

Circuit Complexity, Proof Complexity and Polynomial Identity Testing

Joshua A. Grochow and Toniann Pitassi

April 16, 2014

Abstract

We introduce a new and very natural algebraic proof system, which has tight connections to (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 ($\text{VNP} \neq \text{VP}$). As a corollary to the proof, we also show that super-polynomial lower bounds on the number of lines in Polynomial Calculus proofs (as opposed to the usual measure of number of monomials) imply the Permanent versus Determinant Conjecture. Note that, prior to our work, there was no proof system for which lower bounds on an arbitrary tautology implied *any* computational lower bound.

Our proof system helps clarify the relationships between previous algebraic proof systems, and begins to shed light on why proof complexity lower bounds for various proof systems have been so much harder than lower bounds on the corresponding circuit classes. In doing so, we highlight the importance of polynomial identity testing (PIT) for understanding proof complexity.

More specifically, we introduce certain propositional axioms satisfied by any Boolean circuit computing PIT. (The existence of efficient proofs for our PIT axioms appears to be somewhere in between the major conjecture that $\text{PIT} \in \text{P}$ and the known result that $\text{PIT} \in \text{P/poly}$.) We use these PIT axioms to shed light on $\text{AC}^0[p]$ -Frege lower bounds, which have been open for nearly 30 years, with no satisfactory explanation as to their apparent difficulty. We show that either:

- a. Proving super-polynomial lower bounds on $\text{AC}^0[p]$ -Frege implies $\text{VNP}_{\mathbb{F}_p}$ does not have polynomial-size circuits of depth d —a notoriously open question for any $d \geq 4$ —thus explaining the difficulty of lower bounds on $\text{AC}^0[p]$ -Frege, or
- b. $\text{AC}^0[p]$ -Frege cannot efficiently prove the depth d PIT axioms, and hence we have a lower bound on $\text{AC}^0[p]$ -Frege.

We also prove many variants on this statement for other proof systems and other computational lower bounds.

Finally, using the algebraic structure of our proof system, we propose a novel way to extend techniques from algebraic circuit complexity to prove lower bounds in proof complexity. Although we have not yet succeeded in proving such lower bounds, this proposal should be contrasted with the difficulty of extending $\text{AC}^0[p]$ circuit lower bounds to $\text{AC}^0[p]$ -Frege lower bounds.

1 Extended abstract

1.1 Introduction

NP versus coNP is the very natural question of whether, for every graph that doesn't have a Hamiltonian path, there is a short proof of this fact. One of the arguments for the utility of

proof complexity is that by proving lower bounds against stronger and stronger proof systems, we “make progress” towards proving $\text{NP} \neq \text{coNP}$. However, until now this argument has 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 $\text{NP} \neq \text{coNP}$.

We remedy this situation by introducing a very natural algebraic proof system, which has tight connections to (algebraic) circuit complexity. 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 ($\text{VNP} \neq \text{VP}$). 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 $\text{AC}^0[p]$ -Frege, and on number-of-lines in Polynomial Calculus-style proofs—have been so apparently difficult.

1.1.1 Background and Motivation

Algebraic Circuit Complexity. The most natural way to compute a polynomial function $f(x_1, \dots, x_n)$ is with a sequence of instructions $g_1, \dots, g_m = f$, starting from the inputs x_1, \dots, 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 $(f_n(x_1, \dots, x_{\text{poly}(n)}))_{n=1}^\infty$, typically in terms of size and depth. In addition to the intrinsic interest in these questions, since Valiant’s work [Val79a, Val79b, Val82] 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 [vzG87]); several direct connections have been found between algebraic and Boolean complexity [KI04, Bür00b, JS12, Mul99]; and the Geometric Complexity Theory Program (see, e.g., the survey [Mul12] 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. Valiant’s Permanent versus Determinant Conjecture [Val79a] states that the permanent of an $n \times n$ matrix, as a polynomial in n^2 variables, cannot be written 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 $\text{P} \neq \text{NP}$, although it is in fact much closer to $\text{FNC}^2 \neq \#P$. In addition to this analogy, the Permanent versus Determinant Conjecture is also known to be a formal consequence of the nonuniform lower bound $\text{NP} \not\subseteq \text{P/poly}$ [Bür00b], and is thus thought to be an important step towards showing $\text{P} \neq \text{NP}$.

Unlike in Boolean circuit complexity, (slightly) non-trivial lower bounds for the size of algebraic circuits are known [Str73, BS83]. 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 exciting work [AV08, Koi12, Tav13] has shown that polynomial-size algebraic circuits computing functions of polynomial degree can in fact be computed by subexponential-size depth 4 algebraic circuits. Thus, strong enough lower bounds for depth 4 algebraic circuits for the permanent would already prove $\text{VP} \neq \text{VNP}$.

Proof Complexity. Despite considerable progress obtaining super-polynomial lower bounds for many weak proof systems (resolution, cutting planes, bounded-depth Frege systems), 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 $\text{AC}^0[p]$ -Frege system. Note that in contrast, the analogous result in circuit complexity—proving super-polynomial $\text{AC}^0[p]$ lower bounds for an explicit function—was resolved by Smolensky over 25 years ago [Smo87]. 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 [BGS75], Razborov–Rudich natural proofs [RR97], or algebrization [AW08] that exist in Boolean function complexity. Moreover, there has not even been progress by way of conditional lower bounds. That is, trivially $\text{NP} \neq \text{coNP}$ implies superpolynomial lower bounds for $\text{AC}^0[p]$ -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 [BPR00, BDG⁺04]. In the converse direction, there are essentially no implications at all. For example, we do not know if $\text{AC}^0[p]$ -Frege lower bounds—nor even Frege nor Extended Frege lower bounds—imply any nontrivial circuit lower bounds.

1.1.2 Our Results

In this paper, 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 previously studied, but is different in a crucial way that we explain below.

Given a set of polynomials F_1, \dots, F_m in n variables x_1, \dots, x_n over a field \mathbb{F} without a common zero over the algebraic closure of \mathbb{F} , Hilbert’s Nullstellensatz says that there exist polynomials $G_1, \dots, G_m \in \mathbb{F}[x_1, \dots, x_n]$ 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 which 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 $F_1(\vec{x}) = F_2(\vec{x}) = \dots = F_m(\vec{x}) = 0$ is unsatisfiable over $\overline{\mathbb{F}}$ is a polynomial $C(\vec{x}, \vec{y})$ in the variables x_1, \dots, x_n and y_1, \dots, y_m such that

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

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

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

For any class \mathcal{C} of polynomial families, we may speak of \mathcal{C} -IPS proofs of a family of systems of equations (\mathcal{F}_n) where \mathcal{F}_n is $F_{n,1}(\vec{x}) = \dots = F_{n,\text{poly}(n)}(\vec{x}) = 0$. When we refer to IPS without further qualification, we mean VP-IPS, that is, the family of IPS proofs should be computed by circuits of polynomial size *and polynomial degree*, unless specified otherwise.

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

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 $\kappa_1, \dots, \kappa_m$ denote all the clauses of φ , and let F_i be the corresponding polynomials. Then the system of equations we consider is $F_1(\vec{x}) = \dots = F_m(\vec{x}) = x_1^2 - x_1 = \dots = x_n^2 - x_n = 0$. The latter equations force any solution to this 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.

Like previously defined algebraic systems [BIK⁺96, CEI96, Pit96, Pit98], 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 $F_1 = \dots = F_m = 0$. In contrast our system has no such requirement; an IPS proof can compute potentially unsound sub-computations (whose vanishing does not follow from $F_1 = \dots = F_m = 0$), 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 3.1. *Super-polynomial lower bounds for the Ideal Proof System imply that the permanent does not have polynomial-size algebraic circuits, that is, $\text{VNP} \neq \text{VP}$.*

From the proof of this result, together with one of our simulation results (Proposition 2.2), we also get:

Corollary 1.2. *Super-polynomial lower bounds on the number of lines in Polynomial Calculus proofs imply the Permanent versus Determinant Conjecture.¹*

Under a reasonable assumption on polynomial identity testing (PIT), which we discuss further below, we are able to show that Extended Frege is equivalent to the Ideal Proof System. 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 4.1. *Let K be a family of polynomial-size Boolean circuits for PIT such that the PIT axioms for K (see Definition 1.7) have polynomial-size EF proofs. Then EF polynomially simulates IPS, and hence the EF and IPS are polynomially equivalent.*

Under this assumption about PIT, Theorems 3.1 and 4.1 in combination suggest a precise reason that proving lower bounds on Extended Frege is so difficult, namely, that doing so implies $\text{VP} \neq \text{VNP}$. Theorem 4.1 also suggests that to make progress toward proving lower bounds in proof complexity, it may be necessary to prove lower bounds for the Ideal Proof System, which we feel is more natural, and creates the possibility of harnessing tools from algebra, representation theory, and algebraic circuit complexity. We give a specific suggestion of how to apply these tools towards proof complexity lower bounds in Section 1.6.

¹Although Corollary 1.2 may seem to be saying that lower bounds on PC imply a circuit lower bound, this is not precisely the case, because complexity in PC is emphatically not measured by the number of lines, but rather by the total number of monomials appearing in a PC proof. This is true both definitionally and in practice, in that all previous papers on PC use the number-of-monomials complexity measure.

Remark 1.3. Given that $PIT \in P$ is known to imply lower bounds, one may wonder if the combination of the above two theorems really gives any explanation at all for the difficulty of proving lower bounds on Extended Frege. There are at least two reasons that it does.

First, 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 $\text{NEXP} \cap \text{coNEXP}$ [JS12] (via personal communication we have learned that Impagliazzo and Williams have also proved similar results), whereas our conclusion is a lower bound on algebraic circuit-size for an integer polynomial computable in $\#P \subseteq \text{PSPACE}$.

Second, the hypothesis that our PIT axioms can be proven efficiently in Extended Frege seems to be somewhat orthogonal to, and may be no stronger than, the widely-believed hypothesis that PIT is in P . As Extended Frege is a nonuniform proof system, efficient Extended Frege proofs of our PIT axioms are unlikely to have any implications about the uniform complexity of PIT (and given that we already know unconditionally that PIT is in P/poly , uniformity is what the entire question of derandomizing PIT is about). In the opposite direction, it's a well-known observation in proof complexity that nearly all natural uniform polynomial-time algorithms have feasible (Extended Frege) correctness proofs. If this phenomenon doesn't apply to PIT, it would be interesting for both proof complexity and circuit complexity, as it indicates the difficulty of proving that PIT is in P . \triangleleft

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 4.1 can be scaled down to obtain similar results for weaker Frege systems, and discuss some of its more striking consequences.

Theorem 4.5. *Let \mathcal{C} be any of the standard circuit classes $\text{AC}^k, \text{AC}^k[p], \text{ACC}^k, \text{TC}^k, \text{NC}^k$. Let K be a family of polynomial-size Boolean circuits for PIT (not necessarily in \mathcal{C}) such that the PIT axioms for K have polynomial-size \mathcal{C} -Frege proofs. Then \mathcal{C} -Frege is polynomially equivalent to IPS, and consequently to Extended Frege as well.*

Theorem 4.5 also highlights the importance of our PIT axioms for getting $\text{AC}^0[p]$ -Frege lower bounds, which has been an open question for nearly thirty years. (For even weaker systems, Theorem 4.5 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 1.8. *For any d , either:*

- *There are polynomial-size $\text{AC}^0[p]$ -Frege proofs of the depth d PIT axioms, in which case any superpolynomial lower bounds on $\text{AC}^0[p]$ -Frege imply $\text{VNP}_{\mathbb{F}_p}$ does not have polynomial-size depth d algebraic circuits, thus explaining the difficulty of obtaining such lower bounds, or*
- *There are no polynomial-size $\text{AC}^0[p]$ -Frege proofs of the depth d PIT axioms, in which case we've gotten $\text{AC}^0[p]$ -Frege lower bounds.*

Finally, in Section 1.6 we suggest a new framework for proving lower bounds for the Ideal Proof System which we feel has promise. Along the way, we make precise the difference in difficulty between proof complexity lower bounds (on IPS, which may also apply to Extended Frege via Theorem 4.1) and algebraic circuit lower bounds. 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.

1.1.3 Related Work

We will see in Section 1.3.3 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 \mathcal{C} -IPS for certain classes \mathcal{C} . In particular, the Nullstellensatz system [BIK⁺96], the Polynomial Calculus (or Gröbner) proof system [CEI96], and Polynomial Calculus with Resolution [ABSRW02] are all particular measures on IPS, and Pitassi's previous algebraic systems [Pit96, Pit98] are subsystems of IPS.

Raz and Tzameret [RT08] 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. Amongst 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 3.1, that lower bounds on a single system imply algebraic circuit lower bounds.

1.1.4 Outline

The remainder of Section 1 gives proofs of some foundational results, and summarizes the rest of the paper, giving detailed versions of all statements and discussing their proofs and significance. In Section 1 many proofs are only sketched or are delayed until later in the paper, but all proofs of all results are present either in Section 1 or in Sections 2–4.

We start in Section 1.3, by proving several basic facts about IPS (some proofs are deferred to Section 2). 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 [Str73] and the chasms at depth 3 [GKKS13, Tav13] and 4 [AV08, Koi12, Tav13].

In Section 1.4, we outline the proof that lower bounds on IPS imply algebraic circuit lower bounds (Theorem 3.1; full proof in Section 3). We also show how this result gives as a corollary a new, simpler proof that $\text{NP} \not\subseteq \text{coMA} \Rightarrow \text{VNP}^0 \neq \text{VP}^0$. In Section 1.5 we introduce our PIT axioms in detail and outline the proof of Theorems 4.1 and 4.5 (full proofs in Section 4.1). We also discuss in detail many variants of Theorem 4.5 and their consequences, as briefly mentioned above. In Section 1.6 we suggest a new framework for transferring techniques from algebraic circuit complexity to (algebraic) proof complexity lower bounds. Finally, in Section 1.7 we gather a long list of open questions raised by our work, many of which we believe may be quite approachable.

Appendix A contains more complete preliminaries. In Appendices B and C 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.

1.2 A few preliminaries

In this section we cover the bare bones preliminaries that we think may be less familiar to some of our readers. Remaining background material on algebraic complexity, proof complexity, and commutative algebra can be found in Appendix A. As general references, we refer the reader to Bürgisser–Clausen–Shokrollahi [BCS97] and the surveys [SY09, CKW10] for algebraic complexity, to Krajíček [Kra95] for proof complexity, and to any of the standard books [Eis95, AM69, Mat80, Rei95] for commutative algebra.

