Proof Complexity Lower Bounds from Algebraic Circuit Complexity*

Michael A. Forbes[1], Amir Shpilka[2], Iddo Tzameret[3], and Avi Wigderson[4]

1 **Department of Computer Science, Princeton University, Princeton, USA**
miforbes@csail.mit.edu

2 **Department of Computer Science, Tel Aviv University, Tel Aviv, Israel**
shpilka@post.tau.ac.il

3 **Department of Computer Science, Royal Holloway, University of London, Egham, UK**
iddo.tzameret@rhul.ac.uk

4 **School of Mathematics, Institute for Advanced Study, Princeton, USA**
avi@math.ias.edu

Abstract

We give upper and lower bounds on the power of subsystems of the Ideal Proof System (IPS), the algebraic proof system recently proposed by Grochow and Pitassi [26], where the circuits comprising the proof come from various restricted algebraic circuit classes. This mimics an established research direction in the boolean setting for subsystems of Extended Frege proofs whose lines are circuits from restricted boolean circuit classes. Essentially all of the subsystems considered in this paper can simulate the well-studied Nullstellensatz proof system, and prior to this work there were no known lower bounds when measuring proof size by the algebraic complexity of the polynomials (except with respect to degree, or to sparsity).

Our main contributions are two general methods of converting certain algebraic lower bounds into proof complexity ones. Both require stronger arithmetic lower bounds than common, which should hold not for a specific polynomial but for a whole family defined by it. These may be likened to some of the methods by which Boolean circuit lower bounds are turned into related proof-complexity ones, especially the "feasible interpolation" technique. We establish algebraic lower bounds of these forms for several explicit polynomials, against a variety of classes, and infer the relevant proof complexity bounds. These yield separations between IPS subsystems, which we complement by simulations to create a partial structure theory for IPS systems.

Our first method is a *functional lower bound*, a notion of Grigoriev and Razborov [25], which is a function $\hat{f} : \{0,1\}^n \to \mathbb{F}$ such that any polynomial f agreeing with $\hat{f}$ on the boolean cube requires large algebraic circuit complexity. We develop functional lower bounds for a variety of circuit classes (sparse polynomials, depth-3 powering formulas, read-once algebraic branching programs and multilinear formulas) where $\hat{f}(\vec{x})$ equals $1/p(\vec{x})$ for a constant-degree polynomial p depending on the relevant circuit class. We believe these lower bounds are of independent interest in algebraic complexity, and show that they also imply lower bounds for the size of the corresponding IPS refutations for proving that the relevant polynomial p is non-zero over the boolean cube. In particular, we show super-polynomial lower bounds for refuting variants of the subset-sum axioms in these IPS subsystems.

Our second method is to give *lower bounds for multiples*, that is, to give explicit polynomials whose all (non-zero) multiples require large algebraic circuit complexity. By extending known techniques, we give lower bounds for multiples for various restricted circuit classes such

* The research leading to these results has received funding from the Princeton Center for Theoretical Computer Science, the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement number 257575, and NSF grant CCF-1412958.

31st Conference on Computational Complexity (CCC 2016).
Editor: Ran Raz; Article No. 32; pp. 32:1–32:17
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

sparse polynomials, sums of powers of low-degree polynomials, and roABPs. These results are of independent interest, as we argue that lower bounds for multiples is the correct notion for instantiating the algebraic hardness versus randomness paradigm of Kabanets and Impagliazzo [31]. Further, we show how such lower bounds for multiples extend to lower bounds for refutations in the corresponding IPS subsystem.

1998 ACM Subject Classification F.2.2 Nonnumerical Algorithms and Problems – Complexity of Proof Procedures, F.2.1 Numerical Algorithms and Problems, I.1.1 Expressions and their Representation

Keywords and phrases Proof Complexity, Algebraic Complexity, Nullstellensatz, Subset-Sum

Digital Object Identifier 10.4230/LIPIcs.CCC.2016.32

1 Introduction

Propositional proof complexity aims to understand and analyze the computational resources required to prove propositional tautologies, in the same way that circuit complexity studies the resources required to compute boolean functions. A typical goal would be to establish, for a given proof system, super-polynomial lower bounds on the *size* of any proof of some propositional tautology. The seminal work of Cook and Reckhow [13] showed that this goal relates quite directly to fundamental hardness questions in computational complexity such as the NP vs. coNP question: establishing super-polynomial lower bounds for *every* propositional proof system would separate NP from coNP (and thus also P from NP). We refer the reader to Krajíček [35] for more on this subject.

Propositional proof systems come in a large variety, as different ones capture different forms of reasoning, either reasoning used to actually prove theorems, or reasoning used by algorithmic techniques for different types of search problems (as failure of the algorithm to find the desired object constitutes a proof of its nonexistence). Much of the research in proof complexity deals with propositional proof systems originating from logic and from geometry. Logical proof systems include such systems as *resolution* (whose variants are related to popular algorithms for automated theory proving and SAT solving), as well as the *Frege* proof system (capturing the most common logic text-book systems) and its many subsystems. Geometric proof systems include *cutting-plane proofs*, capturing reasoning used in algorithms for integer programming, as well as proof systems arising from systematic strategies for rounding linear- or semidefinite-programming such as the lift-and-project or sum-of-squares hierarchies.

In this paper we focus on algebraic proof systems, in which propositional tautologies (or rather contradictions) are expressed as unsatisfiable systems of polynomial equations and algebraic tools are used to refute them. This study originates with the work of Beame, Impagliazzo, Krajíček, Pitassi and Pudlák [6], who introduced the Nullstellensatz refutation system (based on Hilbert's Nullstellensatz), followed by the Polynomial Calculus system of Clegg-Edmonds-Impagliazzo [10], which is a "dynamic version" of Nullstellensatz. In both systems the main measures of proof size that have been studied are the *degree* and *sparsity* of the polynomials appearing in the proof. Substantial work has lead to a very good understanding of the power of these systems with respect to these measures (see for example [9, 50, 23, 30, 8, 4] and references therein).

However, the above measures of degree and sparsity are rather rough measures of a complexity of a proof. As such, Grochow and Pitassi [26] have recently advocated measuring the complexity of such proofs by their algebraic circuit size and shown that the resulting proof system can polynomially simulate strong proof systems such as the Frege system. This naturally leads to the question of establishing lower bounds for this stronger proof system, even for restricted classes of algebraic circuits.

In this work we establish such lower bounds for previously studied restricted classes of algebraic circuits, and show these lower bounds are interesting by providing non-trivial *upper* bounds in these proof systems for refutations of interesting sets of polynomial equations. This provides what are apparently the first examples of lower bounds on the algebraic circuit size of propositional proofs in the ideal proof system (IPS) framework of Grochow and Pitassi [26].

We note that obtaining proof complexity lower bounds from circuit complexity lower bounds is an established tradition, and takes many forms. Most prominent are the lower bounds for susbsystems of the Frege proof system defined by low-depth Boolean circuits, and lower bounds on Resolution and Cutting Planes system using the so-called feasible interpolation method [44]. We refer the reader again to the monograph [35] for more details. Our approach here for algebraic systems shares features with both of these approaches.

The rest of this long introduction is arranged as follows. In Subsection 1.1 we give the necessary background in algebraic proof complexity, and explain the IPS system. In subsection 1.2 we define the algebraic complexity classes that will underlie the subsystems of IPS we will study. In subsection 1.3 we state our results and explain our techniques, for both the algebraic and proof complexity worlds.

