proof system for which we only know a *randomized* efficient verification procedure). We show that any super-polynomial lower bound on any Boolean tautology in our proof system implies that the permanent does not have polynomial-size algebraic circuits (VNP ≠ VP). Additionally, lower bounds on bounded-depth versions of our system imply the corresponding algebraic circuit lower bounds, e.g., lower bounds on the logarithmic-depth version of our proof system imply VNP $\not\subseteq$ VNC^1. Note that, prior to our work, essentially all implications went the opposite direction: a circuit complexity lower bound implying a proof complexity lower bound. We use this result to begin to explain why several long-open lower bound questions in proof complexity—lower bounds on Extended Frege, on $\mathsf{AC}^0[p]$-Frege, and on number-of-lines in Polynomial Calculus-style proofs—have been so apparently difficult.

1.1 Background and Motivation

Algebraic Circuit Complexity. The most natural way to compute a polynomial function $f(x_1, \ldots, x_n)$ is with a sequence of instructions $g_1, \ldots, g_m = f$, starting from the inputs $x_1, \ldots, x_n$, and where each instruction g_i is of the form $g_j \circ g_k$ for some $j, k < i$, where $\circ$ is either a linear combination or multiplication. Such computations are called algebraic circuits or straight-line programs. The goal of algebraic complexity is to understand the optimal asymptotic complexity of computing a given polynomial family $(f_n(x_1, \ldots, x_{\mathrm{poly}(n)}))_{n=1}^{\infty}$, typically in terms of size (=number of instructions) and depth (the depth of the natural directed acyclic graph associated to the instruction sequence of a straight-line program) of algebraic circuits. In addition to the intrinsic interest in these questions, since Valiant's work [102–104] algebraic complexity has become more and more important for Boolean computational complexity. Valiant argued that understanding algebraic complexity could give new intuitions that may lead to better understanding of other models of computation (see also [108]); several direct connections have been found between algebraic and Boolean complexity [23, 48, 50, 74]; and the Geometric Complexity Theory Program (see, e.g., the overview [76] and references therein) suggests how algebraic techniques might be used to resolve major Boolean complexity conjectures.

Two central functions in this area are the determinant and permanent polynomials, which are fundamental both because of their prominent role in many areas of mathematics and because they are complete for various natural complexity classes. In particular, the permanent of $\{0, 1\}$-matrices is #P-complete, and the permanent of arbitrary matrices is VNP-complete in odd characteristic. Valiant's Permanent versus Determinant Conjecture [102] states that the permanent of an $n \times n$ matrix, as a polynomial in n^2 variables, cannot be written efficiently as the determinant of any polynomially larger matrix all of whose entries are variables or constants. In some ways this is an algebraic analog of P ≠ NP, although it is in fact much closer to FNC^2 ≠ #P. In addition to this analogy, the Permanent versus Determinant Conjecture is also known to be a formal consequence

of the nonuniform lower bound NP $\not\subseteq$ P/poly [23] and is thus thought to be an important step towards showing P $\neq$ NP.

Unlike in Boolean circuit complexity, (slightly) non-trivial lower bounds for the size of algebraic circuits are known [9, 97]. Their methods, however, only give lower bounds up to $\Omega(n \log n)$. Moreover, their methods are based on a degree analysis of certain algebraic varieties and do not give lower bounds for polynomials of constant degree. Recent work [2, 55, 99] has shown that polynomial-size algebraic circuits computing functions of polynomial degree can in fact be computed by sub-exponential-size depth 4 algebraic circuits. Thus, strong-enough lower bounds for depth 4 algebraic circuits for the permanent would already prove VP $\neq$ VNP.

Effective Nullstellensatzë. A special case of Hilbert's famous Nullstellensatz says that over a field $\mathbb{F}$, a set of polynomials $F_1(x_1, \ldots, x_n), \ldots, F_m(x_1, \ldots, x_n)$ of degree at most d has no common zero over the algebraic closure $\overline{\mathbb{F}}$ if and only if the ideal they generate contains 1 or, equivalently, if there exists polynomials G_i such that $\sum G_i F_i = 1$. Doubly exponential upper bounds on the degree of the $G_i F_i$ were shown as early as 1983 [68], of the form $d^{O(2^n)}$. These were later improved to singly exponential bounds of the form $O(d^n)$ [20] (in characteristic zero) and subsequently improved to bounds that hold over an arbitrarily algebraically closed field, have tighter dependence on the degrees of each F_i, and are essentially tight [56]. Since then, more refined geometric information than merely degree bounds has been obtained by a number of authors [21, 30, 49, 57, 96]. For a good overview of this work, see the introduction of Ein and Lazarsfeld [30] and references therein.

In this article, we raise the question of extending Effective Nullstellensatzë from degree bounds to bounds on the algebraic circuit complexity of Nullstellensatz certificates. (This question was perhaps implicit in Pitassi [81, 82], and a syntactically more complicated variant of this question was raised in Grigoriev-Hirsch [35].) It has long been known [42]—although perhaps not well known—that bounds on algebraic circuit complexity can have geometric consequences; indeed, this is one of the philosophical underpinnings of the current Geometric Complexity Theory Program towards resolving questions like P versus NP (see, e.g., References [36, 64, 75–78] and references therein). Here, we show that the algebraic circuit complexity of the Nullstellensatz has deep connections to Boolean proof complexity, and we use ideas motivated by this question to forge new bridges between proof complexity and computational complexity.

Proof Complexity. Despite considerable progress obtaining super-polynomial lower bounds for many weak proof systems (resolution [40], cutting planes [17], and bounded-depth Frege systems [58]), there has been essentially no progress in the last 25 years for stronger proof systems such as Extended Frege systems or Frege systems. More surprisingly, no nontrivial lower bounds are known for the seemingly weak $\mathrm{AC}^0[p]$-Frege system. In contrast, the analogous result in circuit complexity—proving super-polynomial $\mathrm{AC}^0[p]$ lower bounds for an explicit function—was resolved by Razborov and Smolensky over 25 years ago [86, 94]. To date, there has been no satisfactory explanation for this state of affairs.

In proof complexity, there are no known formal barriers such as relativization [8], Razborov–Rudich-natural proofs [87], or algebrization [1] that exist in Boolean function complexity. Moreover, there has not even been progress by way of conditional lower bounds. That is, trivially NP $\neq$ coNP implies superpolynomial lower bounds for $\mathrm{AC}^0[p]$-Frege, but we know of no weaker complexity assumption that implies such lower bounds. The only formal implication in this direction shows that certain circuit lower bounds imply lower bounds for proof systems that admit feasible interpolation, but unfortunately only weak proof systems (not Frege nor even AC^0-Frege) have this property, under standard complexity-theoretic assumptions [18, 19]. In the converse direction, there are essentially no implications at all. For example, we do not know if $\mathrm{AC}^0[p]$-Frege lower bounds—nor even Frege nor Extended Frege lower bounds—imply any nontrivial circuit lower bounds.

1.2 Our Results

In this article, we define a simple and natural proof system that we call the Ideal Proof System (IPS) based on Hilbert's Nullstellensatz. Our system is similar in spirit to related algebraic proof systems that have been studied previously but is different in a crucial way that we explain below.

Given a set of polynomials $F_1, \ldots, F_m$ in n variables $x_1, \ldots, x_n$ over a field $\mathbb{F}$ without a common zero over the algebraic closure $\overline{\mathbb{F}}$ of $\mathbb{F}$, Hilbert's Nullstellensatz says that there exist polynomials $G_1, \ldots, G_m \in \mathbb{F}[x_1, \ldots, x_n]$ such that $\sum F_i G_i = 1$, i.e., that 1 is in the ideal generated by the F_i. In the Ideal Proof System, we introduce new variables y_i that serve as placeholders into which the original polynomials F_i will eventually be substituted:

Definition 1.1 (Ideal Proof System). An *IPS certificate* that a system of $\mathbb{F}$-polynomial equations $F_1(\vec{x}) = F_2(\vec{x}) = \cdots = F_m(\vec{x}) = 0$ is unsatisfiable over $\overline{\mathbb{F}}$ is an $\mathbb{F}$-polynomial $C(\vec{x}, \vec{y})$ in the variables $x_1, \ldots, x_n$ and $y_1, \ldots, y_m$ such that

(1) $C(x_1, \ldots, x_n, \vec{0}) = 0$, and
(2) $C(x_1, \ldots, x_n, F_1(\vec{x}), \ldots, F_m(\vec{x})) = 1$.

The first condition is equivalent to C being in the ideal generated by $y_1, \ldots, y_m$, and the two conditions together therefore imply that 1 is in the ideal generated by the F_i, and hence that $F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$ is unsatisfiable.

An *IPS proof* of the unsatisfiability of the polynomials F_i is an $\mathbb{F}$-algebraic circuit on inputs $x_1, \ldots, x_n, y_1, \ldots, y_m$ computing some IPS certificate of unsatisfiability.

For any class C of polynomial families, we may speak of C-IPS proofs of a family of systems of equations $(\mathcal{F}_n)$, where $\mathcal{F}_n$ is $F_{n,1}(\vec{x}) = \cdots = F_{n,\mathrm{poly}(n)}(\vec{x}) = 0$. When we refer to IPS without further qualification, we mean IPS certificates whose proofs are computed by circuits of polynomial size (with *no a priori bound on the degree*), unless specified otherwise.[1]

The Ideal Proof System is easily shown to be sound, and (without any size bounds) its completeness follows from the Nullstellensatz.

We note that although the Nullstellensatz says that if $F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$ is unsatisfiable, then there always exists a certificate that is linear in the y_i—that is, of the form $\sum y_i G_i(\vec{x})$—our definition of IPS certificate does not enforce $\vec{y}$-linearity. The definition of IPS certificate allows certificates with $\vec{y}$-monomials of higher degree, and it is conceivable that one could achieve a savings in size by considering such certificates rather than only considering $\vec{y}$-linear ones. (Subsequent to this work it was shown [34] that, for *general* IPS, super-polynomial savings are not possible; but the result does not rule out a savings for C-IPS proofs for various restricted classes C; see Section 8.2 below for details.) As the linear form is closer to the original way Hilbert expressed the Nullstellensatz (see, e.g., the translation [44]), we refer to certificates of the form $\sum y_i G_i(\vec{x})$ as *Hilbert-like IPS certificates*.

We typically consider IPS as a propositional proof system by translating a CNF tautology φ into a system of equations as follows. We translate a clause κ of φ into a single algebraic equation $F(\vec{x})$ as follows: $x \mapsto 1 - x$, $x \vee y \mapsto xy$. This translation has the property that a $\{0,1\}$ assignment satisfies κ if and only if it satisfies the equation $F = 0$. Let $\kappa_1, \ldots, \kappa_m$ denote all the clauses of φ, and let F_i be the corresponding polynomials. Then the system of equations we consider is $F_1(\vec{x}) = \cdots = F_m(\vec{x}) = x_1^2 - x_1 = \cdots = x_n^2 - x_n = 0$. The latter equations force any solution to this

[1] In an earlier version of this article, we defined IPS to be VP-IPS, that is, with polynomially bounded degree as well; however, it was pointed out in References [34, 65] and by an anonymous referee that the simulation of EF by IPS seemed to need unbounded degree. Most of our results go through unchanged; those that change are only in the statement of the result, not in the proof.

system of equations to be $\{0, 1\}$-valued. Despite our indexing here, when we speak of the system of equations corresponding to a tautology, we always assume that the $x_i^2 - x_i$ are among the equations, unless explicitly stated otherwise (and, indeed, there are a few situations where we do not need the equations $x_i^2 - x_i$).

Like previously defined algebraic systems [14, 27, 81, 82], proofs in our system can be checked in randomized polynomial time. The key difference between our system and previously studied ones is that those systems are axiomatic in the sense that they require that *every* sub-computation (derived polynomial) be in the ideal generated by the original polynomial equations F_i and thus be a sound consequence of the equations $F_1 = \cdots = F_m = 0$. In contrast our system has no such requirement: An IPS proof can compute potentially "unsound" sub-computations (whose vanishing does not follow from $F_1 = \cdots = F_m = 0$), as long as the *final polynomial* is in the ideal generated by the equations. This key difference allows IPS proofs to be *ordinary algebraic circuits*, and thus nearly all results in algebraic circuit complexity apply directly to the Ideal Proof System. To quote the tagline of a common US food chain, the Ideal Proof System is a "No rules, just right" proof system.

Our first main theorem shows one of the advantages of this close connection with algebraic circuits. To the best of our knowledge, this is the first implication showing that a proof complexity lower bound implies any sort of computational complexity lower bound.

Theorem 1.2 (Brief; see Section 4 for the Full Statement). *Super-polynomial lower bounds for the Ideal Proof System imply that the permanent does not have polynomial-size algebraic circuits, that is,* VNP $\neq$ VP.

The preceding theorem is perhaps somewhat unsurprising—though not completely immediate—given the definition of IPS, because of the close connection between the definition of IPS proofs and algebraic circuits. However, the following result is significantly more surprising—showing a relation between a standard rule-based algebraic proof system and algebraic circuit lower bounds—and we believe we would not have come to this result had we not first considered the rule-less Ideal Proof System.

Corollary 1.3. *Super-polynomial lower bounds on the number of lines in Polynomial Calculus proofs imply the Permanent versus Determinant Conjecture.*[2,3]

Corollary 1.3 follows from the proof of Theorem 1.2 together with one of our simulation results (Proposition 3.4).

Under a reasonable assumption on polynomial identity testing, which we discuss further below, we are able to show that Extended Frege is equivalent to the Ideal Proof System. Polynomial Identity Testing (PIT) is the problem of deciding whether a given algebraic circuit computes the identically zero polynomial or not; unless otherwise stated, we take PIT to allow arbitrary circuits as input, with no restriction on their degree. Even without degree restriction, the standard randomized algorithm places PIT into coRP by working over an extension field of polynomial degree

[2] Although Corollary 1.3 may seem to be saying that lower bounds on PC imply a circuit lower bound, this is not precisely the case, because size complexity in PC is typically measured not by the number of lines but rather by the total number of monomials appearing in a PC proof.

[3] A folklore result might mislead one to think that the premise of our Corollary 1.3 is false: Proposition 2.3 of the ECCC preprint of Miksa–Nordström [72] states the folklore result that every unsatisfiable k-CNF tautology has a PC refutation using only a linear number of lines over $\mathbb{F}_2$, in the version of PC *in which the axioms* $x_i^2 = x_i$ *may be applied for free.* In contrast, even over $\mathbb{F}_2$, we are considering the number of lines here *for proofs in which each use of the axioms* $x_i^2 = x_i$ *counts as a line.* As shown in this article, a polynomial upper bound on the latter system would imply NP $\subseteq$ coAM and hence that PH collapses. See also Open Question 7.9.

if needed [29, 89, 109]. Extended Frege (EF) is the strongest natural deduction-style propositional proof system that has been proposed and is the proof complexity analog of P/poly (that is, Extended Frege = P/poly-Frege).

Theorem 1.4. *Let K be a family of polynomial-size Boolean circuits for PIT such that the PIT axioms for K (see Definition 5.1) have polynomial-size EF proofs. Then EF polynomially simulates IPS, and hence EF and IPS are polynomially equivalent.*

In light of Theorem 1.4, a promising direction for proving EF lower bounds is to try to prove lower bounds for IPS instead. Since this suggestion seems counter to the usual philosophy of trying to prove lower bounds on the next-hardest proof system, and proving IPS lower bounds may be much harder, this suggestion deserves a little discussion. First, by considering C-IPS for restricted circuit classes C, we recover some of the usual philosophy of trying to prove lower bounds on incrementally harder proof systems first rather than jumping straight to (full) IPS lower bounds.

Second, and more importantly, IPS gives a new way of thinking about propositional proof systems and creates the possibility of harnessing tools from algebra, representation theory, and algebraic circuit complexity. Indeed, tools from algebraic circuit complexity have already been used to prove lower bounds for some restricted IPS systems [34] , and in Section 6 we give one suggestion of how to apply tools from algebraic geometry to obtain IPS lower bounds.

Remark 1.5. The combination of Thereoms 1.2 and 1.4 together state that if the PIT axioms are provable in EF, then EF lower bounds imply circuit lower bounds (VNP $\neq$ VP). The hypothesis that the PIT axioms are provable in EF is orthogonal but still closely related to the more standard hypothesis that PIT is in P. Since upper bounds on PIT are also known to imply lower bounds, we would like to address the differences between the two conclusions. The best lower bound known to follow from PIT $\in$ P is an algebraic circuit-size lower bound on an integer polynomial that can be evaluated in NEXP $\cap$ coNEXP [25, 48] (via personal communication we have learned that Impagliazzo and R. Williams have also proved similar results), whereas our conclusion is a lower bound on algebraic circuit-size for an integer polynomial computable in the much smaller class #P $\subseteq$ PSPACE.

Although PIT has long been a central problem of study in computational complexity—both because of its importance in many algorithms, as well as its strong connection to circuit lower bounds—our theorems highlight the importance of PIT in proof complexity. Next we prove that Theorem 1.4 can be scaled down to obtain similar results for weaker Frege systems and discuss some of its more striking consequences.

Theorem 1.6. *Let C be any of the standard circuit classes AC^k, $\mathrm{AC}^k[p]$, ACC^k, TC^k, NC^k. Let K be a family of polynomial-size Boolean circuits for PIT (not necessarily in C) such that the PIT axioms for K have polynomial-size C-Frege proofs. Then C-Frege is polynomially equivalent to IPS and, consequently, to Extended Frege as well.*

Theorem 1.6 also highlights the importance of our PIT axioms for getting $\mathrm{AC}^0[p]$-Frege lower bounds, which has been an open question for nearly 30 years. (For even weaker systems, Theorem 1.6 in combination with known results yields an unconditional lower bound on AC^0-Frege proofs of the PIT axioms.) In particular, we are in the following win-win scenario:

Corollary 1.7. *For any d, either:*

- *There are polynomial-size $\mathrm{AC}^0[p]$-Frege proofs of the depth-d PIT axioms, in which case* any superpolynomial lower bounds on $\mathrm{AC}^0[p]$-Frege imply $\mathrm{VNP}_{\mathbb{F}_p}$ does not have polynomial-size depth-d algebraic circuits*, thus explaining the difficulty of obtaining such lower bounds, or*

- *There are no polynomial-size* $\mathrm{AC}^0[p]$*-Frege proofs of the depth-d PIT axioms, in which case we have* $\mathrm{AC}^0[p]$*-Frege lower bounds.*

Finally, in Section 6 we show what obstacles must be overcome in any attempt to extend proof techniques from lower bounds on (C-)algebraic circuits to lower bounds on (C-)IPS proofs—which may also apply to Extended Frege via Theorem 1.4. We then use the algebraic structure of IPS to suggest a new approach to proving lower bounds that we feel has promise. In particular, the set of *all IPS-certificates* for a given unsatisfiable system of equations is, in a certain precise sense, "finitely generated." We suggest how one might take advantage of this finite generation to transfer techniques from algebraic circuit complexity to prove lower bounds on IPS, and, consequently, on Extended Frege (since IPS p-simulates Extended Frege unconditionally), giving hope for the long-sought length-of-proof lower bounds on an algebraic proof system. We hope to pursue this approach in future work.

1.3 Related Work

Other proof systems. We will see in Section 3.3 that many previously studied proof systems can be p-simulated by IPS and, furthermore, can be viewed simply as different complexity measures on IPS proofs or as C-IPS for certain classes C. In particular, the Nullstellensatz system [14], the Polynomial Calculus (or Gröbner) proof system [27], and Polynomial Calculus with Resolution [4] are all particular measures on IPS, and Pitassi's previous algebraic systems [81, 82] are subsystems of IPS.

Raz and Tzameret [85] introduced various multilinear algebraic proof systems. Although their systems are not so easily defined in terms of IPS, the Ideal Proof System nonetheless p-simulates all of their systems. Among other results, they show that a super-polynomial separation between two variants of their system—one representing lines by multilinear circuits, and one representing lines by general algebraic circuits—would imply a super-polynomial separation between general and multilinear circuits computing multilinear polynomials. However, they only get implications to lower bounds on multilinear circuits rather than general circuits, and they do not prove a statement analogous to our Theorem 1.2, that lower bounds on a single system imply algebraic circuit lower bounds.

Grigoriev and Hirsch [35] introduced two proof systems, F-NS and F-PC, analogous to Nullstellensatz and Polynomial Calculus, respectively, but in which the basic objects of the proofs are allowed to be algebraic formulae, rather than only sums of monomials, and equivalence of formulae must be verified line-by-line using the axioms for a polynomial ring (associativity, commutativity, distributivity, etc.). Again, although these systems are not so easily defined in terms of IPS, they are easily p-simulated by IPS; indeed, in IPS the standard axioms for a polynomial ring come nearly for free—the main cost is that the verification is randomized instead of deterministic. They did not draw connections between algebraic circuit lower bounds and lower bounds on their proof systems.

Finally, we mention that Hrubeš and Tzameret [46] have studied the proof complexity of polynomial identity testing (PIT). In particular, they studied the question of how many basic ring identities—associativity, distributivity, and so on—are needed to verify a polynomial identity. While one of the messages of our article is the importance of the proof complexity of PIT, the way it shows up in our work is in proving that a Boolean circuit deciding PIT is correct (see our PIT axioms, Definition 5.1), whereas in Hrubeš and Tzameret [46] they study the complexity of proving individual polynomial identities.

Ideal Membership and Effective Nullstellensatzë. Prior to our work, much work was done on bounds for the Ideal Membership Problem (EXPSPACE-complete [70, 71]), the so-called Effective

Nullstellensatz (where exponential degree bounds are known, and known to be tight [20, 30, 56, 96]), and the arithmetic Nullstellensatz over $\mathbb{Z}$, where one wishes to bound not only the degree of the polynomials but also the sizes of the integer coefficients appearing [61]. The viewpoint afforded by the Ideal Proof Systems raises new questions about potential strengthening of these results (or at least simplifies and highlights questions that were implicit in References [35, 81, 82]).

In particular, the following is a natural extension of Definition 1.1.

Definition 1.8. An *IPS certificate* that a polynomial $G(\vec{x}) \in \mathbb{F}[\vec{x}]$ is in the ideal (respectively, radical of the ideal) generated by $F_1(\vec{x}), \ldots, F_m(\vec{x})$ is a polynomial $C(\vec{x}, \vec{y})$ such that

(1) $C(\vec{x}, \vec{0}) = 0$, and
(2) $C(\vec{x}, F_1(\vec{x}), \ldots, F_m(\vec{x})) = G(\vec{x})$ (respectively, $G(\vec{x})^k$ for any $k > 0$).

An *IPS derivation* of G (respectively, G^k) from $F_1, \ldots, F_m$ is a circuit computing some IPS certificate that $G \in \langle F_1, \ldots, F_m \rangle$ (respectively, $G \in \sqrt{\langle F_1, \ldots, F_m \rangle}$).

Grigoriev and Hirsch [35, Section 2.5] introduced a related system, denoted (F-)PC$_{\sqrt{}}$, for proving that a polynomial is in the radical of an ideal. The key difference between (F-)PC$_{\sqrt{}}$ and (F-)PC being that they add the rule from which derives a polynomial P from P^2. But otherwise, their system has similar tradeoffs relative to IPS: On the one hand, their system can be deterministically verified; on the other hand, it is restricted to syntactic derivations.

Observation 1.9. *There is no sub-exponential ($\bigcap_{\varepsilon>0} O(2^{n^\varepsilon})$) upper bound on the size of constant-free circuits computing IPS-certifices of ideal membership. Similarly for general algebraic circuits in characteristic zero, assuming the Generalized Riemann Hypothesis (GRH).*

Under special circumstances, of course, one may be able to achieve better upper bounds.

Proof. Suppose that for every $G(\vec{x}) \in \langle F_1(\vec{x}), \ldots, F_m(\vec{x}) \rangle$ there were a constant-free circuit of sub-exponential size computing some IPS certificate for the membership of G in $\langle F_1, \ldots, F_m \rangle$. Then guessing that circuit and verifying its correctness using PIT gives a $\mathsf{MA}_{\mathrm{subexp}} \subseteq \mathsf{SUBEXPSPACE}$ algorithm for the Ideal Membership Problem. The EXPSPACE-completeness of Ideal Membership [70, 71] would then imply that $\mathsf{EXPSPACE} \subseteq \mathsf{SUBEXPSPACE}$, contradicting the Space Hierarchy Theorem [41]. In characteristic zero, if we assume GRH, we may drop the assumption that the circuits are constant free, using essentially the same argument as in Proposition 3.2. □

The preceding observation, however, does not seem to apply to effective Nullstellensatzë, which are generally about showing that a function G is in the *radical* of an ideal (which, in particular, always applies for $G = 1$). We thus raise the following question:

Open Question 1.10. For any $G(\vec{x}) \in \sqrt{\langle F_1(\vec{x}), \ldots, F_m(\vec{x}) \rangle}$, is there always an IPS-certificate, as in Definition 1.8, of sub-exponential size that G is in the *radical* of $\langle F_1, \ldots, F_m \rangle$? Similarly, for $G, F_1, \ldots, F_m \in \mathbb{Z}[x_1, \ldots, x_n]$, is there a constant-free IPS$_{\mathbb{Z}}$-certificate of sub-exponential size that $aG(\vec{x})$ is in the *radical* of the ideal $\langle F_1, \ldots, F_m \rangle$ for some integer a?

1.4 Outline

In Section 2, we give the necessary preliminaries from algebraic circuit complexity, proof complexity, and commutative algebra. We really begin in Section 3 by proving several basic facts about IPS. We discuss the relationship between IPS and previously studied proof systems. We also highlight several consequences of results from algebraic complexity theory for the Ideal Proof System, such as division elimination [98] and the chasms at depths 3 [39, 99] and 4 [2, 55, 99].

In Section 4, we prove that lower bounds on IPS imply algebraic circuit lower bounds (Theorem 1.2). We also show how this result gives as a corollary a new, simpler proof that

$\mathsf{NP} \not\subseteq \mathsf{coMA} \Rightarrow \mathsf{VNP}^0_{\mathbb{F}} \neq \mathsf{VP}^0_{\mathbb{F}}$ over any field $\mathbb{F}$ (Corollary 4.1). In Section 5 we introduce our PIT axioms in detail and prove Theorems 1.4 and 1.6.

We also discuss in detail many variants of Theorem 1.6 and their consequences, as briefly mentioned above. In Section 6, we show what obstacles need to be overcome to extend lower bounds from algebraic circuit complexity to (algebraic) proof complexity; we also suggest a new framework for transferring techniques in this direction. Finally, in Section 7, we gather a long list of open questions raised by our work, many of which we believe may be quite approachable.

In Section 8, we discuss some developments that occurred subsequent to the appearance of the preliminary version of this article [37]. Namely, Li, Tzameret, and Wang [65] showed—along the lines suggested in Section 5—that non-commutative formula IPS is unconditionally quasi-polynomially equivalent to Frege. We discuss their result and its significance for proving Frege lower bounds. Also, Forbes, Shpilka, Tzameret, and Wigderson [34] showed several fundamental results about IPS as well as how to transfer some techniques from circuit complexity to prove lower bounds on some simple polynomials in C-IPS for various C.

In Appendices A and B, we introduce two variants of the Ideal Proof System—one of which allows certificates to be rational functions and not only polynomials and one of which has a more geometric flavor—and discuss their relationship to IPS. These systems further suggest that tools from geometry and algebra could potentially be useful for understanding the complexity of various propositional tautologies and more generally the complexity of individual instances of NP-complete problems.

2 PRELIMINARIES

As general references, we refer the reader to Sipser [93] or Arora–Barak [5] for Boolean computational complexity, to Bürgisser–Clausen–Shokrollahi [24] and two surveys [26, 92] for algebraic complexity, to Krajíček [59] for proof complexity, and to any of the standard books [6, 31, 69, 88] for commutative algebra.

We use $\mathrm{poly}(n)$ as a synonym for $n^{O(1)}$, i.e., any function $\mathbb{N} \to \mathbb{N}$ that is eventually bounded by n^k for some k. Different instances of "$\mathrm{poly}(n)$," even in the same sentence, may mean different polynomials. We use the quantifier $\exists^p$ to mean "there exists a string of length $\mathrm{poly}(n)$" and $\forall^p$ to mean "for all strings of length $\mathrm{poly}(n)$." Similarly, $Pr(X)$ denotes the probability of an event X, and $Pr^p_r(X)$ denotes the probability of the event X, taken over a uniformly random choice of strings r of length $\mathrm{poly}(n)$.

2.1 Algebraic Complexity

Over a ring R, VP_R is the class of families $f = (f_n)_{n=1}^{\infty}$ of formal polynomials—that is, considered as symbolic polynomials rather than as functions—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 is the class of families g of polynomials g_n such that g_n has $\mathrm{poly}(n)$ input variables and is of $\mathrm{poly}(n)$ degree and can be written as

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

for some family $(f_n) \in \mathsf{VP}_R$.

A *linear combination gate* is a gate g, with incoming edges from gates $f_1, \ldots, f_k$, and with scalar weights $w_i \in \mathbb{F}$ on its incoming edges, which computes the linear combination $\sum_i w_i f_i$. For the definitions of VP and VNP it does not matter whether we use gates of bounded fan-in or unbounded fan-in, and whether we allow general linear combination gates or merely addition gates (with no

weights). But when we consider families of algebraic circuits of bounded-depth, we will by default allow linear combination gates and product gates of unbounded fan-in.

A family of algebraic circuits is said to be *constant free* if the only constants used in the circuit are $\{0, 1, -1\}$. Other constants can be used but must be built up using algebraic operations, which then count towards the size of the circuit. The class VP^0 is defined by restricting the circuits used in the definition of VP to be constant free and similarly for VNP^0. We note that over a fixed finite field $\mathbb{F}_q$, $\mathsf{VP}^0_{\mathbb{F}_q} = \mathsf{VP}_{\mathbb{F}_q}$, since there are only finitely many possible constants. Consequently, $\mathsf{VNP}^0_{\mathbb{F}_q} = \mathsf{VNP}_{\mathbb{F}_q}$ as well. Over the integers, $\mathsf{VP}^0_{\mathbb{Z}}$ coincides with those families in $\mathsf{VP}_{\mathbb{Z}}$ that are computable by algebraic circuits of polynomial total *bit-size*: Note that any integer of polynomial bit-size can be constructed by a constant-free circuit by using its binary expansion $b_n \cdots b_1 = \sum_{i=0}^{n-1} b_i 2^i$, and computing the powers of 2 by linearly many successive multiplications. A similar trick shows that over the algebraic closure $\overline{\mathbb{F}}_p$ of a finite field, $\mathsf{VP}^0_{\overline{\mathbb{F}}_p}$ coincides with those families in $\mathsf{VP}_{\overline{\mathbb{F}}_p}$ that are computable by algebraic circuits of polynomial total bit-size or equivalently where the constants they use lie in subfields of $\overline{\mathbb{F}}_p$ of total size bounded by $2^{n^{O(1)}}$. (Recall that $\mathbb{F}_{p^a}$ is a subfield of $\mathbb{F}_{p^b}$ whenever $a|b$, and that the algebraic closure $\overline{\mathbb{F}}_p$ is just the union of $\mathbb{F}_{p^a}$ over all integers a.)