1.2.1 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 $\text{poly}(n)$ input variables, is of $\text{poly}(n)$ degree, and can be computed by algebraic circuits over R of $\text{poly}(n)$ size. VNP_R is the class of families g of polynomials g_n such that g_n has $\text{poly}(n)$ input variables and is of $\text{poly}(n)$ degree, and can be written as

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

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

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. We note that over a fixed finite field \mathbb{F}_q , $\text{VP}_{\mathbb{F}_q}^0 = \text{VP}_{\mathbb{F}_q}$, since there are only finitely many possible constants. Consequently, $\text{VNP}_{\mathbb{F}_q}^0 = \text{VNP}_{\mathbb{F}_q}$ as well. Over the integers, $\text{VP}_{\mathbb{Z}}^0$ coincides with those families in $\text{VP}_{\mathbb{Z}}$ 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, $\text{VP}_{\overline{\mathbb{F}}_p}^0$ coincides with those families in $\text{VP}_{\overline{\mathbb{F}}_p}$ 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 .)

1.2.2 Proof Complexity

In brief, a *proof system* for a language $L \in \text{coNP}$ is a nondeterministic algorithm for L , or equivalently a deterministic polynomial-time verifier P such that $x \in L \Leftrightarrow (\exists y)[P(x, y) = 1]$, and we refer to any such y as a P -proof that $x \in L$.² We say that P is *polynomially bounded* if for every $x \in L$ there is a P -proof of length polynomially bounded in $|x|$: $|y| \leq \text{poly}(|x|)$. We will generally be considering proof systems for the coNP -complete language TAUT consisting of all propositional tautologies; there is a polynomially bounded proof system for TAUT if and only if $\text{NP} = \text{coNP}$.

Given two proof systems P_1 and P_2 for the same language $L \in \text{coNP}$, we say that P_1 *polynomially simulates* or *p-simulates* P_2 if there is a polynomial-time function f that transforms P_1 -proofs into P_2 -proofs, that is, $P_1(x, y) = 1 \Leftrightarrow P_2(x, f(y)) = 1$. We say that P_1 and P_2 are *polynomially equivalent* or *p-equivalent* if each p-simulates the other. (This is the proof complexity version of Levin reductions between NP problems.)

²This notion is essentially due to Cook and Reckhow [CR79]; although their definition was formalized slightly differently, it is essentially equivalent to the one we give here.

For TAUT (or UNSAT), there are a variety of standard and well-studied proof systems. In this paper we will be primarily concerned with Frege—a standard, school-style line-by-line deductive system—and its variants such as Extended Frege (EF) and AC^0 -Frege. Bounded-depth Frege or AC^0 -Frege are Frege proofs but with the additional restriction that each formula appearing in the proof has bounded depth *syntactically* (the *syntactic* nature of this condition is crucial: since every formula appearing in a proof is a tautology, semantically all such formulas are the constant-true function and can be computed by trivial circuits). As with AC^0 circuits, AC^0 -Frege has rules for handling unbounded fan-in AND and OR connectives, in addition to negations.

For almost any syntactically-defined class of circuits \mathcal{C} , one can similarly define \mathcal{C} -Frege. For example, NC^1 -Frege is p-equivalent to Frege. However, despite the seeming similarities, there are some differences between a circuit class and its corresponding Frege system. Exponential lower bounds are known for AC^0 -Frege [BIK⁺92], which use the Switching Lemma as for lower bounds on AC^0 circuits, but in a more complicated way. However, unlike the case of $\text{AC}^0[p]$ circuits for which we have exponential lower bounds [Raz87, Smo87], essentially no nontrivial lower bounds are known for $\text{AC}^0[p]$ -Frege.

Extended Frege systems generalize Frege systems by allowing, in addition to all of the Frege rules, a new axiom schema of the form $y \leftrightarrow A$, where A can be any formula, and y is a new variable not occurring in A . Whereas polynomial-size Frege proofs allow a polynomial number of lines, each of which must be a polynomial-sized formula, using the new axiom, polynomial-size EF proofs allow a polynomial number of lines, each of which can essentially be a polynomial-sized circuit (you can think of the new variables introduced by this axiom schema as names for the gates of a circuit, in that once a formula is named by a single variable, it can be reused without having to create another copy of the whole formula). In particular, a natural definition of P/poly-Frege is equivalent to Extended Frege. Extended Frege is the strongest natural system known for proving propositional tautologies. One may also consider seemingly much stronger systems such as Peano Arithmetic or ZFC, but it is unclear and unknown if these systems can prove Boolean tautologies (with no quantifiers) any more efficiently than Extended Frege.

We define all of the algebraic systems we consider in Section 1.3.3 below.

1.3 Foundational results

1.3.1 Relation with coMA

Proposition 1.4. *For any field \mathbb{F} , if every propositional tautology has a polynomial-size constant-free IPS _{\mathbb{F}} -proof, then $\text{NP} \subseteq \text{coMA}$, and hence the polynomial hierarchy collapses to its second level.*

If we wish to drop the restriction of “constant-free” (which, recall, is no restriction at all over a finite field), 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 2.4).

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 Schwarz–Zippel–DeMillo–Lipton coRP algorithm for PIT. \square

1.3.2 Chasms, depth reduction, and other circuit transformations

Recently, many strong depth reduction theorems have been proved for circuit complexity [AV08, Koi12, GKKS13, Tav13], which have been called “chasms” since Agrawal and Vinay [AV08]. 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 the type of circuits used in the Polynomial Calculus or other previous algebraic systems [Pit98], and indeed this was part of the motivation to move to our more general notion of IPS proof.

Observation 1.5 (Chasms for IPS proof size). If a system of $n^{O(1)}$ polynomial equations in n variables has an IPS proof of unsatisfiability of size s and (semantic) degree d , then it also has:

1. A $O(\log d(\log s + \log d))$ -depth IPS proof of size $\text{poly}(ds)$ (follows from Valiant–Skyum–Berkowitz–Rackoff [VSB83]);
2. A depth 4 IPS formula proof of size $n^{O(\sqrt{d})}$ (follows from Koiran [Koi12]) or a depth 4 IPS proof of size $2^{O(\sqrt{d \log(ds) \log n})}$ (follows from Tavenas [Tav13]).
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 [GKKS13]) or even $2^{O(\sqrt{d \log n \log s})}$ (follows from Tavenas [Tav13]). \triangleleft

This observation helps explain why size lower bounds for algebraic proofs for the stronger notion of size—namely number of lines, used here and in Pitassi [Pit96], rather than number of monomials—have been difficult to obtain. This also 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 [Str73]). 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 B.

1.3.3 Simulations and 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) [CEI96], Polynomial Calculus with Resolution (PCR) [ABSRW02], the Nullstellensatz proof system [BIK⁺96], and Pitassi’s algebraic systems [Pit96, Pit98], as we explain below.

Before explaining these, we note that although the Nullstellensatz says that if $F_1(\vec{x}) = \dots = F_m(\vec{x}) = 0$ 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. As the linear form is closer to the original way Hilbert expressed the Nullstellensatz (see, e.g., the translation [Hil78]), we refer to certificates of the form $\sum y_i G_i(\vec{x})$ as *Hilbert-like IPS certificates*.

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 $F_1(\vec{x}), \dots, F_m(\vec{x})$. 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, \dots, y_m having linear combination gates and, for each $i = 1, \dots, n$, gates that multiply their input by x_i . (Alternatively, one may view the x_i as inputs, require that the circuit be syntactically *linear* in the y_i , and that each x_i is only an input to multiplication gates, each of which syntactically depends on at least one y_i . Again alternatively, one may view the x_i as inputs, but with the requirement that the polynomial computed *at each gate* is a polynomial of y_i -degree one in the ideal $\langle y_1, \dots, y_m \rangle \subseteq \mathbb{F}[\vec{x}, \vec{y}]$.) In particular, rule-based circuits necessarily produce Hilbert-like certificates.

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, or “Nullstellensatz degree,” is simply 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” 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” 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 1997 algebraic proof system [Pit98] 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.
- Polynomial Calculus with Resolution (PCR) [ABSRW02] also allows variables \bar{x}_i and adds the equations $\bar{x}_i = 1 - x_i$ and $x_i \bar{x}_i = 0$. This is easily accommodated into the Ideal Proof System: add the \bar{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 $\bar{x}_i - 1 + x_i$ and $x_i \bar{x}_i$ 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.

Proposition 2.2. *Pitassi’s 1996 algebraic proof system [Pit96] is p-equivalent to Hilbert-like IPS.*

Pitassi’s 1997 algebraic proof system [Pit98]—equivalent to the number-of-lines measure on PC proofs—is p-equivalent to Hilbert-like det-IPS or VP_{ws} -IPS.

Combining Proposition 2.2 with the techniques used in Theorem 3.1 shows that super-polynomial lower bounds on the number of lines in PC proofs would positively resolve the Permanent Versus Determinant Conjecture, explaining the difficulty of such proof complexity lower bounds.

In light of this proposition (which we prove in Section 2.2), we henceforth refer to the systems from [Pit96] and [Pit98] as Hilbert-like IPS and Hilbert-like det-IPS, respectively. Pitassi [Pit96, Theorem 1] 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 [Pit96] does not seem to generalize to give a depth-preserving simulation. Nonetheless, our next proposition shows that there is indeed a depth-preserving simulation.

Theorem 2.3. *For any $d(n)$, depth- $(d+2)$ $\text{IPS}_{\mathbb{F}_p}$ p-simulates depth- d Frege proofs with unbounded fan-in $\vee, \wedge, \text{MOD}_p$ connectives (for $d = O(1)$, this is $\text{AC}_d^0[p]$ -Frege).*

1.4 Lower bounds on IPS imply circuit lower bounds

Theorem 3.1. *A super-polynomial lower bound on [constant-free] Hilbert-like IPS_R proofs of any family of tautologies implies $\text{VNP}_R \neq \text{VP}_R$ [respectively, $\text{VNP}_R^0 \neq \text{VP}_R^0$], for any ring R .*

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

Together with Proposition 1.4, this immediately gives an alternative, and we believe simpler, proof of the following result:

Corollary 1.6. *If $\text{NP} \not\subseteq \text{coMA}$, then $\text{VNP}_R^0 \neq \text{VP}_R^0$, for any ring R .*

For comparison, here is a brief sketch of the only previous proof of this result that we are aware of, which only seems to work when R is a finite field or, assuming the Generalized Riemann Hypothesis, a field of characteristic zero, and uses several other significant results. The previous proof combines: 1) Bürgisser's results [Bür00b] relating VP and VNP over various fields to standard Boolean complexity classes such as NC/poly, #P/poly (uses GRH), and Mod_pP/poly, and 2) the implication $\text{NP} \not\subseteq \text{coMA} \Rightarrow \text{NC/poly} \neq \#P/\text{poly}$ (and similarly with #P/poly replaced by Mod_pP/poly), which uses the downward self-reducibility of complete functions for #P/poly (the permanent [Val79a]) and Mod_pP/poly [FF93], as well as Valiant–Vazirani [VV86].

The following lemma is the key to Theorem 3.1.

Lemma 3.2. *Every family of CNF tautologies (φ_n) has a Hilbert-like family of IPS certificates (C_n) in VNP_R^0 .*

Here we show how Theorem 3.1 follows from Lemma 3.2. Lemma 3.2 is proved in Section 3.

Proof of Theorem 3.1, assuming Lemma 3.2. For a given set \mathcal{F} of unsatisfiable polynomial equations $F_1 = \dots = F_m = 0$, 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 3.2—cannot be computed by polynomial-size algebraic circuits, and hence that $\text{VNP} \neq \text{VP}$. Since Lemma 3.2 even guarantees a constant-free certificate, we get the analogous consequence for constant-free lower bounds.

The second part of Theorem 3.1 follows from the fact that number of lines in a PC proof is p-equivalent to Hilbert-like det-IPS (Proposition 2.2). 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. \square

1.5 PIT as a bridge between circuit complexity and proof complexity

In this section we state our PIT axioms and give an outline of the proof of Theorems 4.1 and 4.5, which say that Extended Frege (EF) (respectively, AC⁰- or AC⁰[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, AC⁰- or AC⁰[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 4.5 begins to explain why AC⁰[p]-Frege

lower bounds have been so difficult to obtain, and highlights the importance of our PIT axioms for $\text{AC}^0[p]$ -Frege lower bounds. We begin by describing and discussing these axioms.

Fix some standard Boolean encoding of constant-free algebraic circuits, so that the encoding of any size- m constant-free algebraic circuit has size $\text{poly}(m)$. 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, \dots, 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]”, this refers to a fixed Boolean string that is encoding the (fixed) algebraic circuit C . In contrast, if we write [C], 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 $\vec{x}, \vec{y}, \vec{z}, \dots$ 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 , [$C(\vec{x})$] is a string of propositional variables encoding an unspecified algebraic circuit C , and [$C(\vec{p})$] 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 1.7. 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 isn’t particularly interesting or useful.)

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

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

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 [$C(\vec{p})$] 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.

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

Here, if \vec{q} are the propositional variables describing C , 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.

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

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.

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 .

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

We can now state and discuss two of our main theorems precisely.

Theorem 4.1. *If there is a family K of polynomial-size Boolean circuits that correctly compute 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 uniformly random points—and we do not see how to prove the above axioms 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, especially in light of Remark 1.3.

To prove the theorem (which we do in Section 4.1), we 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 show that efficient EF proofs of *Soundness_{IPS}* follows from efficient EF proofs for the PIT axioms.

Our next main result shows that the previous result can be scaled down to much weaker proof systems than EF.

Theorem 4.5. *Let \mathcal{C} be any 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 \mathcal{C} -Frege proofs, then \mathcal{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 family K to be in the class \mathcal{C} . This requires one more (standard) technical device compared to the proof of Theorem 4.1, namely the use of auxiliary variables for the gates of K . Here we prove and discuss some corollaries of Theorem 4.5; the proof of Theorem 4.5 is given in Section 4.2.

As AC^0 is known unconditionally to be strictly weaker than Extended Frege [Ajt94], 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 4.5, 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, for any $d \geq 4$ super-polynomial lower bounds on depth d algebraic circuits are a notoriously open problem. (The chasm at depth 4 says that depth 4 lower bounds of size $2^{\omega(\sqrt{n} \log n)}$ imply super-polynomial size lower bounds on general algebraic circuits, but this does not give any indication of why merely super-polynomial lower bounds on depth 4 circuits should be difficult.)

