Ideals, Determinants, and Straightening: Proving and Using Lower Bounds for Polynomial Ideals

Robert Andrews
University of Illinois Urbana-Champaign
Department of Computer Science
Urbana, IL, USA
rgandre2@illinois.edu

Michael A. Forbes
University of Illinois Urbana-Champaign
Department of Computer Science
Urbana, IL, USA
miforbes@illinois.edu

ABSTRACT

We show that any nonzero polynomial in the ideal generated by the $r \times r$ minors of an $n \times n$ matrix X can be used to efficiently approximate the determinant. Specifically, for any nonzero polynomial f in this ideal, we construct a small depth-three f-oracle circuit that approximates the $\Theta(r^{1/3}) \times \Theta(r^{1/3})$ determinant in the sense of border complexity. For many classes of algebraic circuits, this implies that every nonzero polynomial in the ideal generated by $r \times r$ minors is at least as hard to approximately compute as the $\Theta(r^{1/3}) \times \Theta(r^{1/3})$ determinant. We also prove an analogous result for the Pfaffian of a $2n \times 2n$ skew-symmetric matrix and the ideal generated by Pfaffians of $2r \times 2r$ principal submatrices.

This answers a recent question of Grochow about complexity in polynomial ideals in the setting of border complexity. Leveraging connections between the complexity of polynomial ideals and other questions in algebraic complexity, our results provide a generic recipe that allows lower bounds for the determinant to be applied to other problems in algebraic complexity. We give several such applications, two of which are highlighted below.

We prove new lower bounds for the Ideal Proof System of Grochow and Pitassi. Specifically, we give super-polynomial lower bounds for refutations computed by low-depth circuits. This extends the recent breakthrough low-depth circuit lower bounds of Limaye et al. to the setting of proof complexity. Moreover, we show that for many natural circuit classes, the approximative proof complexity of our hard instance is governed by the approximative circuit complexity of the determinant.

We also construct new hitting set generators for the closure of low-depth circuits. For any $\varepsilon > 0$, we construct generators with seed length $O(n^{\varepsilon})$ that hit n-variate low-depth circuits. Our generators attain a near-optimal tradeoff between their seed length and degree, and are computable by low-depth circuits of near-linear size (with respect to the size of their output). This matches the seed length of the generators recently obtained by Limaye et al., but improves on the degree and circuit complexity of the generator.

STOC '22, June 20–24, 2022, Rome, Italy

ACM ISBN 978-1-4503-9264-8/22/06…$15.00
https://doi.org/10.1145/3519935.3520025

CCS CONCEPTS

• **Theory of computation** → **Algebraic complexity theory**; **Problems, reductions and completeness**; *Pseudorandomness and derandomization*; *Proof complexity*.

KEYWORDS

Determinantal ideals, straightening law, polynomial identity testing, Ideal Proof System

ACM Reference Format:
Robert Andrews and Michael A. Forbes. 2022. Ideals, Determinants, and Straightening: Proving and Using Lower Bounds for Polynomial Ideals. In *Proceedings of the 54th Annual ACM SIGACT Symposium on Theory of Computing (STOC '22), June 20–24, 2022, Rome, Italy*. ACM, New York, NY, USA, 14 pages. https://doi.org/10.1145/3519935.3520025

1 INTRODUCTION

A central goal of algebraic complexity theory is to understand the resources needed to compute multivariate polynomials in algebraic models of computation. Typically, one attempts to determine the complexity of a single family of polynomials $\{f_n(\overline{x}) : n \in \mathbb{N}\}$, such as the $n \times n$ determinant or permanent. A generalization of this task is to examine the complexity of a family of *ideals* $\{I_n \subseteq \mathbb{F}[\overline{x}] : n \in \mathbb{N}\}$ of polynomials. Recall that in a commutative ring R, an ideal $I \subseteq R$ is a subset of R such that (1) if $a, b \in I$, then $a + b \in I$, and (2) if $a \in I$ and $r \in R$, then $ar \in I$. Ideals naturally arise in commutative algebra and algebraic geometry; for example, the set of polynomials that vanish on a subset $V \subseteq \mathbb{F}^n$ is an ideal. Closer to computer science and algebraic complexity, ideals appear in the study of polynomial identity testing, polynomial factorization, and algebraic proof complexity, though these appearances are not always made explicit. Due to the prominence of ideals in algebra and algebraic complexity, it is both natural and worthwhile to study them from a complexity-theoretic perspective.