A polynomial $f(\vec{x})$ is a *projection* of a polynomial $g(\vec{y})$ if $f(\vec{x}) = g(L(\vec{x}))$ identically as polynomials in $\vec{x}$, for some map L that assigns to each y_i either a variable or a constant. A family of polynomials (f_n) is a polynomial projection or *p-projection* of another family (g_n), denoted $(f_n) \leq_p (g_n)$, 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 primary value of projections is that they are very simple and thus preserve bounds on nearly all natural complexity measures. Valiant [102, 104] was the first to point out not only their value but also their ubiquity in computational complexity—nearly all problems that are known to be complete for some natural class, even in the Boolean setting, are complete under p-projections. We say that two families $f = (f_n)$ and $g = (g_n)$ are of the same p-degree if each is a p-projection of the other, which we denote $f \equiv_p g$.

Despite its central role in computation, and the fact that $\mathsf{VP} = \mathsf{VNC}^2$ [105], the determinant is not known to be VP-complete under p-projections. The determinant is VQP-complete (VQP is defined just like VP but with a quasi-polynomial $n^{(\log n)^{O(1)}}$ bound on the size and degree of the circuits) under qp-projections (like p-projections, but with a quasi-polynomial bound). The complexity of the determinant is clarified by skew and weakly skew circuits. An algebraic circuit is *skew* if every multiplication gate has only two inputs, one of which is a variable or a constant. An algebraic circuit is *weakly skew* if every multiplication gate has at least one input that is computed solely for the purposes of that multiplication gate; more precisely, each multiplication gate $f = g_0 \times g_1$ has the property that for at least one of its input g_i, removing the edge connecting g_i to f in the directed acyclic graph corresponding to the circuit results in a disconnected graph. VP_s is the class of polynomials of poly(n) variables computed by skew circuits of poly(n) size; VP_{ws} is defined analogously with "skew" replaced by "weakly skew." In both these classes, poly(n) bounded degree follows from the definition for free, thus $\mathsf{VP}_s \subseteq \mathsf{VP}$ and $\mathsf{VP}_{ws} \subseteq \mathsf{VP}$. Let VP_{det} denote the class of polynomials that are p-projections of the determinant. It turns out that $\mathsf{VP}_s = \mathsf{VP}_{ws} = \mathsf{VP}_{\text{det}}$ [101] (see also Malod and Portier [67]). We will use weakly skew circuits and VP_{det} in Proposition 3.4.

The *semantic degree* of any gate in an algebraic circuit is just the degree of the polynomial it computes; the semantic degree of a (single-output) algebraic circuit is the semantic degree of its output gate. The *syntactic degree* of an algebraic circuit is defined inductively as follows: The syntactic degree of a constant is 0; the syntactic degree of a variable is 1; the syntactic degree of a product gate with children $f_1, \ldots, f_k$ is the sum of the syntactic degrees of the f_i; and the syntactic degree of a sum or linear combination gate with children $f_1, \ldots, f_k$ is the maximum of

the syntactic degrees of the f_i. Semantic degree can be exponentially smaller than syntactic degree due to cancellations.

2.2 Proof Complexity

Here we give formal definitions of proof systems and probabilistic proof systems for coNP languages and discuss several important and standard proof systems for TAUT.

Definition 2.1. Let $L \subseteq \{0,1\}^*$ be a coNP language. A *proof system P for L* is a polynomial-time function of two inputs $x, \pi \in \{0,1\}^*$ ("π" for "proof") with the following properties:

(1) (Perfect Soundness) If x is not in L, then for every π, $P(x,\pi) = 0$.
(2) (Completeness) If x is in L, then there exists a π such that $P(x,\pi) = 1$.

P is *polynomially bounded* if for every $x \in L$, there exists a π such that $|\pi| \leq \text{poly}(|x|)$ and $P(x,\pi) = 1$.

As this is just the definition of an NP procedure for L, it follows that for any coNP-complete language L, L has a polynomially bounded proof system if and only if coNP $\subseteq$ NP.

Cook and Reckhow [28] formalized proof systems for the language TAUT (all Boolean tautologies) in a slightly different way, although their definition is essentialy equivalent to the one above. We mildly prefer the above definition as it is consistent with definitions of interactive proofs.

Definition 2.2. A *Cook–Reckhow proof system* is a polynomial-time function P' of just one input π, and whose range is the set of all yes-instances of L. If $x \in L$, then any π such that $P'(\pi) = x$ is called a P'-proof of x. P' must satisfy the following properties:

(1) (Soundness) For every $x, \pi \in \{0,1\}^*$, if $P'(\pi) = x$, then $x \in L$.
(2) (Completeness) For every $x \in L$, there exists a π such that $P'(\pi) = x$.

(That is, the image of P' must be exactly L.)

P' is *polynomially bounded* if for every $x \in L$, there exists a π such that $|\pi| \leq \text{poly}(|x|)$ and $P'(\pi) = x$.

Intuitively, we think of P' as a procedure for verifying that π is a proof that some $x \in L$ and if so, it outputs x. (For all strings z that do not encode valid proofs, $P'(z)$ may just output some fixed $x_0 \in L$.) It is a simple exercise to see that for every language L, any propositional proof system P according to our definition can be converted to a Cook–Reckow proof system P', and vice versa, and furthermore the runtime properties of P and P' will be the same. In the forward direction, say P is a proof system for L according to our definition. Define π as encoding a pair (x, π'); on input $\pi = (x, \pi')$, P' runs P on the pair (x, π'). If P accepts, then $P'(\pi)$ outputs x, and if P rejects, then $P'(\pi)$ outputs (the encoding of) a canonical x_0 in L. Conversely, say that P' is a Cook–Reckhow proof system for L. $P(x,\pi)$ runs P' on π and accepts if and only if $P'(\pi) = x$.

Definition 2.3. Let P_1 and P_2 be two proof systems for a language L in coNP. P_1 *polynomially simulates* or *p-simulates* P_2 if for every $x \in L$ and for every π such that $P_2(x,\pi) = 1$, there exists π' such that $|\pi'| \leq \text{poly}(|\pi|)$, and $P_1(x,\pi') = 1$.

Informally, P_1 p-simulates P_2 if proofs in P_1 are no longer than proofs in P_2 (up to polynomial factors).

Definition 2.4. Let P_1 and P_2 be two proof systems for a language L in coNP. P_1 and P_2 are *polynomially equivalent* or *p-equivalent* if P_1 p-simulates P_2 and P_2 p-simulates P_1.

Standard Propositional Proof Systems. For TAUT (or UNSAT), there are a variety of standard and well-studied proof systems, the most important ones including Extended Frege (EF), Frege,

Bounded-depth Frege, and Resolution. A Frege rule is an inference rule of the form: $B_1, \ldots, B_n \Rightarrow B$, where $B_1, \ldots, B_n, B$ are propositional formulas. If $n = 0$, then the rule is an axiom. For example, $A \vee \neg A$ is a typical Frege axiom, and $A, \neg A \vee B \Longrightarrow B$ is a typical Frege rule. A Frege system is specified by a finite set, R, of rules. Given a collection R of rules, a derivation of a 3DNF formula f is a sequence of formulas $f_1, \ldots, f_m$ such that each f_i is either an instance of an axiom scheme or follows from previous formulas by one of the rules in R and such that the final formula f_m is f. In order for a Frege system to be a proof system in the Cook–Reckhow sense, its corresponding set of rules must be sound and complete. Work by Cook and Reckhow in the 1970s [28] showed that Frege systems are very robust in the sense that all Frege systems are polynomially equivalent.

Bounded-depth Frege proofs (AC^0-Frege) are Frege proofs but with the additional restriction that each formula in the proof has bounded depth. (Because our connectives are AND, OR, and negation, by depth we assume the formula has all negations at the leaves, and we count the maximum number of alternations of AND/OR connectives in the formula.) Polynomial-sized AC^0-Frege proofs correspond to the complexity class AC^0 because such proofs allow a polynomial number of lines, each of which must be "syntactically in AC^0" (that is, syntactically it must be described by a bounded-depth circuit).

Bounded-depth Frege proofs with mod p connectives ($AC^0[p]$-Frege) are bounded-depth Frege proofs that also allow unbounded fan-in MOD_p connectives, namely MOD_p^i for $i \in \{0, \ldots, p-1\}$. $MOD_p^i(x_1, \ldots, x_k)$ evaluates to true if the number of x_i that are true is congruent to $i \pmod p$ and evaluates to false otherwise.

Extended Frege systems generalize Frege systems by allowing, in addition to all of the Frege rules, a new axiom of the form $y \leftrightarrow A$, where A is a formula and y is a new variable not occurring in A nor in the final formula (i.e., the formula being proved). Whereas polynomial-size Frege proofs allow a polynomial number of lines, each of which must be a polynomial-size formula, using the new axiom, polynomial-size EF proofs allow a polynomial number of lines, each of which can be a polynomial-size circuit. See Krajíček [59] for precise definitions of Frege, AC^0-Frege, and EF proof systems.

Probabilistic Proof Systems. The concept of a proof system for a language in coNP can be generalized in the natural way to obtain randomized Merlin–Arthur-style proof systems.

Definition 2.5. Let L be a language in coNP, and let V (for "verifier") be a probabilistic polynomial-time algorithm with two inputs $x, \pi \in \{0,1\}^*$. V is a *probabilistic proof system* for L if:

(1) (Perfect Soundness) For every x that is not in L, and for every π,
$$Pr_r[P(x, \pi) = 1] = 0,$$
where the probability is over the random coin tosses, r of P.
(2) (Completeness) For every x in L, there exists a π such that
$$Pr_r[P(x, \pi) = 1] \geq 3/4.$$

P is *polynomially bounded* if for every $x \in L$, there exists π such that $|\pi| \leq \text{poly}(|x|)$ and $Pr_r[P(x, \pi) = 1] \geq 3/4$.

It is clear that for any coNP-complete language L, there is a polynomially bounded probabilistic proof system for L if and only if coNP $\subseteq$ MA (which implies the collapse of PH).

Again, we have chosen to define our probabilitic proof systems to match the definition of MA. The probabilistic proof system that would be analogous to the standard Cook–Reckhow proof system would be somewhat different, as defined below. Again, a simple argument like the one

above shows that our probablistic proof systems are essentially equivalent to probabilistic Cook–Reckhow proof systems.

Definition 2.6. A *probabilistic Cook–Reckhow proof system* for a language $L \in \text{coNP}$ is a probabilistic polynomial-time algorithm A (whose runtime is independent of its random choices) such that

(1) There is a surjective function $f : \Sigma^* \to L$ such that $A(x) = f(x)$ with probability at least 2/3 (over A's random choices), and
(2) Regardless of A's random choices, its output is always in L.

Such a proof system is *polynomially bounded* or *p-bounded* if for every $x \in L$, there is some π such that $f(\pi) = x$ and $|\pi| \leq \text{poly}(|x|)$.

We note that both Pitassi's algebraic proof system [81] and the Ideal Proof System are probabilistic Cook–Reckhow systems. The algorithm P takes as input a description of a (constant-free) algebraic circuit C together with a tautology φ and then verifies that the circuit is indeed an IPS-certificate for φ by using the standard Schwartz–Zippel–DeMillo–Lipton [29, 89, 109] coRP algorithm for polynomial identity testing. The proof that Pitassi's algebraic proof system is a probabilistic Cook–Reckhow system is essentially the same.

2.3 Commutative Algebra

The following preliminaries from commutative algebra are needed only in Section 6 and Appendix A.

A *module* over a ring R is defined just like a vector space, except over a ring instead of a field. That is, a module M over R is a set with two operations: addition (making M an abelian group) and multiplication by elements of R ("scalars"), satisfying the expected axioms (see any textbook on commutative algebra, e.g., References [6, 31]). A module over a field $R = \mathbb{F}$ is precisely a vector space over $\mathbb{F}$. Every ring R is naturally an R-module (using the ring multiplication as the scalar multiplication), as is R^n, the set of n-tuples of elements of R. Every ideal $I \subseteq R$ is an R-module—indeed, an ideal could be defined, if one desired, as an R-submodule of R—and every quotient ring R/I is also an R-module, by $r \cdot (r_0 + I) = rr_0 + I$.

Unlike vector spaces, however, there is not so nice a notion of "dimension" for modules over arbitrary rings. Two differences will be particularly relevant in our setting. First, although every vector subspace of $\mathbb{F}^n$ is finite-dimensional, hence finitely generated, this need not be true of every submodule of R^n for an arbitrary ring R. Second, every (finite-dimensional) vector space V has a basis, and every element of V can be written as a *unique* $\mathbb{F}$-linear combination of basis elements, but this need not be true of every R-module, even if the R-module is finitely generated, as in the following example.

Example 2.7. Let $R = \mathbb{C}[x, y]$ and consider the ideal $I = \langle x, y \rangle$ as an R-module. For clarity, let us call the generators of this R-module $g_1 = x$ and $g_2 = y$. First, I cannot be generated as an R-module by fewer than two elements: If I were generated by a single element, say, f, then we would necessarily have $x = r_1 f$ and $y = r_2 f$ for some $r_1, r_2 \in R$, and thus f would be a common divisor of x and y in R (here we are using the fact that I is both a module and a subset of R). But the GCD of x and y is 1, and the only submodule of R containing 1 is $R \neq I$. So $\{g_1, g_2\}$ is a minimum generating set of I. But not every element of I has a unique representation in terms of this (or, indeed, any) generating set: For example, $xy \in I$ can be written either as $r_1 g_1$ with $r_1 = y$ or $r_2 g_2$ with $r_2 = x$.

A ring R is *Noetherian* if there is no strictly increasing, infinite chain of ideals $I_1 \subsetneq I_2 \subsetneq I_3 \subsetneq \cdots$. Fields are Noetherian (every field has only two ideals: the zero ideal and the whole field), as are the integers $\mathbb{Z}$. Hilbert's Basis Theorem says that every ideal in a Noetherian ring is finitely generated. Hilbert's (other) Basis Theorem says that if R is finitely generated, then so is the polynomial ring $R[x]$ (and hence so is any polynomial ring $R[\vec{x}]$). Quotient rings of Noetherian rings are Noetherian, so every ring that is finitely generated over a field (or more generally, over a Noetherian ring R) is Noetherian.

Similarly, an R-module M is Noetherian if there is no strictly increasing, infinite chain of submodules $M_1 \subsetneq M_2 \subsetneq M_3 \subsetneq \cdots$. If R is Noetherian as a ring, then it is Noetherian as an R-module. It is easily verified that finite direct sums of Noetherian modules are Noetherian, so if R is a Noetherian ring, then it is a Noetherian R-module, and, consequently, R^n is a Noetherian R-module for any finite n. Just as for ideals, every submodule of a Noetherian module is finitely generated.

3 FOUNDATIONAL RESULTS AND SIMULATIONS

3.1 Relation with coMA and coAM

Proposition 3.1 (Cf. Pitassi [81, Theorem 4]). *For any field $\mathbb{F}$, if every propositional tautology has a polynomial-size constant-free $IPS_{\mathbb{F}}$-proof, then* NP $\subseteq$ coMA*, and hence the polynomial hierarchy collapses to its second level.*

This result and its proof are essentially the same as Pitassi [81, Theorem 4]; here we mainly take advantage of history that PIT and coMA are now much more standard than they were in 1996. We also note that the proof allows arbitrary fields, as long as one is careful about the use of constant freeness.

If we wish to drop the restriction of "constant free" (which, recall, is no restriction at all over a finite field), then we may do so either by using the Blum–Shub–Smale analogs of NP and coMA using essentially the same proof or over fields of characteristic zero using the Generalized Riemann Hypothesis (Proposition 3.2).

Proof. Merlin nondeterministically guesses the polynomial-size constant-free IPS proof, and then Arthur must check conditions (1) and (2) of Definition 1.1. (We need constant free so that the algebraic proof has polynomial bit-size and thus can in fact be guessed by a Boolean Merlin.) Both conditions of Definition 1.1 are instances of Polynomial Identity Testing (PIT), which can thus be solved in randomized polynomial time by the standard Schwartz–Zippel–DeMillo–Lipton [29, 89, 109] coRP algorithm for PIT. □

Proposition 3.2. *Over any field $\mathbb{F}$ of characteristic zero, if every propositional tautology has a polynomial-size $IPS_{\mathbb{F}}$-proof, then* NP $\subseteq$ coAM*, assuming the Generalized Riemann Hypothesis.*

The key difference between this result and Proposition 3.1 is that we do not need to assume the proofs are constant free. The price we pay is the use of GRH and that we do not know how to improve this result from coAM to coMA (as in Proposition 3.1). We thank Pascal Koiran for the second half of the proof.

Proof (with P. Koiran). The key fact we will use is that deciding Hilbert's Nullstellensatz—that is, given a system of integer polynomials over $\mathbb{Z}$, deciding if they have a solution over $\mathbb{C}$—is in AM [54]. Rather than looking at solvability of the original set of equations $F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$, we consider solvability of a set of equations whose solutions describe all of the polynomial-size IPS-certficiates for F.

The equations we consider will come from a *generic* polynomial-size circuit; here we use the model of generic circuits from Mulmuley–Sohoni [77, Section 6]. The generic circuit will have

depth $d \leq \text{poly}(n)$ and width $n + m \leq w \leq \text{poly}(n)$, consisting of $d + 1$ layers of gates, the 0th layer consisting of the inputs to a potential IPS certificate—$x_1, \ldots, x_n, y_1, \ldots, y_m$—the dth layer consisting of a single output gate, and each intermediate layer containing w nodes. Each node in level ℓ is connected to every node in level $\ell + 1$. There are also new variables $z_{i,j,k}$. We define the circuit to compute as follows: We use f_k to denote the function computed at gate k. If k is an input gate, then f_k is equal to the appropriate input variable; otherwise, $f_k(\vec{x}, \vec{y}, \vec{z}) \stackrel{def}{=} \sum_{i,j} z_{i,j,k} f_i f_j$, where the sum is over all pairs of gates i, j in the layer immediately preceding the layer of k. The output gate of this generic circuit computes a polynomial $C(\vec{x}, \vec{y}, \vec{z})$, and for any setting of the $z_{i,j,k}$ variables to constants $\zeta_{i,j,k}$, we get a particular polynomial $C_{\vec{\zeta}}(\vec{x}, \vec{y}) \stackrel{def}{=} C(\vec{x}, \vec{y}, \vec{\zeta})$ that is easily seen to be computed by circuits of polynomial size. Furthermore, any function computed by a polynomial-size circuit is equal to $C_{\vec{\zeta}}(\vec{x}, \vec{y})$ for some setting of $\vec{\zeta}$. In particular, there is a polynomial-size IPS proof C' for F if and only if there is some $\vec{\zeta} \in \mathbb{F}^n$ such that $C' = C_{\vec{\zeta}}(\vec{x}, \vec{y})$.

We will translate the conditions that a circuit be an IPS certificate into *equations* on the new z variables. Pick sufficiently many random values $\vec{\xi}^{(1)}, \vec{\xi}^{(2)}, \ldots, \vec{\xi}^{(h)}$ to be substituted into $\vec{x}$. Heintz and Schnorr [42, Theorem 4.4] showed that by picking $h \sim \text{poly}(n)$ random values from $[N]^n$ for some $N \leq \exp(n^{O(1)})$ (which therefore have poly(n) bit-size), with high probability $\{\vec{\xi}^{(1)}, \ldots, \vec{\xi}^{(h)}\}$ will be a hitting set against all n-variable polynomials of circuit-size $\leq \text{poly}(n)$. Then we consider the solvability of the following set of $2h$ equations in $\vec{z}$:

$$\begin{aligned} &(\text{For } i = 1, \ldots, h) \qquad C(\vec{\xi}^{(i)}, \vec{0}, \vec{z}) = 0 \\ &(\text{For } i = 1, \ldots, h) \qquad C(\vec{\xi}^{(i)}, \vec{F}(\vec{\xi}^{(i)}), \vec{z}) = 1. \end{aligned}$$

Determining whether a system of polynomial equations, given by circuits over a field $\mathbb{F}$ of characteristic zero, has a solution in the algebraic closure $\overline{\mathbb{F}}$ can be done in AM [54]. If there is an IPS proof, then let $\vec{\zeta}$ be such that $C_{\vec{\zeta}}(\vec{x}, \vec{y}) = C(\vec{x}, \vec{y}, \vec{\zeta})$ is an IPS proof. Then the preceding equalities will be satisfied regardless of the choices of the $\vec{\xi}^{(i)}$. Conversely, suppose that $\vec{\zeta}$ is a solution to the above system of equations. Since $C(\vec{\xi}^{(i)}, 0, \vec{\zeta}) = 0$ for each $\vec{\xi}^{(i)}$, and $\{\vec{\xi}^{(1)}, \ldots, \vec{\xi}^{(h)}\}$ is a hitting set, it follows that $C(\vec{x}, 0, \vec{\zeta})$ is identically zero as a polynomial in $\vec{x}$. Similarly for $C(\vec{x}, \vec{F}(\vec{x}), \vec{\zeta}) - 1$. Hence, $C_{\vec{\zeta}}(\vec{x}, \vec{y})$ is an IPS proof.

Composing Koiran's AM algorithm for the Nullstellensatz with the random guesses for the $\vec{\xi}^{(i)}$, and assuming that every family of propositional tautologies has polynomial-size IPS certificates, we get an AM algorithm for TAUT. □

3.2 Chasms, Depth Reduction, and Other Circuit Transformations

Recently, many strong depth reduction theorems have been proved for circuit complexity [2, 39, 55, 99], which have been called "chasms" since Agrawal and Vinay [2]. In particular, they imply that sufficiently strong lower bounds against depth 3 or 4 circuits imply super-polynomial lower bounds against arbitrary circuits. Since an IPS proof is just a circuit, these depth reduction chasms apply equally well to IPS proof size. Note that it was not clear to us how to adapt the proofs of these chasms to proofs in the Polynomial Calculus or other previous algebraic systems [82], and indeed this was part of the motivation to move to our more general notion of IPS proof.

Observation 3.3 (Chasms for IPS Proof Size). *If a system of* poly(n) *polynomial equations in n variables has an IPS proof of unsatisfiability of size $s = s(n)$ and (semantic) degree $d = d(n)$, then it also has the following:*

(1) *A* $O(\log d(\log s + \log d))$*-depth IPS proof of size* $\mathrm{poly}(ds)$ *(follows from Valiant–Skyum–Berkowitz–Rackoff [105]);*
(2) *A depth 4 IPS formula proof of size* $n^{O(\sqrt{d})}$ *(follows from Koiran [55]) or a depth 4 IPS proof of size* $2^{O(\sqrt{d \log(ds) \log n})}$ *(follows from Tavenas [99]).*
(3) *(Over fields of characteristic zero) A depth 3 IPS proof of size* $2^{O(\sqrt{d \log d \log n \log s})}$ *(follows from Gupta, Kayal, Kamath, and Saptharishi [39]) or even* $2^{O(\sqrt{d \log n \log s})}$ *(follows from Tavenas [99]).*

This suggests that size lower bounds for IPS proofs in restricted circuit classes would be interesting, even for restricted kinds of depth 3 circuits.

Similarly, since IPS proofs are just circuits, any IPS certificate family of polynomially bounded degree that is computed by a polynomial-size family of algebraic circuits with divisions can also be computed by a polynomial-size family of algebraic circuits without divisions (follows from Strassen [98]). We note, however, that one could in principle consider IPS certificates that were not merely polynomials, but even rational functions, under suitable conditions; divisions for computing these cannot always be eliminated. We discuss this "Rational Ideal Proof System," the exact conditions needed, and when such divisions can be effectively eliminated in Appendix A.

3.3 Definitions of Other Algebraic Proof Systems in Terms of IPS

Previously studied algebraic proof systems can be viewed as particular complexity measures on the Ideal Proof System, including the Polynomial Calculus (or Gröbner) proof system (PC) [27], Polynomial Calculus with Resolution (PCR) [4], the Nullstellensatz proof system [14], and Pitassi's algebraic systems [81, 82], as we explain below.

All of the previous algebraic proof systems are rule-based systems, in that they syntactically enforce the condition that every line of the proof is a polynomial in the ideal of the original polynomials $F_1(\vec{x}), \ldots, F_m(\vec{x})$. Typically, they do this by allowing two derivation rules: (1) from G and H, derive $\alpha G + \beta H$ for α, β constants and (2) from G, derive Gx_i for any variable x_i. By "rule-based circuits," we mean circuits with inputs $y_1, \ldots, y_m$ having linear combination gates and, for each $i = 1, \ldots, n$, gates that multiply their input by x_i. In particular, rule-based circuits necessarily produce Hilbert-like certificates.

In Pitassi's 1998 system [82], a proof is a rule-based derivation of 1, as above, starting from the F_i, with size measured by number of lines. This is essentially the same as the Polynomial Calculus, but with size measured by the number of lines rather than by the total number of monomials appearing.

In Pitassi's 1996 system [81], a proof of the unsatisfiability of $F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$ is a circuit computing a vector $(G_1(\vec{x}), \ldots, G_m(\vec{x}))$ such that $\sum_i F_i(\vec{x})G_i(\vec{x}) = 1$. Size is measured by the size of the corresponding circuit.

Now we come to the definitions of previous algebraic proof systems in terms of complexity measures on the Ideal Proof System:

- Complexity in the Nullstellensatz proof system [14], or "Nullstellensatz degree," is the minimal degree of any Hilbert-like certificate (for systems of equations of constant degree, such as the algebraic translations of tautologies.)
- "Polynomial Calculus size" [27] is the sum of the (semantic) number of monomials at each gate in $C(\vec{x}, \vec{F}(\vec{x}))$, where C ranges over rule-based circuits.
- "PC degree" [27] is the minimum over rule-based circuits $C(\vec{x}, \vec{y})$ of the maximum semantic degree at any gate in $C(\vec{x}, \vec{F}(\vec{x}))$.

- Pitassi's 1998 algebraic proof system [82] is essentially PC, except where size is measured by number of lines of the proof (rather than total number of monomials appearing). This corresponds exactly to the smallest size of any rule-based circuit $C(\vec{x}, \vec{y})$ computing any Hilbert-like IPS certificate. Below we show that this is p-equivalent to $\mathrm{VP}_{\mathrm{det}}$-IPS.
- Polynomial Calculus with Resolution (PCR) [4] also allows variables $\overline{x}_i$ and adds the equations $\overline{x}_i = 1 - x_i$ and $x_i\overline{x}_i = 0$. This is easily accommodated into the Ideal Proof System: Add the $\overline{x}_i$ as new variables, with the same restrictions as are placed on the x_i's in a rule-based circuit, and add the polynomials $\overline{x}_i - 1 + x_i$ and $x_i\overline{x}_i$ to the list of equations F_i. Note that while this may have an effect on the PC size as it can decrease the total number of monomials needed, it has essentially no effect on the number of lines of the proof.
- Pitassi's 1996 algebraic proof system [81] is equivalent to Hilbert-like IPS.

We prove the precise relationships between Pitassi's previous algebraic proof systems [81, 82] and IPS next.

3.4 Number of Lines in Polynomial Calculus Is Equivalent to Determinantal IPS

Recall from Section 2.1 the definitions of $\mathrm{VP}_{\mathrm{det}}$ and VP_{ws}; for readability and ease of speech, we refer to $\mathrm{VP}_{\mathrm{det}}$-IPS = VP_{ws}-IPS as "determinantal IPS" or "det-IPS," for short.

Proposition 3.4. *The number-of-lines measure on PC proofs—equivalent to Pitassi's 1998 algebraic proof system [82]—is p-equivalent to Hilbert-like* det*-IPS or* VP_{ws}*-IPS.*
Furthermore, Pitassi's 1996 algebraic proof system [81] is p-equivalent to Hilbert-like IPS.

In light of this proposition, we henceforth refer to the systems from Pitassi [81] and [82] as Hilbert-like IPS and Hilbert-like det-IPS, respectively. Pitassi [81, Theorem 5] showed that Hilbert-like IPS p-simulates Polynomial Calculus and Frege. Essentially, the same proof shows that Hilbert-like IPS p-simulates Extended Frege as well. Unfortunately, the proof of the simulation in Pitassi [81] does not seem to generalize to give a depth-preserving simulation. In Section 3.5, we show there is indeed a depth-preserving simulation (Theorem 3.5).

Proof. We start with the proof of the second statement, as its proof is a simpler version of the proof of the first statement.

Let C be a proof in the 1996 system [81], namely a circuit computing $(G_1(\vec{x}), \ldots, G_m(\vec{x}))$. Then with m product gates and a single fan-in-m addition gate, we get a circuit C' computing the Hilbert-like IPS certificate $\sum_{i=1}^{m} y_i G_i(\vec{x})$.

Conversely, if C' is a Hilbert-like IPS-proof computing the certificate $\sum_i y_i G'_i(\vec{x})$, then by Baur–Strassen [9] there is a circuit C of size at most $O(|C'|)$ computing the vector $(\frac{\partial C'}{\partial y_1}, \ldots, \frac{\partial C'}{\partial y_m}) = (G'_1(\vec{x}), \ldots, G'_m(\vec{x}))$, which is exactly a proof in the 1996 system. (Alternatively, more simply, but at slightly more cost, we may create m copies of C', and in the ith copy of C' plug in 1 for one of the y_i and 0 for all of the others.

The proof of the first statement takes a bit more work. At this point, the reader may wish to recall the definition of weakly skew circuit from Section 2.1.

Suppose we have a derivation of 1 from $F_1(\vec{x}), \ldots, F_m(\vec{x})$ in the 1998 system [82]. First, replace each $F_i(\vec{x})$ at the beginning of the derivation with the corresponding placeholder variable y_i. Since size in the 1998 system is measured by number of lines in the proof, this has not changed the size. Furthermore, the final step no longer derives 1 but rather derives an IPS certificate. By structural induction on the two possible rules, one easily sees that this is in fact a Hilbert-like IPS-certificate. Convert each linear combination step into a linear combination gate and each "multiply by x_i" step into a product gate one of whose inputs is a new leaf with the variable x_i. As we create a new

leaf for every application of the product rule, these new leaves are clearly cut off from the rest of the circuit by removing their connection to their product gate. As these are the only product gates introduced, we have a weakly skew circuit computing a Hilbert-like IPS certificate.

The converse takes a bit more work, so we first show that a Hilbert-like *formula*-IPS proof can be converted at polynomial cost into a proof in the 1998 system [82] and then explain why the same proof works for VP_{ws}-IPS. This proof is based on a folklore result (see the remark after Definition 2.6 in Raz–Tzameret [85]); we thank Iddo Tzameret for a conversation clarifying it, which led us to realize that the result also applies to weakly skew circuits.

Let C be a formula computing a Hilbert-like IPS-certificate $\sum_{i=1}^{m} y_i G_i(\vec{x})$. Using the trick above of substituting in $\{0,1\}$-values for the y_i (one 1 at a time), we find that each $G_i(\vec{x})$ can be computed by a formula Γ_i no larger than $|C|$. For each i we show how to derive $F_i(\vec{x})G_i(\vec{x})$ in the 1998 system. These can then be combined using the linear combination rule. Thus, for simplicity, we drop the subscript i and refer to y, $F(\vec{x})$, $G(\vec{x})$, and the formula Γ computing G. Without loss of generality (with a polynomial blow-up if needed), we can assume that all of Γ's gates have fan-in at most 2.