Corollary 1.8. *For any d , if there is a family of tautologies with no polynomial-size $\text{AC}^0[p]$ -Frege proof, and $\text{AC}^0[p]$ -Frege has polynomial-size proofs of the [depth d] PIT axioms for some K , then $\text{VNP}_{\mathbb{F}_p}$ does not have polynomial-size [depth d] algebraic circuits.*

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

Open Question 1.9. For some $d \geq 4$, is there some K computing depth d PIT, for which the depth d PIT axioms have $\text{AC}^0[p]$ -Frege proofs of polynomial size?

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

- If $\text{AC}^0[p]$ -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 $\text{AC}^0[p]$ -Frege, answering a question that has been open for nearly thirty years.
- Otherwise, super-polynomial size lower bounds on $\text{AC}^0[p]$ -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 $\text{AC}^0[p]$ -Frege has been so difficult.

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

Similarly, along with Theorem 2.3, we get the following corollary.

Corollary 1.10. *If for every constant d , there is a constant d' such that the depth d PIT axioms have polynomial-size depth d' $\text{AC}_{d'}^0[p]$ -Frege proofs , then $\text{AC}^0[p]$ -Frege is polynomially equivalent to constant-depth $\text{IPS}_{\mathbb{F}_p}$.*

Using the chasms at depth 3 and 4 for algebraic circuits [AV08, Koi12, Tav13] (see Observation 1.5 above), we can also help explain why sufficiently strong exponential lower bounds for AC^0 -Frege—that is, lower bounds that don’t depend on the depth, or don’t depend so badly on the depth, which have also been open for nearly thirty years—have been difficult to obtain:

Corollary 1.11. *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 $\text{PIT}_{\mathbb{F}}$ axioms for some K , then $\text{VP}_{\mathbb{F}}^0 \neq \text{VNP}_{\mathbb{F}}^0$.*

If \mathbb{F} has characteristic zero, we may replace “depth 4” above with “depth 3.”

Proof. Suppose that AC^0 -Frege can efficiently prove the depth 4 $\text{PIT}_{\mathbb{F}}$ axioms for some Boolean circuit K . Let (φ_n) be a family of tautologies. If $\text{VNP}_{\mathbb{F}}^0 = \text{VP}_{\mathbb{F}}^0$, then there is a polynomial-size IPS proof of φ_n . By Observation 1.5, 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 $\text{PIT}_{\mathbb{F}}$ 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, 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. \square

As with Corollary 1.8, 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 $\text{VP}^0 \neq \text{VNP}^0$.

1.6 Towards lower bounds

Theorem 3.1 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 more precise, and use this precision to suggest a direction for proving new proof complexity lower bounds, aimed at proving the long-sought-for length-of-proof lower bounds on an algebraic proof system.

The key fact we use is embodied in Lemma 1.12, 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; these are reviewed in Appendix A.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 $(G_1(\vec{x}), \dots, G_m(\vec{x}))$, 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 $(G_1(\vec{x}), \dots, G_m(\vec{x}))$. 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, we may consider Hilbert-like certificates, and 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:

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

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:

$$\left(\begin{array}{ccc} F_1(\vec{x}) & \cdots & F_m(\vec{x}) \end{array} \right) \begin{pmatrix} g_1 \\ \vdots \\ g_m \end{pmatrix} = 0 \tag{1}$$

(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” amongst 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 1.12. *For a given set of unsatisfiable polynomial equations $F_1(\vec{x}) = \dots = F_m(\vec{x}) = 0$ 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 (1). 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. \square

Lemma 1.12 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 giving 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 $\langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$. 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 [EH12] (see also [Eis95, Section 15.5]).

Lemma 1.12 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 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 [Raz87, Smo87] 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

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

which is either empty or a coset of the ideal of zero-certificates: $\langle y_1, \dots, y_m \rangle \cap \langle y_1 - F_1(\vec{x}), \dots, y_m - F_m(\vec{x}) \rangle$. The intersection ideal $\langle y_1, \dots, y_m \rangle \cap \langle y_1 - F_1(\vec{x}), \dots, y_m - F_m(\vec{x}) \rangle$ plays the role here that

the set of syzygies played for Hilbert-like IPS-certificates.³

A finite generating set for the ideal of zero-certificates can be computed using Gröbner bases (see, e.g., [EH12, 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 giving 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 1.7.

1.7 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 computational 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 depth 3 and 4 in algebraic circuit complexity imply that lower bounds on even depth 3 or 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 thirty-year open question of $\text{AC}^0[p]$ -Frege lower bounds: either there are $\text{AC}^0[p]$ -Frege lower bounds on the PIT axioms, or any $\text{AC}^0[p]$ -Frege lower bounds are as hard as showing $\text{VP} \neq \text{VNP}$ over a field of characteristic p . 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 2.1 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.

³Note that the ideal of zero-certificates is not merely the set of all functions in the ideal $\langle y_1 - F_1(\vec{x}), \dots, y_m - F_m(\vec{x}) \rangle$ that only involve the y_i , since the ideal $\langle y_1, \dots, y_m \rangle \subseteq R[\vec{x}, \vec{y}]$ 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 C.

Open Question 1.13. 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?

Open Question 1.14 (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 [Bro87, Kol88, Som99].

Open Question 1.15. Are there tautologies for which the certificate family constructed in Theorem 3.1 is the one of minimum complexity (under p-projections or c-reductions, see Appendix A.1)?

If there is any family $\varphi = (\varphi_n)$ of tautologies for which Question 1.15 has a positive answer and for which the certificates constructed in Theorem 3.1 are VNP-complete (Question 1.19 below), then super-polynomial size lower bounds on IPS-proofs of φ would be equivalent to $\text{VP} \neq \text{VNP}$. 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 $(f_n^{(0)} + I_n)$ ($I_n \subseteq R[x_1, \dots, x_{\text{poly}(n)}]$), the relationships under various 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 1.16. Given a family of cosets of ideals $f_n^{(0)} + I_n$ (or more generally modules) of polynomials, with $I_n \subseteq R[x_1, \dots, x_{\text{poly}(n)}]$, 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 1.15)? 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 1.16 is of interest even when $f^{(0)} = 0$, that is, for ideals and modules of functions rather than their nontrivial cosets.

Open Question 1.17. 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 B). It seems unlikely that all ideals coming from propositional tautologies are prime, however.

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 [MM82] showed that the doubly-exponential upper bound on the degree of a Gröbner basis [Her26] (see also [Sei74, MW83]) 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 [Bro87, Kol88, Som99]. 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 papers [BS87a, BS87b, BS88] or the survey [BM93].

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 1.18. 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 3.1 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 Appendix A.1), there is some family of tautologies for which the above polynomials are of that degree. However, we have not yet proved this for even a single degree.

Open Question 1.19. Are there tautologies for which the certificates constructed in Theorem 3.1 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?

Prior to our work, much work was done on bounds for the Ideal Membership Problem—EXPSPACE-complete [MM82, May89]—the so-called Effective Nullstellensatz—where exponential degree bounds are known, and known to be tight [Bro87, Kol88, Som99, EL99]—and the arithmetic Nullstellensatz over \mathbb{Z} , where one wishes to bound not only the degree of the polynomials but the sizes of the integer coefficients appearing [KPS01]. The viewpoint afforded by the Ideal Proof Systems raises new questions about potential strengthening of these results.

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

Definition 1.20. 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 $F_1(\vec{x}), \dots, F_m(\vec{x})$ is a polynomial $C(\vec{x}, \vec{y})$ such that

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

An *IPS derivation* of G from F_1, \dots, F_m is a circuit computing some IPS certificate that $G \in \langle F_1, \dots, F_m \rangle$.

For the Ideal Membership Problem, the EXPSPACE lower bound [MM82, May89] implies an subexponential-size lower bound on constant-free circuits computing IPS-certificates of ideal membership (or non-constant-free circuits in characteristic zero, assuming GRH, see Proposition 2.4). Here by “sub-exponential” we mean a function $f(n) \in \bigcap_{\varepsilon > 0} O(2^{n^\varepsilon})$. Indeed, if for every $G(\vec{x}) \in \langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$ there were a constant-free circuit of subexponential size computing some IPS certificate for the membership of G in $\langle F_1, \dots, F_m \rangle$, then guessing that circuit and verifying its correctness using PIT gives a $\text{MA}_{\text{subexp}} \subseteq \text{SUBEXPSPACE}$ algorithm for the Ideal Membership Problem. The EXPSPACE-completeness of Ideal Membership would then imply that $\text{EXPSPACE} \subseteq \text{SUBEXPSPACE}$, contradicting the Space Hierarchy Theorem [HS65]. Under special circumstances, of course, one may be able to achieve better upper bounds.

However, for the effective Nullstellensatz and its arithmetic variant, we leave the following open:

Open Question 1.21. For any G, F_1, \dots, F_m on x_1, \dots, x_n , as in Definition 1.20, is there always an IPS-certificate of subexponential size that G is in the *radical* of $\langle F_1, \dots, F_m \rangle$? Similarly, if $G, F_1, \dots, F_m \in \mathbb{Z}[x_1, \dots, x_n]$ is there a constant-free $\text{IPS}_{\mathbb{Z}}$ -certificate of subexponential size that $aG(\vec{x})$ is in the *radical* of the ideal $\langle F_1, \dots, F_m \rangle$ for some integer a ?

2 Simulations

In this section we start with a result we haven’t yet mentioned relating general IPS to Hilbert-like IPS, and then complete the proofs of any remaining simulation results that we’ve stated previously. Namely, we relate Pitassi’s previous algebraic systems [Pit96, Pit98] and number-of-lines in Polynomial Calculus proofs with subsystems of IPS; we show that $\text{IPS}_{\mathbb{F}_p}$ p-simulates $\text{AC}^0[p]\text{-Frege}$ in a depth-preserving way; and we show that over fields of characteristic zero, IPS-proofs of polynomial size *with arbitrary constants* can be simulated in coAM , assuming the Generalized Riemann Hypothesis.

2.1 General versus Hilbert-like IPS

Proposition 2.1. *Let $F_1 = \dots = F_m = 0$ be a polynomial system of equations in n variables x_1, \dots, 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 $\text{poly}(D, t, n, s)$.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 [Zip79]. 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 [BOT88] or Kaltofen and Yagati [KY89]. We note that all of these interpolation algorithms only give us 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.

⁴If 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 $(\log T)^k$.

Proof. Using a sparse multivariate interpolation algorithm such as Zippel's [Zip79], 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 [Str73] (or alternatively consider $\frac{1}{e_{i_0}} \frac{\partial C'}{\partial y_{i_0}}$ using Baur–Strassen [BS83]). 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. \square

2.2 Number of lines in Polynomial Calculus is equivalent to determinantal IPS

We begin by recalling Pitassi's 1996 and 1997 algebraic proof systems [Pit96, Pit98]. In the 1996 system, a proof of the unsatisfiability of $F_1(\vec{x}) = \cdots = F_m(\vec{x}) = 0$ is a circuit computing a vector $(G_1(\vec{x}), \dots, G_m(\vec{x}))$ such that $\sum_i F_i(\vec{x})G_i(\vec{x}) = 1$. Size is measured by the size of the corresponding circuit.

In the 1997 system, a proof is a rule-based derivation of 1 starting from the F_i . Recall that rule-based algebraic derivations have the following two rules: 1) from G and H , derive $\alpha G + \beta H$ for any fields elements $\alpha, \beta \in \mathbb{F}$, and 2) from G , derive Gx_i for any variable x_i . 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.

Proposition 2.2. *Pitassi's 1996 algebraic proof system [Pit96] is p-equivalent to Hilbert-like IPS.*

Pitassi's 1997 algebraic proof system [Pit98]—equivalent to the number-of-lines measure on PC proofs—is p-equivalent to Hilbert-like det-IPS or VP_{ws}-IPS.

Proof. Let C be a proof in the 1996 system [Pit96], namely a circuit computing $(G_1(\vec{x}), \dots, G_m(\vec{x}))$. 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 [BS83] there is a circuit C of size at most $O(|C'|)$ computing the vector $(\frac{\partial C'}{\partial y_1}, \dots, \frac{\partial C'}{\partial y_m}) = (G'_1(\vec{x}), \dots, G'_m(\vec{x}))$, 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 i -th copy of C' plug in 1 for one of the y_i and 0 for all of the others.)

The proof of the second statement takes a bit more work. At this point the reader may wish to recall the definition of weakly skew circuit from Appendix A.1.

Suppose we have a derivation of 1 from $F_1(\vec{x}), \dots, F_m(\vec{x})$ in the 1997 system [Pit98]. First, replace each $F_i(\vec{x})$ at the beginning of the derivation with the corresponding placeholder variable y_i . Since size in the 1997 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 1997 system [Pit98], 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 [RT08]); 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 1997 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: for all formulas Γ' of size $|\Gamma'| < |\Gamma|$, for all polynomials $P(\vec{x})$, in the 1997 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, 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'd be effectively 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: for all weakly skew circuits Γ' of size $|\Gamma'| < |\Gamma|$, possibly with multiple outputs that we denote $\Gamma'_{out,1}, \dots, \Gamma'_{out,s}$, from any $P(\vec{x})$ one can derive the tuple $P\Gamma'_{out,1}, \dots, P\Gamma'_{out,s}$ in the 1997 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've 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 don't 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 . \square

It is interesting to note that the condition of being weakly skew is precisely the condition we

needed to make this proof go through.

2.3 Depth-preserving simulation of Frege systems by the Ideal Proof System

Theorem 2.3. *For any $d(n)$, depth- $(d+2)$ $\text{IPS}_{\mathbb{F}_p}$ p -simulates depth- d Frege proofs with unbounded fan-in \vee, \neg, MOD_p connectives.*

Proof. For simplicity we will present the proof for $p = 2$. The generalization to other values of p is straightforward. We will use a small modification of the formalization of $\text{AC}^0[2]$ -Frege as given by Maciel and Pitassi [MP98]. The underlying connectives are: negation, unbounded fanin OR, unbounded fanin AND, and unbounded fanin XOR gates. We will work in a sequent calculus style proof system, where lines are cedents of the form $\Gamma \rightarrow \Delta$, where both Γ and Δ are $\{\vee, \neg, \text{MOD}_p\}$ -formulas, where each of $\neg\Gamma_i$ (for $\Gamma_i \in \Gamma$) and $\Delta_i \in \Delta$ has depth at most $d(n)$; the intended meaning is that the conjunction of the formulas in Γ implies the disjunction of the formulas in Δ . For notational convenience, we state the rest of the proof only for $\text{AC}^0[2]$ -Frege, but it will be clear that nothing in the proof depends on the depth being constant. The axioms are as follows.