2 Algebraic Proof Systems

We now describe the algebraic proof systems that are the subject of this paper. If one has a set of polynomials (called *axioms*) $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ over some field $\mathbb{F}$, then (the weak version of) Hilbert's Nullstellensatz shows that the system $f_1(\vec{x}) = \cdots = f_m(\vec{x}) = 0$ is unsatisfiable (over the algebraic closure of $\mathbb{F}$) if and only if there are polynomials $g_1, \ldots, g_m \in \mathbb{F}[\vec{x}]$ such that $\sum_j g_j(\vec{x}) f_j(\vec{x}) = 1$ (as a formal identity), or equivalently, that 1 is in the ideal generated by the $\{f_j\}_j$.

Beame, Impagliazzo, Krajíček, Pitassi, and Pudlák [6] suggested to treat these $\{g_j\}_j$ as a *proof* of the unsatisfiability of this system of equations, called a *Nullstellensatz refutation*. This is particular relevant for complexity theory as one can restrict attention to *boolean* solutions to this system by adding the *boolean axioms*, that is, adding the polynomials $\{x_i^2 - x_i\}_{i=1}^n$ to the system. As such, one can then naturally encode NP-complete problems such as the satisfiability of 3CNF formulas as the satisfiability of a system of constant-degree polynomials, and a Nullstellensatz refutation is then an equation of the form $\sum_{j=1}^m g_j(\vec{x}) f_j(\vec{x}) + \sum_{i=1}^n h_i(\vec{x})(x_i^2 - x_i) = 1$ for $g_j, h_i \in \mathbb{F}[\vec{x}]$. This proof system is sound (only refuting unsatisfiable systems over $\{0,1\}^n$) and complete (refuting any unsatisfiable system, by Hilbert's Nullstellensatz).

Given that the above proof system is sound and complete, it is then natural to ask what is its power to refute unsatisfiable systems of polynomial equations over $\{0,1\}^n$. To understand this question one must define the notion of the *size* of the above refutations. Two popular notions are that of the *degree*, and the *sparsity* (number of monomials). One can then show (see for example Pitassi [43]) that for any unsatisfiable system which includes the boolean axioms, there exist a refutation where the g_j are multilinear and where the h_i have degree at most $O(n + d)$, where each f_j has degree at most d. In particular, this implies, when $d = O(n)$, that for any unsatisfiable system there is a refutation of degree $O(n)$ and involving at most $\exp(O(n))$ monomials. This intuitively agrees with the fact that coNP is a subset of non-deterministic exponential time.

Building on the suggestion of Pitassi [43], Grochow and Pitassi [26] have recently considered more *succinct* descriptions of polynomials where one measures the size of a polynomial by the size of an algebraic circuit needed to compute it. This is potentially much more powerful as there are polynomials such as the determinant which are of high degree and involve exponentially many monomials and yet can be computed by small algebraic circuits. They named the resulting system the *Ideal Proof System (IPS)* which we now define.

▶ **Definition 2.1** (Ideal Proof System (IPS), Grochow-Pitassi [26]). Let $f_1(\vec{x}), \ldots, f_m(\vec{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ be a system of polynomials. An **IPS refutation** for showing that the polynomials $\{f_j\}_j$ have no common solution in $\{0,1\}^n$ is an algebraic circuit $C(\vec{x}, \vec{y}, \vec{z}) \in \mathbb{F}[\vec{x}, y_1, \ldots, y_m, z_1, \ldots, z_n]$, such that

1. $C(\vec{x}, \vec{0}, \vec{0}) = 0$.
2. $C(\vec{x}, f_1(\vec{x}), \ldots, f_m(\vec{x}), x_1^2 - x_1, \ldots, x_n^2 - x_n) = 1$.

The **size** of the IPS refutation is the size of the circuit C. If C is of individual degree ≤ 1 in each y_j and z_i, then this is a **linear** IPS refutation (called *Hilbert* IPS by Grochow-Pitassi [26]), which we will abbreviate as IPS_{LIN} . If C is of individual degree ≤ 1 only in the y_j then we say this is a $\text{IPS}_{\text{LIN}'}$ refutation. If C comes from a restricted class of algebraic circuits $\mathcal{C}$, then this is a called a $\mathcal{C}$-IPS refutation, and further called a $\mathcal{C}$-IPS_{LIN} refutation if C is linear in $\vec{y}, \vec{z}$, and a $\mathcal{C}$-$\text{IPS}_{\text{LIN}'}$ refutation if C is linear in $\vec{y}$.

Notice also that our definition here adds the equations $\{x_i^2 - x_i\}_i$ to the system $\{f_j\}_j$. For convenience we will often denote the equations $\{x_i^2 - x_i\}_i$ as $\vec{x}^2 - \vec{x}$. One need not add the equations $\vec{x}^2 - \vec{x}$ to the system in general, but this is the most interesting regime for proof complexity and thus we adopt it as part of our definition.

though it is a complete refutation system for the standard polynomial translation of unsatisfiable CNFs) but that the $\text{IPS}_{\text{LIN}'}$ version is complete.

Grochow-Pitassi [26] proved the following theorem, showing that the IPS system has surprising power and that lower bounds on this system give rise to *computational* lower bounds.

▶ **Theorem 2.2** (Grochow-Pitassi [26]). *Let φ be a 3CNF. If there is an Extended Frege proof (Frege proof) that φ is unsatisfiable in size-s, then there is an IPS refutation of circuit (formula) size* $\mathsf{poly}(|\varphi|, s)$ *that is checkable in randomized* $\mathsf{poly}(|\varphi|, s)$ *time. Conversely, if every IPS refutation requires circuit (formula) size $\geq s$ then there is an explicit polynomial (that is, in* VNP*) that requires $\geq s$-size algebraic circuits (formulas).*[1]

▶ Remark. One point to note is that the transformation from Extended Frege to IPS refutations yields circuits of polynomial size but without any guarantee on their degree. In particular, such circuits can compute polynomials of exponential degree. In contrast, the conversion from Frege to IPS refutations yields polynomial sized algebraic formulas and those compute polynomials of polynomially bounded degree. This range of parameters, polynomials of polynomially bounded degree, is the more common setting studied in algebraic complexity.

The fact that $\mathcal{C}$-IPS refutations are efficiently checkable (with randomness) follows from the fact that we need to verify the polynomial identities stipulated by the definition. That is, one needs to solve an instance of the *polynomial identity testing (PIT)* problem for the class $\mathcal{C}$: given a circuit from the class $\mathcal{C}$ decide whether it computes the identically zero polynomial.

[1] We note that Grochow and Pitassi [26] proved this for Extended Frege and circuits, but essentially the same proof relates Frege and formula size.

This problem is solvable in probabilistic polynomial time (BPP) for general algebraic circuits, and there are various restricted classes for which deterministic algorithms are known.

Motivated by the fact that PIT of non-commutative formulas[2] can be solved deterministically ([47]) and admit exponential-size lower bounds ([38]), Li, Tzameret and Wang [37] have shown that IPS over *non-commutative* polynomials can simulate Frege (they also provided a quasipolynomial simulation of IPS over non-commutative formulas by Frege; see Li, Tzameret and Wang [37] for more details).

▶ **Theorem 2.3** (Li, Tzameret and Wang [37])**.** *Let φ be a 3CNF. If Frege can prove that φ is unsatisfiable in size-s, then there is a non-commutative IPS refutation of formula size* $\mathsf{poly}(|\varphi|, s)$ *computing a polynomial of degree* $\mathsf{poly}(|\varphi|, s)$*, where the commutator axioms $x_i x_j - x_j x_i$ are also included in the polynomial system being refuted. Further, this refutation is checkable in deterministic* $\mathsf{poly}(|\varphi|, s)$ *time.*

The above results naturally motivate studying $\mathcal{C}$-IPS for various restricted classes of algebraic circuits, as lower bounds for such proofs then intuitively correspond to restricted lower bounds for the Extended Frege proof system. In particular, as exponential lower bounds are known for non-commutative formulas ([38]), this possibly suggests that such methods could even attack the full Frege system itself.