We proceed by induction on the size of the formula Γ. Our inductive hypothesis is as follows: For all formulas Γ' of size $|\Gamma'| < |\Gamma|$, for all polynomials $P(\vec{x})$, in the 1998 system one can derive $P(\vec{x})\Gamma'(\vec{x})$ starting from $P(\vec{x})$, using at most $|\Gamma'|$ lines. The base case is $|\Gamma| = 1$, in which case $G(\vec{x})$ is a single variable x_i, and from $P(\vec{x})$ we can compute $P(\vec{x})x_i$ in a single step using the variable-product rule.

If Γ has a linear combination gate at the top, say, $\Gamma = \alpha\Gamma_1 + \beta\Gamma_2$. By induction, from $P(\vec{x})$, we can derive $P(\vec{x})\Gamma_i(\vec{x})$ in $|\Gamma_i|$ steps for $i = 1, 2$. Do those two derivations and then apply the linear combination rule to derive $\alpha P(\vec{x})\Gamma_1(\vec{x}) + \beta P(\vec{x})\Gamma_2(\vec{x}) = P(\vec{x})\Gamma(\vec{x})$ in one additional step. The total length of this derivation is then $|\Gamma_1| + |\Gamma_2| + 1 = |\Gamma|$.

If Γ has a product gate at the top, say, $\Gamma = \Gamma_1 \times \Gamma_2$. Unlike the case of linear combinations where we proceeded in parallel, here we proceed sequentially and use more of the strength of our inductive assumption. Starting from $P(\vec{x})$, we derive $P(\vec{x})\Gamma_1(\vec{x})$ in $|\Gamma_1|$ steps. Now, starting from $P'(\vec{x}) = P(\vec{x})\Gamma_1(\vec{x})$, we derive $P'(\vec{x})\Gamma_2(\vec{x})$ in $|\Gamma_2|$ steps. But $P'\Gamma_2 = P\Gamma_1\Gamma_2 = P\Gamma$, which we derived in $|\Gamma_1| + |\Gamma_2| \leq |\Gamma|$ steps. This completes the proof of this direction for Hilbert-like *formula*-IPS.

For Hilbert-like weakly skew IPS the proof is similar. However, because gates can now be reused, we must also allow lines in our constructed proof to be reused (otherwise, we would effectively be unrolling our weakly skew circuit into a formula, for which the best-known upper bound is only quasi-polynomial). We still induct on the size of the weakly skew circuit, but now we allow circuits with multiple outputs. We change the induction hypothesis to the following: for all weakly skew circuits Γ' of size $|\Gamma'| < |\Gamma|$, possibly with multiple outputs that we denote $\Gamma'_{out,1}, \ldots, \Gamma'_{out,s}$, from any $P(\vec{x})$ one can derive the tuple $P\Gamma'_{out,1}, \ldots, P\Gamma'_{out,s}$ in the 1998 system using at most $|\Gamma'|$ lines.

To simplify matters, we assume that every multiplication gate in a weakly skew circuit has a label indicating which one of its children is separated from the rest of the circuit by this gate.

The base case is the same as before, since a circuit of size one can only have one output, a single variable.

Linear combinations are similar to before, except now we have a multi-output weakly skew circuit of some size, say, s, that outputs Γ_1 and Γ_2. By the induction hypothesis, there is a derivation of size $\leq s$ that derives both $P\Gamma_1$ and $P\Gamma_2$. Then we apply one additional linear combination rule, as before.

For a product gate $\Gamma = \Gamma_1 \times \Gamma_2$, suppose without loss of generality that Γ_2 is the child that is isolated from the larger circuit by this product gate (recall that we assumed Γ comes with an indicator of which child this is). Then we proceed as before, first computing $P\Gamma_1$ from P and then $(P\Gamma_1)\Gamma_2$ from $(P\Gamma_1)$. Because we apply "multiplication by Γ_1" and "multiplication by Γ_2" in sequence,

it is crucial that the gates computing Γ_2 do not depend on those computing Γ_1, for the gates g in Γ_1 get translated into lines computing Pg, and if we reused *that* in computing Γ_2, rather than getting g as needed, we would be getting Pg. □

It is interesting to note that the condition of being weakly skew is precisely the condition we needed to make this proof go through.

3.5 Depth-Preserving Simulation of Frege Systems by the Ideal Proof System

Throughout this section, all algebraic circuits may have linear combination gates—with weights on their incoming edges (see Section 2.1)—and product gates of unbounded fan-in. We measure the size of all circuits C (Boolean and algebraic), denoted size(C), by the number of gates. The depth of a circuit C (Boolean or algebraic), denoted depth(C), is the maximum number of gates encountered on any path from a leaf to the root.

Theorem 3.5. *Let p be prime and $\mathbb{F}$ any field of characteristic p. Then $IPS_{\mathbb{F}}$ p-simulates Frege with MOD_p connectives in such a way that depth-d Frege proofs are simulated by depth-$O(d)$ $IPS_{\mathbb{F}}$ proofs.*

In particular, $\mathsf{AC}^0[p]$-Frege is p-simulated by bounded-depth $IPS_{\mathbb{F}}$, and Frege is p-simulated by logarithmic-depth $IPS_{\mathbb{F}}$, i.e., $\mathsf{VNC}^1_{\mathbb{F}}$-IPS.

The "$O(d)$" above is $\lesssim 4d$; in a forthcoming preprint [38], we prove a tighter depth-preserving simulation, which has the advantage of drawing connections to depth-six algebraic circuits; the two proofs follow the same outline, but the tighter result requires several new technical ingredients.

Proof. We will use a small modification of the sequent-calculus formalization of $\mathsf{AC}^0[p]$-Frege as given by Maciel and Pitassi [66]. Changing between a Frege system and sequent calculus does not increase the depth by more than an additive constant. The underlying connectives are unbounded fan-in OR, unbounded fan-in MOD_p^i for $i \in \{0, \ldots, p-1\}$, and unary negation. The inputs are x_i and the constants 0,1.

We will work in a sequent calculus style proof system, where lines are cedents of the form $\Gamma \to \Delta$, where both Γ and Δ are sets of $\{\vee, \neg, MOD_p^0, \ldots, MOD_p^{p-1}\}$-formulas whose inputs are $x_i, 0, 1$, where each of $\Gamma_i \in \Gamma$ and $\Delta_i \in \Delta$ has depth at most d; the intended meaning is that the conjunction of the formulas in Γ implies the disjunction of the formulas in Δ. The commutativity of the arguments to each connective is implicit. Throughout we use Γ, Δ to denote sets of formulae and A, A_i to denote individual formulae. Although Γ and Δ are sets, we use sequence notation for convenience, viz. "Δ, A" actually means "$\Delta \cup \{A\}$."

Because we are working with gates of unbounded fan-in, we use the prefix notation $\vee(A_1, \ldots, A_k)$ for a single OR gate whose inputs are $A_1, \ldots, A_k$, and similarly $MOD_p^i(A_1, \ldots, A_k)$. In particular, $\vee()$ is the OR with no inputs, which is equal to 0 by convention, $MOD_p^0()$ is equal to 1 by convention, and $MOD_p^i()$ for $i \neq 0$ is equal to 0 by convention.

The axioms are as follows:

(1) $A \to A$
(2) $\vee() \to$
(3) $\to MOD_p^0()$
(4) $MOD_p^i() \to$ for $i \neq 0$.

The rules of inference are as follows; throughout, "$i-1$" is to be interpreted modulo p.

$$\text{Weakening}\quad \frac{\Gamma \to \Delta}{\Gamma \to \Delta, A} \qquad \frac{\Gamma \to \Delta}{A, \Gamma \to \Delta} \qquad \text{Cut}\ \frac{\to A, \Gamma \qquad A \to \Gamma}{\to \Gamma}$$

$$\text{Negation}\quad \frac{\Gamma, A \to \Delta}{\Gamma \to \neg A, \Delta} \qquad \frac{\Gamma \to A, \Delta}{\Gamma, \neg A \to \Delta}$$

$$\text{Or-Left}\quad \frac{A_1, \Gamma \to \Delta \qquad \vee(A_2, \ldots, A_k), \Gamma \to \Delta}{\vee(A_1, \ldots, A_k), \Gamma \to \Delta}$$

$$\text{Or-Right}\quad \frac{\Gamma \to A_1, \vee(A_2, \ldots, A_k), \Delta}{\Gamma \to \vee(A_1, \ldots, A_k), \Delta}$$

$$\text{Mod-}p\text{-Left}\quad \frac{A_1, MOD_p^{i-1}(A_2, \ldots, A_k), \Gamma \to \Delta \qquad \neg A_1, MOD_p^{i}(A_2, \ldots, A_k), \Gamma \to \Delta}{MOD_p^{i}(A_1, \ldots, A_k), \Gamma \to \Delta}$$

$$\text{Mod-}p\text{-Right}\quad \frac{\Gamma \to \neg A_1, MOD_p^{i-1}(A_2, \ldots, A_k), \Delta \qquad \Gamma \to A_1, MOD_p^{i}(A_2, \ldots, A_k), \Delta}{\Gamma \to MOD_p^{i}(A_1, \ldots, A_k), \Delta}$$

$$\text{Mod-}p\text{ Constants}\quad \frac{\Gamma \to MOD_p^{i}(1, A_2, \ldots, A_k), \Delta}{\Gamma \to MOD_p^{i-1}(A_2, \ldots, A_k), \Delta} \qquad \frac{\Gamma \to MOD_p^{i}(0, A_2, \ldots, A_k), \Delta}{\Gamma \to MOD_p^{i}(A_2, \ldots, A_k), \Delta}.$$

A refutation of a 3CNF formula $\varphi = \kappa_1 \wedge \kappa_2 \wedge \cdots \wedge \kappa_m$ in Frege with mod p connectives is a sequence of cedents, where each cedent is either one of the κ_i's, or an instance of an axiom scheme or follows from two earlier cedents by one of the above inference rules, and the final cedent is the empty cedent.

We define a translation $t(A)$ from Boolean formulas to algebraic formulae over $\mathbb{F}$ such that for any assignment α, $A(\alpha) = 1$ if and only if $t(A)(\alpha) = 0$. The translation is defined inductively as follows:

(1) $t(x) = 1 - x$ for x atomic (a Boolean variable),
(2) $t(\neg A) = 1 - t(A)$,
(3) $t(\vee(A_1, \ldots, A_k)) = t(A_1)t(A_2)\cdots t(A_k)$,
(4) $t(MOD_p^i(A_1, \ldots, A_k)) = (k - i - t(A_1) - t(A_2) \cdots - t(A_k))^{p-1}$.

Note that

$$\text{depth}(t(A)) \leq 2\text{depth}(A) + 1 \qquad \text{and} \qquad \text{size}(t(A)) \leq 2\text{size}(A) + 1.$$

The factor of 2 comes from (4), and the +1 comes from (1). (Here we are counting negation gates towards the size and depth of a Boolean circuit; even if we do not, we will still incur a constant-factor increase in the size and depth, since we may assume without loss of generality that there are no two successive negation gates).

For a cedent $\Gamma \to \Delta$, we will translate the cedent by moving everything to the right of the arrow. That is, the cedent $L = A_1, \ldots, A_k \to B_1, \ldots, B_\ell$ will be translated to $t(L) = t(\neg A_1 \vee \cdots \vee \neg A_k \vee B_1 \vee \cdots \vee B_\ell) = (1 - t(A_1))(1 - t(A_2)) \cdots (1 - t(A_k))t(B_1) \cdots t(B_\ell)$. This may again increase the depth by 1, since the product gate used to simulate the $\to$ was not counted in the depth of the A_i, B_i.

Let R be a Frege refutation (with mod p connectives) of φ. Without loss of generality, we may assume that R is treelike. Recall that a Frege (or sequent calculus) proof is *treelike* if the underlying directed acyclic graph structure of the proof is a tree, and therefore every cedent in the refutation, other than the final empty cedent, is used exactly once. Any Frege proof can be efficiently converted into a treelike proof at a polynomial increase in size, and increasing the depth by one

Krajíček [58, Proposition 1.1] (although not stated there for Frege with mod p connectives, the same proof still works, see Segerlind [90, Section 3.2]).

We will prove by induction on the number of cedents of R that for each cedent L in the refutation, we can derive $t(L)$ via a Hilbert-like IPS proof (see Definition 1.8) whose size is polynomial in the original size, and whose depth is at most a constant factor greater than the depth of R.

For the base case, each initial cedent of the form $\to \kappa_i$ translates to y_i, and thus has the right form.

Axioms (2), (3), and (4) translate to the identically zero polynomial, so they have the right form. The axiom $A \to A$ translates to $t(A)(1 - t(A))$; we need to show that $t(A)(1 - t(A))$ can be derived from the Boolean axioms $x_i^2 - x_i$ by an IPS proof of appropriate size and depth.

Lemma 3.6. *Let p be any prime and $\mathbb{F}$ any field of characteristic p. For any Boolean formula A of size s and depth d with connectives $\neg$, $\vee$, MOD_p^0, $\ldots$, MOD_p^{p-1}, $t(A)(1 - t(A))$ can be derived from $\{x_i^2 - x_i : i \in [n]\}$ by a Hilbert-like IPS$_{\mathbb{F}}$ derivation of size $O(s^2)$ and depth $O(d)$.* □

Proof. To make our formulae clearer, we write $b(A_j)$ for the IPS circuit that derives $t(A_j)^2 - t(A_j)$ from the placeholder variables y_i for the Boolean axioms $x_i^2 - x_i$. We build up the IPS circuit starting from the leaves (inputs) of A; in fact, we will derive $t(g)^2 - t(g)$ for every gate g of A. Our IPS circuit will include a single copy of the (natural) circuit for $t(A)$; whenever we write $t(g_i)$ inside an expression for some $b(g)$, we mean to use the gate $t(g_i)$ in this single copy of the circuit for $t(A)$. This incurs an additive cost of $\text{size}(t(A)) \leq \text{size}(A)$ and means the depth of $b(A)$ is at least $\text{depth}(t(A)) \leq 2\text{depth}(A) + 1$.

Case 0: For an input gate x_i, we have $t(x_i)(1 - t(x_i)) = (1 - x_i)x_i = x_i^2 - x_i$, so its IPS derivation is just $b(x_i) = y_i$. Both the input gate and its IPS derivation have depth zero and size one (or size zero, depending on how you count, but either way will not affect the rest of the result).

Case 1: For a negation gate, say, $g = \neg g_1$. Then $t(g)(1 - t(g)) = (1 - t(g_1))t(g_1)$, so $b(g) = b(g_1)$ and the depth and size do not increase at all.

Case 2: $g = \vee(g_1, \ldots, g_k)$. First, we claim that the following is a polynomial identity which holds over any ring:

$$\left(\prod_{i=1}^{k} z_i\right)^2 - \prod_{i=1}^{k} z_i = \sum_{i=1}^{k} (z_i^2 - z_i)\left(\prod_{j<i} z_j^2\right)\left(\prod_{j>i} z_j\right). \tag{1}$$

This is readily verified by induction on the number of variables. Indeed, for $k = 1$, (1) becomes $z_1^2 - z_1 = z_1^2 - z_1$. For $k > 1$, suppose inductively that (1) holds for $k - 1$, then we have

$$\left(\prod_{i=1}^{k-1} z_i\right)^2 - \prod_{i=1}^{k-1} z_i = \sum_{i=1}^{k-1} (z_i^2 - z_i)\left(\prod_{1\leq j<i} z_j^2\right)\left(\prod_{i<j\leq k-1} z_j\right)$$

$$z_k\left(\prod_{i=1}^{k-1} z_i\right)^2 - \prod_{i=1}^{k} z_i = \sum_{i=1}^{k-1} (z_i^2 - z_i)\left(\prod_{1\leq j<i} z_j^2\right)\left(\prod_{i<j\leq k} z_j\right) \qquad \text{(multiply by } z_k\text{)}$$

$$z_k\left(\prod_{i=1}^{k-1} z_i\right)^2 - \prod_{i=1}^{k} z_i + (z_k^2 - z_k)\prod_{j<k} z_j^2 = \sum_{i=1}^{k} (z_i^2 - z_i)\left(\prod_{1\leq j<i} z_j^2\right)\left(\prod_{i<j\leq k} z_j\right) \qquad \text{(add to both sides)}$$

$$\left(\prod_{i=1}^{k} z_i\right)^2 - \prod_{i=1}^{k} z_i = \sum_{i=1}^{k} (z_i^2 - z_i)\left(\prod_{j<i} z_j^2\right)\left(\prod_{j>i} z_j\right),$$

as claimed.

Now, as $t(g) = \prod_{i=1}^{k} t(g_i)$, plugging in $t(g_i)$ for z_i into the above identity, we find that

$$b(g) = \sum_{i=1}^{k} b(g_i) \left(\prod_{j<i} t(g_j)^2 \right) \left(\prod_{j>i} t(g_j) \right)$$

is an IPS derivation of $t(g)^2 - t(g)$ from the Boolean axioms. The preceding formula shows that we increase both the depth and size by at most 2.

Case 3: $g = MOD_p^i(g_1, \ldots, g_k)$. The idea of the proof in this case as follows: First, we show that adding up a bunch of $\{0,1\}$ values over $\mathbb{F}$, possibly with a constant from $\{0, \ldots, p-1\}$, results in an element z of the prime field $\mathbb{F}_p$ (which sits inside any field $\mathbb{F}$ of characteristic p); in other words, we derive $z^p - z$, where $z = -i + \sum_j t(g_j)$. (Note that the proof holds for symbolic polynomials in characteristic p, not only with the corresponding functions, so it is allowed within the framework of $\mathrm{IPS}_{\mathbb{F}}$.) Then we show that $z^{p-1} \in \{0,1\}$ by noting that $(z^{p-1})^2 - z^{p-1} = z^{2p-2} - z^{p-1} = z^{p-2}(z^p - z)$.

Let us now implement the preceding idea carefully: Let $z = -i + \sum_j t(g_j)$. Then we have

$$\begin{aligned} z^p - z &= \left(-i + \sum_j t(g_j) \right)^p - \left(-i + \sum_j t(g_j) \right) \\ &= (-i)^p + \sum_j t(g_j)^p - \left(-i + \sum_j t(g_j) \right) \\ &= ((-i)^p + i) + \sum_j (t(g_j)^p - t(g_j)), \end{aligned}$$

where the second inequality follows from the fact that for any two polynomials f, g in a field of characteristic p, $(f+g)^p = f^p + g^p$, symbolically. The first term, $(-i)^p + (-i)$, is identically zero, since $i \in \{0, \ldots, p-1\}$. The remaining terms can be derived as follows:

$$\begin{aligned} t(g_j)^p - t(g_j) &= \left(\sum_{\ell=0}^{p-2} t(g_j)^\ell \right) (t(g_j)^2 - t(g_j)) \\ &= \sum_{\ell=0}^{p-2} t(g_j)^\ell b(g_j). \end{aligned}$$

Putting these together with the idea outlined above, we then get our derivation of $(z^{p-1})^2 - z^{p-1}$ as

$$b(g) = \left(-i + \sum_j t(g_j) \right)^{p-2} \left(\sum_{j \in [k]} \sum_{\ell=0}^{p-2} t(g_j)^\ell b(g_j) \right). \tag{2}$$

The preceding formula shows that we can derive $b(g)$ by a formula whose depth has increased from that of $t(g_j), b(g_j)$ by at most 3 (the formula is a product of sums of products of the $t(g_j), b(g_j)$) and whose size has increased by at most $k(p-1) + 3 \le O(ps)$.

In total, we have included a copy of $t(A)$, and for each gate we add at most $O(ps) = O(s)$ gates to our IPS derivation, so the total size is at most $s + sO(s) = O(s^2)$, and the depth has increased by a factor of at most 3. □

The preceding lemma handled the only nontrivial axiom; we now conclude the proof of Theorem 3.5. For the inductive step, it is a matter of going through all of the rules. We assume inductively that we have an IPS proof of appropriate size and depth of all the antecedents.

(1) (Weakening) Let $C(\vec{x},\vec{y})$ be an Hilbert-like IPS derivation of $t(\Gamma \to \Delta)$. We want to obtain a derivation of $t(\Gamma \to \Delta, A)$. Since we move everything to the right when we translate, this is equivalent to showing that if C is a derivation of $t(\to A_1,\ldots,A_k) = t(A_1)t(A_2)\cdots t(A_k)$, that we can obtain a derivation of $t(\to A_1,\ldots,A_k,B) = t(A_1)t(A_2)\cdots t(A_k)t(B)$. Multiplying C by $t(B)$ achieves this and is still Hilbert-like. The depth has become $1+\max\{\text{depth}(C),\text{depth}(t(B))\}$, and the size has increased by at most $\text{size}(t(B))+1$.

(2) (Cut) Let $C(\vec{x},\vec{y})$ be a Hilbert-like IPS derivation of $t(\to \neg A, B_1,\ldots,B_k) = (1-t(A))t(B_1)\cdots t(B_k)$ and $C'(\vec{x},\vec{y})$ be a Hilbert-like IPS derivation of $t(\to A, B_1,\ldots B_k) = t(A)t(B_1)\cdots t(B_k)$. We want to derive $t(\to B_1 \ldots B_k) = t(B_1)\cdots t(B_k)$, which is easily done as $C + C'$; the depth and size have both increased by 1, and the result is still Hilbert-like.

(3) (Negation) Because our translation moves everything to the right, the translated versions become syntactically identical, and there is nothing to do for the negation rules.

(4) (Or-Left) Let $C(\vec{x},\vec{y})$ be a derivation of $t(\to \neg A_1,\Delta)$, and $C'(\vec{x},\vec{y})$ be a derivation of $t(\to \neg \vee (A_2,\ldots,A_k),\Delta)$. We want to derive $t(\to \neg \vee (A_1,\ldots,A_k),\Delta)$. We have

$$C(\vec{x},\vec{F}) = t(\to \neg A_1,\Delta) = (1-t(A_1))t(\Delta),$$

$$C'(\vec{x},\vec{F}) = t(\to \neg \vee (A_2,\ldots,A_k),\Delta) = (1-t(A_2)t(A_3)\cdots t(A_k))t(\Delta).$$

The desired derivation is $C'\cdot t(A_1) + C$, which increases the size and depth by at most 2.

(5) (Or-Right) The translation of the derived formula is syntactically identical to the original formula, so there is nothing to do.

(6) (Mod-p-Right) Let C be a derivation of $t(\to \neg A_1, MOD_p^{i-1}(A_2,\ldots,A_k),\Delta)$ and C' be a derivation of $t(\to A_1, MOD_p^{i}(A_2,\ldots,A_k),\Delta)$. We want to derive $t(\to MOD_p^{i}(A_1,\ldots,A_k),\Delta)$.

For notational convenience, let $a = 1-t(A_1)$ and let $b = (k-i) - t(A_2) - t(A_3) - \cdots - t(A_k)$. In terms of a and b, our antecedents and consequent are

$$t(\to \neg A_1, MOD_p^{i-1}(A_2,\ldots,A_k),\Delta) = a(b+1)^{p-1}t(\Delta)$$
$$t(\to A_1, MOD_p^{i}(A_2,\ldots,A_k),\Delta) = (1-a)b^{p-1}t(\Delta)$$
$$t(\to MOD_p^{i}(A_1,\ldots,A_k),\Delta) = (a+b)^{p-1}t(\Delta).$$

Next, we rewrite the consequent $(a+b)^{p-1}t(\Delta)$ in a way that tries to leverage the antecedents as much as possible. We start by adding and subtracting $t(\to \neg A_1, MOD_p^{i-1}(A_2,\ldots,A_k),\Delta)$, and then expand and combine terms, we find that $(a+b)^{p-1}t(\Delta)$ is equal to

$$\begin{aligned}
&= \left((a+b)^{p-1} - a(b+1)^{p-1} + a(b+1)^{p-1}\right)t(\Delta)\\
&= \left(\sum_{k=0}^{p-1}\binom{p-1}{k}b^k a^{p-1-k} - a\sum_{k=0}^{p-1}\binom{p-1}{k}b^k 1^{p-1-k} + a(b+1)^{p-1}\right)t(\Delta)\\
&= \left(\sum_{k=0}^{p-1}\binom{p-1}{k}b^k (a^{p-1-k}-a) + a(b+1)^{p-1}\right)t(\Delta).
\end{aligned}$$

Every term of the sum has $a^{p-1-k}-a$ as a factor, and for $k < p-2$, we have $p-1-k \geq 2$, so in these cases we may derive $a^{p-1-k}-a$ from a^2-a as $a^{p-1-k}-a = (\sum_{\ell=0}^{p-k-3} a^\ell)(a^2-a)$, and we can derive a^2-a using Lemma 3.6. For $k = p-2$, note that the term is identically zero, since it is multiplied by $a-a=0$. And for $k = p-1$, the term is $b^{p-1}(1-a)$,

which is exactly our other antecedent. Thus we are left with a derivation of the following form:

$$\left(\sum_{k=0}^{p-3}\binom{p-1}{k}b^k\left(\sum_{\ell=0}^{p-k-3}a^\ell\right)(a^2-a)+(1-a)b^{p-1}+(1-a)b^{p-1}\right)t(\Delta)$$
$$=\left(\sum_{k=0}^{p-3}\binom{p-1}{k}b^k\left(\sum_{\ell=0}^{p-k-3}a^\ell\right)(a^2-a)\right)t(\Delta)+C+C'. \tag{3}$$

The preceding derivation is a ΣΠΣΠ formula whose inputs are a, b, the derivation of a^2-a from Lemma 3.6, and the derivations C, C' of the antecedents. Thus, the depth has increased by at most 4, and the size has increased by at most $O(p^2)=O(1)$.

(7) (Mod-p-Left) This case is similar to Mod-p-Right, but with "1−" floating around in various places; because of the latter, we write it out here, but we do so quickly, since the explanation for the derivation is essentially identical to the preceding case. Let $a=1-t(A_1)$ and $b=k-i-t(A_2)-\cdots-t(A_k)$, as above. Then from

$$t(A_1, MOD_p^{i-1}(A_2,\ldots,A_k)\to\Delta)=a(1-(b+1)^{p-1})t(\Delta) \text{ and}$$
$$t(\neg A_1, MOD_p^{i}(A_2,\ldots,A_k)\to\Delta)=(1-a)(1-b^{p-1})t(\Delta),$$

we want to derive

$$t(\to MOD_p^{i}(A_1,\ldots,A_k),\Delta)=(1-(a+b)^{p-1})t(\Delta).$$

Deriving as before, we begin with the conclusion and modify it to see how to derive it:

$$\begin{aligned}(1-(a+b)^{p-1})t(\Delta)&=\left(1-(a+b)^{p-1}-a(1-(b+1)^{p-1})+a(1-(b+1)^{p-1})\right)t(\Delta)\\&=\left(1-a+a\sum_{k=0}^{p-1}\binom{p-1}{k}b^k-\sum_{k=0}^{p-1}\binom{p-1}{k}b^k a^{p-1-k}+a(1-(b+1)^{p-1})\right)t(\Delta)\\&=\left(1-a-\sum_{k=0}^{p-1}\binom{p-1}{k}b^k(a^{p-1-k}-a)+a(1-(b+1)^{p-1})\right)t(\Delta)\\&=\left(1-a-b^{p-1}(1-a)-\sum_{k=0}^{p-3}\binom{p-1}{k}b^k(a^{p-1-k}-a)+a(1-(b+1)^{p-1})\right)t(\Delta)\\&=\left(\left[(1-a)(1-b^{p-1})\right]-\sum_{k=0}^{p-3}\binom{p-1}{k}b^k(a^{p-1-k}-a)+\left[a(1-(b+1)^{p-1})\right]\right)t(\Delta).\end{aligned}$$

As with Mod-p-Right, the depth increases by at most 4 and the size by at most $O(p^2)$.

(8) (Mod-p Constants) The translation of the derived formula is syntactically identical to the original formula, so there is nothing to do.

At each step, we have increased the depth by at most 4 and, except for the axiom $A\to A$, the size by at most a constant as well. Since Lemma 3.6 can increase the size quadratically, our overall size increase is quadratic, and the depth has been multiplied by at most 4. □

3.6 General versus Hilbert-like IPS

Proposition 3.7. *Let $F_1=\cdots=F_m=0$ be a polynomial system of equations in n variables $x_1,\ldots,x_n$, and let $C(\vec{x},\vec{y})$ be an IPS-certificate of the unsatisfiability of this system. Let*

$D = \max_i \deg_{y_i} C$ and let t be the number of terms of C, when viewed as a polynomial in the y_i with coefficients in $\mathbb{F}[\vec{x}]$. Suppose C and each F_i can be computed by a circuit of size $\leq s$. Then a Hilbert-like IPS-certificate for this system can be computed by a circuit of size $\mathrm{poly}(D, t, n, s)$.[4]

The proof uses known sparse multivariate polynomial interpolation algorithms. The threshold T is essentially the number of points at which the polynomial must be evaluated in the course of the interpolation algorithm. Here we use one of the early, elegant interpolation algorithms due to Zippel [110]. Although Zippel's algorithm chooses random points at which to evaluate polynomials for the interpolation, in our nonuniform setting it suffices merely for points with the required properties to exist (which they do as long as $|\mathbb{F}| \geq T$). Better bounds may be achievable using more recent interpolation algorithms such as those of Ben-Or and Tiwari [16] or Kaltofen and Yagati [51]. We note that all of these interpolation algorithms only give limited control on the *depth* of the resulting Hilbert-like IPS-certificate (as a function of the depth of the original IPS-certificate f), because they all involve solving linear systems of equations, which is not known to be computable efficiently in constant depth.

Forbes, Shpilka, Tzameret, and Wigderson [34] subsequently improved on this result; see Section 8.

Proof. Using a sparse multivariate interpolation algorithm such as Zippel's [110], for each monomial in the placeholder variables $\vec{y}$ that appears in C, there is a polynomial-size algebraic circuit for its coefficient, which is an element of $\mathbb{F}[\vec{x}]$. For each such monomial $\vec{y}^{\vec{e}} = y_1^{e_1} \cdots y_m^{e_m}$, with coefficient $c_{\vec{e}}(\vec{x})$, there is a small circuit C' computing $c_{\vec{e}}(\vec{x})\vec{y}^{\vec{e}}$. Since every $\vec{y}$-monomial appearing in C is non-constant, at least one of the exponents $e_i > 0$. Let i_0 be the least index of such an exponent. Then we get a small circuit computing $c(\vec{e})(\vec{x})y_{i_0}F_{i_0}(\vec{x})^{e_{i_0}-1}F_{i_0+1}(\vec{x})^{e_{i_0+1}} \cdots F_m(\vec{x})^{e_m}$ as follows. Divide C' by y_{i_0}, and then eliminate this division using Strassen [98] (or, alternatively, consider $\frac{1}{e_{i_0}} \frac{\partial C'}{\partial y_{i_0}}$ using Baur–Strassen [9]). In the resulting circuit, replace each input y_i by a small circuit computing $F_i(\vec{x})$. Then multiply the resulting circuit by y_{i_0}. Repeat this procedure for each monomial appearing (the list of monomials appearing in C is one of the outputs of the sparse multivariate interpolation algorithm), and then add them all together. □

4 LOWER BOUNDS ON IPS IMPLY CIRCUIT LOWER BOUNDS

Theorem 1.2. *A super-polynomial lower bound on [constant-free] Hilbert-like* VP*-IPS*$_R$ *proofs of any family of tautologies implies* $\mathsf{VNP}_R \neq \mathsf{VP}_R$ *[respectively,* $\mathsf{VNP}^0_R \neq \mathsf{VP}^0_R$*], for any ring R.*

A super-polynomial lower bound on the number of lines in Polynomial Calculus proofs implies the Permanent versus Determinant Conjecture ($\mathsf{VNP} \neq \mathsf{VP}_{ws}$*).*

Together with Proposition 3.1, this immediately gives an alternative, and we believe simpler, proof of the following result:

Corollary 4.1. *If* $\mathsf{NP} \not\subseteq \mathsf{coMA}$*, then* $\mathsf{VNP}^0_{\mathbb{F}} \neq \mathsf{VP}^0_{\mathbb{F}}$*, for any field* $\mathbb{F}$*.*

The previous proofs we are aware of all depend crucially on the random self-reducibility of the permanent or of some function complete for $\mathsf{Mod}_p\mathsf{P}/\mathrm{poly}$. In contrast, our proof is quite different, in that it avoids random self-reduciblity altogether and does not need any completeness results: Indeed, we do not even know if there exist tautologies and a choice of ordering of the clauses such that the VNP-IPS certificates of Lemma 4.2 are random self-reducible nor (separately) VNP-complete.