1. $A \rightarrow A$
2. (false implies nothing) $\vee() \rightarrow$
3. $\rightarrow \neg \oplus()$

The rules of inference are as follows:

Weakening	$\frac{\Gamma \rightarrow \Delta}{\Gamma \rightarrow \Delta, A}$ $\frac{\Gamma \rightarrow \Delta}{A, \Gamma \rightarrow \Delta}$
Cut	$\frac{\rightarrow A, \Gamma \quad \rightarrow \neg A, \Gamma}{\rightarrow \Gamma}$
Negation	$\frac{\Gamma, A \rightarrow \Delta}{\Gamma \rightarrow \neg A, \Delta}$
	$\frac{\Gamma \rightarrow A, \Delta}{\Gamma, \neg A \rightarrow \Delta}$
Or-Left	$\frac{A_1, \Gamma \rightarrow \Delta \quad \vee(A_2, \dots, A_n), \Gamma \rightarrow \Delta}{\vee(A_1, \dots, A_n), \Gamma \rightarrow \Delta}$
Or-Right	$\frac{\Gamma \rightarrow A_1, \vee(A_2, \dots, A_n), \Delta}{\Gamma \rightarrow \vee(A_1, \dots, A_n), \Delta}$
Parity-Left	$\frac{A_1, \neg \oplus(A_2, \dots, A_n), \Gamma \rightarrow \Delta \quad \oplus(A_2, \dots, A_n), \Gamma \rightarrow A_1, \Delta}{\oplus(A_1, \dots, A_n), \Gamma \rightarrow \Delta}$
Parity-Right	$\frac{A_1, \Gamma \rightarrow \neg \oplus(A_2, \dots, A_n), \Delta \quad \Gamma \rightarrow A_1, \oplus(A_2, \dots, A_n), \Delta}{\Gamma \rightarrow \oplus(A_1, \dots, A_n), \Delta}$

A refutation of a 3CNF formula $\varphi = \kappa_1 \wedge \kappa_2 \wedge \dots \wedge \kappa_m$ in $\text{AC}^0[2]$ -Frege 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. It is well known that any Frege refutation can be efficiently converted into a tree-like proof.⁵

We define a translation $t(A)$ from Boolean formulas to algebraic circuits (over \mathbb{F}_2) such that for any assignment α , $A(\alpha) = 1$ if and only if $t(A)(\alpha) = 0$. The translation is defined inductively as follows.

⁵By tree-like, we mean that the underlying directed acyclic graph structure of the proof is a tree, and therefore every cedent, other than the final empty cedent, in the refutation is used exactly once.

1. $t(x) = 1 - x$ for x atomic (a Boolean variable).
2. $t(\neg A) = 1 - t(A)$
3. $t(\vee(A_1, \dots, A_n)) = t(A_1)t(A_2) \cdots t(A_n)$
4. $t(\oplus(A_1, \dots, A_n)) = n - t(A_1) - t(A_2) \cdots - t(A_n)$ (recall n will be interpreted mod 2).

Note that the depth of $t(A)$ as an algebraic formula is at most the depth of A as a Boolean formula.

For a cedent $\Gamma \rightarrow \Delta$, we will translate the cedent by moving everything to the right of the arrow. That is, the cedent $L = A_1, \dots, A_n \rightarrow B_1, \dots, B_m$ will be translated to $t(L) = t(\neg A_1 \vee \cdots \vee \neg A_n \vee B_1 \vee \cdots \vee B_m) = (1 - t(A_1))(1 - t(A_2)) \cdots (1 - t(A_n))t(B_1) \cdots t(B_m)$.

Let R be a tree-like $\text{AC}^0[2]$ -Frege refutation of φ . 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 1.20) of the form $\sum_i G_i y_i$, where the y_i 's are the placeholder variables for the initial polynomials (the sum may contain each y_i more than once), each G_i is a depth d formula, and the overall size is polynomial in the size of the original $\text{AC}^0[2]$ -Frege refutation. (NB: as will become clear below, in order to preserve the depth, we wait to gather like terms in the sum until the end of the proof.) The placeholder variables y_1, \dots, y_m correspond to $t(\kappa_1), \dots, t(\kappa_m)$, and y_{m+1}, \dots, y_{m+n} correspond to the Boolean axioms $x_1^2 - x_1, \dots, x_n^2 - x_n$.

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

The axiom $A \rightarrow A$ translates to $t(A)(1 - t(A))$. A simple induction on the structure of A shows that $t(A)(1 - t(A))$ can be derived from the $x_i^2 - x_i$ by an IPS derivation of depth at most the depth of A . The other axioms translate to the identically zero polynomial, so again have the right form.

For the inductive step, it is a matter of going through all of the rules. We assume inductively that we have a list L of circuits each of the form $G_i y_i$, such that each G_i has a product gate at its output, and $\sum_L G_i y_i$ is a derivation of the antecedents of the rule (note that, as L is a list, each y_i may appear more than once in this sum).

1. (Weakening) Assume $\sum G_i y_i$ is a derivation of $t(\Gamma \rightarrow \Delta)$. We want to obtain a derivation of $t(\Gamma \rightarrow \Delta, A)$. Since we move everything to the right when we translate, this is equivalent to showing that if $\sum G_i y_i$ is a derivation of $t(\rightarrow A_1, \dots, A_n) = t(A_1)t(A_2) \cdots t(A_n)$, that we can obtain a derivation of $t(\rightarrow A_1, \dots, A_n, B) = t(A_1)t(A_2) \cdots t(A_n)t(B)$. Multiplying each $G_i y_i$ by $t(B)$ achieves this. The resulting derivation is equivalent to $\sum G'_i y_i$, where the depth of G'_i is $\max\{\text{depth}(G_i), \text{depth}(B)\}$ (we do not need to add 1 to the depth because we've assumed that G_i has a product gate at the top).
2. (Cut) We want to show that if $\sum G_i y_i$ is a derivation of $t(\rightarrow \neg A, B_1, \dots, B_n) = (1 - t(A))t(B_1) \cdots t(B_n)$ and $\sum G'_i y_i$ is a derivation of $t(\rightarrow A, B_1, \dots, B_n) = t(A)t(B_1) \cdots t(B_n)$, that we can derive $t(\rightarrow B_1 \dots B_n) = t(B_1) \cdots t(B_n)$. Semantically, adding these two derivations gives what we want. In order to preserve the inductive assumption, we do *not* gather terms, but rather concatenate the two lists $(G_i y_i)$ and $(G'_i y_i)$, so that each term still has a product gate at the top without increasing the depth.
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) We want to show that if $\sum G_i y_i$ is a derivation of $t(\rightarrow \neg A_1, \Delta)$, and $\sum G'_i y_i$ is a derivation of $t(\rightarrow \neg \vee(A_2, \dots, A_n), \Delta)$, then we can derive $t(\rightarrow \neg \vee(A_1, \dots, A_n), \Delta)$. We have

$$\sum G_i F_i = t(\rightarrow \neg A_1, \Delta) = (1 - t(A_1))t(\Delta),$$

$$\sum G'_i F_i = t(\rightarrow \neg \vee(A_2, \dots, A_n), \Delta) = (1 - t(A_2)t(A_3) \cdots t(A_n))t(\Delta).$$

Multiplying the second by $t(A_1)$ and “adding” to the first gives the desired derivation. Again, when we “add” we do not gather terms, but rather just concatenate lists, so that each G_i has a product gate at the top.

5. (Or-Right) The translation of the derived formula is syntactically identical to the original formula, so there is nothing to do.
6. (Parity-Left) We want to show that if $\sum G_i y_i$ is a derivation of $t(\rightarrow \neg A_1, \oplus(A_2, \dots, A_n), \Delta)$ and $\sum G'_i y_i$ is a derivation of $t(\rightarrow A_1, \neg \oplus(A_2, \dots, A_n), \Delta)$, then we can derive $t(\rightarrow \neg \oplus(A_1, \dots, A_n), \Delta)$. We have

$$t(\rightarrow \neg A_1, \oplus(A_2, \dots, A_n), \Delta) = (1 - t(A_1))(n - 1 - t(A_2) - t(A_3) - \cdots - t(A_n))t(\Delta),$$

$$t(\rightarrow A_1, \neg \oplus(A_2, \dots, A_n), \Delta) = t(A_1)(1 - (n - 1 - t(A_2) - t(A_3) - \cdots - t(A_n)))t(\Delta).$$

It is easily verified that subtracting the latter from the former yields $t(\rightarrow \neg \oplus(A_1, \dots, A_n), \Delta)$. To perform “subtraction” while maintaining a product gate at the top, we multiply the latter by -1 and then concatenate the two lists.

7. (Parity-Right) This case is similar to Parity-left.

In all cases, we can derive the bottom cedent as $\sum_i G_i y_i$, where each G_i has constant depth (in fact, depth at most one greater than the depth of the original proof), and the overall size is polynomial in the original proof size. Since we’ve actually just been maintaining a list of terms $G_i y_i$ in which the y_i may appear multiple times, the final step is to add these all together and gather terms, leading to our final derivation of polynomial size, and depth at most $d + 2$, where d was the original depth. \square

2.4 Simulating IPS-proofs with arbitrary constants in coAM

The following proposition shows how we may conclude that $\text{NP} \subseteq \text{coAM}$ from the assumption of polynomial-size IPS proofs for all tautologies, *without* assuming the IPS proofs are constant-free (but using the Generalized Riemann Hypothesis). We thank Pascal Koiran for the second half of the proof.

Proposition 2.4. *Assuming the Generalized Riemann Hypothesis, over any field \mathbb{F} of characteristic zero, if every propositional tautology has a polynomial-size $\text{IPS}_{\mathbb{F}}$ -proof of polynomial degree, then $\text{NP} \subseteq \text{coAM}$.*

We do not know how to improve this result from coAM to coMA (as in Proposition 1.4).

Proof (with P. Koiran). We reduce to the fact 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 [Koi96]. 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-certificates for F . Namely, consider a *generic* polynomial-size circuit, meaning a layered circuit of $\text{poly}(n)$ depth and $\text{poly}(n)$ width, with n inputs $x_1, \dots, x_n, y_1, \dots, y_m$, and alternating layers of linear combination and product gates, where every edge e terminating at any linear combination gate gets its own independent variable z_e . The output gate of this generic circuit computes a polynomial $C(\vec{x}, \vec{y}, \vec{z})$, and for any setting of the z_e variables to constants ζ_e , we get a particular polynomial-size circuit computing a polynomial $C_{\vec{\zeta}}(\vec{x}, \vec{y}) := C(\vec{x}, \vec{y}, \vec{\zeta})$. 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, if there is a polynomial size IPS proof C' for F , then 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 $\vec{\xi}^{(1)}, \vec{\xi}^{(2)}, \dots, \vec{\xi}^{(h)}$ to be substituted into \vec{x} ; think of the $\vec{\xi}^{(i)}$ as a hitting set for the x -variables. Then we consider the solvability of the following set of $2h$ equations in \vec{z} :

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

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 [Koi96]. If $\vec{\zeta}$ is such that $C_{\vec{\zeta}}(\vec{x}, \vec{y}) = C(\vec{x}, \vec{y}, \vec{\zeta})$ is in fact an IPS certificate, then the preceding equalities will be satisfied regardless of the choices of the $\vec{\xi}^{(i)}$. Otherwise, at least one monomial in $C(\vec{x}, \vec{0}, \vec{\zeta})$ or $C(\vec{x}, \vec{F}(\vec{x}), \vec{\zeta}) - 1$ will be nonzero. Since all the monomials have polynomial degree, the usual DeMillo–Lipton–Schwarz–Zippel lemma implies that with high probability, a random point $\vec{\xi}$ will make any such nonzero monomial evaluate to a nonzero value. Choosing polynomially many points thus suffices. 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 VP-IPS certificates, we get an AM algorithm for TAUT. \square

3 Lower bounds on IPS imply circuit lower bounds

Here we complete the proof of the following theorem:

Theorem 3.1. *A super-polynomial lower bound on [constant-free] Hilbert-like IPS_R proofs of any family of tautologies implies $\text{VNP}_R \neq \text{VP}_R$ [respectively, $\text{VNP}_R^0 \neq \text{VP}_R^0$], for any ring R .*

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

In Section 1.4 we proved this theorem assuming the following key lemma, which we now prove in full.

Lemma 3.2. *Every family of CNF tautologies (φ_n) has a Hilbert-like family of IPS certificates (C_n) in VNP_R^0 .*

Proof. We mimic one of the proofs of completeness for Hilbert-like IPS [Pit96, Theorem 1] (recall Proposition 2.2), 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 $\varphi_n(\vec{x}) = \kappa_1(\vec{x}) \wedge \dots \wedge \kappa_m(\vec{x})$ 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 $C_1(\vec{x}) = \dots = C_m(\vec{x}) = x_1^2 - x_1 = \dots = x_n^2 - x_n = 0$ is unsatisfiable. In fact, as we'll see in the course of the proof, we won't 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

$$1 = \sum_{\vec{e} \in \{0,1\}^n} \prod_{i=1}^n b(e_i, x_i). \quad (2)$$

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 $C_i(\vec{x})$.

The idea is to partition the assignments $\{0, 1\}^n$ into m parts A_1, \dots, A_m , where all assignments in the i -th part A_i falsify clause i . This will then allow us to rewrite equation (2) as

$$1 = \sum_{i=1}^m C_i(\vec{x}) \left(\sum_{\vec{e} \in A_i} \prod_{j: x_j \notin \kappa_i} b(e_j, x_j) \right), \quad (3)$$

where “ $x_j \notin \kappa_i$ ” means that neither x_j nor its negation appears in κ_i . Equation (3) then becomes the IPS-certificate $\sum_{i=1}^m c_i \cdot \left(\sum_{\vec{e} \in A_i} \prod_{j: x_j \notin \kappa_i} b(e_j, x_j) \right)$. What remains is to show that the sum can indeed be rewritten this way, and that there is some partition (A_1, \dots, A_m) as above such that the resulting certificate is in fact in VNP.