3 Algebraic Circuit Classes

Having motivated $\mathcal{C}$-IPS for restricted circuit classes $\mathcal{C}$, we now give formal definitions of the algebraic circuit classes of interest to this paper, all of which were studied independently in algebraic complexity. Some of them define the state-of-art in our ability to prove lower bounds and provide efficient deterministic identity tests, so it is natural to attempt converting these to the proof complexity framework. We define each and briefly explain what we know about it. As the list is long, the reader may consider skipping to the results (Section 4), and refer to the definitions of these classes as they arise.

Algebraic circuits and formula (over a fixed chosen field) compute polynomials via addition and multiplication gates, starting from the input variables and constants from the field. For background on algebraic circuits in general and their complexity measures we refer the reader to the survey [54]. We next define the restricted circuit classes that we will be studying in this paper.

3.1 Low Depth Classes

We start by defining what are the simplest and most restricted classes of algebraic circuits. The first class simply represents polynomials as a sum of monomials. This is also called the *sparse representation* of the polynomial. Notationally we call this model $\sum\prod$ formulas (to capture the fact that polynomials computed in the class are represented simply as sums of products), but we will more often call these polynomials “sparse”.

▶ **Definition 3.1.** The class $\mathcal{C} = \sum\prod$ compute polynomials in their sparse representation, i.e., as sum of monomials. The graph of computation has two layers with an addition gate at the top and multiplication gates at the bottom. The size of a $\sum\prod$ circuit of a polynomial f is the number of monomials in f.

[2] These are formulas over a set of non-commuting variables.

This class of circuits is what is used in the Nullstellensatz proof system. In our terminology $\sum\prod$-IPS$_{\text{LIN}}$ is exactly the Nullstellensatz proof system.

Another restricted class of algebraic circuits is that of *depth-3 powering formulas* (sometimes also called "diagonal depth-3 circuits"). We will sometimes abbreviate this name as a "$\sum\bigwedge\sum$ formula", where $\bigwedge$ denotes the powering operation. Specifically, polynomials that are efficiently computed by small formulas from this class can be represented as sum of powers of linear functions. This model appears implicitly in Shpilka [53] and explicitly in the work of Saxena [52].

▶ **Definition 3.2.** The class of depth-3 powering formulas, denoted $\sum\bigwedge\sum$, computes polynomials of the following form

$$f(\vec{x}) = \sum_{i=1}^{s} \ell_i(\vec{x})^{d_i},$$

where $\ell_i(\vec{x})$ are linear functions. The degree of this $\sum\bigwedge\sum$ representation of f is $\max_i\{d_i\}$ and its size is $n \cdot \sum_{i=1}^{s}(d_i + 1)$.

One reason for considering this class of circuits is that it is a simple, but non-trivial model that is somewhat well-understood. In particular, the partial derivative method of Nisan-Wigderson [40] implies lower bounds for this model and efficient PIT algorithms are known ([52, 3, 21, 22, 19]).

We also consider a generalization of this model where we allow powering of low-degree polynomials.

▶ **Definition 3.3.** The class $\sum\bigwedge\sum\prod^t$ computes polynomials of the following form

$$f(\vec{x}) = \sum_{i=1}^{s} f_i(\vec{x})^{d_i} \ ,$$

where the degree of the $f_i(\vec{x})$ is at most t. The size of this representation is $\binom{n+t}{t} \cdot \sum_{i=1}^{s}(d_i+1)$.

We remark that the reason for defining the size this way is that we think of the f_i as represented as sum of monomials (there are $\binom{n+t}{t}$ n-variate monomials of degree at most t) and the size captures the complexity of writing this as an algebraic formula. This model is the simplest that requires the method of *shifted partial derivatives* of Kayal [34, 27] to establish lower bounds, and this has recently been generalized to obtain PIT algorithms ([16]).

3.2 Oblivious Algebraic Branching Programs

Algebraic branching programs (ABPs) form a model whose computational power lies between that of algebraic circuits and algebraic formulas, and certain *read-once* and *oblivious* ABPs are a natural setting for studying the *partial derivative matrix* lower bound technique of Nisan [38].

▶ **Definition 3.4** (Nisan [38]). An **algebraic branching program (ABP) with unrestricted weights** of depth D and width $\leq r$, on the variables $x_1, \ldots, x_n$, is a directed acyclic graph such that:

- The vertices are partitioned in $D + 1$ layers $V_0, \ldots, V_D$, so that $V_0 = \{s\}$ (s is the source node), and $V_D = \{t\}$ (t is the sink node). Further, each edge goes from V_{i-1} to V_i for some $0 < i \leq D$.

- $\max |V_i| \leq r$.
- Each edge e is weighted with a polynomial $f_e \in \mathbb{F}[\vec{x}]$.

Each s-t path is said to compute the polynomial which is the product of the labels of its edges, and the algebraic branching program itself computes the sum over all s-t paths of such polynomials.

- An algebraic branching program is said to be **oblivious** if for every layer ℓ, all the edge labels in that layer are univariate polynomials in a variable x_{i_ℓ}.
- An oblivious branching program is said to be a **read-once** oblivious ABP (roABP) if the x_{i_ℓ}'s are distinct variables, so that $D = n$. That is, each x_i appears in the edge labels in at exactly one layer. The layers thus define a **variable order**, which will be $x_1 < \cdots < x_n$ if not otherwise specified.
- An oblivious branching program is said to be a **read-k** oblivious ABP if each variable x_i appears in the edge labels of at most k layers, so that $D = kn$.
- An ABP is non-commutative if it is defined over the ring of non-commuting variables.

Intuitively, roABPs are the algebraic analog of read-once boolean branching program, the non-uniform model of the class RL. Nisan [38] proved lower bounds for non-commutative ABPs (and thus also for roABPs, in any order) and in a sequence of papers polynomial identity testing algorithms were devised for it ([47, 22, 19, 2]). Recently Anderson, Forbes, Saptharishi, Shpilka, and Volk [5] obtained exponential lower bounds for read-k oblivious ABPs (when $k = o(\log n/\log\log n)$) as well as a slightly subexponential PIT algorithm.

We note that roABPs are known to simulate non-commutative formulas ([38]). Thus, the result of Li, Tzameret and Wang [37] (see Theorem 2.3) demonstrates the importance of studying IPS proofs over roABPs (see also Tzameret [56]).

3.3 Multilinear Formulas

The last model that we consider is that of multilinear formulas.

▶ **Definition 3.5** (Multilinear formula)**.** An algebraic formula is *a multilinear formula* (or equivalently, *multilinear algebraic formula*) if the polynomial computed by *each* gate of the formula is multilinear (as a formal polynomial, that is, as an element of $\mathbb{F}[x_1, \ldots, x_n]$).

Raz [46, 45] proved quasi-polynomial lower bounds for multilinear formulas and separated multilinear formulas from multilinear circuits. Raz and Yehudayoff proved exponential lower bounds for small depth multilinear formulas [49]. Only slightly sub-exponential polynomial identity testing algorithms are known for small-depth multilinear formulas ([42]).

4 Our Results and Techniques

We now briefly summarize our results and techniques, stating some results in less than full generality to more clearly convey the result. We present the results in the order that we later prove them. We start by giving upper bounds for the IPS (Subsection 4.1). We then describe our functional lower bounds and the IPS_{LIN} lower bounds they imply (Subsection 4.2). Finally, we discuss lower bounds for multiples and state our lower bounds for IPS (Subsection 4.3).