[4] If the base field $\mathbb{F}$ has size less than $T = Dt\binom{n}{2}$, and the original circuit had multiplication gates of fan-in bounded by k, then the size of the resulting Hilbert-like certificate should be multiplied by $(\log T)^k$.

For comparison, here is a brief sketch of three previous proofs (we thank Lance Fortnow for one and an anonymous reviewer for the other two). Note that all of these proofs rely on several seminal results from computational complexity (all of them rely on Valiant's completeness result [102], and each relies on a subset of References [7, 23, 33, 50, 100, 102, 106]), whereas our proof uses little more than the Nullstellensatz, a result over 100 years old.

Proof 1: This proof seems to only work when $\mathbb{F}$ is a finite field or, assuming the Generalized Riemann Hypothesis, a field of characteristic zero. First, Bürgisser's results [23] relate VP and VNP over various fields to standard Boolean complexity classes such as NC/poly, #P/poly (uses GRH), and $\mathsf{Mod}_p\mathsf{P}$/poly. The result then follows from the implication $\mathsf{NP} \not\subseteq \mathsf{coMA} \Rightarrow \mathsf{NC/poly} \neq \#\mathsf{P/poly}$ (and similarly with #P/poly replaced by $\mathsf{Mod}_p\mathsf{P}$/poly), which uses the downward self-reducibility of complete functions for #P/poly (the permanent [102]) and $\mathsf{Mod}_p\mathsf{P}$/poly [33], as well as Valiant–Vazirani [106].

Proof 2: This proof seems to only work when R is a finite ring of odd characteristic. If $\mathsf{VNP}_R \subseteq \mathsf{VP}_R$ for a finite ring R of odd characteristic, then the permanent over R has polynomial-size R-algebraic circuits, and hence—since R is finite—polynomial-size Boolean circuits. This implies $\mathsf{P}^{\mathsf{Mod}_m\mathsf{P}} \subseteq \mathsf{coMA}$ by Babai–Fortnow–Nisan–Wigderson [7], where m is the characteristic of R. The proof concludes by using either Toda's Theorem [100]—or the slightly weaker result of Valiant and Vazirani [106]—to show that $\mathsf{NP} \subseteq \mathsf{P}^{\mathsf{Mod}_m\mathsf{P}}$.

Proof 3: Similar to Proof 2, but instead of Babai–Fortnow–Nisan–Wigderson [7], uses Kabanets–Impagliazzo [50] to conclude from the polynomial-size R-algebraic circuits for the permanent that $\mathsf{NP} \subseteq \mathsf{coNP}^{\mathsf{RP}} \subseteq \mathsf{coMA}$.

The following lemma is the key to Theorem 1.2.

Lemma 4.2. *Every family of CNF tautologies* (φ_n) *has a Hilbert-like family of IPS certificates* (C_n) *in* VNP^0_R.

We first show how Theorem 1.2 follows from Lemma 4.2, and then return to the proof of the lemma.

Proof of Theorem 1.2, Assuming Lemma 4.2. For a given set $\mathcal{F}$ of unsatisfiable polynomial equations $F_1 = \cdots = F_m = 0$, a lower bound on 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 Hilbert-like IPS implies that some function in VNP—namely, the VNP-IPS certificate guaranteed by Lemma 4.2—cannot be computed by polynomial-size algebraic circuits and hence that $\mathsf{VNP} \neq \mathsf{VP}$. Since Lemma 4.2 even guarantees a constant-free certificate, we get the analogous consequence for constant-free lower bounds.

The second part of Theorem 1.2 follows from the fact that number of lines in a PC proof is p-equivalent to Hilbert-like det-IPS (Proposition 3.4). As in the first part, a super-polynomial lower bound on Hilbert-like det-IPS implies that some function family in VNP is not a p-projection of the determinant. Since the permanent is VNP-complete under p-projections, the result follows. □

Proof of Lemma 4.2. We mimic one of the proofs of completeness for Hilbert-like IPS [81, Theorem 1] (recall Proposition 3.4) and then show that this proof can in fact be carried out in VNP^0. We omit any mention of the ground ring, as it will not be relevant.

Let $\varphi_n(\vec{x}) = \kappa_1(\vec{x}) \wedge \cdots \wedge \kappa_m(\vec{x})$ be an unsatisfiable CNF, where each κ_i is a disjunction of literals. Let $C_i(\vec{x})$ denote the (negated) polynomial translation of κ_i via $\neg x \mapsto x$, $x \mapsto 1 - x$ and $f \vee g \mapsto fg$; in particular, $C_i(\vec{x}) = 0$ if and only if $\kappa_i(\vec{x}) = 1$, and thus φ_n is unsatisfiable if and only if the system of equations $C_1(\vec{x}) = \cdots = C_m(\vec{x}) = x_1^2 - x_1 = \cdots = x_n^2 - x_n = 0$ is unsatisfiable. In fact, as we will see in the course of the proof, we will not need the equations $x_i^2 - x_i = 0$. It will be convenient to introduce the function $b(e, x) = ex + (1 - e)(1 - x)$, i.e., $b(1, x) = x$ and

$b(0,x) = 1 - x$. For example, the clause $\kappa_i(\vec{x}) = (x_1 \vee \neg x_{17} \vee x_{42})$ gets translated into $C_i(\vec{x}) = (1-x_1)x_{17}(1-x_{42}) = b(0,x_1)b(1,x_{17})b(0,x_{42})$, and therefore an assignment falsifies κ_i if and only if $(x_1, x_{17}, x_{42}) \mapsto (0,1,0)$.

Just as $1 = x_1x_2 + x_1(1-x_2) + (1-x_2)x_1 + (1-x_2)(1-x_1)$, an easy induction shows that

$$1 = \sum_{\vec{e}\in\{0,1\}^n} \prod_{i=1}^{n} b(e_i, x_i). \tag{4}$$

We will show how to turn this expression—which is already syntactically in VNP^0 form—into a VNP certificate refuting φ_n.

Let c_i be the placeholder variable corresponding to $C_i(\vec{x})$. For any property Π, we write $[\![\Pi(\vec{e})]\!]$ for the indicator function of Π: $[\![\Pi(\vec{e})]\!] = 1$ if and only if $\Pi(\vec{e})$ holds and 0 otherwise. We claim that the following is a VNP^0-IPS certificate:

$$\sum_{i=1}^{m} c_i \cdot \left(\sum_{\vec{e}\in\{0,1\}^n} [\![\vec{e} \text{ falsifies } \kappa_i \text{ and satisfies } \kappa_j \text{ for all } j < i]\!] \prod_{j : x_j \notin \kappa_i} b(e_j, x_j) \right). \tag{5}$$

First, let us prove that this is indeed a certificate, and then we will show it is in VNP^0.

To see that Equation (5) is a certificate, we claim that on substituting $C_i(\vec{x})$ for c_i, the resulting sum becomes syntactically identical to Equation (4) and therefore sums to 1. (It is clear from its form that it is in the ideal generated by the c_i.) Note that an assignment $\vec{e}$ falsifies clause κ_i if and only if $C_i(x) = \prod_{j : x_j \in \kappa_i} b(e_j, x_j)$. Let A_i be the set of assignments $\vec{e}$ satisfying the ith condition in brackets: $A_i = \{\vec{e} \in \{0,1\}^n : \vec{e} \text{ falsifies } \kappa_i \text{ and satisfies } \kappa_j \text{ for all } j < i\}$. Then, on substituting the $C_i(\vec{x})$ for the c_i, Equation (5) becomes

$$\begin{aligned}
&\sum_{i=1}^{m} C_i(\vec{x}) \cdot \left(\sum_{\vec{e}\in\{0,1\}^n} [\![\vec{e} \text{ falsifies } \kappa_i \text{ and satisfies } \kappa_j \text{ for all } j < i]\!] \prod_{j : x_j \notin \kappa_i} b(e_j, x_j) \right) \\
&= \sum_{i=1}^{m} \sum_{\vec{e}\in\{0,1\}^n} \left([\![\vec{e} \text{ falsifies } \kappa_i \text{ and satisfies } \kappa_j \text{ for all } j < i]\!] C_i(\vec{x}) \prod_{j : x_j \notin \kappa_i} b(e_j, x_j) \right) \\
&= \sum_{i=1}^{m} \sum_{\vec{e}\in\{0,1\}^n} \left([\![\vec{e} \text{ falsifies } \kappa_i \text{ and satisfies } \kappa_j \text{ for all } j < i]\!] \left(\prod_{j : x_j \in \kappa_i} b(e_j, x_j) \right) \left(\prod_{j : x_j \notin \kappa_i} b(e_j, x_j) \right) \right) \\
&= \sum_{i=1}^{m} \sum_{\vec{e}\in\{0,1\}^n} \left([\![\vec{e} \text{ falsifies } \kappa_i \text{ and satisfies } \kappa_j \text{ for all } j < i]\!] \prod_{j\in[n]} b(e_j, x_j) \right) \\
&= \sum_{i=1}^{m} \sum_{\vec{e}\in A_i} \prod_{j\in[n]} b(e_j, x_j).
\end{aligned} \tag{6}$$

Now, note that the condition defining A_i automatically excludes any $\vec{e} \in A_j$, so the A_i are disjoint from one another. Furthermore, as φ was unsatisfiable, every assignment $\vec{e}$ must falsify some clause and therefore must appear in some A_i. Thus the A_i form a partition of $\{0,1\}^n$, so the sum $\sum_{i=1}^{m} \sum_{\vec{e}\in A_i}$ is the same as $\sum_{\vec{e}\in\{0,1\}^n}$, and Equation (6) becomes syntactically identical to the right-hand side of Equation (4). Therefore, Equation (5) is a certificate, as claimed.

Indeed, as noted in Pitassi [81, Theorem 1], the same proof would have worked had the A_i been *any* partition of $\{0,1\}^n$ such that every $\vec{e} \in A_i$ falsified clause κ_i; we will now use this particular

partition to show that the certificate (5) is in VNP^0. We have

$$\begin{aligned}
&\sum_{i=1}^{m} c_i \cdot \left(\sum_{\vec{e}\in\{0,1\}^n} [\![\vec{e} \text{ falsifies } \kappa_i \text{ and satisfies } \kappa_j \text{ for all } j < i]\!] \prod_{j : x_j \notin \kappa_i} b(e_j, x_j) \right) \\
&= \sum_{i=1}^{m} c_i \cdot \left(\sum_{\vec{e}\in\{0,1\}^n} [\![C_i(\vec{e}) = 1 \text{ and } C_j(\vec{e}) = 0 \text{ for all } j < i]\!] \prod_{j : x_j \notin \kappa_i} b(e_j, x_j) \right) \\
&= \sum_{i=1}^{m} c_i \cdot \left(\sum_{\vec{e}\in\{0,1\}^n} \left(C_i(\vec{e}) \prod_{j<i} (1 - C_j(\vec{e})) \right) \prod_{j : x_j \notin \kappa_i} b(e_j, x_j) \right) \\
&= \sum_{e\in\{0,1\}^n} \sum_{i=1}^{m} c_i C_i(\vec{e}) \left(\prod_{j<i} (1 - C_j(\vec{e})) \right) \left(\prod_{j : x_j \notin \kappa_i} b(e_j, x_j) \right).
\end{aligned}$$

Finally, it is readily visible that the polynomial function of $\vec{c}$, $\vec{e}$, and $\vec{x}$ that is the summand of the outermost sum $\sum_{\vec{e}\in\{0,1\}^n}$ is computed by a polynomial-size circuit of polynomial degree, and thus the entire certificate is in VNP. Indeed, the expression as written exhibits it as a small *formula* of constant depth with unbounded fan-in gates. By inspection, this circuit only uses the constants $0, 1, -1$, and hence the certificate is in VNP^0. □

5 PIT AS A BRIDGE BETWEEN CIRCUIT COMPLEXITY AND PROOF COMPLEXITY

In this section, we state our PIT axioms and prove Theorems 1.4 and 1.6, which say that Extended Frege (EF) (respectively, AC^0- or $\mathrm{AC}^0[p]$-Frege) is polynomially equivalent to the Ideal Proof System if there are polynomial-size circuits for PIT whose correctness—suitably formulated—can be efficiently proved in EF (respectively, AC^0- or $\mathrm{AC}^0[p]$-Frege). More precisely, we identify a small set of natural axioms for PIT and show that if these axioms can be proven efficiently in EF, then EF is p-equivalent to IPS. Theorem 1.6 begins to explain why $\mathrm{AC}^0[p]$-Frege lower bounds have been so difficult to obtain and highlights the importance of our PIT axioms for $\mathrm{AC}^0[p]$-Frege lower bounds. We begin by describing and discussing these axioms.

5.1 Axioms for Circuits for Polynomial Identity Testing

Fix some standard Boolean encoding of constant-free algebraic circuits, so that the encoding of any size-m constant-free algebraic circuit has size $\mathrm{poly}(m)$. We use "$[C]$" to denote the encoding of the algebraic circuit C. Let $K = (K_{m,n})$ denote a family of Boolean circuits for solving polynomial identity testing. That is, $K_{m,n}$ is a Boolean function that takes as input the encoding of a size m constant-free algebraic circuit, C, over variables $x_1, \ldots, x_n$, and if C has polynomial degree, then K outputs 1 if and only if the polynomial computed by C is the 0 polynomial.

Notational convention: We underline parts of a statement that involve propositional variables. For example, if in a propositional statement we write "$[C]$," then this refers to a fixed Boolean string that is encoding the (fixed) algebraic circuit C. In contrast, if we write $\underline{[C]}$, then this denotes a Boolean string *of propositional variables*, which is to be interpreted as a description of an as-yet-unspecified algebraic circuit C; any setting of the propositional variables corresponds to a particular algebraic circuit C. Throughout, we use $\vec{p}$ and $\vec{q}$ to denote propositional variables (which we do not bother underlining except when needed for emphasis) and $\vec{x}, \vec{y}, \vec{z}, \ldots$ to denote the algebraic variables that are the inputs to algebraic circuits. Thus, $C(\vec{x})$ is an algebraic circuit with inputs $\vec{x}$, $[C(\vec{x})]$ is a fixed Boolean string encoding some particular algebraic circuit C, $\underline{[C(\vec{x})]}$ is a string of propositional variables encoding an unspecified algebraic circuit C, and $[C(\underline{\vec{p}})]$ denotes

a Boolean string together with propositional variables $\vec{p}$ that describes a fixed algebraic circuit C whose inputs have been set to the propositional variables $\vec{p}$.

Definition 5.1. Our PIT axioms for a Boolean circuit K are as follows. (This definition makes sense even if K does not correctly compute PIT, but that case is not particularly interesting or useful.)

(1) Intuitively, the first axiom states that if C is a circuit computing the identically 0 polynomial, and then the polynomial evaluates to 0 on all Boolean inputs,

$$K(\underline{[C(\vec{x})]}) \to K(\underline{[C(\vec{p})]}).$$

Note that the only variables on the left-hand side of the implication are Boolean propositional variables, $\vec{q}$, that encode an algebraic circuit of size m over n algebraic variables $\vec{x}$ (these latter are *not* propositional variables of the above formula). The variables on the right-hand side are $\vec{q}$ plus Boolean variables $\vec{p}$, where some of the variables in $\vec{q}$—those encoding the x_i—have been replaced by constants or $\vec{p}$ in such a way that $[C(\vec{p})]$ encodes a circuit that plugs in the $\{0,1\}$-valued p_i for its algebraic inputs x_i. In other words, when we say $\underline{[C(\vec{p})]}$ we mean the encoding of the circuit C where Boolean constants are plugged in for the original algebraic $\vec{x}$ variables, as specified by the variables $\vec{p}$.

(2) Intuitively, the second axiom states that if C is a circuit computing the zero polynomial, then the circuit $1 - C$ does not compute the zero polynomial,

$$K(\underline{[C(\vec{x})]}) \to \neg K(\underline{[1 - C(\vec{x})]}).$$

Here, if $\vec{q}$ are the propositional variables describing C, then these are the only variables that appear in the above statement. We abuse syntax slightly in writing $[1 - C]$: It is meant to denote a Boolean formula $\varphi(\vec{q})$ such that if $\vec{q} = [C]$ describes a circuit C, then $\varphi(\vec{q})$ describes the circuit $1 - C$ (with one subtraction gate more than C).

(3) Intuitively, the third axiom states that PIT circuits respect certain substitutions. More specifically, if the polynomial computed by circuit G is 0, then G can be substituted for the constant 0,

$$K(\underline{[G(\vec{x})]}) \wedge K(\underline{[C(\vec{x},0)]}) \to K(\underline{[C(\vec{x},G(\vec{x}))]}).$$

Here the notations $[C(\vec{x},0)]$ and $[C(\vec{x},G(\vec{x}))]$ are similar abuses of notation to above; we use these and similar shorthands without further mention. (In particular, just as all instances of $\underline{[C]}$ across the statement are encoded using the same variables, so are all instances of $\underline{[G]}$: In "$K(\underline{[C(\vec{x},G(\vec{x}))]})$," where the circuit $G(\vec{x})$ is being plugged into an input for C, the variables used to encode $G(\vec{x})$ are the *same* as the variables used to encode $G(\vec{x})$ in the antecedent "$K(\underline{[G(\vec{x})]})$.")

(4) Intuitively, the last axiom states that PIT is closed under permutations of the (algebraic) variables. More specifically if $C(\vec{x})$ is identically 0, then so is $C(\pi(\vec{x}))$ for all permutations π,

$$K(\underline{[C(\vec{x})]}) \to K(\underline{[C(\pi(\vec{x}))]}).$$

5.2 Extended Frege Is p-equivalent to IPS If PIT Is EF-provably Easy

Theorem 1.4. *If there is a family K of polynomial-size Boolean circuits computing PIT, such that the PIT axioms for K have polynomial-size EF proofs, then EF is polynomially equivalent to IPS.*

Note that the issue is not the existence of small circuits for PIT, since we would be happy with nonuniform polynomial-size PIT circuits, which do exist. Unfortunately, the known constructions are highly nonuniform—they involve picking random points—and we do not see how to prove

axiom 1 for these constructions. Nonetheless, it seems very plausible to us that there exists a polynomial-size family of PIT circuits where the above axioms are efficiently provable in EF.

To prove the theorem, we will first show that EF is p-equivalent to IPS if a family of propositional formulas expressing soundness of IPS are efficiently EF provable. Then we will show that efficient EF proofs of $Soundness_{\mathrm{IPS}}$ follows from efficient EF proofs for our PIT axioms.

Remark 5.2. It is standard for two proof systems P_1 and P_2 that if P_2 can prove the soundness of P_1, then P_2 can p-simulate P_1. What is more interesting here is that we show (Lemma 5.4) that a *natural* set of axioms for PIT (Definition 5.1) imply $Soundness_{\mathrm{IPS}}$. This allows us to draw on intuitions (and, hopefully, results) about PIT to get a better sense of the plausibility of efficient EF proofs of $Soundness_{\mathrm{IPS}}$. The power of this connection to PIT has already led to new results building on ours: Li, Tzameret, and Wang [65] showed that noncommutative formula IPS is qp-equivalent to Frege by showing that a noncommutative formula PIT algorithm [84] could be proved correct in Frege (see Section 8 for details).

Soundness of IPS. It is well known that for standard Cook–Reckhow proof systems, a proof system P can p-simulate another proof system P' if and only if P can prove soundness of P'. Our proof system is not standard, because verifying a proof requires probabilistic, rather than deterministic, polynomial-time. Still we will show how to formalize soundness of IPS propositionally, and we will show that if EF can efficiently prove soundness of IPS then EF is p-equivalent to IPS.

Let $\varphi = \kappa_1 \wedge \ldots \wedge \kappa_m$ be an unsatisfiable propositional 3CNF formula over variables $p_1, \ldots, p_n$, and let $Q_1^\varphi, \ldots, Q_m^\varphi$ be the corresponding polynomial equations (each of degree at most 3) such that $\kappa_i(\alpha) = 1$ if and only if $Q_i^\varphi(\alpha) = 0$ for $\alpha \in \{0,1\}^n$. An IPS-refutation of φ is an algebraic circuit, C, which demonstrates that 1 is in the ideal generated by the polynomial equations $\vec{Q}^\varphi$. (This demonstrates that the polynomial equations $\vec{Q}^\varphi = 0$ are unsolvable, which is equivalent to proving that φ is unsatisfiable.) In particular, recall that C has two types of inputs: $x_1, \ldots, x_n$ (corresponding to the propositional variables $p_1, \ldots, p_n$) and the placeholder variables $y_1, \ldots, y_m$ (corresponding to the equations $Q_1^\varphi, \ldots, Q_m^\varphi$) and satisfies the following two properties:

(1) $C(\vec{x}, \vec{0}) = 0$. This property essentially states that the polynomial computed by $C(\vec{x}, \vec{Q}(\vec{x}))$ is in the ideal generated by $Q_1^\varphi, \ldots, Q_m^\varphi$.
(2) $C(\vec{x}, \vec{Q}^\varphi(\vec{x})) = 1$. This property states that the polynomial computed by C, when we substitute the Q_i^φ's for the y_i's, is the identically 1 polynomial.

Encoding IPS Proofs. Let K be a family of polynomial-size circuits for PIT. Using $K_{m,n}$, we can create a polynomial-size Boolean circuit, $Proof_{\mathrm{IPS}}([C], [\varphi])$ that is true if and only if C is an IPS-proof of the unsatisfiability of $\vec{Q}^\varphi = 0$. The polynomial-sized Boolean circuit $Proof_{\mathrm{IPS}}([C], [\varphi])$ first takes the encoding of the algebraic circuit C (which has x-variables and placeholder variables), and creates the encoding of a new algebraic circuit, $[C']$, where C' is like C but with each y_i variable replaced by 0. Second, it takes the encoding of C and $[\varphi]$ and creates the encoding of a new circuit C'', where C'' is like C but now with each y_i variable replaced by Q_i^φ. (Note that whereas C has $n + m$ underlying algebraic variables, both C' and C'' have only n underlying variables.) $Proof_{\mathrm{IPS}}([C], [\varphi])$ is true if and only if $K([C'])$—that is, $C'(\vec{x}) = C(\vec{x}, \vec{0})$ computes the 0 polynomial—and $K([1 - C'']) = 0$—that is, $C''(\vec{x}) = C(\vec{x}, \vec{Q}^\varphi(\vec{x}))$ computes the 1 polynomial.

Definition 5.3. Let formula $Truth_{bool}(\vec{p}, \vec{q})$ state that the truth assignment $\vec{q}$ satisfies the Boolean formula coded by $\vec{p}$. The soundness of IPS says that if φ has a refutation in IPS, then φ is unsatisfiable. That is, $Soundness_{\mathrm{IPS},m,n}([C], [\varphi], \vec{p})$ has variables that encode a size m IPS-proof C,

variables that encode a 3CNF formula φ over n variables, and n additional Boolean variables, $\vec{p}$. $Soundness_{\mathrm{IPS},m,n}([C],[\varphi],\vec{p})$ states:

$$Proof_{\mathrm{IPS}}(\underline{[C]},\underline{[\varphi]}) \rightarrow \neg Truth_{bool}(\underline{[\varphi]},\vec{p}).$$

Lemma 5.4. *If EF can efficiently prove $Soundness_{IPS}$ for some polynomial-size Boolean circuit family K computing PIT, then EF is p-equivalent to IPS.*

Proof. Because IPS can p-simulate EF, it suffices to show that if EF can prove Soundness of IPS, then EF can p-simulate IPS. Assume that we have a polynomial-size EF proof of $Soundness_{\mathrm{IPS}}$. Now suppose that C is an IPS-refutation of an unsatisfiable 3CNF formula φ on variables $\vec{p}$. We will show that EF can also prove $\neg\varphi$ with a proof of size polynomial in $|C|$.

First, we claim that it follows from a natural encoding (see Section 5.4) that EF can efficiently prove:

$$\varphi \rightarrow Truth_{bool}([\varphi],\vec{p}).$$

(Variables of this statement just the p variables, because φ is a fixed 3CNF formula, so the encoding $[\varphi]$ is a variable-free Boolean string.)

Second, if C is an IPS-refutation of φ, then EF can prove $Proof_{\mathrm{IPS}}([C],[\varphi])$.[5] This holds because both C and φ are fixed, so this formula is variable free. Thus, EF can just verify that it is true.

Third, by soundness of IPS, which we are assuming is EF-provable, and the fact that EF can prove $Proof_{\mathrm{IPS}}([C],[\varphi])$ (step 2), it follows by modus ponens that EF can prove $\neg Truth_{bool}([\varphi],\vec{p})$. (The statement $Soundness_{\mathrm{IPS}}([C],[\varphi],\vec{p})$ for this instance will only involve variables $\vec{p}$: The other two sets of inputs to the $Soundness_{\mathrm{IPS}}$ statement, $[C]$ and $[\varphi]$, are constants here, since both C and φ are fixed.)

Finally, by modus ponens and the contrapositive of $\varphi \rightarrow Truth_{bool}([\varphi],\vec{p})$, we conclude in EF $\neg\varphi$, as desired. □

Theorem 1.4 follows from Lemma 5.4 and the following lemma.

Lemma 5.5. *If EF can efficiently prove the PIT axioms for some polynomial-size Boolean circuit family K computing PIT, then EF can efficiently prove $Soundness_{IPS}$ (for that same K).*

Proof. Starting with $Truth_{bool}(\underline{[\varphi]},\vec{p}), K(\underline{[C(\vec{x},\vec{0})]}), K(\underline{[1-C(\vec{x},\vec{Q}(\vec{x}))]})$, we will derive a contradiction.

(1) First show for every $i \in [m]$, $Truth_{bool}([\varphi],\vec{p}) \rightarrow K(\underline{[Q_i^{\varphi}(\vec{p})]})$, where Q_i^{φ} is the low degree polynomial corresponding to the clause, κ_i, of φ. Note that, as φ is not a fixed formula but is determined by the propositional variables encoding $\underline{[\varphi]}$, the encoding $\underline{[Q_i^{\varphi}]}$ depends on a subset of these variables.

$Truth_{bool}(\underline{[\varphi]},\vec{p})$ states that each clause κ_i in φ evaluates to true under $\vec{p}$. It is a tautology that if κ_i evaluates to true under $\vec{p}$, then Q_i^{φ} evaluates to 0 at $\vec{p}$. Since K correctly computes PIT,

$$Truth_{bool}(\underline{[\kappa_i]},\vec{p}) \rightarrow K(\underline{[Q_i^{\varphi}(\vec{p})]}) \qquad (*)$$

is a tautology. Furthermore, although both the encoding $\underline{[\kappa_i]}$ and $\underline{[Q_i^{\varphi}]}$ depend on the propositional variables encoding $\underline{[\varphi]}$, since we assume that φ is a 3CNF, these only depend

[5] The fact that $Proof_{\mathrm{IPS}}([C],[\varphi])$ is even true, given that C is an IPS-refutation of φ, follows from the completeness of the circuit K computing PIT—that is, if $C \equiv 0$, then $K([C])$ accepts. This is one of only two places in the proof of Theorem 1.4 that we actually need the assumption that K correctly computes PIT rather than merely assuming that K satisfies our PIT axioms. However, it is clear that this usage of this assumption is crucial. The other usage is in Step 1 of Lemma 5.5.

on *constantly many* of the variables encoding $\underline{[\varphi]}$. Thus the tautology (*) can be proven in EF by brute force. Putting these together, we can derive $Truth_{bool}(\underline{[\varphi]}, \vec{p}) \to K(\underline{[Q_i^{\varphi}(\vec{p})]})$, as desired.

(2) Using the assumption $Truth_{bool}(\underline{[\varphi]}, \vec{p})$ together with (1) we derive $K(\underline{[Q_i^{\varphi}(\vec{p})]})$ for all $i \in [m]$.

(3) Using Axiom 1 we can prove $K(\underline{[C(\vec{x}, \vec{0})]}) \to K(\underline{[C(\vec{p}, \vec{0})]})$. Using modus ponens with the assumption $K(\underline{[C(\vec{x}, \vec{0})]})$, we derive $K(\underline{[C(\vec{p}, \vec{0})]})$.

(4) Repeatedly using Axiom 3 and Axiom 4, we can prove

$$K(\underline{[Q_1^{\varphi}(\vec{p})]}), K(\underline{[Q_2^{\varphi}(\vec{p})]}), \ldots, K(\underline{[Q_m^{\varphi}(\vec{p})]}), K(\underline{[C(\vec{p}, \vec{0})]}) \to K(\underline{[C(\vec{p}, \vec{Q}(\vec{p}))]}).$$

(5) Applying modus ponens repeatedly with (4), (2), and (3), we can prove $K(\underline{[C(\vec{p}, \vec{Q}(\vec{p}))]})$.

(6) Applying Axiom 2 to (5) we get $\neg K(\underline{[1 - C(\vec{p}, \vec{Q}(\vec{p}))]})$.

(7) Using Axiom 1 we can prove $K(\underline{[1 - C(\vec{x}, \vec{Q}(\vec{x}))]}) \to K(\underline{[1 - C(\vec{p}, \vec{Q}(\vec{p}))]})$. Using our assumption $K(\underline{[1 - C(\vec{x}, \vec{Q}(\vec{x}))]})$ and modus ponens, we conclude $K(\underline{[1 - C(\vec{p}, \vec{Q}(\vec{p}))]})$.

Finally, (6) and (7) give a contradiction. □

5.3 $AC^0[p]$-Frege Lower Bounds, PIT, and Circuit Lower Bounds

Theorem 1.6. *Let C be any class of circuits closed under AC^0 circuit reductions. If there is a family K of polynomial-size Boolean circuits for PIT such that the PIT axioms for K have polynomial-size C-Frege proofs, then C-Frege is polynomially equivalent to IPS and, consequently, polynomially equivalent to Extended Frege.*

Note that here we *do not* need to restrict the circuit K to be in the class C. This requires one more technical device compared to the proofs in the previous section. The proof of Theorem 1.6 follows the proof of Theorem 1.4 very closely. The main new ingredient is a folklore technical device that allows even very weak systems such as AC^0-Frege to make statements about arbitrary circuits K—such as those needed to reason about the PIT axioms—together with a careful analysis of what was needed in the proof of Theorem 1.4. Before proving Theorem 1.6, we discuss some of its more interesting consequences.