First, let us see why such a partition allows us to rewrite (2) as (3). The key fact here is that the clause polynomial $C_i(\vec{x})$ divides the term $t_{\vec{e}}(\vec{x}) := \prod_{i=1}^n b(e_i, x_i)$ if and only if $C_i(\vec{e}) = 1$, if and only if \vec{e} falsifies κ_i . Let $C_i(\vec{x}) = \prod_{i \in I} b(f_i, x_i)$, where $I \subseteq [n]$ is the set of indices of the variables appearing in clause i . By the properties of b discussed above, $1 = C_i(\vec{e}) = \prod_{i \in I} b(f_i, e_i)$ if and only if $b(f_i, e_i) = 1$ for all $i \in I$, if and only if $f_i = e_i$ for all $i \in I$. In other words, if $1 = C_i(\vec{e})$ then $C_i = \prod_{i \in I} b(e_i, x_i)$, which clearly divides $t_{\vec{e}}$. Conversely, suppose $C_i(\vec{x})$ divides $t_{\vec{e}}(\vec{x})$. Since $t_{\vec{e}}(\vec{e}) = 1$ and every factor of $t_{\vec{e}}$ only takes on Boolean values on Boolean inputs, it follows that every factor of $t_{\vec{e}}$ evaluates to 1 at \vec{e} , in particular $C_i(\vec{e}) = 1$.

Let A_1, \dots, A_m be a partition of $\{0, 1\}^n$ such that every assignment in A_i falsifies κ_i . Since C_i divides every term $t_{\vec{e}}$ such that \vec{e} falsifies clause i , C_i divides every term $t_{\vec{e}}$ with $\vec{e} \in A_i$, and thus we can indeed rewrite (2) as (3).

Next, we show how to construct a partition A_1, \dots, A_m as above so that the resulting certificate is in VNP. The partition we'll use is a greedy one. A_1 will consist of *all* assignments that falsify κ_1 . A_2 will consist of all *remaining* assignments that falsify κ_2 . And so on. In particular, A_i consists of all assignments that falsify κ_i and *satisfy* all A_j with $j < i$. (If at some clause κ_i before we reach the end, we have used up all the assignments—which happens if and only if the first i clauses on their own are unsatisfiable—that's okay: nothing we've done so far nor anything we do below assumes that all A_i are nonempty.)

Equivalently, $A_i = \{\vec{e} \in \{0, 1\}^n \mid C_i(\vec{e}) = 1 \text{ and } C_j(\vec{e}) = 0 \text{ for all } j < i\}$. For any property Π , we write $\llbracket \Pi(\vec{e}) \rrbracket$ for the indicator function of Π : $\llbracket \Pi(\vec{e}) \rrbracket = 1$ if and only if $\Pi(\vec{e})$ holds, and 0 otherwise.

We thus get the certificate:

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

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 , hence the certificate is in VNP⁰. \square

4 PIT as a bridge between circuit complexity and proof complexity

Having already introduced and discussed our PIT axioms in Section 1.5, here we complete the proofs of Theorems 4.1 and 4.5. We maintain the notations and conventions of Section 1.5.

4.1 Extended Frege is p-equivalent to IPS if PIT is EF-provably easy

Theorem 4.1. *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.*

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_{IPS}* follows from efficient EF proofs for our PIT axioms.

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 $\varphi = \kappa_1 \wedge \dots \wedge \kappa_m$ be an unsatisfiable propositional 3CNF formula over variables p_1, \dots, p_n , and let $Q_1^\varphi, \dots, Q_m^\varphi$ 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}^φ . (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, \dots, x_n

(corresponding to the propositional variables p_1, \dots, p_n) and the placeholder variables y_1, \dots, y_m (corresponding to the equation $Q_1^\varphi, \dots, Q_m^\varphi$), 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 $Q_1^\varphi, \dots, Q_m^\varphi$.
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, $\text{Proof}_{\text{IPS}}([C], [\varphi])$ 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 $\text{Proof}_{\text{IPS}}([C], [\varphi])$ 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. Secondly, 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.) $\text{Proof}_{\text{IPS}}([C], [\varphi])$ 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 4.2. Let formula $\text{Truth}_{\text{bool}}(\vec{p}, \vec{q})$ 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, $\text{Soundness}_{\text{IPS}, m, n}([C], [\varphi], \vec{p})$ 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} . $\text{Soundness}_{\text{IPS}, m, n}([C], [\varphi], \vec{p})$ states:

$$\text{Proof}_{\text{IPS}}(\underline{[C]}, \underline{[\varphi]}) \rightarrow \neg \text{Truth}_{\text{bool}}(\underline{[\varphi]}, \vec{p}).$$

Lemma 4.3. If EF can efficiently prove $\text{Soundness}_{\text{IPS}}$ 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 $\text{Soundness}_{\text{IPS}}$. 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 4.3) that EF can efficiently prove:

$$\varphi \rightarrow \text{Truth}_{\text{bool}}([\varphi], \vec{p}).$$

(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 $\text{Proof}_{\text{IPS}}([C], [\varphi])$.⁶ This holds because both C and φ are fixed, so this formula is variable-free. Thus, EF can just verify that it is true.

⁶The fact that $\text{Proof}_{\text{IPS}}([C], [\varphi])$ 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 4.1 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 4.4.

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

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

Theorem 4.1 follows from the preceding lemma and the next one.

Lemma 4.4. *If EF can efficiently prove the PIT axioms for some polynomial-size Boolean circuit family K computing PIT, then EF can efficiently prove $\text{Soundness}_{\text{IPS}}$ (for that same K).*

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

1. First show for every $i \in [m]$, $\text{Truth}_{\text{bool}}([\varphi], \vec{p}) \rightarrow K(\underline{[Q_i^\varphi(\vec{p})]})$, 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 $[\varphi]$, the encoding $\underline{[Q_i^\varphi]}$ depends on a subset of these variables.

$\text{Truth}_{\text{bool}}([\varphi], \vec{p})$ 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,

$$\text{Truth}_{\text{bool}}(\underline{[\kappa_i]}, \vec{p}) \rightarrow K(\underline{[Q_i^\varphi(\vec{p})]}) \quad (*)$$

is a tautology. Furthermore, although both the encoding $\underline{[\kappa_i]}$ and $\underline{[Q_i^\varphi]}$ depend on the propositional variables encoding $[\varphi]$, since we assume that φ is a 3CNF, these only depend on *constantly many* of the variables encoding $[\varphi]$. Thus the tautology $(*)$ can be proven in EF by brute force. Putting these together we can derive $\text{Truth}_{\text{bool}}([\varphi], \vec{p}) \rightarrow K(\underline{[Q_i^\varphi(\vec{p})]})$, as desired.

2. Using the assumption $\text{Truth}_{\text{bool}}([\varphi], \vec{p})$ together with (1) we derive $K(\underline{[Q_i^\varphi(\vec{p})]})$ for all $i \in [m]$.
3. Using Axiom 1 we can prove $K(\underline{[C(\vec{x}, \vec{0})]}) \rightarrow K(\underline{[C(\vec{p}, \vec{0})]})$. 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

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

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 $K(\underline{[1 - C(\vec{x}, \vec{Q}(\vec{x}))]}) \rightarrow K(\underline{[1 - C(\vec{p}, \vec{Q}(\vec{p}))]})$. 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. \square

4.2 Proofs relating $\text{AC}^0[p]$ -Frege lower bounds, PIT, and circuit lower bounds

Having already discussed the corollaries and consequences of Theorem 4.5, here we merely complete its proof.

Theorem 4.5. *Let \mathcal{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 \mathcal{C} -Frege proofs, then \mathcal{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 \mathcal{C} . This requires one more technical device compared to the proofs in the previous section. The proof of Theorem 4.5 follows the proof of Theorem 4.1 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 , together with a careful analysis of what was needed in the proof of Theorem 4.1.

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 ($=\text{NC}^1$ -Frege), $\text{AC}^0[p]$ -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 3 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 $\text{Proof}_{\text{IPS}}(\underline{[C]}, \underline{[\varphi]})$ now takes the following form. Recall that $\text{Proof}_{\text{IPS}}$ 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 it’s i -th bit, which may be a constant, a propositional variable, or even a propositional formula. Then $\text{Proof}_{\text{IPS}}(\underline{[C]}, \underline{[\varphi]})$ 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 $\text{Proof}_{\text{IPS}}(\underline{[C]}, \underline{[\varphi]})$ as before to mean this modified statement (we will no longer be referring to the original, EF-style statement). The modified statement $\text{Soundness}_{\text{IPS}}(\underline{[C]}, \underline{[\varphi]}, \vec{p})$ will now take the form

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

using the new version of $\text{Proof}_{\text{IPS}}$. Here “dummy statements” refers to certain statements that we will explain in Lemma 4.7. These dummy statements will only involve variables that do not

appear in the rest of $\text{Soundness}_{\text{IPS}}$, and therefore will be immediately seen not to affect its truth or provability.

The proofs

Lemmata 4.7 and 4.8 are the AC^0 -analogs of Lemmata 4.3 and 4.4, respectively. The proof of Lemma 4.7 will cause no trouble, and the proof of Lemma 4.8 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 4.6. *Let (φ_n) be a sequence of tautologies on $\text{poly}(n)$ 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 $\text{poly}(n)$ -size AC^0 -Frege proof of φ .

Proof sketch. The basic idea is that AC^0 -Frege can brute force over all $\text{poly}(n)$ -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 $\varphi = \bigvee_i (\neg \alpha_i) \vee \omega = \left(\bigvee_g \neg(k_g^{(i)} \leftrightarrow k_{g\ell}^{(i)} \text{op}_g k_{gr}^{(i)}) \right) \vee \left(\bigvee_{\text{remaining clauses } i} \neg \alpha_i \right) \vee \omega$. Thus, if AC^0 -Frege can prove that the rest of φ , namely $\left(\bigvee_{\text{remaining clauses } i} \neg \alpha_i \right) \vee \omega$ is a tautology, then it can prove that φ is a tautology. \square

Now we state the analogs of Lemmata 4.3 and 4.4 for \mathcal{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 4.7 (AC^0 analog of Lemma 4.3). *Let \mathcal{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 \mathcal{C} -Frege proofs, then \mathcal{C} -Frege is polynomially equivalent to IPS.*

Proof. Mimic the proof of Lemma 4.3. 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 \mathcal{C} -Frege can prove $\varphi \rightarrow \text{Truth}_{\text{bool}}([\varphi], \vec{p})$. This follows directly from the details of the encoding of $[\varphi]$ and the full definition of $\text{Truth}_{\text{bool}}$; see Lemma 4.9.

The second step is to show that \mathcal{C} -Frege can prove $\text{Proof}_{\text{IPS}}([C], [\varphi])$ for a fixed C, φ . In Lemma 4.3, 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, $\text{Proof}_{\text{IPS}}([C], [\varphi])$ satisfies the requirements of Lemma 4.6, and applying that lemma we are done. \square

Lemma 4.8 (AC^0 analog of Lemma 4.4). *Let \mathcal{C} be a class of circuits closed under AC^0 circuit reductions. If \mathcal{C} -Frege can efficiently prove the PIT axioms for some polynomial-sized family of circuits K computing PIT, then \mathcal{C} -Frege can efficiently prove $\text{Soundness}_{\text{IPS}}$ (for that same K).*

Proof. We mimic the proof of Lemma 4.4. 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 $m \leq \text{poly}(n)$. In order to talk about these copies of K in \mathcal{C} -Frege, however, the K variables must already be present in the statement we wish to prove in \mathcal{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 $K^{(3)}, \dots, K^{(m+2)}$ (recall that the first two copies $K^{(1)}$ and $K^{(2)}$ are already used in the statement of *Proof_{IPS}*). We won’t actually need these clauses anywhere in the proof, we just need their variables to be present from the beginning.

Starting with $\text{Truth}_{\text{bool}}([\varphi], \vec{p}), K^{(1)}([C(\vec{x}, \vec{0})]), K^{(2)}([1 - C(\vec{x}, \vec{Q}(\vec{x}))])$ we’ll derive a contradiction. The only step of the proof of Lemma 4.4 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 $i \in [m]$, $\text{Truth}_{\text{bool}}([\varphi], \vec{p}) \rightarrow K([Q_i^\varphi(\vec{p})])$, 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.

$\text{Truth}_{\text{bool}}([\varphi], \vec{p})$ 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,

$$\text{Truth}_{\text{bool}}([\kappa_i], \vec{p}) \rightarrow K^{(i+2)}([Q_i^\varphi(\vec{p})]) \quad (**)$$

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

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

which is equivalent to

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

Thus (**) satisfies the conditions of Lemma 4.6 and has a short AC^0 -Frege proof. Since $\text{Truth}_{\text{bool}}([\varphi], \vec{p})$ is defined as $\bigwedge_i \text{Truth}_{\text{bool}}([\kappa_i], \vec{p})$ (see Section 4.3), we can then derive

$$\text{Truth}_{\text{bool}}([\varphi], \vec{p}) \rightarrow K^{(i+2)}([Q_i^\varphi(\vec{p})]),$$

as desired. \square

4.3 Some details of the encodings

For an $\leq m$ -clause, $\leq n$ -variable 3CNF $\varphi = \kappa_1 \wedge \dots \wedge \kappa_m$, 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

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

For a single 3-literal clause κ , we define $Truth_{bool,n}([\kappa], \vec{p})$ as follows. For an integer i , let $[i]$ denote the standard binary encoding of $i - 1$ (so that the numbers $1, \dots, 2^k$ are put into bijective correspondence with $\{0, 1\}^k$). Let $[\kappa] = \vec{q}_1 s_1 \vec{q}_2 s_2 \vec{q}_3 s_3$ 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:

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

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

Lemma 4.9. *For any 3CNF φ on n variables, there are $\text{poly}(n)$ -size AC⁰-Frege proofs of $\varphi(\vec{p}) \rightarrow Truth_{bool}([\varphi], \vec{p})$.*

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 $Truth_{bool}([\kappa], \vec{p})$ in fact becomes *syntactically identical* to $\kappa(\vec{p})$. By the definition of $Truth_{bool}([\varphi], \vec{p})$, we get the same conclusion for any fixed CNF φ . Simplifying constants can easily be carried out in AC⁰-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 $\kappa(\vec{p})$. \square

Acknowledgments

We thank David Liu for many interesting discussions, and for collaborating with us on some of the open questions posed in this paper. 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 4.1. We thank Pascal Koiran for providing the second half of the proof of Proposition 2.4. We thank Iddo Tzameret for useful discussions that led to Proposition 2.2. 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 didn’t even hold a candle to this one. We gratefully acknowledge financial support from NSERC; in particular, J. A. G. was supported by A. Borodin’s NSERC Grant # 482671.