4.1 Upper Bounds for Proofs within Subclasses of IPS

Grochow and Pitassi [26] showed that the full IPS proof system can simulate powerful proof systems such as Extended Frege. This left open the extent to which $\mathcal{C}$-IPS can refute interesting sets of polynomial equations for restricted classes $\mathcal{C}$. We establish here that even restricted classes of IPS are powerful, such as being able to refute interesting unsatisfiable systems of equations arising from particular instances of NP-complete problems.

Our first upper bound is to show that linear-IPS can simulate the full IPS proof system when the axioms are computationally simple.

▶ **Theorem 4.1.** *For $|\mathbb{F}| \ge \mathsf{poly}(d)$, if $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are degree-d polynomials computable by size-s algebraic circuits and they have a size-t IPS refutation, then they also have a size-$\mathsf{poly}(d, s, t)$ IPS$_{LIN}$ refutation.*

This theorem is established by pushing the "non-linear" dependencies on the axioms into the IPS refutation itself, which is possible as the axioms are assumed to themselves be computable by small circuits. We note that Grochow and Pitassi [26] showed such a conversion, but only for IPS refutations computable by sparse polynomials.

We then turn our attention to IPS involving only restricted classes of algebraic circuits, and show that they are complete proof systems. This is clear for complete models of algebraic circuits such as sparse polynomials, depth-3 powering formulas [3] and roABPs. For multilinear formulas this is more subtle as not every polynomial is multilinear, however we can show a simulation of sparse-IPS$_{\text{LIN}}$ by a careful multilinearization.

▶ **Theorem 4.2.** *The proof systems of sparse-IPS$_{LIN}$, $\sum\bigwedge\sum$-IPS$_{LIN}$ (in large characteristic fields), and roABP-IPS$_{LIN}$ are complete proof systems (for systems of polynomials with no boolean solutions). The multilinear-formula-IPS$_{LIN}$ proof system is not complete, but the depth-2 multilinear-formula-IPS$_{LIN'}$ proof system is complete (for multilinear axioms) and can polynomially simulate sparse-IPS$_{LIN}$ (for low-degree axioms). For standard polynomial* translation of CNFs*, multilinear-formula-IPS$_{LIN}$ is complete (even without using the boolean axioms).*

We next consider the equation $\sum_{i=1}^n \alpha_i x_i - \beta$ along with the boolean axioms $\{x_i^2 - x_i\}_i$. Deciding whether this system of equations is satisfiable is the NP-complete *subset-sum* problem, and as such we do not expect small refutations in general (unless NP = coNP). Indeed, Impagliazzo, Pudlák, and Sgall [30] have shown lower bounds for refutations in the *polynomial calculus* system (and thus also the Nullstellensatz system) even when $\vec{\alpha} = \vec{1}$. Specifically, they showed that such refutations require both $\Omega(n)$-degree and $\exp(\Omega(n))$-many monomials. In the language of this paper, they gave $\exp(\Omega(n))$-size lower bounds for refuting this system in $\sum\prod$-IPS$_{\text{LIN}}$ (which is equivalent to the Nullstellensatz proof system). In contrast, we establish here $\mathsf{poly}(n)$-size refutations for $\vec{\alpha} = \vec{1}$ in the restricted proof systems of roABP-IPS$_{\text{LIN}}$ and *depth-3* multilinear-formula-IPS$_{\text{LIN}}$ (despite the fact that multilinear-formula-IPS$_{\text{LIN}}$ is not complete).

▶ **Theorem 4.3.** *Let $\mathbb{F}$ be a field of characteristic $\operatorname{char}(\mathbb{F}) > n$. Then the system of polynomial equations $\sum_{i=1}^n x_i - \beta$, $\{x_i^2 - x_i\}_{i=1}^n$ is unsatisfiable for $\beta \in \mathbb{F} \setminus \{0, \ldots, n\}$, and there are explicit $\mathsf{poly}(n)$-size refutations within roABP-IPS$_{LIN}$, as well as within depth-3 multilinear-formula-IPS$_{LIN}$.*

This theorem is proven by noting that the polynomial $p(t) := \prod_{k=0}^n (t - k)$ vanishes on $\sum_i x_i$ modulo $\{x_i^2 - x_i\}_{i=1}^n$, but $p(\beta)$ is a non-zero constant. This implies that $\sum_i x_i - \beta$ divides $p(\sum_i x_i) - p(\beta)$. Denoting the quotient by $f(\vec{x})$, it follows that $\frac{1}{-p(\beta)} \cdot f(\vec{x}) \cdot (\sum_i x_i - \beta) \equiv 1$

[3] Showing that depth-3 powering formulas are complete (in large characteristic) can be seen from the fact that any multilinear monomial can be computed in this model, see for example Fischer [15].

mod $\{x_i^2 - x_i\}_{i=1}^n$, which is nearly a linear-IPS refutation except for the complexity of establishing this relation over the boolean cube. We show that the quotient f is easily expressed as a depth-3 powering circuit. Unfortunately, proving the above equivalence to 1 modulo the boolean cube is not possible in the depth-3 powering circuit model. However, by moving to more powerful models (such as roABPs and multilinear formulas) we can give proofs of this multilinearization to 1 and thus give proper IPS refutations.

4.2 Linear-IPS Lower Bounds via Functional Lower Bounds

Having demonstrated the power of various restricted classes of IPS refutations by refuting the subset-sum axiom, we now turn to lower bounds. We give two paradigms for establishing lower bounds, the first of which we discus here, named a *functional circuit lower bound.* This term appeared in the work of Grigoriev and Razborov [25] as well as in the recent work of Forbes, Kumar and Saptharishi [18]. We briefly motivate this type of lower bound as a topic of independent interest in algebraic circuit complexity, and then discuss the lower bounds we obtain and their implications to obtaining proof complexity lower bounds.

In algebraic complexity one computes polynomials *syntactically* as objects in the ring $\mathbb{F}[x_1, \ldots, x_n]$. Thus, even if one is only interested in evaluating the polynomial over the boolean cube, yielding a function $\hat{f} : \{0,1\}^n \to \mathbb{F}$, an algebraic computation of the polynomial necessarily gives a method for evaluating the polynomial over $\mathbb{F}$ as well as any extension of $\mathbb{F}$. However, some polynomials such as the permanent are known in boolean complexity to have complex behavior as functions even over boolean inputs, so one would expect that *any* polynomial f that agrees with the permanent on boolean inputs must require large algebraic circuits. We call such results functional circuit lower bounds. Prior work ([24, 25, 36]) has established functional lower bounds over fixed-size finite fields, and the recent work of Forbes, Kumar and Saptharishi [18] has established some lower bounds for any field.

▶ **Goal 4.4** (Functional Circuit Lower Bound ([25, 18])). *Obtain explicit functions $\hat{f} : \{0,1\}^n \to \mathbb{F}$ such that for any polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ such that $f(\vec{x}) = \hat{f}(\vec{x})$ for all $\vec{x} \in \{0,1\}^n$, it must be that f requires large algebraic circuits.*

While it is natural to hope that existing methods would yield such lower bounds, many lower bound techniques inherently use that algebraic computation is *syntactic.* In particular, techniques involving partial derivatives (which include the partial derivative method of Nisan-Wigderson [40] and the shifted partial derivative method of Kayal [34, 27]) cannot as is yield functional lower bounds as knowing a polynomial on $\{0,1\}^n$ is not enough to conclude information about its partial derivatives.