As AC^0-Frege is known unconditionally to be strictly weaker than Extended Frege [3], we immediately get that AC^0-Frege cannot efficiently prove the PIT axioms for any Boolean circuit family K correctly computing PIT.

Using essentially the same proof as Theorem 1.6, we also get the following result. By "depth-d PIT axioms," we mean a variant where the algebraic circuits C (encoded as $[C]$ in the statement of the axioms) have depth at most d. Note that, even over finite fields, super-polynomial lower bounds on depth-d algebraic circuits are notoriously open problems even for d as small as 4 or 5.[6]

Corollary 1.7. *For any d, if there is a family of tautologies with no polynomial-size $AC^0[p]$-Frege proof, and $AC^0[p]$-Frege has polynomial-size proofs of the [depth-d] PIT axioms for some K, then $VNP_{\mathbb{F}_p}$ does not have polynomial-size [depth-d] algebraic circuits.*

[6]Lower bounds of $2^{\Omega(\sqrt{n}\log n)}$ on *homogeneous* depth four circuits are known [52, 63]—and, furthermore, *any* asymptotic improvement to these lower bounds implies VP ≠ VNP [99]—but for unrestricted depth four algebraic circuits nothing better than Strassen's degree bound of $\Omega(n \log n)$ [97] is known. Some lower bounds are also known for depth five circuits, but again, only homogeneous circuits [53, 62]. Even if we restrict attention to homogeneous circuits, depth six is completely open.

This corollary makes the following question of central importance in getting lower bounds on $\mathsf{AC}^0[p]$-Frege:

Open Question 5.6. For some $d \geq 4$, is there some K computing depth-d PIT, for which the depth-d PIT axioms have $\mathsf{AC}^0[p]$-Frege proofs of polynomial size?

This question has the virtue that answering it either way is highly interesting:

- If $\mathsf{AC}^0[p]$-Frege does not have polynomial-size proofs of the [depth-d] PIT axioms for any K, then we have super-polynomial size lower bounds on $\mathsf{AC}^0[p]$-Frege, answering a question that has been open for nearly thirty years.
- Otherwise, super-polynomial size lower bounds on $\mathsf{AC}^0[p]$-Frege imply that the permanent does not have polynomial-size algebraic circuits [of depth d] over any finite field of characteristic p. This would then explain why getting superpolynomial lower bounds on $\mathsf{AC}^0[p]$-Frege has been so difficult.

This dichotomy is in some sense like a "completeness result for $\mathsf{AC}^0[p]$-Frege, modulo proving strong algebraic circuit lower bounds on VNP": If one hopes to prove $\mathsf{AC}^0[p]$-Frege lower bounds *without proving* strong lower bounds on VNP, then one must prove $\mathsf{AC}^0[p]$-Frege lower bounds on the PIT axioms. For example, if you believe that proving $\mathsf{VP} \neq \mathsf{VNP}$ [or that proving VNP does not have bounded-depth polynomial-size circuits] is very difficult, and that proving $\mathsf{AC}^0[p]$-Frege lower bounds is comparatively easy, then to be consistent you must also believe that proving $\mathsf{AC}^0[p]$-Frege lower bounds *on the [bounded-depth] PIT axioms* is easy.

Similarly, by combining Theorems 1.6 and 3.5, we get the following corollary.

Corollary 5.7. *If for every constant d, there is a constant d' such that the depth-d PIT axioms have polynomial-size depth-d' $\mathsf{AC}^0_{d'}[p]$-Frege proofs , then $\mathsf{AC}^0[p]$-Frege is polynomially equivalent to constant-depth $\mathrm{IPS}_{\mathbb{F}_p}$.*

Using the chasms at depths 3 and 4 for algebraic circuits [2, 55, 99] (see Observation 3.3), we can also help explain why sufficiently strong exponential lower bounds for AC^0-Frege—that is, lower bounds that do not depend on the depth, or do not depend so badly on the depth (the current best bounds are of the form $\exp(\Omega(n^{\exp(-d+O(1))}))$ [15, 60, 83]), which have also been open for nearly thirty years—have been difficult to obtain:

Corollary 5.8. *Let $\mathbb{F}$ be any field, and let c be a sufficiently large constant. If there is a family of tautologies (φ_n) such that any AC^0-Frege proof of φ_n has size at least $2^{c\sqrt{n}\log n}$, and AC^0-Frege has polynomial-size proofs of the depth 4 $\mathrm{PIT}_{\mathbb{F}}$ axioms for some K, then $\mathsf{VP}^0_{\mathbb{F}} \neq \mathsf{VNP}^0_{\mathbb{F}}$.*

If $\mathbb{F}$ has characteristic zero, then we may replace "depth 4" above with "depth 3."

Proof. Suppose that AC^0-Frege can efficiently prove the depth-4 $\mathrm{PIT}_{\mathbb{F}}$ axioms for some Boolean circuit K. Let (φ_n) be a family of tautologies. If $\mathsf{VNP}^0_{\mathbb{F}} = \mathsf{VP}^0_{\mathbb{F}}$, then there is a polynomial-size IPS proof of φ_n. By Observation 3.3, the same certificate is computed by a depth 4 $\mathbb{F}$-algebraic circuit of size $2^{O(\sqrt{n}\log n)}$. By assumption, AC^0-Frege can efficiently prove the depth 4 $\mathrm{PIT}_{\mathbb{F}}$ axioms for K, and therefore AC^0-Frege p-simulates depth 4 IPS. Thus there are AC^0-Frege proofs of φ_n of size $2^{O(\sqrt{n}\log n)}$.

If $\mathbb{F}$ has characteristic zero, then we may instead use the best-known chasm at depth 3, for which we only need depth-3 PIT and depth-3 IPS and yields the same bounds. □

As with Corollary 1.7, we conclude a similar dichotomy: Either AC^0-Frege can efficiently prove the depth-4 PIT axioms (depth 3 in characteristic zero) or proving $2^{\omega(\sqrt{n}\log n)}$ lower bounds on AC^0-Frege implies $\mathsf{VP}^0 \neq \mathsf{VNP}^0$.

Encoding K into Weak Proof Systems. Extended Frege can easily reason about arbitrary circuits K: For each gate g of K (or even each gate of each instance of K in a statement, if so desired), with children g_ℓ, g_r, EF can introduce a new variable k_g together with the requirement that $k_g \leftrightarrow k_{g_\ell}\, op_g\, k_{g_r}$, where op_g is the corresponding operation $g = g_\ell\, op_g\, g_r$ (e.g., $\wedge$, $\vee$, etc.). But weaker proof systems such as Frege (=NC^1-Frege), $\mathsf{AC}^0[p]$-Frege, or AC^0-Frege do not have this capability. We thus need to help them out by introducing these new variables and formulae ahead of time.

For each gate g, the statement $k_g \leftrightarrow k_{g_\ell}\, op_g\, k_{g_r}$ only involves three variables and thus can be converted into a 3CNF of constant size. We refer to these clauses as the "K-clauses." Note that the K-clauses do not set the inputs of K to any particular values nor require its output to be any particular value. We denote the variables corresponding to K's inputs as $k_{in,i}$ and the variable corresponding to K's output as k_{out}.

The modified statement $Proof_{\mathrm{IPS}}(\underline{[C]}, \underline{[\varphi]})$ now takes the following form. Recall that $Proof_{\mathrm{IPS}}$ involves two uses of K: $K(\underline{[C(\vec{x}, \vec{0})]})$ and $K(\underline{[1 - C(\vec{x}, \vec{Q}^{\varphi}(\vec{x}))]})$. Each of these instances of K needs to get its own set of variables, which we denote $k_g^{(1)}$ for gate g in the first instance and $k_g^{(2)}$ for gate g in the second instance, together with their own copies of the K-clauses. For an encoding $[C]$ or $[\varphi]$, let $[C]_i$ denote its ith bit, which may be a constant, a propositional variable, or even a propositional formula. Then $Proof_{\mathrm{IPS}}(\underline{[C]}, \underline{[\varphi]})$ is

$$\begin{aligned}
&\bigwedge_g \left(k_g^{(1)} \leftrightarrow k_{g_\ell}^{(1)}\, op_g\, k_{g_r}^{(1)}\right) \wedge \bigwedge_i \left(k_{in,i}^{(1)} \leftrightarrow \underline{[C(\vec{x}, \vec{0})]}_i\right) \\
\wedge &\bigwedge_g \left(k_g^{(2)} \leftrightarrow k_{g_\ell}^{(2)}\, op_g\, k_{g_r}^{(2)}\right) \wedge \bigwedge_i \left(k_{in,i}^{(2)} \leftrightarrow \underline{[1 - C(\vec{x}, \vec{Q}^{\varphi}(\vec{x}))]}_i\right) \\
\rightarrow &k_{out}^{(1)} \wedge k_{out}^{(2)}.
\end{aligned}$$

Throughout, we use the same notation $Proof_{\mathrm{IPS}}(\underline{[C]}, \underline{[\varphi]})$ as before to mean this modified statement (we will no longer be referring to the original, EF-style statement). The modified statement $Soundness_{\mathrm{IPS}}(\underline{[C]}, \underline{[\varphi]}, \underline{\vec{p}})$ will now take the form

$$\left((\text{dummy statements}) \wedge Proof_{\mathrm{IPS}}(\underline{[C]}, \underline{[\varphi]})\right) \rightarrow \neg Truth_{bool}(\underline{[\varphi]}, \underline{\vec{p}}),$$

using the new version of $Proof_{\mathrm{IPS}}$. Here "dummy statements" refers to certain statements that we will explain in Lemma 5.10. These dummy statements will only involve variables that do not appear in the rest of $Soundness_{\mathrm{IPS}}$ and therefore will be immediately seen not to affect its truth or provability.

The Proofs. Lemmata 5.10 and 5.11 are the AC^0-analogs of Lemmata 5.4 and 5.5, respectively. The proof of Lemma 5.10 will cause no trouble, and the proof of Lemma 5.11 will need one additional technical device (the "dummy statements" above).

Before getting to their proofs, we state the main additional lemma that we use to handle the new K variables. We say that a variable $k_{in,j}^{(i)}$ corresponding to an input gate of K is *set to* ψ by a propositional statement if $k_{in,j}^{(i)} \leftrightarrow \psi$ occurs in the statement.

Lemma 5.9. *Let (φ_n) be a sequence of tautologies on* $\mathrm{poly}(n)$ *variables, including any number of copies of the K variables, of the form $\varphi = ((\bigwedge_i \alpha_i) \rightarrow \omega)$. Let $\vec{p}$ denote the other (non-K) variables. Suppose that (1) there are at most $O(\log n)$ non-K variables in φ; (2) for each copy of K, the corresponding K-clauses appear amongst the α_i; (3) the only K variables that appear in ω are output variables $k_{out}^{(i)}$; and (4) if $k_{out}^{(i)}$ appears in ω, then all the inputs to $K^{(i)}$ are set to formulas that syntactically depend on at most $\vec{p}$.*

Then there is a $\mathrm{poly}(n)$*-size* AC^0*-Frege proof of φ.*

Proof sketch. The basic idea is that AC^0-Frege can brute force over all poly(n)-many assignments to the $O(\log n)$ non-K variables and for each such assignment can then just evaluate each copy of K gate by gate to verify the tautology. Any copy $K^{(i)}$ of K all of whose input variables are unset must not affect the truth of φ, since none of the $k^{(i)}$ variables can appear in the consequent ω of φ. In fact, for such copies of K, the K-clauses merely appear as disjuncts of φ, since it then takes the form $\varphi = \bigvee_i(\neg\alpha_i) \vee \omega = (\bigvee_g \neg(k_g^{(i)} \leftrightarrow k_{g_\ell}^{(i)}\, op_g\, k_{g_r}^{(i)})) \vee (\bigvee_{\text{remaining clauses } i} \neg\alpha_i) \vee \omega$. Thus, if AC^0-Frege can prove that the rest of φ, namely $(\bigvee_{\text{remaining clauses } i} \neg\alpha_i) \vee \omega$, is a tautology, then it can prove that φ is a tautology. □

Now we state the analogs of Lemmata 5.4 and 5.5 for C-Frege. Because of the similarity of the proofs to the previous case, we merely indicate how their proofs differ from the Extended Frege case.

Lemma 5.10 (AC^0 analog of Lemma 5.4). *Let C be a class of circuits closed under AC^0 circuit reductions. If there is a family K of polynomial-size Boolean circuits computing PIT, such that the PIT axioms for K have polynomial-size C-Frege proofs, then C-Frege is polynomially equivalent to IPS.*

Proof. Mimic the proof of Lemma 5.4. The third and fourth steps of that proof are just modus ponens, so we need only check the first two steps.

The first step is to show that C-Frege can prove $\varphi \to Truth_{bool}([\varphi], \vec{p})$. This follows directly from the details of the encoding of $[\varphi]$ and the full definition of $Truth_{bool}$; see Lemma 5.12.

The second step is to show that C-Frege can prove $Proof_{\text{IPS}}([C], [\varphi])$ for a fixed C, φ. In Lemma 5.4, this followed because this statement was variable free. Now this statement is no longer variable free, since it involve two copies of K and the corresponding variables and K-clauses. However, $Proof_{\text{IPS}}([C], [\varphi])$ satisfies the requirements of Lemma 5.9, and applying that lemma we are done. □

Lemma 5.11 (AC^0 analog of Lemma 5.5). *Let C be a class of circuits closed under AC^0 circuit reductions. If C-Frege can efficiently prove the PIT axioms for some polynomial-sized family of circuits K computing PIT, then C-Frege can efficiently prove $Soundness_{IPS}$ (for that same K).*

Proof. We mimic the proof of Lemma 5.5. In steps (1), (2), and (4) of that proof we used m additional copies of K, where m is the number of clauses in the CNF φ encoded by $[\varphi]$, and thus $m \le \text{poly}(n)$. To talk about these copies of K in C-Frege, however, the K variables must already be present in the statement we wish to prove in C-Frege. The "dummy statements" in the new version of soundness are the K-clauses—with inputs and outputs not set to anything—for each of m new copies of K, which we denote $K^{(3)}, \ldots, K^{(m+2)}$ (recall that the first two copies $K^{(1)}$ and $K^{(2)}$ are already used in the statement of $Proof_{\text{IPS}}$). We will not actually need these clauses anywhere in the proof, we just need their variables to be present from the beginning.

Starting with $Truth_{bool}([\varphi], \vec{p}), K^{(1)}([C(\vec{x}, \vec{0})]), K^{(2)}([1 - C(\vec{x}, \vec{Q}(\vec{x}))])$ we derive a contradiction. The only step of the proof of Lemma 5.5 that was not either the use of an axiom or modus ponens was step (1), so it suffices to verify that this can be carried out in AC^0-Frege with the K-clauses.

Step (1) was to show for every $i \in [m]$, $Truth_{bool}([\varphi], \vec{p}) \to K([Q_i^{\varphi}(\vec{p})])$, where Q_i^{φ} is the low-degree polynomial corresponding to the clause, κ_i, of φ. Note that, as φ is not a fixed formula but is determined by the propositional variables encoding $[\varphi]$, the encoding $[Q_i^{\varphi}]$ depends on a subset of these variables.

$Truth_{bool}(\underline{[\varphi]}, \vec{p})$ states that each clause κ_i in φ evaluates to true under $\vec{p}$. It is a tautology that if κ_i evaluates to true under $\vec{p}$, then Q_i^{φ} evaluates to 0 at $\vec{p}$. Since K correctly computes PIT,

$$Truth_{bool}(\underline{[\kappa_i]}, \vec{p}) \to K^{(i+2)}(\underline{[Q_i^{\varphi}(\vec{p})]}) \tag{**}$$

is a tautology. Furthermore, although both the encoding $\underline{[\kappa_i]}$ and $\underline{[Q_i^{\varphi}]}$ depend on the propositional variables encoding $\underline{[\varphi]}$, since we assume that φ is a 3CNF, these only depend on *constantly many* of the variables encoding $\underline{[\varphi]}$. Writing out (**) it has the form

$$Truth_{bool} \to \left((K^{(i+2)}\text{-clauses }) \wedge (\text{ setting inputs of } K^{(i+2)} \text{ to } \underline{[Q_i^{\varphi}(\vec{p})]}) \to k_{out}^{(i+2)} \right),$$

which is equivalent to

$$Truth_{bool} \wedge (K^{(i+2)}\text{-clauses}) \wedge \left(\text{setting inputs of } K^{(i+2)} \text{ to } \underline{[Q_i^{\varphi}(\vec{p})]} \right) \to k_{out}^{(i+2)}.$$

Thus (**) satisfies the conditions of Lemma 5.9 and has a short AC^0-Frege proof. Since $Truth_{bool}(\underline{[\varphi]}, \vec{p})$ is defined as $\bigwedge_i Truth_{bool}(\underline{[\kappa_i]}, \vec{p})$ (see Section 5.4), we can then derive

$$Truth_{bool}(\underline{[\varphi]}, \vec{p}) \to K^{(i+2)}(\underline{[Q_i^{\varphi}(\vec{p})]}),$$

as desired. □

5.4 Some Details of the Encodings

For an $\leq m$-clause, $\leq n$-variable 3CNF $\varphi = \kappa_1 \wedge \cdots \wedge \kappa_m$, its encoding is a Boolean string of length $3m(\lceil \log_2(n) \rceil + 1)$. Each literal x_i or $\neg x_i$ is encoded as the binary encoding of i ($\lceil \log_2(n) \rceil$ bits) plus a single other bit indicating whether the literal is positive (1) or negative (0). The encoding of a single clause is just the concatenation of the encodings of the three literals, and the encoding of φ is the concatenation of these encodings.

We define

$$Truth_{bool,n,m}(\underline{[\varphi]}, \vec{p}) \stackrel{def}{=} \bigwedge_{i=1}^{m} Truth_{bool,n}(\underline{[\kappa_i]}, \vec{p}).$$

For a single 3-literal clause κ, we define $Truth_{bool,n}(\underline{[\kappa]}, \vec{p})$ as follows. For an integer i, let $[i]$ denote the standard binary encoding of $i-1$ (so that the numbers $1, \ldots, 2^k$ are put into bijective correspondence with $\{0,1\}^k$). Let $\underline{[\kappa]} = \vec{q}_1 s_1 \vec{q}_2 s_2 \vec{q}_3 s_3$, where each s_i is the sign bit (positive/negative) and each $\vec{q}_i$ is a length-$\lceil \log_2 n \rceil$ string of variables corresponding to the encoding of the index of a variable. We write $\vec{q} = [k]$ as shorthand for $\bigwedge_{i=1}^{\lceil \log_2 n \rceil} (q_i \leftrightarrow [k]_i)$, where $x \leftrightarrow y$ is shorthand for $(x \wedge y) \vee (\neg x \wedge \neg y)$. Finally, we define:

$$Truth_{bool,n}(\underline{[\kappa]}, \vec{p}) \stackrel{def}{=} \bigvee_{j=1}^{3} \bigvee_{i=1}^{n} (\vec{q}_j = [i] \wedge (p_i \leftrightarrow s_j)).$$

(Hereafter we drop the subscripts n, m; they should be clear from context.)

Lemma 5.12. *For any 3CNF φ on n variables, there are* poly(n)*-size* AC^0*-Frege proofs of* $\varphi(\vec{p}) \to Truth_{bool}(\underline{[\varphi]}, \vec{p})$.

Proof. In fact, we will see that for a fixed clause κ, after simplifying constants—that is, $\varphi \wedge 1$ and $\varphi \vee 0$ both simplify to φ, $\varphi \wedge 0$ simplifies to 0, and $\varphi \vee 1$ simplifies to 1—that $Truth_{bool}(\underline{[\kappa]}, \vec{p})$ in fact becomes *syntactically identical* to $\kappa(\vec{p})$. By the definition of $Truth_{bool}(\underline{[\varphi]}, \vec{p})$, we get the same conclusion for any fixed CNF φ. Simplifying constants can easily be carried out in AC^0-Frege.

For a fixed κ, $\vec{q}_j$ and s_j become fixed to constants for $j = 1, 2, 3$. Denote the indices of the three variables in κ by i_1, i_2, i_3. The only variables left in the statement $Truth_{bool}([\kappa], \vec{p})$ are $\vec{p}$. Since the $\vec{q}_j$ and $[i]$ are all fixed, every term in $\bigvee_i (\vec{q}_j = [i] \wedge (p_i \leftrightarrow s_j))$ except for the i_j term simplifies to 0, so this entire disjunction simplifies to $(p_{i_j} \leftrightarrow s_j)$. Since the s_j are also fixed, if $s_j = 1$, then $(p_{i_j} \leftrightarrow s_j)$ simplifies to p_{i_j}, and if $s_j = 0$, then it simplifies to $\neg p_{i_j}$. With this understanding, we write $\pm p_{i_j}$ for the corresponding literal. Then $Truth_{bool}([\kappa], \vec{p})$ simplifies to $(\pm p_{i_1} \vee \pm p_{i_2} \vee \pm p_{i_3})$ (with signs as described previously). This is exactly $\kappa(\vec{p})$. □

6 ON LOWER BOUNDS FOR IPS

Theorem 1.2 shows that proving lower bounds on (even Hilbert-like) IPS, or on the number of lines in Polynomial Calculus proofs (equivalent to Hilbert-like det-IPS), is at least as hard as proving algebraic circuit lower bounds. In this section, we begin to make the difference between proving proof complexity lower bounds and proving circuit lower bounds precise.

The key difference, which any technique must grapple with, is that while an algebraic circuit complexity lower bound is a lower bound on a single function family $(f_n)_{n=1,2,3,\ldots}$, one for each n, an IPS lower bound is instead a lower bound on $(C_n)_{n=1,2,3,\ldots}$, where C_n is the set of *all* IPS certificates for the nth system of equations $\mathcal{F}_n$. Even over finite fields, C_n will be *infinite* for each n (when it is not empty). However, we observe that C_n is finitely generated, and we use this to suggest a direction for proving new proof complexity lower bounds, aimed at proving the long-sought length-of-proof lower bounds on an algebraic proof system.

6.1 The Difference Between Proof Complexity and Circuit Complexity Lower Bounds

The key fact we use is embodied in Lemma 6.1, which says that the set of (Hilbert-like) certificates for a given unsatisfiable system of equations is, in a precise sense, "finitely generated." The basic idea is then to leverage this finite generation to extend lower bound techniques from individual polynomials to entire "finitely generated" sets of polynomials.

Because Hilbert-like certificates are somewhat simpler to deal with, we begin with those and then proceed to general certificates. But keep in mind that all our key conclusions about Hilbert-like certificates will also apply to general certificates. For this section, we will need the notion of a module over a ring (the ring-analogue of a vector space over a field) and a few basic results about such modules (see Section 2.3).

Recall that a *Hilbert-like* IPS-certificate $C(\vec{x}, \vec{y})$ is one that is linear in the y-variables, that is, it has the form $\sum_{i=1}^{m} G_i(\vec{x}) y_i$. Each function of the form $\sum_i G_i(\vec{x}) y_i$ is completely determined by the tuple $(G_1(\vec{x}), \ldots, G_m(\vec{x}))$, and the set of all such tuples is exactly the $R[\vec{x}]$-module $R[\vec{x}]^m$.

The algebraic circuit size of a Hilbert-like certificate $C = \sum_i G_i(\vec{x}) y_i$ is equivalent (up to a small constant factor and an additive $O(n)$) to the algebraic circuit size of computing the entire tuple $(G_1(\vec{x}), \ldots, G_m(\vec{x}))$. A circuit computing the tuple can easily be converted to a circuit computing C by adding m times gates and a single plus gate. Conversely, for each i we can recover $G_i(\vec{x})$ from $C(\vec{x}, \vec{y})$ by plugging in 0 for all y_j with $j \neq i$ and 1 for y_i. So from the point of view of lower bounds on Hilbert-like certificates, we may consider their representation as tuples essentially without loss of generality. This holds even in the setting of Hilbert-like depth 3 IPS-proofs.

Using the representation of Hilbert-like certificates as tuples, we find that Hilbert-like IPS-certificates are in bijective correspondence with $R[\vec{x}]$ solutions (in the new variables g_i) to the following $R[\vec{x}]$-linear equation:

$$\begin{pmatrix} F_1(\vec{x}) & \cdots & F_m(\vec{x}) \end{pmatrix} \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix} = 1.$$

Just as in linear algebra over a field, the set of such solutions can be described by taking one solution and adding to it all solutions to the associated homogeneous equation:

$$\left(F_1(\vec{x}) \cdots F_m(\vec{x})\right)\begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix} = 0. \tag{7}$$

(To see why this is so, mimic the usual linear algebra proof: Given two solutions of the inhomogeneous equation, consider their difference.) Solutions to the latter equation are commonly called "syzygies" among the F_i. Syzygies and their properties are well studied—though not always well understood—in commutative algebra and algebraic geometry, so lower and upper bounds on Hilbert-like IPS-proofs may benefit from known results in algebra and geometry.

We now come to the key lemma for Hilbert-like certificates.

Lemma 6.1. *For a given set of unsatisfiable polynomial equations $F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$ over a Noetherian ring R (such as a field or $\mathbb{Z}$), the set of Hilbert-like IPS-certificates is a coset of a finitely generated submodule of $R[\vec{x}]^m$.*

Proof. The discussion above shows that the set of Hilbert-like certificates is a coset of a $R[\vec{x}]$-submodule of $R[\vec{x}]^m$, namely the solutions to Equation (7). As R is a Noetherian ring, so is $R[\vec{x}]$ (by Hilbert's Basis Theorem). Thus $R[\vec{x}]^m$ is a Noetherian $R[\vec{x}]$-module, and hence every submodule of it is finitely generated. □

Lemma 6.1 seems so conceptually important that it is worth re-stating:

> **The set of all Hilbert-like IPS-certificates for a given system of equations can be described by a single Hilbert-like IPS-certificate, together with a finite generating set for the syzygies.**

Its importance may be underscored by contrasting the preceding statement with the structure (if any?) of the set of all proofs in other proof systems, particularly non-algebraic ones.

Note that a finite generating set for the syzygies (indeed, even a Gröbner basis) can be found in the process of computing a Gröbner basis for the $R[\vec{x}]$-ideal $\langle F_1(\vec{x}), \ldots, F_m(\vec{x})\rangle$. This process is to Buchberger's Gröbner basis algorithm as the extended Euclidean algorithm is to the usual Euclidean algorithm; an excellent exposition can be found in the book by Ene and Herzog [32] (see also Eisenbud [31, Section 15.5]).

6.2 Towards Lower Bounds

Lemma 6.1 suggests that one might be able to prove size lower bounds on Hilbert-like-IPS along the following lines: (1) Find a single family of Hilbert-like IPS-certificates $(G_n)_{n=1}^{\infty}$, $G_n = \sum_{i=1}^{\text{poly}(n)} y_i G_i(\vec{x})$ (one for each input size n); (2) use your favorite algebraic circuit lower bound technique to prove a lower bound on the polynomial family G; (3) find a (hopefully nice) generating set for the syzygies; and (4) show that when adding to G any $R[\vec{x}]$-linear combinations of the generators of the syzygies, whatever useful property was used in the lower bound on G still holds. Although this indeed seems significantly more difficult than proving a single algebraic circuit complexity lower bound, it at least suggests a recipe for proving lower bounds on Hilbert-like IPS (and its subsystems such as homogeneous depth 3, depth 4, multilinear, etc.), which should be contrasted with the amorphous difficulty of transferring lower bounds for a circuit class to lower bounds on previous related proof systems, e.g., transferring $\text{AC}^0[p]$ lower bounds [86, 94] to $\text{AC}^0[p]$-Frege.

This entire discussion also applies to general IPS-certificates, with the following modifications. We leave a certificate $C(\vec{x}, \vec{y})$ as is, and instead of a module of syzygies we get an ideal (still finitely generated) of what we call zero-certificates. The difference between any two IPS-certificates is a zero-certificate; equivalently, a *zero-certificate* is a polynomial $C(\vec{x}, \vec{y})$ such that $C(\vec{x}, \vec{0}) = 0$ and $C(\vec{x}, \vec{F}(\vec{x})) = 0$ as well (contrast with the definition of IPS certificate, which has $C(\vec{x}, \vec{F}(\vec{x})) = 1$). The set of IPS-certificates is then the coset intersection

$$\langle y_1, \ldots, y_m \rangle \cap (1 + \langle y_1 - F_1(\vec{x}), \ldots, y_m - F_m(\vec{x}) \rangle),$$

which is either empty or a coset of the ideal of zero-certificates: $\langle y_1, \ldots, y_m \rangle \cap \langle y_1 - F_1(\vec{x}), \ldots, y_m - F_m(\vec{x}) \rangle$. The intersection ideal $\langle y_1, \ldots, y_m \rangle \cap \langle y_1 - F_1(\vec{x}), \ldots, y_m - F_m(\vec{x}) \rangle$ plays the role here that the set of syzygies played for Hilbert-like IPS-certificates.[7]

A finite generating set for the ideal of zero-certificates can be computed using Gröbner bases (see, e.g., Ene and Herzog [32, Section 3.2.1]).

Just as for Hilbert-like certificates, we get:

> **The set of all IPS-certificates for a given system of equations can be described by a single IPS-certificate, together with a finite generating set for the ideal of zero-certificates.**

Our suggestions above for lower bounds on Hilbert-like IPS apply *mutatis mutandis* to general IPS-certificates, suggesting a route to proving true size lower bounds on IPS using known techniques from algebraic complexity theory.

The discussion here raises many basic and interesting questions about the complexity of sets of (families of) functions in an ideal or module, which we propose in Section 7.

7 SUMMARY AND OPEN QUESTIONS

We introduced the Ideal Proof System IPS (Definition 1.1) and showed that it is a very close algebraic analog of Extended Frege—the most powerful, natural system currently studied for proving propositional tautologies. We showed that lower bounds on IPS imply (algebraic) circuit lower bounds, which to our knowledge is the first time that lower bounds on a proof system have been shown to imply any sort of complexity class lower bounds. Using the same techniques, we were also able to show that lower bounds on the number of *lines* (rather than the usual measure of number of monomials) in Polynomial Calculus proofs also imply strong algebraic circuit lower bounds. Because proofs in IPS are just algebraic circuits satisfying certain polynomial identity tests, many results from algebraic circuit complexity apply immediately to IPS. In particular, the chasms at depths 3 and 4 in algebraic circuit complexity imply that lower bounds on even depth 3 or depth 4 IPS proofs would be very interesting.