References

- [ABSRW02] Michael Alekhnovich, Eli Ben-Sasson, Alexander A. Razborov, and Avi Wigderson, *Space complexity in propositional calculus*, SIAM J. Comput. **31** (2002), no. 4, 1184–1211.
- [Ajt94] Miklós Ajtai, *The complexity of the pigeonhole principle*, Combinatorica **14** (1994), no. 4, 417–433, A preliminary version appeared in FOCS ’88, pp. 346–355.

- [AM69] M. F. Atiyah and I. G. Macdonald, *Introduction to commutative algebra*, Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1969.
- [AV08] Manindra Agrawal and V. Vinay, *Arithmetic circuits: A chasm at depth four*, FOCS '08: 49th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, 2008, pp. 67–75.
- [AW08] Scott Aaronson and Avi Wigderson, *Algebrization: a new barrier in complexity theory*, STOC '08: 40th Annual ACM Symposium on Theory of Computing, ACM, New York, 2008, pp. 731–740.
- [BCS97] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi, *Algebraic complexity theory*, Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], vol. 315, Springer-Verlag, Berlin, 1997, With the collaboration of Thomas Lickteig.
- [BDG⁺04] Maria Luisa Bonet, Carlos Domingo, Ricard Gavaldà, Alexis Maciel, and Toniann Pitassi, *Non-automatizability of bounded-depth Frege proofs*, Comput. Complexity **13** (2004), no. 1-2, 47–68, A preliminary version appeared in CCC '99.
- [BGS75] Ted Baker, John Gill, and Robert Solovay, *Relativizations of the P =? NP question*, SIAM J. Comput. **4** (1975), 431–442.
- [BIK⁺92] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, Pavel Pudlák, and Alan Woods, *Exponential lower bounds for the pigeonhole principle*, STOC '92: 24th Annual ACM Symposium on Theory of Computing (New York, NY, USA), ACM, 1992, pp. 200–220.
- [BIK⁺96] Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák, *Lower bounds on Hilbert's Nullstellensatz and propositional proofs*, Proc. London Math. Soc. (3) **73** (1996), no. 1, 1–26, A preliminary version appeared in FOCS '94, pp. 794–806.
- [BM93] Dave Bayer and David Mumford, *What can be computed in algebraic geometry?*, Computational algebraic geometry and commutative algebra (Cortona, 1991), Sympos. Math., XXXIV, Cambridge Univ. Press, Cambridge, 1993, Preprint available as arXiv:alg-geom/9304003, pp. 1–48.
- [BOT88] Michael Ben-Or and Prasoon Tiwari, *A deterministic algorithm for sparse multivariate polynomial interpolation*, STOC '88: 20th Annual ACM Symposium on Theory of Computing (New York, NY, USA), ACM, 1988, pp. 301–309.
- [BPR00] Maria Luisa Bonet, Toniann Pitassi, and Ran Raz, *On interpolation and automatization for Frege systems*, SIAM J. Comput. **29** (2000), no. 6, 1939–1967, A preliminary version appeared in FOCS '97.
- [Bro87] W. Dale Brownawell, *Bounds for the degrees in the Nullstellensatz*, Ann. of Math. (2) **126** (1987), no. 3, 577–591.
- [BS83] Walter Baur and Volker Strassen, *The complexity of partial derivatives*, Theoret. Comput. Sci. **22** (1983), no. 3, 317–330.

- [BS87a] David Bayer and Michael Stillman, *A criterion for detecting m -regularity*, Invent. Math. **87** (1987), no. 1, 1–11.
- [BS87b] David Bayer and Michael Stillman, *A theorem on refining division orders by the reverse lexicographic order*, Duke Math. J. **55** (1987), no. 2, 321–328.
- [BS88] David Bayer and Michael Stillman, *On the complexity of computing syzygies*, J. Symbolic Comput. **6** (1988), no. 2-3, 135–147.
- [Bür00a] Peter Bürgisser, *Completeness and reduction in algebraic complexity theory*, Algorithms and Computation in Mathematics, vol. 7, Springer-Verlag, Berlin, 2000.
- [Bür00b] Peter Bürgisser, *Cook’s versus Valiant’s hypothesis*, Theoret. Comput. Sci. **235** (2000), no. 1, 71–88, Selected papers in honor of Manuel Blum (Hong Kong, 1998).
- [CEI96] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo, *Using the Gröbner basis algorithm to find proofs of unsatisfiability*, STOC ’96: 28th Annual ACM Symposium on Theory of Computing, ACM, New York, 1996, pp. 174–183.
- [CKW10] Xi Chen, Neeraj Kayal, and Avi Wigderson, *Partial derivatives in arithmetic complexity and beyond*, Found. Trends Theor. Comput. Sci. **6** (2010), no. 1–2.
- [CR79] Stephen A. Cook and Robert A. Reckhow, *The relative efficiency of propositional proof systems*, J. Symbolic Logic **44** (1979), no. 1, 36–50.
- [EH12] Viviana Ene and Jürgen Herzog, *Gröbner bases in commutative algebra*, Graduate Studies in Mathematics, vol. 130, American Mathematical Society, Providence, RI, 2012.
- [Eis95] David Eisenbud, *Commutative algebra*, Graduate Texts in Mathematics, vol. 150, Springer-Verlag, New York, 1995.
- [EL99] Lawrence Ein and Robert Lazarsfeld, *A geometric effective Nullstellensatz*, Invent. Math. **137** (1999), no. 2, 427–448.
- [FF93] Joan Feigenbaum and Lance Fortnow, *Random-self-reducibility of complete sets*, SIAM J. Comput. **22** (1993), no. 5, 994–1005.
- [GKKS13] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi, *Arithmetic circuits: A chasm at depth three*, FOCS ’13: 54th Annual IEEE Symposium on Foundations of Computer Science, 2013.
- [Her26] Grete Hermann, *Die Frage der endlich vielen Schritte in der Theorie der Polynomideale*, Math. Ann. **95** (1926), no. 1, 736–788.
- [Hil78] David Hilbert, *Hilbert’s invariant theory papers*, Lie Groups: History, Frontiers and Applications, VIII, Math Sci Press, Brookline, Mass., 1978, Translated from the German by Michael Ackerman, With comments by Robert Hermann.
- [HS65] Juris Hartmanis and Richard E. Stearns, *On the computational complexity of algorithms*, Trans. Amer. Math. Soc. **117** (1965), 285–306.

- [HT12] Pavel Hrubeš and Iddo Tzameret, *Short proofs for the determinant identities*, STOC '12: 44th Annual ACM Symposium on Theory of Computing, ACM, New York, 2012, pp. 193–212.
- [JS12] Maurice Jansen and Rahul Santhanam, *Stronger lower bounds and randomness-hardness trade-offs using associated algebraic complexity classes*, STACS '12: 29th Annual Symposium on Theoretical Aspects of Computer Science, LIPIcs. Leibniz Int. Proc. Inform., vol. 14, Schloss Dagstuhl. Leibniz-Zent. Inform., Wadern, 2012, pp. 519–530.
- [KI04] Valentine Kabanets and Russell Impagliazzo, *Derandomizing polynomial identity tests means proving circuit lower bounds*, Comput. Complexity **13** (2004), no. 1-2, 1–46.
- [Koi96] Pascal Koiran, *Hilbert's Nullstellensatz is in the polynomial hierarchy*, J. Complexity **12** (1996), no. 4, 273–286, Special issue for the Foundations of Computational Mathematics Conference (Rio de Janeiro, 1997).
- [Koi12] Pascal Koiran, *Arithmetic circuits: the chasm at depth four gets wider*, Theoret. Comput. Sci. **448** (2012), 56–65.
- [Kol88] János Kollár, *Sharp effective Nullstellensatz*, J. Amer. Math. Soc. **1** (1988), no. 4, 963–975.
- [KPS01] Teresa Krick, Luis Miguel Pardo, and Martín Sombra, *Sharp estimates for the arithmetic Nullstellensatz*, Duke Math. J. **109** (2001), no. 3, 521–598.
- [Kra95] Jan Krajíček, *Bounded arithmetic, propositional logic, and complexity theory*, Encyclopedia of Mathematics and its Applications, vol. 60, Cambridge University Press, Cambridge, 1995.
- [KY89] Erich Kaltofen and Lakshman Yagati, *Improved sparse multivariate polynomial interpolation algorithms*, ISSAC '88: International Symposium on Symbolic and Algebraic Computation, Lecture Notes in Computer Science, vol. 358, Springer, Berlin, 1989, pp. 467–474.
- [Mat80] Hideyuki Matsumura, *Commutative algebra*, second ed., Mathematics Lecture Note Series, vol. 56, Benjamin/Cummings Publishing Co., Inc., Reading, Mass., 1980.
- [May89] Ernst Mayr, *Membership in polynomial ideals over \mathbb{Q} is exponential space complete*, STACS '89: 6th Annual Symposium on Theoretical Aspects of Computer Science, Lecture Notes in Computer Science, vol. 349, Springer, Berlin, 1989, pp. 400–406.
- [MM82] Ernst W. Mayr and Albert R. Meyer, *The complexity of the word problems for commutative semigroups and polynomial ideals*, Adv. in Math. **46** (1982), no. 3, 305–329.
- [MP98] Alexis Maciel and Toniann Pitassi, *Towards lower bounds for bounded-depth Frege proofs with modular connectives*, Proof Complexity and Feasible Arithmetics (Paul Beame and Sam Buss, eds.), DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 39, American Mathematical Society, 1998, A preliminary version appeared in “On $\text{ACC}^0[p^k]$ Frege Proofs,” STOC '97, 720–729, pp. 195–227.
- [MP08] Guillaume Malod and Natacha Portier, *Characterizing Valiant's algebraic complexity classes*, J. Complexity **24** (2008), no. 1, 16–38.

- [Mul99] Ketan D. Mulmuley, *Lower bounds in a parallel model without bit operations*, SIAM J. Comput. **28** (1999), no. 4, 1460–1509 (electronic).
- [Mul12] Ketan D. Mulmuley, *The GCT program toward the P vs. NP problem*, Commun. ACM **55** (2012), no. 6, 98–107.
- [Mum76] David Mumford, *Algebraic geometry I. complex projective varieties*, Grundlehren der Mathematischen Wissenschaften, no. 221, Springer-Verlag, Berlin, 1976.
- [MW83] D. W. Masser and G. Wüstholz, *Fields of large transcendence degree generated by values of elliptic functions*, Invent. Math. **72** (1983), no. 3, 407–464.
- [Pit96] Toniann Pitassi, *Algebraic propositional proof systems*, Descriptive Complexity and Finite Models, Proceedings of the DIMACS Workshop held at Princeton University, Princeton, NJ, January 14–17, 1996. Edited by Neil Immerman and Phokion G. Kolaitis, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, vol. 31, American Mathematical Society, 1996, pp. 215–244.
- [Pit98] Toniann Pitassi, *Propositional proof complexity and unsolvability of polynomial equations*, Proceedings of the International Congress of Mathematicians. Vol. III. Sections 10–19. Held in Berlin, August 18–27, 1998, 1998, pp. 215–244.
- [Raz87] Alexander A. Razborov, *Lower bounds on the dimension of schemes of bounded depth in a complete basis containing the logical addition function*, Mat. Zametki **41** (1987), no. 4, 598–607, 623, English translation: Mathematical Notes of the Academy of Sci. of the USSR, 41(4):333–338, 1987.
- [Rei95] Miles Reid, *Undergraduate commutative algebra*, London Mathematical Society Student Texts, vol. 29, Cambridge University Press, Cambridge, 1995.
- [RR97] Alexander A. Razborov and Steven Rudich, *Natural proofs*, J. Comput. System Sci. **55** (1997), no. 1, part 1, 24–35.
- [RT08] Ran Raz and Iddo Tzameret, *The strength of multilinear proofs*, Comput. Complexity **17** (2008), no. 3, 407–457, A preliminary version appeared in ECCC, Tech. Report TR06-001, 2006.
- [SC04] Michael Soltys and Stephen Cook, *The proof complexity of linear algebra*, Ann. Pure Appl. Logic **130** (2004), no. 1–3, 277–323.
- [Sei74] Abraham Seidenberg, *Constructions in algebra*, Trans. Amer. Math. Soc. **197** (1974), 273–313.
- [Smo87] Roman Smolensky, *Algebraic methods in the theory of lower bounds for Boolean circuit complexity*, STOC ’87: 19th Annual ACM Symposium on Theory of Computing, ACM, 1987, pp. 77–82.
- [Som99] Martín Sombra, *A sparse effective Nullstellensatz*, Adv. in Appl. Math. **22** (1999), no. 2, 271–295.
- [Str73] Volker Strassen, *Vermeidung von Divisionen*, J. Reine Angew. Math. **264** (1973), 184–202.

- [Str73] Volker Strassen, *Die Berechnungskomplexität von elementarsymmetrischen Funktionen und von Interpolationskoeffizienten*, Numer. Math. **20** (1972/73), 238–251.
- [SY09] Amir Shpilka and Amir Yehudayoff, *Arithmetic circuits: a survey of recent results and open questions*, Found. Trends Theor. Comput. Sci. **5** (2009), no. 3–4, 207–388 (2010).
- [Tav13] Sébastien Tavenas, *Improved bounds for reduction to depth 4 and depth 3*, MFCS '13: Symposium on Mathematical Foundations of Computer Science (Krishnendu Chatterjee and Jirí Sgall, eds.), Lecture Notes in Computer Science, vol. 8087, Springer Berlin Heidelberg, 2013, pp. 813–824.
- [Tod92] Seinosuke Toda, *Classes of arithmetic circuits capturing the complexity of computing the determinant*, IEICE Trans. Inf. & Syst. **E75D** (1992), no. 1, 116–124.
- [Val79a] Leslie G. Valiant, *Completeness classes in algebra*, STOC '79: 11th Annual ACM Symposium on Theory of Computing, ACM, 1979, pp. 249–261.
- [Val79b] Leslie G. Valiant, *The complexity of computing the permanent*, Theoret. Comput. Sci. **8** (1979), no. 2, 189–201.
- [Val82] Leslie G. Valiant, *Reducibility by algebraic projections*, Enseign. Math. (2) **28** (1982), no. 3–4, 253–268.
- [VSBR83] Leslie G. Valiant, S. Skyum, S. Berkowitz, and Charles Rackoff, *Fast parallel computation of polynomials using few processors*, SIAM J. Comput. **12** (1983), no. 4, 641–644.
- [VV86] Leslie G. Valiant and Vijay V. Vazirani, *NP is as easy as detecting unique solutions*, Theoret. Comput. Sci. **47** (1986), no. 1, 85–93.
- [vzG87] Joachim von zur Gathen, *Feasible arithmetic computations: Valiant's hypothesis*, J. Symbolic Comput. **4** (1987), no. 2, 137–172.
- [Zip79] Richard Zippel, *Probabilistic algorithms for sparse polynomials*, Symbolic and algebraic computation (EUROSAM '79, Internat. Sympos., Marseille, 1979), Lecture Notes in Computer Science, vol. 72, Springer, Berlin, 1979, pp. 216–226.