We now explain how functional lower bounds imply lower bounds for linear-IPS refutations in certain cases. Suppose one considers refutations of the unsatisfiable polynomial system $f(\vec{x}), \{x_i^2 - x_i\}_{i=1}^n$. A linear-IPS refutation would yield an equation of the form $g(\vec{x}) \cdot f(\vec{x}) + \sum_i h_i(\vec{x}) \cdot (x_i^2 - x_i) = 1$ for some polynomials $g, h_i \in \mathbb{F}[\vec{x}]$. Viewing this equation modulo the boolean cube, we have that $g(\vec{x}) \cdot f(\vec{x}) \equiv 1 \mod \{x_i^2 - x_i\}_i$. Equivalently, since $f(\vec{x})$ is unsatisfiable over $\{0,1\}^n$, we see that $g(\vec{x}) = {}^1\!/\!_{f(\vec{x})}$ for $\vec{x} \in \{0,1\}^n$, as $f(\vec{x})$ is never zero so this fraction is well-defined. It follows that *if* the function $\vec{x} \mapsto {}^1\!/\!_{f(\vec{x})}$ induces a functional lower bound then $g(\vec{x})$ must require large complexity, yielding the desired linear-IPS lower bound.

Thus, it remains to instantiate this program. While we are successful, we should note that this approach as is seems to only yield proof complexity lower bounds for systems with one non-boolean axiom and thus cannot encode polynomial systems arising from 3CNFs in a meaningful way.

Our starting point is to observe that the subset-sum axiom already induces a weak form of functional lower bound, where the complexity is measured by degree.

▶ **Theorem 4.5.** *Let $\mathbb{F}$ be a field of a characteristic at least $\mathsf{poly}(n)$ and $\beta \notin \{0, \ldots, n\}$. Then $\sum_i x_i - \beta, \{x_i^2 - x_i\}_i$ is unsatisfiable and any polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ with $f(\vec{x}) = \frac{1}{\sum_i x_i - \beta}$ for $\vec{x} \in \{0,1\}^n$, satisfies $\deg f \geq n$.*

A lower bound of $\lceil \frac{n}{2} \rceil$ was previously established by Impagliazzo, Pudlák, and Sgall [30], but the bound of 'n' (which is tight) will be crucial for our results.

We then lift this result to obtain lower bounds for stronger models of algebraic complexity. In particular, by replacing "x_i" by "$x_i y_i$" we show that the function $\frac{1}{\sum_i x_i y_i - \beta}$ has maximal *evaluation dimension* between $\vec{x}$ and $\vec{y}$, which is some measure of correlation. This measure is essentially *functional*, so that one can lower bound this measure by understanding the functional behavior of the polynomial on finite sets such as the boolean cube. Our lower bound for evaluation dimension follows by examining the above degree bound. Using known relations between this complexity measure and algebraic circuit classes, we can obtain lower bounds for depth-3 powering linear-IPS.

▶ **Theorem 4.6.** *Let $\mathbb{F}$ be a field of characteristic $\geq \mathsf{poly}(n)$ and $\beta \notin \{0, \ldots, n\}$. Then $\sum_{i=1}^n x_i y_i - \beta, \{x_i^2 - x_i\}_i, \{y_i^2 - y_i\}_i$ is unsatisfiable and any $\sum \bigwedge \sum$-IPS$_{LIN}$ refutation requires size $\geq \exp(\Omega(n))$.*

The above axiom only gets maximal correlation between a *fixed* partition of the variables. By introducing auxiliary variables we can create such correlation between *any* partition of (some) of the variables. By again invoking results showing such structure implies computational hardness we obtain more linear-IPS lower bounds.

▶ **Theorem 4.7.** *Let $\mathbb{F}$ be a field of characteristic $\geq \mathsf{poly}(n)$ and $\beta \notin \{0, \ldots, \binom{2n}{2}\}$. Then $\sum_{i<j} z_{i,j} x_i x_j - \beta, \{x_i^2 - x_i\}_{i=1}^n, \{z_{i,j}^2 - z_{i,j}\}_{i<j}$ is unsatisfiable, and any roABP-IPS$_{LIN}$ refutation (in any variable order) requires $\exp(\Omega(n))$-size. Further, any multilinear-formula-IPS$_{LIN'}$ refutation requires $n^{\Omega(\log n)}$-size, and any depth-$(2d+1)$ multilinear-formula-IPS$_{LIN'}$ refutation requires $n^{\Omega((n/\log n)^{1/d}/d^2)}$-size.*

Thus, we show that even though roABP-IPS$_{\mathrm{LIN}}$ and depth-3 multilinear formula-IPS$_{\mathrm{LIN'}}$ can refute the subset-sum axiom in polynomial size, slight variants of this axiom do not have polynomial-size refutations.

4.3 Lower Bounds for Multiples

While the above paradigm can establish super-polynomial lower bounds for *linear*-IPS, it does not seem able to establish lower bounds for the general IPS proof system, even for restricted classes. This is because such systems would induce equations such as $h(\vec{x})f(\vec{x})^2 + g(\vec{x})f(\vec{x}) \equiv 1 \mod \{x_i^2 - x_i\}_{i=1}^n$, where we need to design a computationally simple axiom f so that this equation implies at least one of h or g is of large complexity. In the linear-IPS case we could assume h was zero, so that we can uniquely solve for $g(\vec{x})$ for $\vec{x} \in \{0,1\}^n$. However, in general knowing $f(\vec{x})$ does not uniquely determine $g(\vec{x})$ or $h(\vec{x})$, which makes this approach significantly more complicated. Further, even though we can efficiently simulate IPS by linear-IPS in general, this simulation increases the complexity of the proof so that even if one started with a $\mathcal{C}$-IPS proof for a restricted circuit class $\mathcal{C}$ the resulting IPS$_{\mathrm{LIN}}$ proof may not be in $\mathcal{C}$-IPS$_{\mathrm{LIN}}$.

As such, we introduce a second paradigm, called *lower bounds for multiples*, which can yield $\mathcal{C}$-IPS lower bounds for various restricted classes $\mathcal{C}$. We begin by defining this question.

▶ **Goal 4.8** (Lower Bounds for Multiples)**.** *Design an explicit polynomial $f(\vec{x})$ such that for any non-zero $g(\vec{x})$ we have that $g(\vec{x})f(\vec{x})$ is hard to compute.*

We now explain how such lower bounds yield IPS lower bounds. Consider the system $f, \{x_i^2 - x_i\}_i$ with a single non-boolean axiom. An IPS refutation is a circuit $C(\vec{x}, y, \vec{z})$ such that $C(\vec{x}, 0, \vec{0}) = 0$ and $C(\vec{x}, f, \vec{x}^2 - \vec{x}) = 1$, where (as mentioned) $\vec{x}^2 - \vec{x}$ denotes $\{x_i^2 - x_i\}_i$. Expressing $C(\vec{x}, f, \vec{x}^2 - \vec{x})$ as a univariate in f, we obtain that $\sum_{i \geq 1} C_i(\vec{x}, \vec{x}^2 - \vec{x}) f^i = 1 - C(\vec{x}, 0, \vec{x}^2 - \vec{x})$ for some polynomials C_i. For many natural measures of circuit complexity $1 - C(\vec{x}, 0, \vec{x}^2 - \vec{x})$ has complexity roughly bounded by that of C itself. Though not strictly necessary for this method, it is worth noting that the complexity of each of the C_i is not much larger than that of C, as one can compute the C_i by homogenizing or interpolating C in the variable y (see for example the survey of Shpilka and Yehudayoff [54]). Thus, we see that a multiple of f has a small circuit, as $\left(\sum_{i \geq 1} C_i(\vec{x}, \vec{x}^2 - \vec{x}) f^{i-1}\right) \cdot f = 1 - C(\vec{x}, 0, \vec{x}^2 - \vec{x})$. Thus, if we can show that all multiples of f require large circuits then we rule out a small IPS refutation.