We introduced natural propositional axioms for polynomial identity testing (PIT) and showed that these axioms play a key role in understanding the 30-year open question of $\mathrm{AC}^0[p]$-Frege lower bounds: Either there are $\mathrm{AC}^0[p]$-Frege lower bounds on the PIT axioms or any $\mathrm{AC}^0[p]$-Frege lower bounds are as hard as showing VP $\neq$ VNP over a field of characteristic p. We expect PIT to be in P (given the connection to circuit lower bounds [50]); if this is the case, then IPS becomes a deterministic Cook–Reckhow system. Furthermore, in this case there should be some proof that PIT is in P, which we expect to be in ZFC; if the full ZFC proof translates into a ZFC *propositional*

[7] Note that the ideal of zero-certificates is not merely the set of all functions in the ideal $\langle y_1 - F_1(\vec{x}), \ldots, y_m - F_m(\vec{x}) \rangle$ that only involve the y_i, since the ideal $\langle y_1, \ldots, y_m \rangle \subseteq R[\vec{x}, \vec{y}]$ consists of all polynomials in the y_i with coefficients in $R[\vec{x}]$. Certificates only involving the y_i do have a potentially useful geometric meaning, however, which we consider in Appendix B.

proof of the PIT axioms for some specific Boolean circuit family K, then we would have that ZFC (used as a propositional proof system) p-simulates IPS.

In appendices, we discuss a variant of the Ideal Proof System that allows divisions, and its utility and limitations, as well as a geometric variant of the Ideal Proof System which suggests further geometric properties that might be of interest for computational and proof complexity. And finally, through an analysis of the set of all IPS proofs of a given unsatisfiable system of equations, we suggest how one might transfer techniques from algebraic circuit complexity to prove lower bounds on IPS (and thus on Extended Frege).

The Ideal Proof System raises many new questions, not only about itself but also about PIT, new examples of VNP functions coming from propositional tautologies, and the complexity of ideals or modules of polynomials.

In Proposition 3.7 we show that if a general IPS-certificate C has only polynomially many $\vec{y}$-monomials (with coefficients in $\mathbb{F}[\vec{x}]$), and the maximum degree of each y_i is polynomially bounded, then C can be converted to a polynomial-size Hilbert-like certificate. However, without this sparsity assumption general IPS appears to be stronger than Hilbert-like IPS.

Open Question 7.1. What, if any, is the difference in size between the smallest Hilbert-like and general IPS certificates for a given unsatisfiable system of equations? What about for systems of equations coming from propositional tautologies?

For general IPS, the preceding question was essentially answered [34] after an initial version of our article appeared (see Section 8 below); however, for C-IPS for various C, the question remains interesting.

Open Question 7.2 (Degree versus size). Is there a super-polynomial size separation—or indeed any nontrivial size separation—between IPS certificates of degree $\leq d_{small}(n)$ and IPS certificates of degree $\geq d_{large}(n)$ for some bounds $d_{small} < d_{large}$?

This question is particularly interesting in the following cases: (a) certificates for systems of equations coming from propositional tautologies, where $d_{small}(n) = n$ and $d_{large}(n) \geq \omega(n)$, since we know that every such system of equations has *some* (not necessarily small) certificate of degree $\leq n$, and (b) certificates for unsatisfiable systems of equations taking d_{small} to be the bound given by the best-known effective Nullstellensätze, which are all exponential [20, 56, 96].

Open Question 7.3. Are there tautologies for which the certificate family constructed in Theorem 1.2 is the one of minimum complexity (under p-projections or c-reductions[8])?

If there is any family $\varphi = (\varphi_n)$ of tautologies for which Question 7.3 has a positive answer and for which the certificates constructed in Theorem 1.2 are VNP-complete (Question 7.8 below), then super-polynomial size lower bounds on IPS-proofs of φ would be *equivalent* to VP $\neq$ VNP. This highlights the potential importance of understanding the structure of the set of certificates under computational reducibilities.

Since the set of all (Hilbert-like) IPS-certificates is a coset of a finitely generated ideal (respectively, module), the preceding question is a special case of considering, for a given family of cosets of ideals or modules $(f_n^{(0)} + I_n)$ $(I_n \subseteq R[x_1, \ldots, x_{\mathrm{poly}(n)}])$, the relationships under various

[8] A *c-reduction* is the analogue of Turing reductions for algebraic circuits [22]. Explicitly: An oracle computation of f from g is an algebraic circuit C with "oracle gates" such that when g is plugged in for each oracle gate, the resulting circuit computes f. We say that a family (f_n) is a c-reduction of (g_n) if there is a function $t(n) = n^{\Theta(1)}$ such that there is a polynomial-size oracle reduction from f_n to $g_{t(n)}$ for all sufficiently large n. We define c-degrees by analogy with p-degrees and denote them by $\equiv_c$.

reductions between all families of functions (f_n) with $f_n \in f_n^{(0)} + I_n$ for each n. This next question is of a more general nature than the others we ask; we think it deserves further study.

General Question 7.4. Given a family of cosets of ideals $f_n^{(0)} + I_n$ (or more generally modules) of polynomials, with $I_n \subseteq R[x_1, \ldots, x_{\text{poly}(n)}]$, consider the function families $(f_n) \in (f_n^{(0)} + I_n)$ (meaning that $f_n \in f_n^{(0)} + I_n$ for all n) under any computational reducibility $\leq$ such as p-projections. What can the $\leq$ structure look like? When, if ever, is there such a unique $\leq$-minimum (even a single nontrivial example would be interesting, as in Question 7.3)? Can there be infinitely many incomparable $\leq$-minima?

Say a $\leq$-degree $\mathbf{d}$ is "saturated" in $(f_n^{(0)} + I_n)$ if every $\leq$-degree $\mathbf{d'} \geq \mathbf{d}$ has some representative in $f^{(0)} + I$. Must saturated degrees always exist? We suspect yes, given that one may multiply any element of I by arbitrarily complex polynomials. What can the set of saturated degrees look like for a given $(f_n^{(0)} + I_n)$? Must every $\leq$-degree in $f^{(0)} + I$ be *below* some saturated degree? What can the $\leq$-structure of $f^{(0)} + I$ look like below a saturated degree?

Question 7.4 is of interest even when $f^{(0)} = 0$, that is, for ideals and modules of functions rather than their nontrivial cosets.

Open Question 7.5. Can we leverage the fact that the set of IPS certificates is not only a finitely generated coset intersection but also closed under multiplication?

We note that it is not difficult to show that a coset $c + I$ of an ideal is closed under multiplication if and only if $c^2 - c \in I$. Equivalently, this means that c is idempotent ($c^2 = c$) in the quotient ring R/I. For example, if I is a prime ideal, then R/I has no zero-divisors, and thus the only choices for $c + I$ are I and $1 + I$. We note that the ideal generated by the n^2 equations $XY - I = 0$ in the setting of the Hard Matrix Identities is prime (see Appendix A). It seems unlikely that all ideals coming from propositional tautologies are prime, however.

Remark 7.6. An IPS certificate $C(\vec{x}, \vec{y})$ for a system of equations $F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$ can be viewed as an $\mathbb{A}^1$-homotopy [73, 107] as follows. Let $V \subseteq \mathbb{A}^n \times \mathbb{A}^m$ be the graph of the map $F : \mathbb{A}^n \to \mathbb{A}^m$ defined by $\vec{x} \mapsto (F_1(\vec{x}), \ldots, F_m(\vec{x}))$. Let t be a new variable and consider the function $C'(\vec{x}, \vec{f}, t) \stackrel{def}{=} C(\vec{x}, t\vec{y})$. Then C' is an $\mathbb{A}^1$-homotopy from a function on $\mathbb{A}^n \times \mathbb{A}^m$ that vanishes on $\mathbb{A}^n \times \{0\}$ (namely, $C(\vec{x}, 0)$) to a function that is identically 1 on V (namely, $C(\vec{x}, \vec{F}(\vec{x}))$). We have not yet found any use of this fact but hope it might inspire some of our readers.

The complexity of Gröbner basis computations obviously depends on the degrees and the number of polynomials that one starts with. From this point of view, Mayr and Meyer [71] showed that the doubly exponential upper bound on the degree of a Gröbner basis [43] (see also References [68, 91]) could not be improved in general. However, in practice, many Gröbner basis computations seem to work much more efficiently, and even theoretically many classes of instances—such as proving that 1 is in a given ideal—can be shown to have only a singly-exponential degree upper bound [20, 56, 96]. These points of view are reconciled by the more refined measure of the (Castelnuovo–Mumford) *regularity* of an ideal or module. For the definition of regularity and a discussion of its close connection with the complexity of Gröbner basis and syzygy computations, we refer the reader to the original articles [11, 12, 13] or the survey [10].

Given that the syzygy module or ideal of zero-certificates are so crucial to the complexity of IPS-certificates, and the tight connection between these modules/ideals and the computation of the Gröbner basis of the ideal one started with, we ask:

General Question 7.7. Is there a formal connection between the proof complexity of individual instances of TAUT (in, say, the Ideal Proof System), and the Castelnuovo–Mumford regularity of the corresponding syzygy module or ideal of zero-certificates?

The certificates constructed in the proof of Theorem 1.2 provide many new examples of polynomial families in VNP. There are many natural questions one can ask about these polynomials. For example, the construction itself depends on the order of the clauses; does the complexity of the resulting polynomial family depend on this order? As another example, we suspect that, for any $\equiv_p$ or $\equiv_c$-degree within VNP (see Section 2.1), there is some family of tautologies for which the above polynomials are of that degree. However, we do not yet know this for even a single degree.

Open Question 7.8. Are there tautologies for which the certificates constructed in Theorem 1.2 are VNP-complete? More generally, for any given $\equiv_p$ or $\equiv_c$-degree within VNP, are there tautologies for which this certificate is of that degree?

Finally, we wish to highlight an important and very basic question:

Open Question 7.9 (Hrubeš [45]). Find a function f that vanishes on $\{0,1\}^n$ such that any IPS certificate showing that $f \in \langle x_i^2 - x_i | x \in [n] \rangle$ requires super-polynomial algebraic circuit size.

If NP $\not\subseteq$ coAM, then such an f must exist, but even if we assume just VP $\neq$ VNP, then the existence of such an f is currently unknown.

8 SUBSEQUENT DEVELOPMENTS

After the appearance of the preliminary version of this article [37], there were two significant follow-up works [34, 65], whose main results we briefly mention in the next two sections.

8.1 Noncommutative Formula IPS Is Equivalent to Frege

Li, Tzameret, and Wang [65] considered a noncommutative version of the Ideal Proof System. They consider precisely what one would imagine from the name "noncommutative formula IPS," with the one caveat that—because it is designed to consider systems of polynomial equations coming from Boolean formulas—they always include the equations $x_i x_j - x_j x_i$ among the initial equations F_i. Their main result is as follows.

Theorem (Li, Tzameret, and Wang [65]). *Noncommutative formula IPS p-simulates Frege, and Frege quasi-polynomially simulates noncommutative formula IPS. In particular, noncommutative formula IPS is quasi-polynomially equivalent to Frege.*

Their proof follows the conditional proof in this article; they get an unconditional result by giving a quasi-polynomial-size Frege proof for the deterministic polynomial-time algorithm for noncommutative formula PIT [84].

They go on to suggest that proving lower bounds on noncommutative formula IPS is potentially a more promising avenue for getting Frege lower bounds than by considering commutative formula IPS (which is p-equivalent to Frege if the formula PIT axioms have short Frege proofs). Their reasoning is that (a) noncommutative formula IPS is unconditionally quasi-polynomially equivalent to Frege and (b) exponential lower bounds on computing functions by noncommutative formulas have been known for decades [80]. Despite these facts, there are a few issues making this approach more difficult than it might appear in light of known noncommutative formula lower bounds [80]. In particular, although it remains the case that the set of noncommutative IPS certificates for a given tautology is a coset of an ideal—using essentially the same proof as in Section 6—it is now a coset of an ideal in a *noncommutative* polynomial ring. The issue here is that the remaining

discussion in Section 6 does *not* go through *a priori*, because noncommutative polynomial rings are not Noetherian: For example, the ideal $\langle yxy, yx^2y, yx^3y, \dots, \rangle$ in two noncommuting variables is not finitely generated. This raises a potentially important question about noncommutative IPS:

Open Question 8.1. Are the noncommutative analogs of the ideals from Section 6 finitely generated, when the initial system of equations comes from a CNF tautology and includes both $x_i^2 - x_i$ and $x_ix_j - x_jx_i$ for all i, j?

However, Nisan's original noncommutative circuit lower bound applied to the permanent and determinant *regardless of the ordering of variables within a monomial* [80]. This gives some hope that even if the answer to the preceding question is negative, one might be able to prove lower bounds on noncommutative IPS by proving noncommutative circuit lower bounds by considering (the noncommutative versions of) a finite generating set of the commutative version of the relevant coset of an ideal.

8.2 Improved Simulations and Lower Bounds from Circuit Complexity

Forbes, Shpilka, Tzameret, and Wigderson [34] improved some of our foundational simulations and used circuit complexity lower bounds (some of which they developed in their article) to prove lower bounds on simple systems of equations (often the Boolean axioms plus a single equation) in restricted forms of IPS.

First, they show that Hilbert-like IPS is essentially equivalent to IPS:

Theorem (Forbes, Shpilka, Tzameret, and Wigderson [34, Theorem 4.1]). *Let $F_1, \dots, F_m \in \mathbb{F}[x_1, \dots, x_n]$ be an unsatisfiable system of equations of degree at most d, over a sufficiently large field $\mathbb{F}$ ($|\mathbb{F}| \geq \text{poly}(d)$). Let s be such that each F_i can be computed by an algebraic circuit of size s, and such that there is an IPS certificate of the unsatisfiability of $F_1 = \cdots = F_m = 0$ computable by a circuit of size s. Then a Hilbert-like IPS certificate for this system can be computed by a circuit of size* $\text{poly}(d, s)$.

As with our simulation result Proposition 3.7, in their result it is also difficult to get a good handle on the depth, so the result seems to only hold for IPS of unrestricted depth.

In their article [34], they prove many results; here we just highlight the main IPS lower bounds that they get and some open questions that are underscored by their results. Though we do not discuss their techniques, they surely deserve further investigation.

For definitions of the circuit classes considered, we refer to their article [34]. For some of their results, they introduce a new variant of IPS, which we call "weakly Hilbert-like": This is IPS where the initial equations include the Boolean axioms $x_i^2 - x_i$, but the certificate is only required to be linear in the placeholder variables for the initial equations *other than* the Boolean axioms.

Theorem (Forbes, Shpilka, Tzameret, and Wigderson [34]).

- *(Theorem 4.6, subset-sum) Let $\mathbb{F}$ be a field of characteristic $\geq \text{poly}(n)$ and $\beta \notin \{0, \dots, n\}$. Then $\sum_{i \in [n]} x_iy_i - \beta, \{x_i^2 - x_i\}, \{y_i^2 - y_i\}$ is unsatisfiable, and any Hilbert-like $\Sigma \bigwedge \Sigma$-IPS certificate requires size $\geq \exp(\Omega(n))$.*
- *(Theorem 4.7, subset-sum) Let $\mathbb{F}$ be a field of characteristic $\geq \text{poly}(n)$ and $\beta \notin \{0, \dots, \binom{2n}{2}\}$. Then $\sum_{i<j} z_{i,j}x_ix_j - \beta, \{x_i^2 - x_i\}, \{z_{i,j}^2 - z_{i,j}\}$ is unsatisfiable and any Hilbert-like roABP-IPS certificate (in any variable order) requires size $\exp(\Omega(n))$. Any weakly Hilbert-like multilinear-formula-IPS certificate requires size $\exp(\Omega(\log^2 n))$, and any depth $(2d + 1)$ weakly Hilbert-like multilinear-formula-IPS certificate requires size $\exp(\log n(n/\log n)^{1/d}/d^2)$.*

- *(Theorem 4.11, AND vs OR) Let $\mathbb{F}$ be of characteristic zero, $m \neq n$. Then $x_1x_2\cdots x_n - 1 = x_1 + \cdots + x_n - m = x_i^2 - x_i = 0$ (all i) is unsatisfiable, and any $\Sigma\bigwedge\Sigma$-IPS certificate requires size* $\exp(\Omega(n))$.
- *(Theorem 4.11)* $1 + \prod_{i,j\in[n]}(z_{i,j}(x_i + x_j - x_ix_j) + (1 - z_{i,j}) = x_i^2 - x_i = z_{i,j}^2 - z_{i,j} = 0$ *is unsatisfiable, and any roABP-IPS refutation (in any variable order) requires width* $\exp(\Omega(n))$.

As they point out in their article, all of these lower bounds have the form of the Boolean axioms plus a single polynomial involving all of the variables. In particular, this allowed them to use techniques treating these formal polynomials as functions on the Boolean cube, implicitly handling the syzygies between the Boolean axioms and the one other function. But their techniques seem ill suited to handle situations with more complicated syzygies. Even the following question would be an interesting extension of their results:

Open Question 8.2. *Let $\beta \notin \{0,\ldots,2n\}$, and let $\mathbb{F}$ be a field of characteristic at least $2n+1$. Prove lower bounds on restricted versions of IPS certificates for the unsatisfiable system of equations*

$$x_1 + \cdots + x_n - x = x_{n+1} + \cdots + x_{2n} - x' = x + x' - \beta = x_1^2 - x_1 = \cdots = x_n^2 - x_n = 0.$$

APPENDICES

A DIVISIONS: THE RATIONAL IDEAL PROOF SYSTEM

We begin with an example where it is advantageous to include divisions in an IPS-certificate. Note that this is different than merely computing a polynomial IPS-certificate using divisions. In the latter case, divisions can be eliminated [98]. In the case we discuss here, the certificate itself is no longer a polynomial but is a rational function.

Example A.1. The inversion principle, one of the "Hard Matrix Identities" [95], states that

$$XY = I \Rightarrow YX = I.$$

They are called "Hard," because they were proposed as possible examples—over $\mathbb{F}_2$ or $\mathbb{Z}$—of propositional tautologies separating Extended Frege from Frege. Indeed, it was only in the past 15 years that they were shown to have efficient Extended Frege proofs [95], and it was quite nontrivial to show that they have efficient NC^2-Frege proofs [47], despite the fact that the determinant can be computed in NC^2. It is still open whether the Hard Matrix Identities have (NC^1)-Frege proofs, and believed not to be the case, essentially because it is believed that $\mathsf{DET} \not\subseteq \mathsf{NC}^1$.

In terms of ideals, the inversion principle says that the n^2 polynomials $(YX - I)_{i,j}$ (the entries of the matrix $YX - I$) are in the ideal generated by the n^2 polynomials $(XY - I)_{i,j}$. The simplest rational proof of the inversion principle that we are aware of is as follows:

$$X^{-1}(XY - I)X = YX - I.$$

Note that X^{-1} here involves dividing by the determinant. When converted into a certificate, if we write Q for a matrix of placeholder variables $q_{i,j}$ corresponding to the entries of the matrix $XY - I$, then the n^2 entries of $X^{-1}QX$ are the certificates that the entries of $YX - I$ are in the ideal generated by the entries of $XY - I$. Note that each of these certificates is a rational function that has $\det(X)$ in its denominator. Turning this into a proof that does not use divisions is one of the main foci of the article [47]; thus, if we had a proof system that allowed divisions in this manner, then it would potentially allow for significantly simpler proofs. In this particular case, we assure ourselves that this is a valid proof, because if $XY - I = 0$, then X is invertible, so X^{-1} exists (or, equivalently, $\det(X) \neq 0$).

To introduce an IPS-like proof system that allows rational certificates, we generalize the preceding reasoning. We must be careful what we allow ourselves to divide by. If we are allowed to divide

by arbitrary polynomials, then this would yield an unsound proof system, because then from any polynomials $F_1(\vec{x}), \ldots, F_m(\vec{x})$ we could derive *any* other polynomial $G(\vec{x})$ via the false "certificate" $\frac{G(x)}{F_1(x)} y_1$.

Unfortunately, although we try to eschew as many definitions as possible, our definition of the Rational Ideal Proof System and our results about it are made much cleaner by using some additional standard terminology from commutative algebra, which we now review for the reader's convenience, such as prime ideals, irreducible components of algebraic sets, and localization of rings.

A.1 Background from Commutative Algebra

The following preliminaries from commutative algebra are only needed in this appendix. We refer to the standard textbooks [6, 31, 69, 88] for proofs and further details.

The *radical* of an ideal $I \subseteq R$ is the ideal $\sqrt{I}$ consisting of all $r \in R$ such that $r^k \in I$ for some $k > 0$. An ideal I is *prime* if whenever $rs \in P$, at least one of r or s is in P. For any ideal I, its radical is equal to the intersection of the prime ideals containing I: $\sqrt{I} = \bigcap_{\text{prime } P \supseteq I} P$. We refer to prime ideals that are minimal under inclusion, subject to containing I, as "minimal over I"; there are only finitely many such prime ideals. The radical $\sqrt{I}$ is thus also equal to the intersections of the primes minimal over I.

An *algebraic set* in $\mathbb{F}^n$ is any set of the form $\{\vec{x} \in \mathbb{F}^n : F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0\}$, which we denote $V(F_1, \ldots, F_m)$ ("V" for "variety"). The algebraic set $V(F_1, \ldots, F_m)$ depends only on the ideal $\langle F_1, \ldots, F_m \rangle$, and even its radical, in the sense that $V(F_1, \ldots, F_m) = V(\sqrt{\langle F_1, \ldots, F_m \rangle})$. Conversely, the set of all polynomials vanishing on a given algebraic set V is a radical ideal, denoted $I(V)$. An algebraic set is *irreducible* if it cannot be written as a union of two algebraic proper subsets. V is irreducible if and only if $I(V)$ is prime. The *irreducible components* of an algebraic set $V = V(I)$ are the maximal irreducible algebraic subsets of V, which are exactly the algebraic sets corresponding to the prime ideals minimal over I.

If U is any subset of a ring R that is closed under multiplication—$a, b \in U$ implies $ab \in U$—then we may define the *localization* of R at U to be the ring in which we formally adjoin multiplicative inverses to the elements of U. Equivalently, we may think of the localization of R at U as the ring of fractions over R where the denominators are all in U. If P is a prime ideal, then its complement is a multiplicatively closed subset (this is an easy and instructive exercise in the definition of prime ideal). In this case, rather than speak of the localization of R at the complement $R \backslash P$, it is common usage to refer to the localization of R at P, denoted R_P. Similar statements hold for the union of finitely many prime ideals. We will use the fact that the localization of a Noetherian ring is again Noetherian (however, if R is merely finitely generated, then its localizations need not be, e.g., the localization of $\mathbb{Z}$ at $P = \langle 2 \rangle$ consists of all rationals with odd denominators; this is one of the ways in which the condition of being Noetherian is nicer than that of merely being finitely generated).

A.2 The Rational Ideal Proof System

Definition A.2 (Rational Ideal Proof System). A *rational IPS certificate* or *RIPS-certificate* that a polynomial $G(\vec{x}) \in \mathbb{F}[\vec{x}]$ is in the radical of the $\overline{\mathbb{F}}[\vec{x}]$-ideal generated by $F_1(\vec{x}), \ldots, F_m(\vec{x})$ is a rational function $C(\vec{x}, \vec{y})$ such that

(0) Write $C = C'/D$ with C', D relatively prime polynomials. Then $1/D(\vec{x}, \vec{F}(\vec{x}))$ must be in the localization of $\mathbb{F}[\vec{x}]$ at the union of the prime ideals that are minimal subject to containing the ideal $\langle F_1(\vec{x}), \ldots, F_m(\vec{x}) \rangle$ (We give a more elementary explanation of this condition below),
(1) $C(x_1, \ldots, x_n, \vec{0}) = 0$, and
(2) $C(x_1, \ldots, x_n, F_1(\vec{x}), \ldots, F_m(\vec{x})) = G(\vec{x})$.

A *RIPS proof* that $G(\vec{x})$ is in the radical of the ideal $\langle F_1(\vec{x}), \ldots, F_m(\vec{x})\rangle$ is an $\mathbb{F}$-algebraic circuit with divisions on inputs $x_1, \ldots, x_n, y_1, \ldots, y_m$ computing some RIPS certificate.

Condition (0) is equivalent to: If $G(\vec{x})$ is an invertible constant, then $D(\vec{x}, \vec{y})$ is also an invertible constant and thus C is a polynomial; otherwise, after substituting the $F_i(\vec{x})$ for the y_i, the denominator $D(\vec{x}, \vec{F}(\vec{x}))$ does not vanish identically on any of the irreducible components (over the algebraic closure $\overline{\mathbb{F}}$) of the algebraic set $V(F_1(\vec{x}), \ldots, F_m(\vec{x})) \subseteq \overline{\mathbb{F}}^n$. In particular, for proofs of unsatisfiability of systems of equations, the Rational Ideal Proof System reduces by definition to the Ideal Proof System. For derivations of one polynomial from a set of polynomials, this need not be the case, however; indeed, there are examples for which *every* RIPS-certificate has a nonconstant denominator, that is, there is a RIPS-certifiate, but there are no IPS-certificates (see Example A.4).

Grigoriev and Hirsch [35, Section 2.5] introduced a related system, denoted (F-)PC$_{\sqrt{}}$, for proving that a polynomial is in the radical of an ideal. Beyond the differences between IPS and F-PC (discussed just after Definition 1.8), RIPS also allows potentially more general divisions than (F-)PC$_{\sqrt{}}$.

Proposition A.3. *The Rational Ideal Proof System is sound. That is, if there is a RIPS-certificate that $G(\vec{x})$ is in the radical of $\langle F_1(\vec{x}), \ldots, F_m(\vec{x})\rangle$, then $G(\vec{x})$ is in fact in the radical of $\langle F_1(\vec{x}), \ldots, F_m(\vec{x})\rangle$.*

Proof. Let $C(\vec{x}, \vec{y}) = \frac{1}{D(\vec{x}, \vec{y})} C'(\vec{x}, \vec{y})$ be a RIPS certificate that G is in $\sqrt{\langle F_1, \ldots, F_m\rangle}$, where D and C' are relatively prime polynomials. Then $C'(\vec{x}, \vec{y})$ is an IPS-certificate that $G(\vec{x})D(\vec{x}, \vec{F}(\vec{x}))$ is in the ideal $\langle F_1(\vec{x}), \ldots, F_m(\vec{x})\rangle$ (recall Definition 1.8). Let $D_F(\vec{x}) = D(\vec{x}, \vec{F}(\vec{x}))$.

Geometric proof: Since $G(\vec{x})D_F(\vec{x}) \in \langle F_1(\vec{x}), \ldots, F_m(\vec{x})\rangle$, GD_F must vanish identically on every irreducible component of the algebraic set $V(F_1, \ldots, F_m)$. On each irreducible component V_i, since $D_F(\vec{x})$ does not vanish identically on V_i, $G(\vec{x})$ must vanish everywhere except for the proper subset $V(D_F(\vec{x})) \cap V_i$. Since D_F does not vanish identically on V_i, we have $\dim V(D_F) \cap V_i \leq \dim V_i - 1$ (in fact this is an equality). In particular, this means that G must vanish on a dense subset of V_i. Since G is a polynomial, by (Zariski-) continuity, G must vanish on all of V_i. Finally, since G vanishes on every irreducible component of $V(F_1, \ldots, F_m)$, it vanishes on $V(F_1, \ldots, F_m)$ itself, and by the Nullstellensatz, $G \in \sqrt{\langle F_1, \ldots, F_m\rangle}$.

Algebraic proof: For each prime ideal $P_i \subseteq \overline{\mathbb{F}}[\vec{x}]$ that is minimal subject to containing $\langle F_1, \ldots, F_m\rangle$, D_F is not in P_i by the definition of *R*IPS-certificate. Since $GD_F \in \langle F_1, \ldots, F_m\rangle \subseteq P_i$, by the definition of prime ideal G must be in P_i. Hence G is in the intersection $\bigcap_i P_i$ over all minimal prime ideals $P_i \supseteq \langle F_1, \ldots, F_m\rangle$. This intersection is exactly the radical $\sqrt{\langle F_1, \ldots, F_m\rangle}$. □

Any derivation of a polynomial G that is in the radical of an ideal I but not in I itself will require divisions. Although it is not *a priori* clear that RIPS could derive even one such G, the next example shows that this is the case. In other words, the next example shows that certain derivations *require* rational functions.

Example A.4. Let $G(x_1, x_2) = x_1$, $F_1(\vec{x}) = x_1^2$, $F_2(\vec{x}) = x_1 x_2$. Then $C(\vec{x}, \vec{y}) = \frac{1}{x_1 - x_2}(y_1 - y_2)$ is a RIPS-certificate that $G \in \sqrt{\langle F_1, F_2\rangle}$: By plugging in one can verify that $C(\vec{x}, \vec{F}(\vec{x})) = G(\vec{x})$. For Condition (0), we see that $V(F_1, F_2)$ is the entire x_2-axis, on which $x_1 - x_2$ only vanishes at the origin. However, there is no IPS-certificate that $G \in \langle F_1, F_2\rangle$, since G is *not* in $\langle F_1, F_2\rangle$: $\langle F_1, F_2\rangle = \{x_1(H_1(\vec{x})x_1 + H_2(\vec{x})x_2)\}$ where H_1, H_2 may be arbitrary polynomials. Since the only constant of the form $H_1(\vec{x})x_1 + H_2(\vec{x})x_2$ is zero, $G(x) = x \notin \langle F_1, F_2\rangle$.

In the following circumstances a RIPS-certificate can be converted into an IPS-certificate.

Notational convention. Throughout, we continue to use the notation that if D is a function of the placeholder variables y_i (and possibly other variables), then D_F denotes D after substituting in $F_i(\vec{x})$ for the placeholder variable y_i.

Proposition A.5. *If $C = C'/D$ is a RIPS proof that $G(\vec{x}) \in \sqrt{\langle F_1(\vec{x}), \ldots, F_m(\vec{x})\rangle}$, such that $D_F(\vec{x})$ does not vanish* anywhere *on the algebraic set $V(F_1(\vec{x}), \ldots, F_m(\vec{x}))$, then $G(\vec{x})$ is in fact in the ideal $\langle F_1(\vec{x}), \ldots, F_m(\vec{x})\rangle$. Furthermore, there is an IPS proof that $G(\vec{x}) \in \langle F_1(\vec{x}), \ldots, F_m(\vec{x})\rangle$ of size* $\mathrm{poly}(|C|, |E|)$ *where E is an IPS proof of the unsolvability of $D_F(\vec{x}) = F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$.*

Proof. Since $D_F(\vec{x})$ does not vanish anywhere on $V(F_1, \ldots, F_m)$, the system of equations $D_F(\vec{x}) = F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$ is unsolvable.

Geometric proof idea: The preceding means that when restricted to the algebraic set $V(F_1, \ldots, F_m)$, D_F has a multiplicative inverse Δ. Rather than dividing by D, we then multiply by Δ, which, for points on $V(F_1, \ldots, F_m)$, amounts to the same thing.