A Additional Background

A.1 Algebraic Complexity

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 [Val79a, Val82] was the first to point out not only their value but 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 $f \equiv_p g$.

By analogy with Turing reductions or circuit reductions, Bürgisser [Bür00a] introduced the more general, but somewhat messier, notion of c-reduction (“c” for “computation”). 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 .

Despite its central role in computation, and the fact that $\text{VP} = \text{VNC}^2$ [VSBR83], the determinant is not known to be VP -complete. The determinant is VQP -complete (VQP is defined just like VP but with a quasi-polynomial bound on the size and degree of the circuits) under qp -projections (like p -projections, but with a quasi-polynomial bound). Weakly skew circuits help clarify the complexity of the determinant (see Malod and Portier [MP08] for some history of weakly skew circuits and for highlights of their utility). A circuit of fan-in at most 2 is *weakly skew* if for every multiplication gate g receiving inputs from gates g_1 and g_2 , at least one of the subcircuits C_i rooted at g_i is only connected to the rest of the circuit through g . In other words, for every multiplication gate, one of its two incoming factors was computed entirely and solely for the purpose of being used in that multiplication gate. Toda [Tod92] (see also Malod and Portier [MP08]) showed that a polynomial family $f = (f_n)$ is a p -projection of the determinant family (\det_n) if and only if f is computed by polynomial-size weakly skew circuits.

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

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

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

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 $\text{coNP} \subseteq \text{NP}$.

Cook and Reckhow [CR79] formalized proof systems for the language TAUT (all Boolean tautologies) in a slightly different way, although their definition is essentially equivalent to the one above. We prefer the above definition as it is consistent with definitions of interactive proofs.

Definition A.2. A *Cook–Reckhow proof system* is a polynomial-time function P' of just one input y , and whose range is the set of all yes instances of L . If $x \in L$, then any y such that $P'(y) = x$ is called a P' proof of x . P' must satisfy the following properties:

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

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

Intuitively, we think of P' as a procedure for verifying that y is a proof that some $x \in L$ and if so, it outputs x . (For all strings x that do not encode valid proofs, $P'(x)$ may just output some canonical $x_0 \in L$.) 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-R-eckhow 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 Merlin's string y as encoding a pair (x, y') , and on input $y = (x, y')$, P' runs P on the pair (x, y') . If P accepts, then $P'(y)$ outputs x , and if P rejects, then $P'(y)$ outputs (the encoding of) a canonical x^0 in L . Conversely, say that P' is a Cook-Reckhow proof system for L . $P(x, y)$ runs P' on y and accepts if and only if $P'(y) = x$.

Definition A.3. Let P_1 and P_2 be two proof systems for a language L in coNP . P_1 p-simulates P_2 if for every $x \in L$ and for every y such that $P_2(x, y) = 1$, there exists y' such that $|y'| \leq \text{poly}(|y|)$, and $P_1(x, y') = 1$.

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 A.4. Let P_1 and P_2 be two proof systems for a language L in coNP . P_1 and P_2 are 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: $B_1, \dots, B_n \implies B$, where B_1, \dots, 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 $A, \neg A \vee B \implies B$ 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 3DNF formula f is a sequence of formulas f_1, \dots, f_m such that each f_i is either an instance of an axiom scheme, or follows from two 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 70's (REF) 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 proofs that are Frege proofs but with the additional restriction that each formula in the proof has bounded depth. (Because our connectives are binary 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 in AC^0 .

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 . Whereas polynomial-size Frege proofs allow a polynomial number of lines, each of which must be a polynomial-sized formula, using the new axiom, polynomial-size EF proofs allow a polynomial number of lines, each of which can be a polynomial-sized circuit. See [Kra95] 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 Merlin–Arthur style proof systems.

Definition A.5. Let L be a language in coNP , and let V be a probabilistic polynomial-time algorithm with two inputs $x, y \in \{0, 1\}^*$. (We think of V as the verifier.) V is a *probabilistic proof system* for L if:

1. (Perfect Soundness) For every x that is not in L , and for every y ,

$$\Pr_r[P(x, y) = 1] = 0,$$

where the probability is over the random coin tosses, r of P .

2. (Completeness) For every x in L , there exists a y such that

$$\Pr_r[P(x, y) = 1] \geq 3/4.$$

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

It is clear that for any coNP-complete language L , there is a polynomially bounded probabilistic proof system for L if and only if $\text{coNP} \subseteq \text{MA}$.

Again we have chosen to define our probabilistic 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 probabilistic proof systems are essentially equivalent to a probabilistic Cook–Reckhow proof systems.

Definition A.6. A *probabilistic Cook–Reckhow proof system* is a probabilistic polynomial-time algorithm A (whose run time is independent of its random choices) such that

1. There is a surjective function $f: \Sigma^* \rightarrow \text{TAUT}$ 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 a tautology.

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

We note that both Pitassi’s algebraic proof system [Pit96] 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 coRP algorithm for polynomial identity testing. The proof that Pitassi’s algebraic proof system is a probabilistic Cook–Reckhow system is essentially the same.

A.3 Commutative algebra

The following preliminaries from commutative algebra are needed only in Section 1.6 and Appendix B.

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., [AM69, Eis95]). A module over a field $R = \mathbb{F}$ is exactly a vector space over \mathbb{F} . Every ring R is naturally an R -module (using the ring multiplication for 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 A.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 $I_1 \subsetneq I_2 \subsetneq I_3 \subsetneq \dots$. 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 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 $M_1 \subsetneq M_2 \subsetneq M_3 \subsetneq \dots$. If R is Noetherian as a ring, then it is Noetherian as an R -module. It is easily verified that 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.

The remaining preliminaries from commutative algebra are only needed in Appendix B.

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 intersection of the primes minimal over I .

An *algebraic set* in \mathbb{F}^n is any set of the form $\{\vec{x} \in \mathbb{F}^n : F_1(\vec{x}) = \dots = F_m(\vec{x}) = 0\}$, which we denote $V(F_1, \dots, F_m)$ (“ V ” for “variety”). The algebraic set $V(F_1, \dots, F_m)$ depends only on the ideal $\langle F_1, \dots, F_m \rangle$, and even its radical, in the sense that $V(F_1, \dots, F_m) = V(\sqrt{\langle F_1, \dots, F_m \rangle})$. 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 $ab \in U$ —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, 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 $R \setminus P$, it is common usage to refer to the localization of R and 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, even if R is finitely generated 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 being merely finitely generated).

B 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 [Str73]. In the case we discuss here, the certificate itself is no longer a polynomial but is a rational function.

Example B.1. The inversion principle, one of the so-called “Hard Matrix Identities” [SC04], 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 last 10 years that they were shown to have efficient Extended Frege proofs [SC04], and it was quite nontrivial to show that they have efficient NC^2 -Frege proofs [HT12], 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.

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 $XY - I$, we get n^2 certificates from the entries of $X^{-1}QX$. 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 the main focus of the paper [HT12]; thus, if we had a proof system that allowed divisions in this manner, 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, $\det(X) \neq 0$).

In order 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, this would yield an unsound proof system, because then from any polynomials $F_1(\vec{x}), \dots, F_m(\vec{x})$ we could derive *any* other polynomial $G(\vec{x})$ via the false “certificate” $\frac{G(\vec{x})}{F(\vec{x})}y_1$. The following definition is justified by Proposition B.3.

Unfortunately, although we try to eschew as many definitions as possible, the results here are made much cleaner by using some additional (standard) terminology from commutative algebra which is covered in Appendix A.3 such as prime ideals, irreducible components of algebraic sets, and localization of rings.

Definition B.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 $F_1(\vec{x}), \dots, F_m(\vec{x})$ 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 $\langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$ (We give a more elementary explanation of this condition below),
 1. $C(x_1, \dots, x_n, \vec{0}) = 0$, and
 2. $C(x_1, \dots, x_n, F_1(\vec{x}), \dots, F_m(\vec{x})) = G(\vec{x})$.

A *RIPS proof* that $G(\vec{x})$ is in the radical of the ideal $\langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$ is an \mathbb{F} -algebraic circuit with divisions on inputs $x_1, \dots, x_n, y_1, \dots, y_m$ 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 $V(\langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle) \subseteq \overline{\mathbb{F}}^n$. 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-certificate but there are no IPS-certificates (see Example B.4).

The following proposition establishes that Definition B.2 indeed defines a sound proof system.

Proposition B.3. *If there is a RIPS-certificate that $G(\vec{x})$ is in the radical of $\langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$, then $G(\vec{x})$ is in fact in the radical of $\langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$.*

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 $\sqrt{\langle F_1, \dots, F_m \rangle}$, 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 $\langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$ (recall Definition 1.20). Let $D_F(\vec{x}) = D(\vec{x}, \vec{F}(\vec{x}))$.

Geometric proof: since $G(\vec{x})D_F(\vec{x}) \in \langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$, GD_F must vanish identically on every irreducible component of the algebraic set $V(F_1, \dots, F_m)$. 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 $V(F_1, \dots, F_m)$, it vanishes on $V(F_1, \dots, F_m)$ itself, and by the Nullstellensatz, $G \in \sqrt{\langle F_1, \dots, F_m \rangle}$.

Algebraic proof: for each prime ideal $P_i \subseteq \overline{\mathbb{F}}[\vec{x}]$ that is minimal subject to containing $\langle F_1, \dots, F_m \rangle$, D_F is not in P_i , by the definition of RIPS-certificate. Since $GD_F \in \langle F_1, \dots, F_m \rangle \subseteq P_i$, 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 $P_i \supseteq \langle F_1, \dots, F_m \rangle$. This intersection is exactly the radical $\sqrt{\langle F_1, \dots, F_m \rangle}$. \square

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 B.4. Let $G(x_1, x_2) = x_1$, $F_1(\vec{x}) = x_1^2$, $F_2(\vec{x}) = x_1x_2$. 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 $\langle F_1, F_2 \rangle$: $\langle F_1, F_2 \rangle = \{x(H_1(\vec{x})x_1 + H_2(\vec{x})x_2)\}$ 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 B.5. *If $C = C'/D$ is a RIPS proof that $G(\vec{x}) \in \sqrt{\langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle}$, such that $D_F(\vec{x})$ does not vanish anywhere on the algebraic set $V(F_1(\vec{x}), \dots, F_m(\vec{x}))$, then $G(\vec{x})$ is in fact in the ideal $\langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$. Furthermore, there is an IPS proof that $G(\vec{x}) \in \langle F_1(\vec{x}), \dots, F_m(\vec{x}) \rangle$ of size $\text{poly}(|C|, |E|)$ where E is an IPS proof of the unsolvability of $D_F(\vec{x}) = F_1(\vec{x}) = \dots = F_m(\vec{x}) = 0$.*

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

Geometric proof idea: The preceding means that when restricted to the algebraic set $V(F_1, \dots, F_m)$, D_F has a multiplicative inverse Δ . Rather than dividing by D , we then multiply by Δ , which, for points on $V(F_1, \dots, F_m)$, 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 $E'(\vec{x}, \vec{y}) \in \langle y_1, \dots, y_m \rangle$, 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 $\langle F_1, \dots, F_m \rangle$. The idea is then to multiply by Δ instead of dividing by D . More precisely, the following is an IPS-proof that $G \in \langle F_1, \dots, F_m \rangle$:

$$C_\Delta(\vec{x}, \vec{y}) \stackrel{\text{def}}{=} C'(\vec{x}, \vec{y})\Delta(\vec{x}, \vec{y}, D(\vec{x}, \vec{y})) + G(\vec{x})E'(\vec{x}, \vec{y}). \quad (4)$$

Since C' and E' must individually be in $\langle y_1, \dots, y_m \rangle$, 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 [Str73], 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 [Str73]. Combining these, we then easily construct a circuit for the IPS-certificate C_Δ of size $\text{poly}(|C|, |E|)$. \square

Example B.6. Returning to the inversion principle, we find that the certificate from Example B.1 only divided by $\det(X)$, which we’ve 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 at 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 Hrubes and Tzameret provide [HT12].

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 $\text{GL}_n(\mathbb{F})$, which is an irreducible variety.)

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

Example B.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., [Eis95, 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 B.8. A RIPS certificate is *Hilbert-like* if the denominator doesn’t involve the place-holder 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 B.9. *If there is a RIPS certificate that $G \in \sqrt{\langle F_1, \dots, F_m \rangle}$, 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, we replace the denominator by $D_F(\vec{x}) = D(\vec{x}, \vec{F}(\vec{x}))$. Next, for each monomial appearing in C' , we 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. \square

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 2.1.

Corollary B.10. *If $C = C'/D$ is a RIPS proof that $G \in \sqrt{\langle F_1, \dots, F_m \rangle}$, where the numerator C' satisfies the same sparsity condition as in Proposition 2.1, then there is a Hilbert-like RIPS proof that $G \in \sqrt{\langle F_1, \dots, F_m \rangle}$, of size $\text{poly}(|C|)$.*

Proof. We follow the proof of Lemma B.9, making each step effective. As in the last paragraph of the proof of Proposition B.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 2.1. The resulting circuit now computes a Hilbert-like RIPS certificate. \square

B.1 Towards lower bounds

We begin by noting that, since the numerator and denominator can be computed separately (originally due to Strassen [Str73], see the proof of Proposition B.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 1.6), the set of RIPS certificates showing that $G \in \sqrt{\langle F_1, \dots, F_m \rangle}$ is a coset of a finitely generated ideal.

Lemma B.11. *The set of RIPS-certificates showing that $G \in \sqrt{\langle F_1, \dots, F_m \rangle}$ 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 $\langle F_1, \dots, F_m \rangle$.*

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 1.12, 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. \square

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 $R' = \mathbb{F}[\vec{x}]_{P_1 \cup \dots \cup P_k}$.

However, somewhat surprisingly, we seem to be able to go further in the case of RIPS than IPS, as follows. In general, the ring $\mathbb{F}[\vec{x}, \vec{y}]_{P_1 \cup \dots \cup P_k}$ is a Noetherian *semi-local* ring, that is, in addition to being Noetherian, it has finitely many maximal ideals, namely P_1, \dots, P_k . Ideals in and modules over semi-local rings 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., [Eis95, Corollary 4.8] or [Rei95, Section 2.8]). Once 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, \dots, 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 B.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 2.3), 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).