We now turn to methods for obtaining polynomials with hard multiples. Intuitively if a polynomial f is hard then so should small modifications such as $f^2 + x_1 f$, and this intuition is supported by the result of Kaltofen [32] which shows that if a polynomial has a small algebraic circuit then so do all of its factors. As a consequence, if a polynomial requires super-polynomially large algebraic circuits then so do all of its multiples. However, Kaltofen's [32] result is about *general* algebraic circuits, and there are very limited results about the complexity of factors of *restricted* algebraic circuits ([14, 41]) so that obtaining polynomials for hard multiples via factorization results seems difficult.

However, note that lower bound for multiples has a different order of quantifiers than the factoring question. That is, Kaltofen's [32] result speaks about the factors of *any* small circuit, while the lower bound for multiples speaks about the multiples of a *single* polynomial. Thus, it seems plausible that existing methods could yield such explicit polynomials, and indeed we show this is the case.

We begin by noting that obtaining lower bounds for multiples is a natural instantiation of the algebraic *hardness versus randomness* paradigm. In particular, Heintz-Schnorr [28] and Agrawal [1] showed that obtaining deterministic (black-box) PIT algorithms implies lower bounds, and we strengthen that connection here to lower bounds for multiples. We can actually instantiate this connection, and we use slight modifications of existing PIT algorithms to show that multiples of the determinant are hard in some models.

▶ **Theorem 4.9.** *Let $\mathcal{C}$ be a restricted class of n-variate algebraic circuits. Full derandomization of PIT algorithms for $\mathcal{C}$ yields an explicit polynomials all of whose multiples require* $\exp(\Omega(n))$*-size as $\mathcal{C}$-circuits.*

In particular, when $\mathcal{C}$ is the class of sparse polynomials, depth-3 powering formulas, $\sum\bigwedge\sum\prod^{\text{‰}(1)}$ formulas (in characteristic zero), or "every-order" roABPs, then all nonzero multiples of the $n \times n$ determinant are $\exp(\Omega(n))$*-hard in these models.*

The above statement shows that *derandomization* implies *hardness*. We also partly address the converse direction by arguing that hardness-to-randomness construction of Kabanets and Impagliazzo [31] only requires lower bounds for multiples to derandomize PIT. Unfortunately, this direction is harder to instantiate for restricted classes as it requires lower bounds for classes with suitable closure properties.[4]

[4] Although, we note that one can instantiate this connection with depth-3 powering formulas (or even

Unfortunately the above result is slightly unsatisfying from a proof complexity standpoint as the (exponential-size) lower bounds for the subclasses of IPS one can derive from the above result would involve the determinant polynomial as an axiom. While the determinant is efficiently computable, it is not computable by the above restricted circuit classes (indeed, the above result proves that). As such, this would not fit the real goal of proof complexity which seeks to show that there are statements whose proofs must be *super-polynomial larger* than the length of the statement. Thus, if we measure the size of the IPS proof and the axioms with respect to the same circuit measure, the lower bounds for multiples approach *cannot* establish such super-polynomial lower bounds.

However, we believe that lower bounds for multiples could lead, with further ideas, to proof complexity lower bounds in the conventional sense. That is, it seems plausible that by adding *extension variables* we can convert complicated axioms to simple, local axioms by tracing through the computation of that axiom. That is, consider the axiom $xyzw$. This can be equivalently written as $\{a-xy, b-zw, c-ab, c\}$, where this conversion is done by considering a natural algebraic circuit for $xyzw$, replacing each gate with a new variable, and adding an axiom ensuring the new variables respect the computation of the circuit. While we are unable to understand the role of extension variables in this work, we aim to give as simple axioms as possible whose multiples are all hard as this may facilitate future work on extension variables.

We now discuss the lower bounds for multiples we obtain.[5]

▶ **Theorem 4.10.** *We obtain the following lower bounds for multiples.*

- *All non-zero multiples of* $x_1 \cdots x_n$ *require* $\exp(\Omega(n))$*-size as a depth-3 powering formula (over any field), or as a* $\sum\bigwedge\sum\prod^{‰(1)}$ *formula (in characteristic zero).*
- *All non-zero multiples of* $(x_1+1)\cdots(x_n+1)$ *require* $\exp(\Omega(n))$*-many monomials.*
- *All non-zero multiples of* $\prod_i(x_i+y_i)$ *require* $\exp(\Omega(n))$*-width as a roABPs in any variable order where* $\vec{x}$ *precedes* $\vec{y}$*.*
- *All non-zero multiples of* $\prod_{i,j=1}^{n}(z_{i,j}\cdot(x_i+x_j+x_ix_j)+(1-z_{i,j}))$ *require* $\exp(\Omega(n))$*-width as a roABP in any variable order, as well as* $\exp(\Omega(n))$*-width as a read-twice oblivious ABP.*

We now briefly explain our techniques for obtaining these lower bounds, focusing on the simplest case of depth-3 powering formulas. It follows from the partial derivative method of Nisan and Wigderson [39] (see Kayal [33]) that such formulas require exponential size to compute the monomial $x_1 \ldots x_n$ *exactly.* Forbes and Shpilka [21], in giving a PIT algorithm for this class, showed that this lower bound can be *scaled down* and *made robust.* That is, if one has a size-s depth-3 powering formula, it follows that *if* it computes a monomial $x_{i_1}\cdots x_{i_\ell}$ for distinct i_j then $l \le O(\log s)$ (so the lower bound is scaled down). One can then show that regardless of what this formula actually computes the *leading* monomial $x_{i_1}^{a_{i_1}}\cdots x_{i_\ell}^{a_{i_\ell}}$ (for distinct i_j and positive a_{i_j}) must have that $\ell \le O(\log s)$. One then notes that leading monomials are *multiplicative.* Thus, for any non-zero g the leading monomial of $g\cdot x_1\ldots x_n$ involves n variables so that if $g\cdot x_1\ldots x_n$ is computed in size-s then $n \le O(\log s)$, giving $s \ge \exp(\Omega(n))$ as desired. One can then obtain the other lower bounds using the same

$\sum\bigwedge\sum\prod^{‰(1)}$ formulas) using the lower bounds for multiples developed in this paper, building on the work of Forbes [17]. However, the resulting PIT algorithms are worse than those developed by Forbes [17].

[5] While we discussed functional lower bounds for multilinear formulas, this class is not interesting for the lower bounds for multiples question. This is because a multiple of a multilinear polynomial may not be multilinear, and thus clearly cannot have a multilinear formula.

idea, though for roABPs one needs to define a leading *diagonal* (refining an argument of Forbes-Shpilka [20]).

We now conclude our IPS lower bounds.

▶ **Theorem 4.11.** *We obtain the following lower bound for subclasses of IPS.*

- *In characteristic zero, for $m \neq n$, the system of polynomials $x_1 \cdots x_n - 1, x_1 + \cdots + x_n - m, \{x_i^2 - x_i\}_{i=1}^n$ is unsatisfiable, any $\sum\bigwedge\sum$-IPS refutation requires $\exp(\Omega(n))$-size.*
- *The system of polynomials, $1 + \prod_{i,j=1}^n (z_{i,j}(x_i + x_j - x_i x_j) + (1 - z_{i,j})), \{x_i^2 - x_i\}_i, \{z_{i,j}^2 - z_{i,j}\}_{i,j}$ is unsatisfiable, and any roABP-IPS refutation (in any variable order) must be of width $\exp(\Omega(n))$.*

Note that the first result is an encoding that $\mathrm{AND}(x_1, \ldots, x_n) = 1$ but the number of variables that equal 1 is different than n. The second is not as natural, but contains the simpler polynomial $\prod_i (u_i + v_i - u_i v_i) + 1$ (up to renaming, and after appropriate substitution of the $z_{i,j}$ to values from $\{0, 1\}$), which encodes that $\mathrm{AND}(\mathrm{OR}(u_1, v_1), \cdots, \mathrm{OR}(u_n, v_n)) \notin \{0, 1\}$.