Algebraic proof: Let $E(\vec{x}, \vec{y}, d)$ be an IPS-certificate for the unsolvability of this system, where d is a new placeholder variable corresponding to the polynomial $D_F(\vec{x}) = D(\vec{x}, \vec{F}(\vec{x}))$. By separating out all of the terms involving d, we may write $E(\vec{x}, \vec{y}, d)$ as $d\Delta(\vec{x}, \vec{y}, d) + E'(\vec{x}, \vec{y})$. As $E(\vec{x}, \vec{F}(\vec{x}), D_F(\vec{x})) = 1$ (by the definition of IPS), we get:

$$D_F(\vec{x})\Delta(\vec{x}, \vec{F}(\vec{x}), D_F(\vec{x})) = 1 - E'(\vec{x}, \vec{F}(\vec{x})).$$

Since $E'(\vec{x}, \vec{y}) \in \langle y_1, \ldots, y_m\rangle$, this tells us that $\Delta(\vec{x}, \vec{F}(\vec{x}), D_F(\vec{x}))$ is a multiplicative inverse of $D_F(\vec{x})$ modulo the ideal $\langle F_1, \ldots, F_m\rangle$. The idea is then to multiply by Δ instead of dividing by D. More precisely, the following is an IPS-proof that $G \in \langle F_1, \ldots, F_m\rangle$:

$$C_\Delta(\vec{x}, \vec{y}) \stackrel{def}{=} C'(\vec{x}, \vec{y})\Delta(\vec{x}, \vec{y}, D(\vec{x}, \vec{y})) + G(\vec{x})E'(\vec{x}, \vec{y}). \tag{8}$$

Since C' and E' must individually be in $\langle y_1, \ldots, y_m\rangle$, the entirety of C_Δ is as well. To see that we get $G(\vec{x})$ after plugging in the $F_i(\vec{x})$ for the y_i, we compute:

$$\begin{aligned} C_\Delta(\vec{x}, \vec{F}(\vec{x})) &= C'(\vec{x}, \vec{F}(\vec{x}))\Delta(\vec{x}, \vec{F}(\vec{x}), D(\vec{x}, \vec{F}(\vec{x}))) + G(\vec{x})E'(\vec{x}, \vec{F}(\vec{x})) \\ &= C'(\vec{x}, \vec{F}(\vec{x}))\left(\frac{1 - E'(\vec{x}, \vec{F}(\vec{x}))}{D_F(\vec{x})}\right) + G(\vec{x})E'(\vec{x}, \vec{F}(\vec{x})) \\ &= G(\vec{x})\left(1 - E'(\vec{x}, \vec{F}(\vec{x}))\right) + G(\vec{x})E'(\vec{x}, \vec{F}(\vec{x})) \\ &= G(\vec{x}). \end{aligned}$$

Finally, we give an upper bound on the size of a circuit for C_Δ. The numerator and denominator of a rational function computed by a circuit of size s can be computed individually by circuits of size $O(s)$. The basic idea, going back to Strassen [98], is to replace each wire by a pair of wires explicitly encoding the numerator and denominator, to replace a multiplication gate by a pair of multiplication gates—since $(A/B) \times (C/D) = (A \times C)/(B \times D)$—and to replace an addition gate by the appropriate gadget encoding the expression $(A/B) + (C/D) = (AD + BC)/BD$. In particular, we may assume that a circuit computing C'/D has the following form: It first computes C' and D separately and then has a single division gate computing C'/D. Thus from a circuit for C, we can get circuits of essentially the same size for both C' and D. Given a circuit for $E = d'\Delta + E'$, we get a circuit for E' by setting $d' = 0$. We can then get a circuit for $d'\Delta$ as $E - E'$. From a circuit for $d'\Delta$, we can get a circuit for Δ alone by first dividing $d'\Delta$ by d' and then eliminating that division using Strassen [98]. Combining these, we then easily construct a circuit for the IPS-certificate C_Δ of size $\mathrm{poly}(|C|, |E|)$. □

Example A.6. Returning to the inversion principle, we find that the certificate from Example A.1 only divided by $\det(X)$, which we already remarked does not vanish *anywhere* that $XY - I$ vanishes. By the preceding proposition, there is thus an IPS-certificate for the inversion principle of polynomial size, *if* there is an IPS-certificate for the unsatisfiability of $\det(X) = 0 \wedge XY - I = 0$ of polynomial size. In this case, we can guess the multiplicative inverse of $\det(X)$ modulo $XY - I$, namely $\det(Y)$, since we know that $\det(X)\det(Y) = 1$ if $XY = I$. Hence, we can try to find a certificate for the unsatisfiability of $\det(X) = 0 \wedge XY - I = 0$ of the form

$$\det(X)\det(Y) + (\text{something in the ideal of } \langle (XY - I)_{i,j\in[n]} \rangle) = 1.$$

In other words, we want a refutation-style IPS-proof of the implication $XY = I \Rightarrow \det(X)\det(Y) = 1$, which is another one of the Hard Matrix Identities. Such a refutation is exactly what Hrubeš and Tzameret provide [47].

In fact, for this particular example, we could have anticipated that a rational certificate was unnecessary, because the ideal generated by $XY - I$ is prime and hence radical. (Indeed, the ring $\mathbb{F}[X, Y]/\langle XY - I\rangle$ is the coordinate ring of the algebraic group $\mathrm{GL}_n(\mathbb{F})$, which is an irreducible variety.)

Unfortunately, the Rational Ideal Proof System is not complete, as the next example shows.

Example A.7. Let $F_1(x) = x^2$ and $G(x) = x$. Then $G(x) \in \sqrt{\langle F_1(\vec{x})\rangle}$, but any RIPS certificate would show $G(x)D(x) = F_1(x)H(x)$ for some D, H. Plugging in, we get $xD(x) = x^2H(x)$, and by unique factorization we must have that $D(x) = xD'(x)$ for some D'. But then D vanishes identically on $V(F_1)$, contrary to the definition of RIPS-certificate.

To get a more complete proof system, we could generalize the definition of RIPS to allow dividing by any polynomial that does not vanish to appropriate *multiplicity* on each irreducible component (see, e.g., Eisenbud [31, Section 3.6] for the definition of multiplicity). For example, this would allow dividing by x to show that $x \in \sqrt{\langle x^2\rangle}$ but would disallow dividing by x^2 or any higher power of x. However, the proof of soundness of this generalized system is more involved, and the results of the next section seem not to hold for such a proof system. As of this writing, we do not know of any better characterization of when RIPS certificates exist other than the definition itself.

Definition A.8. A RIPS certificate is *Hilbert-like* if the denominator does not involve the placeholder variables y_i and the numerator is $\vec{y}$-linear. In other words, a Hilbert-like RIPS certificate has the form $\frac{1}{D(\vec{x})}\sum_i y_i G_i(\vec{x})$.

Lemma A.9. *If there is a RIPS certificate that $G \in \sqrt{\langle F_1, \ldots, F_m\rangle}$, then there is a Hilbert-like RIPS certificate proving the same.*

Proof. Let $C = C'(\vec{x}, \vec{y})/D(\vec{x}, \vec{y})$ be a RIPS certificate. First, replace the denominator by $D_F(\vec{x}) = D(\vec{x}, \vec{F}(\vec{x}))$. Next, for each monomial appearing in C', replace all but one of the y_i in that monomial with the corresponding $F_i(\vec{x})$, reducing the monomial to one that is $\vec{y}$-linear. □

As in the case of IPS, we only know how to guarantee a size-efficient reduction under a sparsity condition. The following is the RIPS-analogue of Proposition 3.7.

Corollary A.10. *If $C = C'/D$ is a RIPS proof that $G \in \sqrt{\langle F_1, \ldots, F_m\rangle}$, where the numerator C' satisfies the same sparsity condition as in Proposition 3.7, then there is a Hilbert-like RIPS proof that $G \in \sqrt{\langle F_1, \ldots, F_m\rangle}$, of size* $\mathrm{poly}(|C|)$.

Proof. We follow the proof of Lemma A.9, making each step effective. As in the last paragraph of the proof of Proposition A.5, any circuit with divisions computing a rational function C'/D,

where C', D are relatively prime polynomials can be converted into a circuit without divisions computing the pair (C', D). By at most doubling the size of the circuit, we can assume that the subcircuits computing C' and D are disjoint. Now replace each y_i input to the subcircuit computing D with a small circuit computing $F_i(\vec{x})$. Next, we apply sparse multivariate interpolation to the numerator C' exactly as in Proposition 3.7. The resulting circuit now computes a Hilbert-like RIPS certificate. □

A.3 Towards Lower Bounds

We begin by noting that, since the numerator and denominator can be computed separately (originally due to Strassen [98], see the proof of Proposition A.5 above for the idea), it suffices to prove a lower bound on, for each RIPS-certificate, either the denominator or the numerator.

As in the case of Hilbert-like IPS and general IPS (recall Section 6), the set of RIPS certificates showing that $G \in \sqrt{\langle F_1, \ldots, F_m \rangle}$ is a coset of a finitely generated ideal.

Lemma A.11. *The set of RIPS-certificates showing that $G \in \sqrt{\langle F_1, \ldots, F_m \rangle}$ is a coset of a finitely generated ideal in R, where R is the localization of $\mathbb{F}[\vec{x}, \vec{y}]$ at $\bigcup_i P_i$, where the union is over the prime ideals minimal over $\langle F_1, \ldots, F_m \rangle$.*

Similarly, the set of Hilbert-like RIPS certificates is a coset of a finitely generated submodule of R'^m, where $R' = R \cap \mathbb{F}[\vec{x}]$ is the localization of $\mathbb{F}[\vec{x}]$ at $\bigcup_i (P_i \cap \mathbb{F}[\vec{x}])$.

Proof. The proof is essentially the same as that of Lemma 6.1, but with one more ingredient. Namely, we need to know that the rings R and R' are Noetherian. This follows from the fact that polynomial rings over fields are Noetherian, together with the general fact that any localization of a Noetherian ring is again Noetherian. □

Exactly analogous to the the case of IPS certificates, we define general and Hilbert-like RIPS zero-certificates to be those for which, after plugging in the F_i for y_i, the resulting function is identically zero. In the case of Hilbert-like RIPS, these are again syzygies of the F_i but now syzygies with coefficients in the localization $R' = \mathbb{F}[\vec{x}]_{P_1 \cup \cdots \cup P_k}$.

However, somewhat surprisingly, we seem to be able to go further in the case of RIPS than IPS, as follows. In general, the ring $\mathbb{F}[\vec{x}, \vec{y}]_{P_1 \cup \cdots \cup P_k}$ is a Noetherian *semi-local* ring, that is, in addition to being Noetherian, it has finitely many maximal ideals, namely $P_1, \ldots, P_k$. Modules over semi-local rings, including ideals, enjoy properties not shared by ideals and modules over arbitrary rings.

In the special case when there is just a single prime ideal P_1, the localization is a *local* ring (just one maximal ideal). We note that this is the case in the setting of the Inversion Principle, as the ideal generated by the n^2 polynomials $XY - I$ is prime. Local rings are in some ways very close to fields—if R is a local ring with unique maximal ideal P, then R/P is a field—and modules over local rings are much closer to vector spaces than are modules over more general rings. This follows from the fact that M/P is then in fact a vector space over the field R/P, together with Nakayama's Lemma (see, e.g., Eisenbud [31, Corollary 4.8] or Reid [88, Section 2.8]). One nice feature is that, if M is a module over a local ring, then every minimal generating set has the same size, which is the dimension of M/P as an R/P-vector space. We also get that for every minimal generating set $b_1, \ldots, b_k$ of M ("b" for "basis," even though the word basis is reserved for free modules), for each $m \in M$, any two representations $m = \sum_{i=1}^{k} r_i b_i$ with $r_i \in R$ differ by an element in PM. This near-uniqueness could be very helpful in proving lower bounds, as normal forms have proved useful in proving many circuit lower bounds.

Open Question A.12. Does every RIPS proof of the $n \times n$ Inversion Principle $XY = I \Rightarrow YX = I$ require computing a determinant? That is, is it the case that for every RIPS certificate $C = C'/D$, some determinant of size $n^{\Omega(1)}$ reduces to one of C, C', D by a $O(\log n)$-depth circuit reduction?

A positive answer to this question would imply that the Hard Matrix Identities do not have $O(\log n)$-depth RIPS proofs unless the determinant can be computed by a polynomial-size algebraic formula. Since IPS (and hence RIPS) simulates Frege-style systems in a depth-preserving way (Theorem 3.5), a positive answer would also imply that there are not (NC^1-)Frege proofs of the Boolean Hard Matrix Identities unless the determinant has polynomial-size *algebraic* formulas. Although answering this question may be difficult, the fact that we can even *state* such a precise question on this matter should be contrasted with the preceding state of affairs regarding Frege proofs of the Boolean Hard Matrix Identities (which was essentially just a strong intuition that they should not exist unless the determinant is in NC^1).

B GEOMETRIC IPS-CERTIFICATES

B.1 Background from Commutative Algebra

Let R be a ring. A function $(F_1, \ldots, F_m) = F : R^n \to R^m$ is called a polynomial map if each coordinate $F_i(\vec{x})$ is a polynomial. Given a polynomial map $F : R^n \to R^m$, define $F_* : R[y_1, \ldots, y_m] \to R[x_1, \ldots, x_n]$ to be the map of R-algebras—i.e., $F_*(1) = 1$ and $F_*(rf) = rF_*(f)$ for all $r \in R$ and all $f \in R[y_1, \ldots, y_m]$—such that $F_*(y_i) = F_i(\vec{x})$. For convenience, let $A = R[y_1, \ldots, y_m]$ and $B = R[x_1, \ldots, x_n]$. Then the map $F_* : A \to B$ makes B into an A-module by $a \cdot b \stackrel{def}{=} F_*(a)b$. The following is a standard definition in commutative algebra and algebraic geometry:

Definition B.1. The map $F : R^n \to R^m$ is *finite* if the corresponding map F_* makes $R[x_1, \ldots, x_n]$ into a finitely generated module over $R[y_1, \ldots, y_m]$.

The key fact that we will need about finite maps is as follows:

Proposition B.2 (See, e.g., [31, Corollary 9.3]). *Suppose $F : R^n \to R^m$ is a finite map. Then the image of F is Zariski-closed—equivalently, an algebraic set—in R^m.*

B.2 The Geometric Ideal Proof System

We may consider $F_1(x_1, \ldots, x_n), \ldots, F_m(x_1, \ldots, x_n)$ as a polynomial map $F = (F_1, \ldots, F_m) : \mathbb{F}^n \to \mathbb{F}^m$. Then this system of polynomials has a common zero if and only if $\vec{0}$ is the image of F. In fact, we show that for any system of equations coming from a Boolean tautology, the system of polynomials has a common zero if and only if $\vec{0}$ is in the *closure* of the image of F (this is true regardless of whether the equations include $x_i^2 - x_i = 0$, $x_i^2 - 1 = 0$, or neither of these).

The preceding is the geometric picture we pursue in this section; now we describe the corresponding algebra. The set of IPS certificates is the intersection of the ideal $\langle y_1, \ldots, y_m \rangle$ with the coset $1 + \langle y_1 - F_1(\vec{x}), \ldots, y_m - F_m(\vec{x}) \rangle$. The map $a \mapsto 1 - a$ is a bijection between this coset intersection and the coset intersection $(1 + \langle y_1, \ldots, y_m \rangle) \cap \langle y_1 - F_1(\vec{x}), \ldots, y_m - F_m(\vec{x}) \rangle$. In particular, the system of equations $F_1 = \cdots = F_m = 0$ is unsatisfiable if and only if the latter coset intersection is nonempty.

We show below that if the latter coset intersection contains a polynomial involving only the y_i's—that is, its intersection with the subring $\mathbb{F}[\vec{y}]$ (rather than the much larger ideal $\langle \vec{y} \rangle \subseteq \mathbb{F}[\vec{x}, \vec{y}]$) is nonempty—then $\vec{0}$ is not even in the closure of the image of F. Hence we call such polynomials "geometric certificates":

Definition B.3 (The Geometric Ideal Proof System). A *geometric IPS certificate* that a system of $\mathbb{F}$-polynomial equations $F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$ is unsatisfiable over $\overline{\mathbb{F}}$ is a polynomial $C \in \mathbb{F}[y_1, \ldots, y_m]$ such that

(1) $C(0, 0, \ldots, 0) = 1$, and
(2) $C(F_1(\vec{x}), \ldots, F_m(\vec{x})) = 0$. In other words, C is a polynomial relation amongst the F_i.

A *geometric IPS proof* of the unsatisfiability of $F_1 = \cdots = F_m = 0$, or a *geometric IPS refutation* of $F_1 = \cdots = F_m = 0$, is an $\mathbb{F}$-algebraic circuit on inputs $y_1, \ldots, y_m$ computing some geometric certificate of unsatisfiability.

If C is a geometric certificate, then $1 - C$ is an IPS certificate that involves only the y_i's, somewhat the "opposite" of a Hilbert-like certificate. Hence the smallest circuit size of any geometric certificate is at least the smallest circuit size of any algebraic certificate. We do not know, however, if these complexity measures are polynomially related, as highlighted in the next question.

We call a system of equations "standard Boolean" if it includes $x_i^2 = x_i$ for all i and "multiplicative Boolean" if it includes $x_i^2 = 1$ for all i; by "Boolean system of equations" we mean either of these.

Open Question B.4. For Boolean systems of equations, is Geometric IPS polynomially equivalent to IPS? That is, is there always a geometric certificate whose circuit size is at most a polynomial in the circuit size of the smallest algebraic certificate?

Although the Nullstellensatz does not guarantee the existence of geometric certificates for arbitrary unsatisfiable systems of equations—and, indeed, geometric certificates need not always exist—for *Boolean* systems of equations geometric certificates always exist. In fact, this holds for any system of equations that contains at least one polynomial containing only the variable x_i, for each variable x_i:

Proposition B.5. *Let R be any ring. A Boolean system of equations over R—or more generally any system of equations containing, for each variable x_i, at least one non-constant equation involving only x_i—has a common root if and only if it does not have a geometric certificate.*

Proof. Let $F_1, \ldots, F_m$ be an unsatisfiable system of equations over R satisfying the conditions of Proposition B.5, and let $F = (F_1, \ldots, F_m) : R^n \to R^m$ be the corresponding polynomial map.

First, suppose that $F_1 = \cdots = F_m = 0$ has a solution. Then $\vec{0} \in \mathrm{Im}(F)$, so any $C(y_1, \ldots, y_m)$ that vanishes everywhere on $\mathrm{Im}(F)$, as required by condition (2) of Definition B.3, must vanish at $\vec{0}$. In other words, $C(0, \ldots, 0) = 0$, contradicting condition (1). So there are no geometric certificates.

Conversely, suppose that there are no geometric certificates. Then every polynomial $C(\vec{y})$ that vanishes on $\mathrm{Im}(F)$ also vanishes at $\vec{0}$. Hence, by definition, the origin is in the Zariski closure $\overline{\mathrm{Im}(F)}$. But we will now show that in fact the image of F is already closed, so $\mathrm{Im}(F) = \overline{\mathrm{Im}(F)}$, and thus $\vec{0}$ is in the image of F.

Since F contains, for each variable x_i, one equation involving only x_i, F is finite (see Definition B.1). To see this, let d_i be the smallest degree of any of the F_j that depend only on x_i; by assumption each d_i is finite and at least one. Then $R[x_1, \ldots, x_m]$ is generated, as a module over $R[F_1, \ldots, F_m]$, by the finite set $\{\vec{x}^{\vec{e}} | (\forall i)[0 \le e_i \le d_i]\}$. By Proposition B.2, the image of F is thus closed, hence is equal to its closure. By the preceding paragraph, this completes the proof. □

As we see, the preceding proposition followed almost immediately from standard facts in algebraic geometry, without concern for the nature of the equations coming from a Boolean tautology. However, we also show that even *without* the equations $x_i^2 = x_i$ (nor $x_i^2 = 1$), if φ is an unsatisfiable CNF, then the corresponding set of polynomial equations has a geometric IPS certificate:

Proposition B.6. *Let $\mathbb{F}$ be either (1) any algebraically closed field or (2) a dense subfield of $\mathbb{C}$ (in the Euclidean topology). For a Boolean CNF formula $\varphi(x_1, \ldots, x_n)$ with m clauses, let $F_\varphi : \mathbb{F}^n \to \mathbb{F}^m$ denote the corresponding polynomial map. Note here that we did* not *add $x_i^2 - x_i$ (nor $x_i^2 - 1$, nor any similar) to F.*

A Boolean CNF formula φ is unsatisfiable if and only if F_φ has a geometric IPS$_\mathbb{F}$ certificate.

The field of algebraic numbers, and even the field $\mathbb{Q}(i)$ (the smallest subfield of $\mathbb{C}$ containing both the rationals and i) are potentially interesting examples of fields satisfying (2).

Note that in this case we cannot merely apply the idea of Proposition B.5, since the polynomial map F corresponding to φ need not be finite nor have closed image, as the following example shows.

Example B.7 (Unsatisfiable CNF with Non-closed Image). Let φ be the unsatisfiable CNF $\neg IND_2$ (for "induction"), namely $\varphi = x \wedge (x \to y) \wedge (y \to z) \wedge (\neg z) = x \wedge (\neg x \vee y) \wedge (\neg y \vee z) \wedge (\neg z)$. This translates into the polynomials

$$\begin{aligned} F_1 &= 1 - x \\ F_2 &= x(1-y) \\ F_3 &= y(1-z) \\ F_4 &= z. \end{aligned}$$

In this case, we can compute the image exactly to see that it is not closed. (We could also do this for IND_1, but in that case the image is in fact closed.) Namely, suppose (a, b, c, d) is in the image. Then we have $a = 1 - x$ and $d = z$, and, consequently,

$$b = x(1-y) = (1-a)(1-y) \qquad \text{and} \qquad c = y(1-z) = y(1-d).$$

When $a = 1$, b must be 0; similarly, when $d = 1$, c must be 0. When both $a \neq 1$ and $d \neq 1$, we can solve both of the equations above for y and equate the results, to get $(1-d)(1-a-b) = c(1-a)$. Note that the only point in the image with $a = d = 1$ is the point $(1, 0, 0, 1)$, which satisfies the preceding equation. Thus, the image is $V((1-d)(1-a-b) - c(1-a)) \backslash [(V(a-1) \cup V(d-1))] \cup \{(1, 0, c, d) | d \neq 1\} \cup \{(a, b, 0, 1) | a \neq 1\} \cup \{(1, 0, 0, 1)\}$. To see that this is not a closed set, note that its intersection with the closed set $\{(1, 0, c, d)\} = V(a-1, b)$ consists of a non-closed set: the union of the point $(1, 0, 0, 1)$ and the set $\{(1, 0, c, d) | d \neq 1\}$, which is the complement of a line.

(This example can easily be modified to give an example of a satisfiable CNF with non-closed image; namely, un-negate z and consider $\varphi = x \wedge (\neg x \vee y) \wedge (\neg y \vee z) \wedge z$. The intersection of the image of the corresponding F with the $(1, 0, c, d)$-plane is the union of the point $(1, 0, 0, 0)$ and the non-closed set $\{(1, 0, c, d) | d \neq 0\}$.)

Proof of Proposition B.6. Let $F : \mathbb{F}^n \to \mathbb{F}^m$ be the polynomial map corresponding to φ as above. As with Proposition B.5, the key to the proof is that $\vec{0}$ is in the closure $\overline{\mathrm{Im}(F)}$ if and only if $\vec{0}$ is in fact in the image of F. The rest of the reasoning is the same as in Proposition B.5. Because φ is in CNF, each polynomial F_i is a product of terms, each of which is either x_i or $(1 - x_i)$. For such maps F, we will show that $\vec{0} \in \overline{\mathrm{Im}(F)}$ if and only if $\vec{0} \in \mathrm{Im}(F)$. Only the "only if" direction is nontrivial.

So suppose $\vec{0}$ is in the closure of the image of F. We first prove case (2) (the characteristic zero case) using very little beyond arguments about convergence of Cauchy sequences, then we prove case (1), the case of an arbitrary algebraically closed field.

In both cases, the basic idea is that a clause gets mapped to a product like $x_1 x_2 \cdots x_k (1 - x_{k+1}) \cdots (1 - x_\ell)$, and for such a polynomial to approach zero in the limit, one of its factors must approach zero. For each such factor, rather than considering a limit, we simply set its value to zero. (Setting $1 - x_k$ to zero amounts to setting $x_k = 1$.) In both cases (1) and (2), making this idea rigorous requires some technicalities. We do (2) first only because the technicalities in that case may be more familiar to more readers.

(2) (Dense subfields of $\mathbb{C}$.) First, we note that the closure of the image of F in the Zariski topology agrees with its closure in the standard Euclidean topology on $\mathbb{F}^n$, induced by the Euclidean topology on $\mathbb{C}^n$. For $\mathbb{F} = \mathbb{C}$, see, e.g., Mumford [79, Theorem 2.33]. For other dense $\mathbb{F} \subsetneq \mathbb{C}$, suppose $\vec{y}$ is in

the $\mathbb{F}$-Zariski closure of $F(\mathbb{F}^n)$, that is, every $\mathbb{F}$-polynomial that vanishes everywhere on $F(\mathbb{F}^n)$ also vanishes at $\vec{y}$. By the density of $\mathbb{F}$ in $\mathbb{C}$, every $\mathbb{C}$-polynomial that vanishes on $F(\mathbb{F}^n)$ also vanishes at $\vec{y}$, so $\vec{y}$ is in the $\mathbb{C}$-Zariski closure of $F(\mathbb{F}^n)$ and therefore also in the closure of $F(\mathbb{C}^n)$. By the aforementioned result for $\mathbb{C}$, there is a Cauchy sequence of points $\vec{v}^{(1)}, \vec{v}^{(2)}, \ldots \in \mathbb{C}^n$ such that each $\vec{v}^{(i)}$ is in $F(\mathbb{C}^n)$ and $\vec{y} = \lim_{k\to\infty} \vec{v}^{(k)}$. As $\mathbb{F}$ is dense in $\mathbb{C}$ in the Euclidean topology, $F(\mathbb{F}^n)$ is dense in $F(\mathbb{C}^n)$ in the Euclidean topology. Thus there is a Cauchy sequence of points $\vec{v}'^{(1)}, \vec{v}'^{(2)}, \ldots \in F(\mathbb{F}^n)$ such that $|\vec{v}^{(k)} - \vec{v}'^{(k)}| \le 1/k$ for all k. Hence $\lim_{k\to\infty} \vec{v}'^{(k)} = \lim_{k\to\infty} \vec{v}^{(k)} = \vec{y}$.

In particular, $\vec{0}$ is in the (Zariski-)closure of the image of F if and only if there is a Cauchy sequence of points $\vec{v}^{(1)}, \vec{v}^{(2)}, \vec{v}^{(3)}, \ldots$ in $F(\mathbb{F}^n)$ such that $\lim_{k\to\infty} \vec{v}^{(k)} = 0$. As each $\vec{v}^{(k)}$ is in the image of F, there is some point $\vec{v}^{(k)} \in \mathbb{F}^n$ such that $\vec{v}^{(k)} = F(\vec{v}^{(k)})$. As the $\vec{v}^{(k)}$ approach the origin, each $F_i(\vec{v}^{(k)})$ approaches 0, since it is the ith coordinate of $\vec{v}^{(k)}$ ($\vec{v}_i^{(k)} = F_i(\vec{v}^{(k)})$).

We will show how to construct a $\vec{\mu} \in \mathbb{F}^n$ such that $\vec{F}(\vec{\mu}) = \vec{0}$. Without loss of generality, by renumbering if necessary, suppose that $F_1(\vec{x})$ is $x_1 x_2 \cdots x_k (1 - x_{k+1})(1 - x_{k+2}) \cdots (1 - x_\ell)$. As $F_1(\vec{v}^{(k)})$ approaches 0, at least one of its factors must get arbitrarily close to zero infinitely often, say x_1. (The case of $1 - x_i$ approaching zero, for $k + 1 \le i \le \ell$, is handled similarly.) Then there is an infinite subsequence $(\vec{v}^{(k_i)})_{i=1,2,3,\ldots}$ of $(\vec{v}^{(k)})_{k=1,2,3,\ldots}$ such that the first coordinates of this subsequence form a Cauchy sequence in $\mathbb{F}$ whose limit is 0. Since $\vec{F}(\vec{v}^{(k)})$ is a Cauchy sequence whose limit is $\vec{0}$, and $\vec{v}^{(k_i)}$ is an infinite subsequence of $\vec{v}^{(k)}$, we have that $\vec{F}(\vec{v}^{k_i})$ is also a Cauchy sequence whose limit is $\vec{0}$. Replace $\vec{v}^{(k)}$ by its subsequence $\vec{v}^{(k_i)}$ and renumber.

We now have a Cauchy sequence $\vec{F}(\vec{v}^{(k)})$ whose limit is zero, and such that at least one of the factors of F_1 corresponds to a coordinate of $\vec{v}^{(k)}$ that is itself a Cauchy sequence approaching 0 or 1. We now repeat this argument with the new sequence $\vec{v}^{(k)}$ for F_2, then for F_3, and so on. The result is a sequence $\vec{v}^{(k)} \in \mathbb{F}^n$ such that $\vec{F}(\vec{v}^{(k)})$ is a Cauchy sequence with limit $\vec{0}$, and such that each F_i has at least one of its factors corresponding to a coordinate $i \in [n]$ such that $v_i^{(k)}$ is a Cauchy sequence in $\mathbb{F}$ approaching 0 or 1. For each such coordinate, replace $v_i^{(k)}$ with 0 (respectively, 1) for all k. Then each $F_j(\vec{v}^{(k)})$ is identically zero as a function of k. Thus, any coordinates of $\vec{v}^{(k)}$ that were not just set are irrelevant, so we may set them to 0 or 1 arbitrarily. The result is that $\vec{v}^{(k)} = \vec{\mu} \in \{0, 1\}^n$ is constant, and we have that $\vec{F}(\vec{\mu}) = \vec{0}$.

(1) ($\mathbb{F}$ any algebraically closed field.) Here we cannot use an argument based on the Euclidean topology, but there is a suitable purely algebraic analogue, encapsulated in the following lemma:

Lemma (See, e.g., Bürgisser–Clausen–Shokrollahi [24, Lemma 20.28]). *If p is a point in the Zariski closure of the image of a polynomial map $F : \mathbb{F}^n \to \mathbb{F}^m$, then there are formal Laurent series*[9] *$\chi_1(\varepsilon), \ldots, \chi_n(\varepsilon)$ in a new variable ε such that $F_i(\chi_1(\varepsilon), \ldots, \chi_n(\varepsilon))$ is in fact a* power series*—that is, involves no negative powers of ε—for each $i = 1, \ldots, m$, and such that evaluating the power series $(F_1(\vec{\chi}(\varepsilon)), \ldots, F_m(\vec{\chi}(\varepsilon)))$ at $\varepsilon = 0$ yields the point p.*

Note that the evaluation at $\varepsilon = 0$ must occur *after* applying F_i, since each individual χ_i may involve negative powers of ε.

For a Laurent series χ, let $\deg \chi(\varepsilon)$ denote the *lowest* degree of ε that appears in χ with nonzero coefficient. Note that in a product of several Laurent series, the product of the lowest-degree terms is the lowest-degree term of the product, as this term cannot be cancelled by any other term. So for any Laurent series $\chi_1, \ldots, \chi_k$, we have that $\deg \prod_{i\in[k]} \chi_i = \sum_{i\in[k]} \deg \chi_i$. By a natural convention that is consistent with the preceding facts, we define $\deg 0 = \infty$.

[9] A formal Laurent series is a formal sum of the form $\sum_{k=-k_0}^{\infty} a_k \varepsilon^k$. By "formal" we mean that we are paying no attention to issues of convergence (which need not even make sense over various fields), but are just using the degree of ε as an indexing scheme.