C Geometric IPS-certificates

We may consider $F_1(x_1, \dots, x_n), \dots, F_m(x_1, \dots, x_n)$ as a polynomial map $F = (F_1, \dots, F_m): \mathbb{F}^n \rightarrow \mathbb{F}^m$. Then this system of polynomials has a common zero if and only if 0 is the image of F . In fact, we show that for any Boolean system of equations, which are those that include $x_1^2 - x_1 = \dots = x_n^2 - x_n = 0$, or multiplicative Boolean equations—those that include $x_1^2 - 1 = \dots = x_n^2 - 1 = 0$ —the system of polynomials has a common zero if and only if 0 is in the *closure* of the image of F .

The preceding is the geometric picture we pursue in this section; next we describe the corresponding algebra. The set of IPS certificates is the intersection of the ideal $\langle y_1, \dots, y_m \rangle$ with the coset $1 + \langle y_1 - F_1(\vec{x}), \dots, y_m - F_m(\vec{x}) \rangle$. The map $a \mapsto 1 - a$ is a bijection between this coset intersection and the coset intersection $(1 + \langle y_1, \dots, y_m \rangle) \cap \langle y_1 - F_1(\vec{x}), \dots, y_m - F_m(\vec{x}) \rangle$. In particular, the system of equations $F_1 = \dots = F_m = 0$ 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 0 is not even in the closure of the image of F . Hence we call such polynomials “geometric certificates.”

Definition C.1 (The Geometric Ideal Proof System). A *geometric IPS certificate* that a system of \mathbb{F} -polynomial equations $F_1(\vec{x}) = \dots = F_m(\vec{x}) = 0$ is unsatisfiable over $\overline{\mathbb{F}}$ is a polynomial $C \in \mathbb{F}[y_1, \dots, y_m]$ such that

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

A *geometric IPS proof* of the unsatisfiability of $F_1 = \dots = F_m = 0$, or a *geometric IPS refutation* of $F_1 = \dots = F_m = 0$, is an \mathbb{F} -algebraic circuit on inputs y_1, \dots, 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 most the smallest circuit size of any algebraic certificate. We do not know, however, if these complexity measures are polynomially related:

Open Question C.2. For Boolean systems of equations, 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 doesn't 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 (usual or multiplicative) geometric certificates always exist. In fact, this holds for any system of equations which contains at least one polynomial containing only the variable x_i , for each variable x_i :

Proposition C.3. *Let \mathbb{F} be either a (topologically) dense subfield of \mathbb{C} or any algebraically closed field. A Boolean system of equations over \mathbb{F} —or more generally any system of equations containing, for each variable x_i , at least one non-constant equation involving only x_i ⁷—has a common root if and only if it does not have a geometric certificate.*

The condition of this proposition is almost surely more stringent than necessary, but the next example shows that at least some condition is necessary.

Example C.4. Let $F_1(x, y) = xy - 1$ and $F_2(x, y) = x^2y$. There is no solution to $F_1 = F_2 = 0$, as $F_1 = 0$ implies that both x and y are nonzero, but if this is the case then $x^2y = F_2(x, y)$ is also nonzero. Yet 0 is in the closure of the image of the map $F = (F_1, F_2): \mathbb{F}^2 \rightarrow \mathbb{F}^2$. There are (at least) two ways to see this. First, we exhibit 0 as an explicit limit of points in the image. Let $\chi_1(\varepsilon) = \varepsilon$ and $\chi_2(\varepsilon) = 1/\varepsilon$. Then $F_1(\chi_1(\varepsilon), \chi_2(\varepsilon)) = 0$ identically in ε , and $F_2(\chi_1(\varepsilon), \chi_2(\varepsilon)) = \varepsilon$. Thus, if we take the limit as $\varepsilon \rightarrow 0$, we find that 0 is in the closure of the image of F .⁸

Alternatively, in this case we can determine the entire image exactly (usually a very daunting task): it is $\{(a, b) \in \mathbb{F}^2 : a \neq -1 \text{ and } b \neq 0\} \cup \{(-1, 0)\}$. This can be determined by solving the equations by the elementary method of substitution, and careful but not complicated case analysis. It is then clear (geometrically in the case of subfields of \mathbb{C} , and by a dimension argument over an arbitrary algebraically closed field) that the closure of the image is the entirety of \mathbb{F}^2 , and in particular contains 0.

The next example rules out another natural attempt at generalizing Proposition C.3, and also shows that the existence of geometric certificates for a given set of equations can depend on the equations themselves, and not just on the ideal they generate.

⁷We believe that the “correct” generalization here is to systems of equations $F_1 = \dots = F_m = 0$ such that the corresponding map $F: \mathbb{F}^n \rightarrow \text{Im}(F)$ is flat (see, e.g., [Eis95, Chapter 6]) and has zero-dimensional fibers, that is, the inverse image of any point is a finite set. Systems satisfying the hypothesis of Proposition C.3 satisfy these hypotheses as well, but we have not checked carefully if the result extends in this generality.

⁸If \mathbb{F} is a dense subfield of \mathbb{C} , this limit may be taken in the usual sense of the Euclidean topology. For arbitrary algebraically closed fields \mathbb{F} , the same construction works, but must now be interpreted in the context of Lemma C.7.

Example C.5. Let $F_1(x, y) = xy - 1$ and $F_2(x, y) = x^2y$ as before, and now also add $F_3(x, y) = x^2(1 - y)$. We already saw that $F_1 = F_2 = 0$ is unsatisfiable, so $F_1 = F_2 = F_3 = 0$ is unsatisfiable as well. However, $F_1 = F_3 = 0$ has one, and only one, solution, namely $x = y = 1$. Let $F = (F_1, F_2, F_3): \mathbb{F}^2 \rightarrow \mathbb{F}^3$. To see that $\vec{0}$ is in the closure of the image of F , we again consider $\lim_{\varepsilon \rightarrow 0} F(\varepsilon, 1/\varepsilon)$. As before $F_1(\varepsilon, 1/\varepsilon) = 0$ and $F_2(\varepsilon, 1/\varepsilon) = \varepsilon$, whose limit is zero as $\varepsilon \rightarrow 0$. Similarly, we get $F_3(\varepsilon, 1/\varepsilon) = \varepsilon^2(1 - 1/\varepsilon) = \varepsilon(\varepsilon - 1)$, which again goes to 0 as $\varepsilon \rightarrow 0$.

Note that if we replace equations F_1 and F_3 by another set of equations with the same set of solutions (in this case, a singleton set), but satisfying the conditions of Proposition C.3, such as $F'_1 = (x - 1)^k$ and $F'_3 = (y - 1)^\ell$ for some $k, \ell > 0$, then $\vec{0}$ is no longer in the closure of the image. For if (F'_1, F_2, F'_3) approaches $(0, 0, 0)$, then x and y must both approach 1, but then $F_2 = x^2y$ also approaches 1. Furthermore, by the Nullstellensatz, for some $k, \ell > 0$, the polynomials $(x - 1)^k$ and $(y - 1)^\ell$ both in the ideal $\langle F_1, F_3 \rangle$. Thus, although the solvability of a system of equations is determined entirely by (the radical of) the ideal they generate, the geometry of the corresponding map—and even the existence of geometric certificates—can change depending on which elements of the ideal are used in defining the map.

The following lemma is the key to Proposition C.3.

Lemma C.6. *Let \mathbb{F} be (1) a dense subfield of \mathbb{C} (in the Euclidean topology), or (2) any algebraically closed field. Let $F_1(\vec{x}), \dots, F_m(\vec{x})$ be a system of equations over \mathbb{F} , and let $F = (F_1, \dots, F_m): \mathbb{F}^n \rightarrow \mathbb{F}^m$ be the associated polynomial map, as above. If, for $i = 1, \dots, n$, $F_i(\vec{x})$ is a nonzero function of x_i alone, then the set of equations $F_1 = \dots = F_m = 0$ has a solution if and only if 0 is in the closure $\overline{\text{Im}(F)}$.*

Proof. If the system F has a common solutions, then 0 is in the image of F and hence in its closure.

Conversely, suppose 0 is in the closure of the image of F . We first prove case (1) (the characteristic zero case) as it is somewhat simpler and gives the main idea, and then we prove case (2), the case of an arbitrary algebraically closed field.

(1) 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., [Mum76, 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 aforementioned result for \mathbb{C} , there is a sequence of points $\vec{x}_1, \vec{x}_2, \dots \in \mathbb{C}^n$ such that $\vec{y} = \lim_{k \rightarrow \infty} F(\vec{x}_k)$. As \mathbb{F} is dense in \mathbb{C} in the Euclidean topology, there is similarly a sequence of points $\vec{x}'_1, \vec{x}'_2, \dots \in \mathbb{F}^n$ such that $|\vec{x}_k - \vec{x}'_k| \leq 1/k$ for all k . Hence $\lim_{k \rightarrow \infty} \vec{x}_k = \lim_{k \rightarrow \infty} \vec{x}'_k$. Each $F(\vec{x}'_k) \in \mathbb{F}^m$, so we get a sequence of points $F(\vec{x}'_1), F(\vec{x}'_2), \dots \in \mathbb{F}^m$ whose limit is \vec{y} .

In particular, 0 is in the (Zariski-)closure of the image of F if and only if there is a sequence of points $v^{(1)}, v^{(2)}, v^{(3)}, \dots \in \text{Im}(F)$ such that $\lim_{k \rightarrow \infty} v^{(k)} = 0$. As each $v^{(k)}$ is in the image of F , there is some point $\nu^{(k)} \in \mathbb{F}^n$ such that $v^{(k)} = F(\nu^{(k)})$. As the $v^{(k)}$ approach the origin, each $F_i(\nu^{(k)})$ approaches 0, since it is the i -th coordinate of $v^{(k)}$: $v_i^{(k)} = F_i(\nu^{(k)})$.

In particular, since $F_1(\vec{x})$ depends only on x_1 and is nonzero (by assumption), the first coordinates $\nu_1^{(k)}$ must accumulate around the finitely many zeroes of $F_1(x_1)$. Similarly for each coordinate $i = 1, \dots, n$ of $\nu^{(k)}$. Thus there is an infinite subsequence of the $\nu^{(k)}$ that approaches one single solution \vec{z} to $F = 0$. By choosing such a subsequence and re-indexing, we may assume that $\lim_{k \rightarrow \infty} \nu^{(k)} = \vec{z}$.

Finally, by assumption and continuity, we have

$$0 = \lim_{k \rightarrow \infty} v^{(k)} = \lim_{k \rightarrow \infty} F(\nu^{(k)}) = F\left(\lim_{k \rightarrow \infty} \nu^{(k)}\right) = F(\vec{z}),$$

so \vec{z} is a common root of the original system $F_1 = \dots = F_m = 0$. Hence, if 0 is in the closure of the image of F , then 0 is in the image.

(2) \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 C.7 (See, e.g., [BCS97, Lemma 20.28]). *If p is a point in the closure of the image of a polynomial map $F: \mathbb{F}^n \rightarrow \mathbb{F}^m$, then there are formal Laurent series⁹ $\chi_1(\varepsilon), \dots, \chi_n(\varepsilon)$ in a new variable ε such that $F_i(\chi_1(\varepsilon), \dots, \chi_n(\varepsilon))$ is in fact a power series—that is, involves no negative powers of ε —for each $i = 1, \dots, m$, and such that evaluating the power series $(F_1(\vec{\chi}(\varepsilon)), \dots, F_m(\vec{\chi}(\varepsilon)))$ 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 .

As F_1 involves only x_1 , in order for $F_1(\vec{\chi}(\varepsilon)) = F_1(\chi_1(\varepsilon))$ to be a power series in ε , it must be the case that $\chi_1(\varepsilon)$ itself is a power series (contains no negative powers of ε). For if the highest degree term of F_1 is some constant times x_1^d , and the lowest degree term of $\chi_1(\varepsilon)$ is of degree $-D$, then $F_1(\chi_1(\varepsilon))$ contains the monomial ε^{-dD} with nonzero coefficient. A similar argument applies to χ_i for $i = 1, \dots, n$. Thus each χ_i is in fact a power series, involving no negative terms of ε , and hence can be evaluated at 0. Since evaluating at $\varepsilon = 0$ now makes sense even before applying the F_i , and is a ring homomorphism (we might say, “is continuous with respect to the ring operations”), we get that

$$0 = F_i(\vec{\chi}(\varepsilon))|_{\varepsilon=0} = F_i(\vec{\chi}(\varepsilon)|_{\varepsilon=0}) = F_i(\vec{\chi}(0))$$

for each $i = 1, \dots, m$, and hence $\vec{\chi}(0)$ is a solution to $F_1(\vec{x}) = \dots = F_m(\vec{x}) = 0$. \square

Proof of Proposition C.3. Let F_1, \dots, F_m be an unsatisfiable system of equations over \mathbb{F} satisfying the conditions of Lemma C.6, and let $F = (F_1, \dots, F_m): \mathbb{F}^n \rightarrow \mathbb{F}^m$ be the corresponding polynomial map.

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

Conversely, suppose $C(y_1, \dots, y_m)$ is a geometric certificate. Then C vanishes at every point of the image $\text{Im}(F)$ and hence at every point of its closure $\overline{\text{Im}(F)}$, by (Zariski-)continuity. By condition (1) of Definition C.1, $C(0, \dots, 0) = 1$. Since C does not vanish at the origin, $\vec{0} \notin \text{Im}(F)$. Then by Lemma C.6, $\vec{0}$ is not in the image of F and hence $F_1 = \dots = F_m = 0$ has no solution. \square

Finally, as with IPS certificates and Hilbert-like IPS certificates (see Section 1.6), a *geometric zero-certificate* for a system of equations $F_1(\vec{x}), \dots, F_m(\vec{x})$ is a polynomial $C(y_1, \dots, y_m) \in \langle y_1, \dots, y_m \rangle$ —that is, such that $C(0, \dots, 0) = 0$ —and such that $C(F_1(\vec{x}), \dots, F_m(\vec{x})) = 0$ 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 $\langle y_1, \dots, y_m \rangle \cap \langle y_1 - F_1(\vec{x}), \dots, y_m - F_m(\vec{x}) \rangle$ 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.

⁹A 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.

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” we discussed above in relation to Hilbert-like IPS. As in all the other cases we’ve 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 $\langle y_1 - F_1(\vec{x}), \dots, y_m - F_m(\vec{x}) \rangle$ 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.