5 Discussion

In this paper we proved new lower bounds for the Grochow-Pitassi Ideal Proof System (IPS), for various restricted circuit classes underlying this proof system. The main novelty here, as compared with essentially all previous work in algebraic proof complexity, is that lower bounds are proved directly for the most interesting computational complexity measure, namely circuit size, rather than simpler notions of complexity such as degree and sparsity of the polynomials involved. This opens up a path to extending our results to IPS over other circuit classes, in particular ones for which there are already computational lower bounds. A specific challenge is doing so for IPS using depth-4 arithmetic circuits, for which recent exciting work using shifted partial derivatives imply superpolynomial computational lower bounds for natural polynomials.

A different challenge, even for the circuit classes considered here, is that most of our results apply only when the hard contradiction has a specific, and somewhat unnatural structure: aside from the Boolean axioms, there is only one more axiom, which involves all variables (and sometimes also has high degree). Natural tautologies studied in proof complexity arise from k-CNF formulas, where $k = O(1)$, so the contradiction contains many polynomials, each on a constant number of variables (and hence also of constant degree). The techniques in this paper cannot prove IPS lower bounds for such contradictions even with the simplest circuit classes. It would be extremely interesting to devise techniques able to handle such contradictions, arising, e.g., from Tseitin tautologies or random CNFs.

A more specific direction to follow arises from our "PIT technique". In Subsection 4.3 we noted that this technique of using lower bounds for multiples requires including the determinant as an axiom, and it only works for models that cannot efficiently compute the determinant. It would be interesting to weaken this requirement. For example, instead of considering systems that include the determinant as an axiom, one could instead consider the (algebraic) "hard matrix identities" that were suggested by Cook and Rackoff (cf. [7]) and later studied by Soltys and Cook [55]. Recall that the task at hand is, starting from the axioms $XY - I$ (where X and Y are symbolic $n \times n$ matrices), the goal is to derive $YX - I$ in IPS. Here the axioms are easily computable by roABPs, but the derivation is believed to require computing the determinant, so it should be hard for roABP-IPS (see Hrubeš-Tzameret [29] and also Appendix B of the arXiv version of Grochow-Pitassi [26] for more discussion on this.)

Finally, we leave open the question of extending our results from lower bounds on the "static" IPS to lower bounds on a "dynamic" algebraic proof system like the polynomial calculus:

▶ **Open Problem 5.1.** *Can the lower bounds on roABP-IPS$_{LIN}$ and multilinear-formula-IPS$_{LIN}$ from Theorem 4.7 be extended to (tree-like or dag-like) PC over roABPs ([56]) and PC over multilinear formulas (fMC from [48]), respectively?*

Acknowledgments. We would like to thank Rafael Oliviera for helpful discussions, as well as Mrinal Kumar and Ramprasad Saptharishi for conversations [18] clarifying the roles of functional lower bounds in this work. We would also like to thank Joshua Grochow for helpful discussions regarding this work. We are grateful for the CCC'16 reviewers for their careful read of the paper and for their comments.

References

1 Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2005)*, pages 92–105, 2005. doi:10.1007/11590156_6.

2 Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *arXiv*, 1406.7535, 2014. URL: http://arxiv.org/abs/1406.7535.

3 Manindra Agrawal, Chandan Saha, and Nitin Saxena. Quasi-polynomial hitting-set for set-depth-Δ formulas. In *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, pages 321–330, 2013. Full version at arXiv:1209.2333. doi:10.1145/2488608.2488649.

4 Michael Alekhnovich and Alexander A. Razborov. Lower bounds for polynomial calculus: Non-binomial case. In *Proceedings of the 42nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2001)*, pages 190–199, 2001. doi:10.1109/SFCS.2001.959893.

5 Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. *Electronic Colloquium on Computational Complexity (ECCC)*, 22:184, 2015. URL: http://eccc.hpi-web.de/report/2015/184/.

6 Paul Beame, Russell Impagliazzo, Jan Krajíček, Toniann Pitassi, and Pavel Pudlák. Lower bounds on Hilbert's Nullstellensatz and propositional proofs. *Proc. London Math. Soc. (3)*, 73(1):1–26, 1996. Preliminary version in the *35th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1994)*. doi:10.1112/plms/s3-73.1.1.

7 Paul Beame and Toniann Pitassi. Propositional proof complexity: past, present, and future. *Bull. Eur. Assoc. Theor. Comput. Sci. EATCS*, 1998(65):66–89, 1998.

8 Samuel R. Buss, Dima Grigoriev, Russell Impagliazzo, and Toniann Pitassi. Linear gaps between degrees for the polynomial calculus modulo distinct primes. *J. Comput. Syst. Sci.*, 62(2):267–289, 2001. Preliminary version in the *14th Annual IEEE Conference on Computational Complexity (CCC 1999)*. doi:10.1006/jcss.2000.1726.

9 Samuel R. Buss, Russell Impagliazzo, Jan Krajíček, Pavel Pudlák, Alexander A. Razborov, and Jiří Sgall. Proof complexity in algebraic systems and bounded depth Frege systems with modular counting. *Computational Complexity*, 6(3):256–298, 1996. doi:10.1007/BF01294258.

10 Matthew Clegg, Jeff Edmonds, and Russell Impagliazzo. Using the Groebner basis algorithm to find proofs of unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996)*, pages 174–183, 1996. doi:10.1145/237814.237860.

11 Stephen A. Cook and Robert A. Reckhow. On the lengths of proofs in the propositional calculus (preliminary version). In *Proceedings of the 6th Annual ACM Symposium on Theory of Computing (STOC 1974)*, pages 135–148, 1974. For corrections see Cook-Reckhow [12]. doi:10.1145/800119.803893.

12 Stephen A. Cook and Robert A. Reckhow. Corrections for "On the lengths of proofs in the propositional calculus (preliminary version)". *SIGACT News*, 6(3):15–22, July 1974. doi:10.1145/1008311.1008313.

13 Stephen A. Cook and Robert A. Reckhow. The relative efficiency of propositional proof systems. *J. Symb. Log.*, 44(1):36–50, 1979. This is a journal-version of Cook-Reckhow [11] and Reckhow [51]. doi:10.2307/2273702.

14 Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. Hardness-randomness tradeoffs for bounded depth arithmetic circuits. *SIAM J. Comput.*, 39(4):1279–1293, 2009. Preliminary version in the *40th Annual ACM Symposium on Theory of Computing (STOC 2008)*. doi:10.1137/080735850.

15 Ismor Fischer. Sums of like powers of multivariate linear forms. *Mathematics Magazine*, 67(1):59–61, 1994. URL: http://www.jstor.org/stable/2690560.

16 Michael A. Forbes. *Polynomial Identity Testing of Read-Once Oblivious Algebraic Branching Programs*. PhD thesis, Massachusetts Institute of Technology, June 2014. URL: http://hdl.handle.net/1721.1/89843.

17 Michael A. Forbes. Deterministic divisibility testing via shifted partial derivatives. In *Proceedings of the 56th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2015)*, 2015.

18 Michael A. Forbes, Mrinal Kumar, and Ramprasad Saptharishi. Functional lower bounds for arithmetic circuits and boolean circuit complexity. Manuscript, 2015.

19 Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 867–875, 2014. Full version at arXiv:1309.5668. doi:10.1145/2591796.2591816.