Now, assume that $F_1(\vec{x}) = x_1 \cdots x_k(1 - x_{k+1}) \cdots (1 - x_\ell)$. The fact that $F_1(\vec{\chi}(\varepsilon))|_{\varepsilon=0} = 0$ means that $F_1(\vec{\chi}(\varepsilon))$ is a power series in ε whose constant term is 0 or, equivalently, that $\deg F_1(\vec{\chi}(\varepsilon)) > 0$. But this is equivalent to

$$\sum_{i=1}^{k} \deg \chi_i + \sum_{i=k+1}^{\ell} \deg(1 - \chi_i) > 0. \tag{9}$$

Thus at least one of the Laurent series $\chi_1(\varepsilon), \chi_2(\varepsilon), \ldots, \chi_k(\varepsilon), 1 - \chi_{k+1}(\varepsilon), \ldots, 1 - \chi_\ell(\varepsilon)$ is in fact a power series with zero constant term, that is, has strictly positive degree. For each χ_i $(1 \le i \le k)$ with strictly positive degree, set $\mu_i = 0$, and for each χ_j $(k + 1 \le k \le \ell)$ such that $1 - \chi_j$ has strictly positive degree, set $\mu_j = 1$. If we replace those χ_i with μ_i and χ_j with μ_j (perhaps leaving some of the χ's untouched), then it has the effect of replacing some of the summands in Equation (9) by ∞, maintaining the truth of Equation (9). In fact, once at least one of the χ_i appearing in Equation (9) is zero (equivalently, has infinite degree), the rest of the summands are irrelevant to the truth of Equation (9). (This corresponds to the fact that it only takes one literal to satisfy a clause in CNF.)

All that remains to check is that when we make these assignments across all the F_i we do not run into a contradiction. For this, note that if $\deg \chi_i > 0$, then $\deg(1 - \chi_i) = 0$, since $1 - \chi_i$ has constant term 1; similarly, if $\deg(1 - \chi_i) > 0$, then $\deg \chi_i = 0$. Thus, these two possibilities are mutually exclusive, so we arrive at a consistent setting of the μ_i. As argued above, any index i for which μ_i has not been set is irrelevant, so we may set them arbitrarily. Finally, we arrive at $\vec{F}(\vec{\mu}) = \vec{0}$. □

Finally, as with IPS certificates and Hilbert-like IPS certificates (see Section 6), a *geometric zero-certificate* for a system of equations $F_1(\vec{x}), \ldots, F_m(\vec{x})$ is a polynomial $C(y_1, \ldots, y_m) \in \langle y_1, \ldots, y_m \rangle$—that is, such that $C(0, \ldots, 0) = 0$—and such that $C(F_1(\vec{x}), \ldots, F_m(\vec{x})) = 0$ identically as a polynomial in $\vec{x}$. The same arguments as in the case of algebraic certificates show that any two geometric certificates differ by a geometric zero-certificate, and that the geometric certificates are closed under multiplication. Furthermore, the set of geometric zero-certificates is the intersection of the ideal of (algebraic) zero-certificates $\langle y_1, \ldots, y_m \rangle \cap \langle y_1 - F_1(\vec{x}), \ldots, y_m - F_m(\vec{x}) \rangle$ with the subring $\mathbb{F}[\vec{y}] \subset \mathbb{F}[\vec{x}, \vec{y}]$. As such, it is an ideal of $\mathbb{F}[\vec{y}]$ and so is finitely generated. Thus, as in the case of IPS certificates, the set of all geometric certificates can be specified by giving a single geometric certificate and a finite generating set for the ideal of geometric zero-certificates, suggesting an approach to lower bounds on the Geometric Ideal Proof System.

We note that geometric zero-certificates are also called syzygies amongst the F_i—sometimes "geometric syzygies" or "polynomial syzygies" to distinguish them from the "module-type syzygies" or "linear syzygyies" we discussed above in relation to Hilbert-like IPS. As in all the other cases we have discussed, a generating set of the geometric syzygies can be computed using Gröbner bases, this time using elimination theory: compute a Gröbner basis for the ideal $\langle y_1 - F_1(\vec{x}), \ldots, y_m - F_m(\vec{x}) \rangle$ using an order that eliminates the x-variables, and then take the subset of the Gröbner basis that consists of polynomials only involving the y-variables. The ideal of geometric syzygies is exactly the ideal of the closure of the image of the map F, and for this reason this kind of syzygy is also well-studied. This suggests that geometric properties of the image of the map F (or its closure) may be useful in understanding the complexity of individual instances of coNP-complete problems.

ACKNOWLEDGMENTS

We thank David Liu for many interesting discussions and for collaborating with us on some of the open questions posed in this article. We thank Eric Allender and Andy Drucker for asking whether "Extended Frege-provable PIT" implied that IPS was equivalent to Extended Frege, which

led to the results of Section 5.2. We thank Pascal Koiran for providing the second half of the proof of Proposition 3.2. We thank Iddo Tzameret for useful discussions that led to the second half of Proposition 3.4. We thank Pavel Hrubeš, Iddo Tzameret, and anonymous reviewers for useful feedback. Finally, in addition to several useful discussions, we also thank Eric Allender for suggesting the name "Ideal Proof System"—all of our other potential names did not even hold a candle to this one.

REFERENCES

[1] Scott Aaronson and Avi Wigderson. 2009. Algebrization: A new barrier in complexity theory. *ACM Trans. Comput. Theory* 1, 1, Article 2 (Feb. 2009), 2:1–2:54. DOI: http://dx.doi.org/10.1145/1490270.1490272

[2] Manindra Agrawal and V. Vinay. 2008. Arithmetic circuits: A chasm at depth four. In *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS'08)*. IEEE Computer Society, 67–75. DOI: http://dx.doi.org/10.1109/FOCS.2008.32

[3] Miklós Ajtai. 1994. The complexity of the pigeonhole principle. *Combinatorica* 14, 4 (1994), 417–433. DOI: http://dx.doi.org/10.1007/BF01302964.

[4] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson. 2002. Space complexity in propositional calculus. *SIAM J. Comput.* 31, 4 (2002), 1184–1211. DOI: http://dx.doi.org/10.1137/S0097539700366735

[5] Sanjeev Arora and Boaz Barak. 2009. *Computational Complexity: A Modern Approach*. Cambridge University Press, Cambridge.

[6] M. F. Atiyah and I. G. Macdonald. 1969. *Introduction to Commutative Algebra*. Addison-Wesley Publishing Co., Reading, MA.

[7] László Babai, Lance Fortnow, Noam Nisan, and Avi Wigderson. 1993. BPP has subexponential time simulations unless EXPTIME has publishable proofs. *Comput. Complex.* 3, 4 (1993), 307–318. DOI: http://dx.doi.org/10.1007/BF01275486

[8] Ted Baker, John Gill, and Robert Solovay. 1975. Relativizations of the P =? NP question. *SIAM J. Comput.* 4 (1975), 431–442.

[9] Walter Baur and Volker Strassen. 1983. The complexity of partial derivatives. *Theoret. Comput. Sci.* 22, 3 (1983), 317–330. DOI: http://dx.doi.org/10.1016/0304-3975(83)90110-X

[10] Dave Bayer and David Mumford. 1993. What can be computed in algebraic geometry? In *Computational Algebraic Geometry and Commutative Algebra (Cortona, 1991)*. Cambridge University Press, Cambridge, 1–48.

[11] David Bayer and Michael Stillman. 1987. A criterion for detecting *m*-regularity. *Invent. Math.* 87, 1 (1987), 1–11. DOI: http://dx.doi.org/10.1007/BF01389151

[12] David Bayer and Michael Stillman. 1987. A theorem on refining division orders by the reverse lexicographic order. *Duke Math. J.* 55, 2 (1987), 321–328. DOI: http://dx.doi.org/10.1215/S0012-7094-87-05517-7

[13] David Bayer and Michael Stillman. 1988. On the complexity of computing syzygies. *J. Symbol. Comput.* 6, 2–3 (1988), 135–147. DOI: http://dx.doi.org/10.1016/S0747-7171(88)80039-7

[14] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. 1996. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. Lond. Math. Soc.* 73, 1 (1996), 1–26. DOI: http://dx.doi.org/10.1112/plms/s3-73.1.1.

[15] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, Pavel Pudlák, and Alan Woods. 1992. Exponential lower bounds for the pigeonhole principle. In *Proceedings of the 24th Annual ACM Symposium on Theory of Computing (STOC'92)*. ACM, New York, NY, 200–220. DOI: http://dx.doi.org/10.1145/129712.129733

[16] Michael Ben-Or and Prasoon Tiwari. 1988. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC'88)*. ACM, New York, NY, 301–309. DOI: http://dx.doi.org/10.1145/62212.62241

[17] Maria Bonet, Toniann Pitassi, and Ran Raz. 1997. Lower bounds for cutting planes proofs with small coefficients. *J. Symbolic Logic* 62, 3 (1997), 708–728. DOI: http://dx.doi.org/10.2307/2275569.

[18] Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi. 2004. Non-automatizability of bounded-depth Frege proofs. *Comput. Complex.* 13, 1–2 (2004), 47–68. DOI: http://dx.doi.org/10.1007/s00037-004-0183-5.

[19] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz. 2000. On interpolation and automatization for Frege systems. *SIAM J. Comput.* 29, 6 (2000), 1939–1967. DOI: http://dx.doi.org/10.1137/S0097539798353230.

[20] W. Dale Brownawell. 1987. Bounds for the degrees in the Nullstellensatz. *Ann. Math.* 126, 3 (1987), 577–591. DOI: http://dx.doi.org/10.2307/1971361

[21] W. Dale Brownawell. 1998. A pure power product version of the Hilbert Nullstellensatz. *Mich. Math. J.* 45, 3 (1998), 581–597. DOI: http://dx.doi.org/10.1307/mmj/1030132301

[22] Peter Bürgisser. 2000. *Completeness and Reduction in Algebraic Complexity Theory.* Algorithms and Computation in Mathematics, Vol. 7. Springer-Verlag, Berlin.

[23] Peter Bürgisser. 2000. Cook's versus Valiant's hypothesis. *Theoret. Comput. Sci.* 235, 1 (2000), 71–88. DOI: http://dx.doi.org/10.1016/S0304-3975(99)00183-8 Selected papers in honor of Manuel Blum (Hong Kong, 1998).

[24] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. 1997. *Algebraic Complexity Theory.* Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 315. Springer-Verlag, Berlin.

[25] Marco L. Carmosino, Russell Impagliazzo, Valentine Kabanets, and Antonina Kolokolova. 2015. Tighter connections between derandomization and circuit lower bounds. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM'15)*, Leibniz International Proceedings in Informatics (LIPIcs), Vol. 40. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 645–658. DOI: http://dx.doi.org/10.4230/LIPIcs.APPROX-RANDOM.2015.645

[26] Xi Chen, Neeraj Kayal, and Avi Wigderson. 2010. *Partial Derivatives in Arithmetic Complexity and Beyond.* Foundations and Trends in Theoretical Computer Science, Vol. 6. Now Publishers. DOI: http://dx.doi.org/10.1561/0400000043

[27] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. 1996. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC'96).* ACM, New York, NY, 174–183. DOI: http://dx.doi.org/10.1145/237814.237860

[28] Stephen A. Cook and Robert A. Reckhow. 1979. The relative efficiency of propositional proof systems. *J. Symbol. Logic* 44, 1 (1979), 36–50. DOI: http://dx.doi.org/10.2307/2273702

[29] Richard A. DeMillo and Richard J. Lipton. 1978. A probabilistic remark on algebraic program testing. *Inform. Process. Lett.* 7, 4 (1978), 193–195. DOI: http://dx.doi.org/10.1016/0020-0190(78)90067-4

[30] Lawrence Ein and Robert Lazarsfeld. 1999. A geometric effective Nullstellensatz. *Invent. Math.* 137, 2 (1999), 427–448. DOI: http://dx.doi.org/10.1007/s002220050332

[31] David Eisenbud. 1995. *Commutative Algebra.* Graduate Texts in Mathematics, Vol. 150. Springer-Verlag, New York. NY. DOI: http://dx.doi.org/10.1007/978-1-4612-5350-1

[32] Viviana Ene and Jürgen Herzog. 2012. *Gröbner Bases in Commutative Algebra.* Graduate Studies in Mathematics, Vol. 130. American Mathematical Society, Providence, RI.

[33] Joan Feigenbaum and Lance Fortnow. 1993. Random-self-reducibility of complete sets. *SIAM J. Comput.* 22, 5 (1993), 994–1005. DOI: http://dx.doi.org/10.1137/0222061

[34] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. 2016. Proof complexity lower bounds from algebraic circuit complexity. In *Proceedings of the 31st Conference on Computational Complexity (CCC'16),* Ran Raz (Ed.), Leibniz International Proceedings in Informatics (LIPIcs), Vol. 50. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 32:1–32:17. DOI: http://dx.doi.org/10.4230/LIPIcs.CCC.2016.32

[35] Dima Grigoriev and Edward A. Hirsch. 2003. Algebraic proof systems over formulas. *Theoret. Comput. Sci.* 303, 1 (2003), 83–102. DOI: http://dx.doi.org/10.1016/S0304-3975(02)00446-2 Logic and complexity in computer science (Créteil, 2001).

[36] Joshua A. Grochow. 2015. Unifying known lower bounds via geometric complexity theory. *Comput. Complex.* 24, 2 (2015), 393–475. Issue 2. DOI: http://dx.doi.org/10.1007/s00037-015-0103-x Special issue from IEEE CCC 2014. Open access.

[37] Joshua A. Grochow and Toniann Pitassi. 2014. Circuit complexity, proof complexity, and polynomial identity testing. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS'14).* 110–119. DOI: http://dx.doi.org/10.1109/FOCS.2014.20. Also available as arXiv:1404.3820 [cs.CC] and ECCC Technical Report TR14-052.

[38] Joshua A. Grochow and Toniann Pitassi. 2017. Tighter depth-preserving simulation of $AC^0[p]$-Frege by the Ideal Proof System. (unpublished).

[39] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. 2013. Arithmetic circuits: A chasm at depth three. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS'13).* 578–587. DOI: http://dx.doi.org//10.1109/FOCS.2013.68

[40] Armin Haken. 1985. The intractability of resolution. *Theoret. Comput. Sci.* 39, 2–3 (1985), 297–308. DOI: http://dx.doi.org/10.1016/0304-3975(85)90144-6

[41] Juris Hartmanis and Richard E. Stearns. 1965. On the computational complexity of algorithms. *Trans. Am. Math. Soc.* 117 (1965), 285–306.

[42] Joos Heintz and C.-P. Schnorr. 1982. Testing polynomials which are easy to compute. In *Logic and Algorithmic (Zurich, 1980).* Monograph. Enseign. Math., Vol. 30. Univ. Genève, Geneva, 237–254.

[43] Grete Hermann. 1926. Die Frage der endlich vielen Schritte in der Theorie der Polynomideale. *Math. Ann.* 95, 1 (1926), 736–788. DOI: http://dx.doi.org/10.1007/BF01206635

[44] David Hilbert. 1978. *Hilbert's Invariant Theory Papers.* Math Sci Press, Brookline, MA.

[45] Pavel Hrubeš. 2016. Arithmetic circuits and proof complexity. (2016). Talks given at the Workshop on Algebraic Complexity Theory, Tel Aviv, Israel, 2016.

[46] Pavel Hrubeš and Iddo Tzameret. 2009. The proof complexity of polynomial identities. In *Proceedings of the 24th IEEE Conference on Computational Complexity (CCC'09)*. IEEE Computer Soc., Los Alamitos, CA, 41–51. DOI: http://dx.doi.org/10.1109/CCC.2009.9

[47] Pavel Hrubeš and Iddo Tzameret. 2012. Short proofs for the determinant identities. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC'12)*. ACM, New York, NY, 193–212. DOI: http://dx.doi.org/10.1145/2213977.2213998

[48] Maurice Jansen and Rahul Santhanam. 2012. Stronger lower bounds and randomness-hardness trade-offs using associated algebraic complexity classes. In *Proceedings of the 29th Annual Symposium on Theoretical Aspects of Computer Science (STACS'12)*. Leibniz Int. Proc. Inform. (LIPIcs), Vol. 14. Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 519–530.

[49] Zbigniew Jelonek. 2005. On the effective Nullstellensatz. *Invent. Math.* 162, 1 (2005), 1–17. DOI: http://dx.doi.org/10.1007/s00222-004-0434-8

[50] Valentine Kabanets and Russell Impagliazzo. 2004. Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput. Complex.* 13, 1–2 (2004), 1–46. DOI: http://dx.doi.org/10.1007/s00037-004-0182-6

[51] Erich Kaltofen and Lakshman Yagati. 1989. Improved sparse multivariate polynomial interpolation algorithms. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (ISSAC'88)*. Lecture Notes in Computer Science, Vol. 358. Springer, Berlin, 467–474. DOI: http://dx.doi.org/10.1007/3-540-51084-2_44

[52] Neeraj Kayal, Nutan Limaye, Chandan Saha, and Srikanth Srinivasan. 2014. An exponential lower bound for homogeneous depth four arithmetic formulas. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS'14)*. 61–70. DOI: http://dx.doi.org/10.1109/FOCS.2014.15

[53] Neeraj Kayal and Chandan Saha. 2014. Lower Bounds for Depth Three Arithmetic Circuits with Small Bottom Fanin. *ECCC Tech. Report TR14-089*. (2014).

[54] Pascal Koiran. 1996. Hilbert's Nullstellensatz is in the polynomial hierarchy. *J. Complex.* 12, 4 (1996), 273–286. DOI: http://dx.doi.org/10.1006/jcom.1996.0019

[55] Pascal Koiran. 2012. Arithmetic circuits: The chasm at depth four gets wider. *Theoret. Comput. Sci.* 448 (2012), 56–65. DOI: http://dx.doi.org/10.1016/j.tcs.2012.03.041

[56] János Kollár. 1988. Sharp effective Nullstellensatz. *J. Am. Math. Soc.* 1, 4 (1988), 963–975. DOI: http://dx.doi.org/10.2307/1990996

[57] János Kollár. 1999. Effective Nullstellensatz for arbitrary ideals. *J. Eur. Math. Soc.* 1, 3 (1999), 313–337. DOI: http://dx.doi.org/10.1007/s100970050009

[58] Jan Krajíček. 1994. Lower bounds to the size of constant-depth propositional proofs. *J. Symbol. Logic* 59, 1 (1994), 73–86. DOI: http://dx.doi.org/10.2307/2275250

[59] Jan Krajíček. 1995. *Bounded Arithmetic, Propositional Logic, and Complexity Theory*. Encyclopedia of Mathematics and Its Applications, Vol. 60. Cambridge University Press, Cambridge. DOI: http://dx.doi.org/10.1017/CBO9780511529948

[60] Jan Krajíček, Pavel Pudlák, and Alan Woods. 1995. An exponential lower bound to the size of bounded depth Frege proofs of the pigeonhole principle. *Random Struct. Algor.* 7, 1 (1995), 15–39. DOI: http://dx.doi.org/10.1002/rsa.3240070103

[61] Teresa Krick, Luis Miguel Pardo, and Martín Sombra. 2001. Sharp estimates for the arithmetic Nullstellensatz. *Duke Math. J.* 109, 3 (2001), 521–598. DOI: http://dx.doi.org/10.1215/S0012-7094-01-10934-4

[62] Mrinal Kumar and Ramprasad Saptharishi. 2015. An exponential lower bound for homogeneous depth-5 circuits over finite fields. In *Proceedings of the 32nd IEEE Conference on Computational Complexity (CCC'17)*. 31:1–31:30. DOI: http://dx.doi.org/10.4230/LIPIcs.CCC.2017.31 Preprint available as ECCC Tech. Report TR15-109.

[63] Mrinal Kumar and Shubhangi Saraf. 2017. On the power of homogeneous depth 4 arithmetic circuits. *SIAM J. Comput.* 46, 1 (2017), 336–387. DOI: http://dx.doi.org/10.1137/140999335

[64] J. M. Landsberg. 2014. Geometric complexity theory: An introduction for geometers. *Annali Dell'universitá di Ferrara* 61, 1 (2014), 1–53. DOI: http://dx.doi.org/10.1007/s11565-014-0202-7

[65] Fu Li, Iddo Tzameret, and Zhengyu Wang. 2015. Non-commutative formulas and Frege lower bounds: A new characterization of propositional proofs. In *Proceedings of the 30th Conference on Computational Complexity (CCC'15)*, David Zuckerman (Ed.), Leibniz International Proceedings in Informatics (LIPIcs), Vol. 33. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 412–432. DOI: http://dx.doi.org/10.4230/LIPIcs.CCC.2015.412

[66] Alexis Maciel and Toniann Pitassi. 1998. Towards lower bounds for bounded-depth Frege proofs with modular connectives. In *Proof Complexity and Feasible Arithmetics*, Paul Beame and Sam Buss (Eds.). DIMACS Series in Discrete Mathematics and Theoretical Compute Science, Vol. 39. American Mathematical Society, 195–227.

[67] Guillaume Malod and Natacha Portier. 2008. Characterizing Valiant's algebraic complexity classes. *J. Complex.* 24, 1 (2008), 16–38. DOI: http://dx.doi.org/10.1016/j.jco.2006.09.006

[68] D. W. Masser and G. Wüstholz. 1983. Fields of large transcendence degree generated by values of elliptic functions. *Invent. Math.* 72, 3 (1983), 407–464. DOI: http://dx.doi.org/10.1007/BF01398396

[69] Hideyuki Matsumura. 1980. *Commutative Algebra* (2nd ed.). Mathematics Lecture Note Series, Vol. 56. Benjamin/Cummings Publishing Co., Inc., Reading, MA.

[70] Ernst Mayr. 1989. Membership in polynomial ideals over $\mathbb{Q}$ is exponential space complete. In *Proceedings of the 6th Annual Symposium on Theoretical Aspects of Computer Science (STACS'89)*. Lecture Notes in Computer Science, Vol. 349. Springer, Berlin, 400–406. DOI: http://dx.doi.org/10.1007/BFb0029002

[71] Ernst W. Mayr and Albert R. Meyer. 1982. The complexity of the word problems for commutative semigroups and polynomial ideals. *Adv. Math.* 46, 3 (1982), 305–329. DOI: http://dx.doi.org/10.1016/0001-8708(82)90048-2

[72] Mladen Miksa and Jakob Nordström. 2015. A generalized method for proving polynomial calculus degree lower bounds. In *Proceedings of the 30th IEEE Conference on Computational Complexity (CCC'15)*. 467–487. DOI: http://dx.doi.org/10.4230/LIPIcs.CCC.2015.467 Preprint available as ECCC TR15-078.

[73] Fabien Morel and Vladimir Voevodsky. 1999. $\mathbb{A}^1$-homotopy theory of schemes. *Inst. Hautes Études Sci. Publ. Math.* 90 (1999), 45–143 (2001). http://www.numdam.org/item?id=PMIHES_1999__90__45_0

[74] Ketan D. Mulmuley. 1999. Lower bounds in a parallel model without bit operations. *SIAM J. Comput.* 28, 4 (1999), 1460–1509 (electronic). DOI: http://dx.doi.org/10.1137/S0097539794282930

[75] Ketan D. Mulmuley. 2011. *Geometric Complexity Theory VI: The Flip via Positivity*. Technical Report. Department of Computer Science, The University of Chicago. http://gct.cs.uchicago.edu/gct6.pdf.

[76] Ketan D. Mulmuley. 2012. The GCT program toward the P vs. NP problem. *Commun. ACM* 55, 6 (Jun. 2012), 98–107. DOI: http://dx.doi.org/10.1145/2184319.2184341

[77] Ketan D. Mulmuley and Milind Sohoni. 2001. Geometric complexity theory I: An approach to the P vs. NP and related problems. *SIAM J. Comput.* 31, 2 (2001), 496–526. DOI: http://dx.doi.org/10.1137/S009753970038715X

[78] Ketan D. Mulmuley and Milind Sohoni. 2008. Geometric complexity theory. II. Towards explicit obstructions for embeddings among class varieties. *SIAM J. Comput.* 38, 3 (2008), 1175–1206. DOI: http://dx.doi.org/10.1137/080718115

[79] David Mumford. 1976. *Algebraic Geometry I. Complex Projective Varieties*. Number 221 in Grundlehren der Mathematischen Wissenschaften. Springer-Verlag, Berlin.

[80] Noam Nisan. 1991. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC'91)*. ACM, 410–418. DOI: http://dx.doi.org/10.1145/103418.103462

[81] Toniann Pitassi. 1996. Algebraic propositional proof systems. In *Proceedings of the Descriptive Complexity and Finite Models Workshop (DIMACS'96)*. Neil Immerman and Phokion G. Kolaitis (Eds.). DIMACS Series in Discrete Mathematics and Theoretical Computer Science, Vol. 31. American Mathematical Society, 215–244.

[82] Toniann Pitassi. 1998. Unsolvable systems of equations and proof complexity. In *Proceedings of the International Congress of Mathematicians*, Vol. III, Berlin, 451–460. https://www.emis.de/journals/DMJDMV/xvol-icm/14/Pitassi.MAN.html.

[83] Toniann Pitassi, Paul Beame, and Russell Impagliazzo. 1993. Exponential lower bounds for the pigeonhole principle. *Comput. Complex.* 3, 2 (1993), 97–140. DOI: http://dx.doi.org/10.1007/BF01200117

[84] Ran Raz and Amir Shpilka. 2005. Deterministic polynomial identity testing in non-commutative models. *Comput. Complex.* 14, 1 (2005), 1–19. DOI: http://dx.doi.org/10.1007/s00037-005-0188-8

[85] Ran Raz and Iddo Tzameret. 2008. The strength of multilinear proofs. *Comput. Complex.* 17, 3 (2008), 407–457. DOI: http://dx.doi.org/10.1007/s00037-008-0246-0.

[86] Alexander A. Razborov. 1987. Lower bounds on the dimension of schemes of bounded depth in a complete basis containing the logical addition function. *Mat. Zametki* 41, 4 (1987), 598–607, 623.

[87] Alexander A. Razborov and Steven Rudich. 1997. Natural proofs. *J. Comput. Syst. Sci.* 55, 1, part 1 (1997), 24–35. DOI: http://dx.doi.org/10.1006/jcss.1997.1494

[88] Miles Reid. 1995. *Undergraduate Commutative Algebra*. London Mathematical Society Student Texts, Vol. 29. Cambridge University Press, Cambridge.

[89] J. T. Schwartz. 1980. Fast probabilistic algorithms for verification of polynomial identities. *J. Assoc. Comput. Mach.* 27, 4 (1980), 701–717. DOI: http://dx.doi.org/10.1145/322217.322225

[90] Nathan Segerlind. 2007. The complexity of propositional proofs. *Bull. Symbol. Logic* 13, 4 (2007), 417–481. DOI: http://dx.doi.org/10.2178/bsl/1203350879

[91] Abraham Seidenberg. 1974. Constructions in algebra. *Trans. Am. Math. Soc.* 197 (1974), 273–313. DOI: http://dx.doi.org/10.1090/S0002-9947-1974-0349648-2

[92] Amir Shpilka and Amir Yehudayoff. 2009. Arithmetic circuits: A survey of recent results and open questions. *Found. Trends Theor. Comput. Sci.* 5, 3–4 (2009), 207–388 (2010). DOI: http://dx.doi.org/10.1561/0400000039

[93] Michael Sipser. 2005. *Introduction to the Theory of Computation* (second ed.). Course Technology. 431 pages.

[94] Roman Smolensky. 1987. Algebraic methods in the theory of lower bounds for Boolean circuit complexity. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing (STOC'87)*. ACM, 77–82. DOI: http://dx.doi.org/10.1145/28395.28404

[95] Michael Soltys and Stephen Cook. 2004. The proof complexity of linear algebra. *Ann. Pure Appl. Logic* 130, 1–3 (2004), 277–323. DOI: http://dx.doi.org/10.1016/j.apal.2003.10.018

[96] Martín Sombra. 1999. A sparse effective Nullstellensatz. *Adv. Appl. Math.* 22, 2 (1999), 271–295. DOI: http://dx.doi.org/10.1006/aama.1998.0633

[97] Volker Strassen. 1972/73. Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten. *Numer. Math.* 20, 3 (1972/73), 238–251. DOI: http://dx.doi.org/10.1007/BF01436566

[98] Volker Strassen. 1973. Vermeidung von Divisionen. *J. Reine Angew. Math.* 264 (1973), 184–202. http://eudml.org/doc/151394.

[99] Sébastien Tavenas. 2013. Improved bounds for reduction to depth 4 and depth 3. In *Proceedings of the Symposium on Mathematical Foundations of Computer Science (MFCS'13)*, Krishnendu Chatterjee and Jirí Sgall (Eds.). Lecture Notes in Computer Science, Vol. 8087. Springer, Berlin, 813–824. DOI: http://dx.doi.org/10.1007/978-3-642-40313-2_71

[100] Seinosuke Toda. 1991. PP is as hard as the polynomial-time hierarchy. *SIAM J. Comput* 20, 2 (1991), 865–877. DOI: http://dx.doi.org/10.1137/0220053

[101] Seinosuke Toda. 1992. Classes of arithmetic circuits capturing the complexity of computing the determinant. *IEICE Trans. Inf. Syst.* E75D, 1 (Jan. 1992), 116–124.

[102] Leslie G. Valiant. 1979. Completeness classes in algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC'79)*. ACM, 249–261. DOI: http://dx.doi.org/10.1145/800135.804419

[103] Leslie G. Valiant. 1979. The complexity of computing the permanent. *Theoret. Comput. Sci.* 8, 2 (1979), 189–201. DOI: http://dx.doi.org/10.1016/0304-3975(79)90044-6

[104] Leslie G. Valiant. 1982. Reducibility by algebraic projections. *Enseign. Math.* 28, 3–4 (1982), 253–268.

[105] Leslie G. Valiant, S. Skyum, S. Berkowitz, and Charles Rackoff. 1983. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.* 12, 4 (1983), 641–644. DOI: http://dx.doi.org/10.1137/0212043

[106] Leslie G. Valiant and Vijay V. Vazirani. 1986. NP is as easy as detecting unique solutions. *Theoret. Comput. Sci.* 47, 1 (1986), 85–93. DOI: http://dx.doi.org/10.1016/0304-3975(86)90135-0

[107] Vladimir Voevodsky. 1998. A1-homotopy theory. In *Proceedings of the International Congress of Mathematicians*, Vol. I. 579–604. https://www.emis.de/journals/DMJDMV/xvol-icm/00/Voevodsky.MAN.html.

[108] Joachim von zur Gathen. 1987. Feasible arithmetic computations: Valiant's hypothesis. *J. Symbol. Comput.* 4, 2 (1987), 137–172. DOI: http://dx.doi.org/10.1016/S0747-7171(87)80063-9

[109] Richard Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation*, Edward W. Ng (Ed.). Lecture Notes in Computer Science, Vol. 72. Springer, Berlin, 216–226. DOI: http://dx.doi.org/10.1007/3-540-09519-5_73

[110] Richard Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation (EUROSAM'79)*. Lecture Notes in Computer Science, Vol. 72. Springer, Berlin, 216–226. DOI: https://dx.doi.org/10.1007/3-540-09519-5_73

Received January 2017; revised February 2018; accepted June 2018