20 Michael A. Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, pages 163–172, 2012. Full version at arXiv:1111.0663. doi:10.1145/2213977.2213995.

21 Michael A. Forbes and Amir Shpilka. Explicit Noether Normalization for simultaneous conjugation via polynomial identity testing. In *Proceedings of the 17th International Workshop on Randomization and Computation (RANDOM 2013)*, pages 527–542, 2013. Full version at arXiv:1303.0084. doi:10.1007/978-3-642-40328-6_37.

22 Michael A. Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 243–252, 2013. Full version at arXiv:1209.2408. doi:10.1109/FOCS.2013.34.

23 Dima Grigoriev. Tseitin's tautologies and lower bounds for Nullstellensatz proofs. In *Proceedings of the 39th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1998)*, pages 648–652, 1998. doi:10.1109/SFCS.1998.743515.

24 Dima Grigoriev and Marek Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC 1998)*, pages 577–582, 1998. doi:10.1145/276698.276872.

25 Dima Grigoriev and Alexander A. Razborov. Exponential lower bounds for depth 3 arithmetic circuits in algebras of functions over finite fields. *Appl. Algebra Eng. Commun. Comput.*, 10(6):465–487, 2000. Preliminary version in the *39th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1998)*. doi:10.1007/s002009900021.

26 Joshua A. Grochow and Toniann Pitassi. Circuit complexity, proof complexity, and polynomial identity testing. In *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*, pages 110–119, 2014. Full version at arXiv:abs/1404.3820. doi:10.1109/FOCS.2014.20.

27 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Approaching the chasm at depth four. *J. ACM*, 61(6):33:1–33:16, December 2014. Preliminary version in the *28th Annual IEEE Conference on Computational Complexity (CCC 2013)*. doi:10.1145/2629541.

28 Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing (STOC 1980)*, pages 262–272, 1980. doi:10.1145/800141.804674.

29 Pavel Hrubes and Iddo Tzameret. Short proofs for the determinant identities. *SIAM J. Comput.*, 44(2):340–383, 2015. Preliminary version in the *44th Annual ACM Symposium on Theory of Computing (STOC 2012)*. doi:10.1137/130917788.

30 Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. Lower bounds for the polynomial calculus and the gröbner basis algorithm. *Computational Complexity*, 8(2):127–144, 1999. doi:10.1007/s000370050024.

31 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*. doi:10.1007/s00037-004-0182-6.

32 Erich L. Kaltofen. Factorization of polynomials given by straight-line programs. In Silvio Micali, editor, *Randomness and Computation*, volume 5 of *Advances in Computing Research*, pages 375–412. JAI Press, Inc., Greenwich, CT, USA, 1989. URL: http://www.math.ncsu.edu/~kaltofen/bibliography/89/Ka89_slpfac.pdf.

33 Neeraj Kayal. Personal Communication to Saxena [52], 2008.

34 Neeraj Kayal. An exponential lower bound for the sum of powers of bounded degree polynomials. *Electronic Colloquium on Computational Complexity (ECCC)*, 19(81), 2012. URL: http://eccc.hpi-web.de/report/2012/081.

35 Jan Krajíček. *Bounded arithmetic, propositional logic, and complexity theory*, volume 60 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1995. doi:10.1017/CBO9780511529948.

36 Mrinal Kumar and Ramprasad Saptharishi. An exponential lower bound for homogeneous depth-5 circuits over finite fields. *arXiv*, 1507.00177, 2015. URL: http://arxiv.org/abs/1507.00177.

37 Fu Li, Iddo Tzameret, and Zhengyu Wang. Non-commutative formulas and Frege lower bounds: a new characterization of propositional proofs. In *Proceedings of the 30th Computational Complexity Conference (CCC), June 17-19, 2015*, 2015.

38 Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC 1991)*, pages 410–418, 1991. doi:10.1145/103418.103462.

39 Noam Nisan and Avi Wigderson. Hardness vs randomness. *J. Comput. Syst. Sci.*, 49(2):149–167, 1994. Preliminary version in the *29th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1988)*. doi:10.1016/S0022-0000(05)80043-1.

40 Noam Nisan and Avi Wigderson. Lower bounds on arithmetic circuits via partial derivatives. *Computational Complexity*, 6(3):217–234, 1996. Preliminary version in the *36th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1995)*. doi:10.1007/BF01294256.

41 Rafael Oliveira. Factors of low individual degree polynomials. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC 2015)*, volume 33 of *Leibniz Inter-*

national Proceedings in Informatics (LIPIcs), pages 198–216, 2015. `doi:10.4230/LIPIcs.CCC.2015.198`.

42 Rafael Oliveira, Amir Shpilka, and Ben Lee Volk. Subexponential size hitting sets for bounded depth multilinear formulas. In *Proceedings of the 30th Annual Computational Complexity Conference (CCC 2015)*, volume 33 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 304–322, 2015. Full version at `arXiv:1411.7492`. `doi:10.4230/LIPIcs.CCC.2015.304`.

43 Toniann Pitassi. Algebraic propositional proof systems. In *Descriptive complexity and finite models (Princeton, NJ, 1996)*, volume 31 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 215–244. Amer. Math. Soc., Providence, RI, 1997.

44 Pavel Pudlák. Lower bounds for resolution and cutting plane proofs and monotone computations. *The Journal of Symbolic Logic*, 62(3):981–998, Sept. 1997.

45 Ran Raz. Separation of multilinear circuit and formula size. *Theory of Computing*, 2(6):121–135, 2006. Preliminary version in the *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*. `doi:10.4086/toc.2006.v002a006`.

46 Ran Raz. Multi-linear formulas for permanent and determinant are of super-polynomial size. *J. ACM*, 56(2), 2009. Preliminary version in the *36th Annual ACM Symposium on Theory of Computing (STOC 2004)*. `doi:10.1145/1502793.1502797`.

47 Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Comput. Complex.*, 14(1):1–19, April 2005. Preliminary version in the *19th Annual IEEE Conference on Computational Complexity (CCC 2004)*. `doi:10.1007/s00037-005-0188-8`.

48 Ran Raz and Iddo Tzameret. The strength of multilinear proofs. *Computational Complexity*, 17(3):407–457, 2008.

49 Ran Raz and Amir Yehudayoff. Lower bounds and separations for constant depth multilinear circuits. *Computational Complexity*, 18(2):171–207, 2009. Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*. `doi:10.1007/s00037-009-0270-8`.

50 Alexander A. Razborov. Lower bounds for the polynomial calculus. *Computational Complexity*, 7(4):291–324, 1998. `doi:10.1007/s000370050013`.

51 Robert A. Reckhow. *On the lengths of proofs in the propositional calculus.* PhD thesis, University of Toronto, 1976. URL: `https://www.cs.toronto.edu/~sacook/homepage/reckhow_thesis.pdf`.

52 Nitin Saxena. Diagonal circuit identity testing and lower bounds. In *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*, pages 60–71, 2008. Preliminary version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR07-124. `doi:10.1007/978-3-540-70575-8_6`.

53 Amir Shpilka. Affine projections of symmetric polynomials. *J. Comput. Syst. Sci.*, 65(4):639–659, 2002. Preliminary version in the *16th Annual IEEE Conference on Computational Complexity (CCC 2001)*. `doi:10.1016/S0022-0000(02)00021-1`.

54 Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010. `doi:10.1561/0400000039`.

55 Michael Soltys and Stephen Cook. The proof complexity of linear algebra. *Ann. Pure Appl. Logic*, 130(1-3):277–323, 2004.

56 Iddo Tzameret. Algebraic proofs over noncommutative formulas. *Information and Computation*, 209(10):1269–1292, 2011.