Every nonzero ideal contains polynomials of arbitrarily large circuit complexity. This is a straightforward consequence of the fact that ideals are closed under multiplication by arbitrary polynomials. A more interesting task, then, is to determine the minimum possible complexity of a nonzero polynomial in an ideal.

Unfortunately, little is known about the complexity of ideals aside from what is implicit in their connection to other problems of algebraic complexity. A recent column by Grochow [33] surveyed these connections and posed some open questions, both general and concrete, about the complexity of ideals. In particular, he raised the following question regarding an explicit family of ideals.

Conjecture ([33, Conjecture 6.3]). *Let X be a $n \times n$ matrix of variables and let I_n be the ideal generated by the $n/2 \times n/2$ minors of X. For every nonzero polynomial $f(X) \in I_n$, there is a small algebraic circuit with f-oracle gates that computes the $m \times m$ determinant for some $m = n^{\Theta(1)}$.*

Due to the close relationship between the non-vanishing of minors and matrix rank, it is natural to conjecture that such a circuit exists. If the oracle circuit is not restricted in any manner, then the desired circuit exists simply because the determinant can be computed efficiently by algebraic circuits. However, if the oracle circuit is required to be, for example, a formula, then this question becomes nontrivial, as the determinant is not known to be computable by small formulas.

The main contribution of our work is to resolve this conjecture in the setting of approximate algebraic computation.

Theorem. *Grochow's conjecture is true (with respect to border complexity).*

Specifically, we show that for any nonzero polynomial $f \in I_n$, the $\Theta(n^{1/3}) \times \Theta(n^{1/3})$ determinant can be approximately computed by a small depth-three f-oracle circuit with a single oracle gate. A direct consequence of this is that for many circuit classes C, if the determinant cannot be approximated by polynomial-size C-circuits, then neither can any polynomial in the ideal I_n. Naturally, this has applications to polynomial identity testing and algebraic proof complexity by employing the supporting role played by the complexity of ideals in those areas.

Before describing our results in more detail, we briefly survey what is known about the complexity of ideals and its connections to polynomial identity testing and algebraic proof complexity.

1.1 The Complexity of Ideals

Most of what is known about the complexity of ideals is limited to ideals generated by a single polynomial. The ideal $\langle f \rangle$ generated by a polynomial $f(\overline{x})$ consists of all multiples of f, so questions about the complexity of this ideal become questions about the complexity of f and its multiples. Determining the minimum complexity of a polynomial in $\langle f \rangle$ amounts to determining whether there is a multiple of f that is significantly easier to compute than f itself. This leads to the question of factoring algebraic circuits: given a small circuit computing a polynomial $g(\overline{x})$, can the factors of $g(\overline{x})$ be computed by small circuits?

This question was addressed in a celebrated result of Kaltofen [42] (with alternate proofs by Bürgisser [12, Theorem 2.21] and Chou et al. [16]), who showed that factors (of low multiplicity) of small circuits can be computed by small circuits. Taking the contrapositive, if $f(\overline{x})$ cannot be computed by small circuits, then neither can any polynomial $g \in \langle f \rangle$ which has f as a factor of low multiplicity. Polynomial factorization has since been studied in restricted algebraic circuit classes, including low-depth circuits [17, 27], formulas [25, 55], algebraic branching programs [25, 72], and sparse polynomials [8]. This is motivated in part by the use of Kaltofen's theorem to establish hardness-to-pseudorandomness results for polynomial identity testing, as done in the work of Kabanets and Impagliazzo [41].

Kaltofen's result gives us a strong understanding of the complexity of the low-degree polynomials in a principal ideal. Because algebraic complexity theory is primarily interested in the computation of low-degree polynomials, this suffices for most applications. However, the situation would be cleaner if lower bounds on the complexity of a polynomial f implied comparable lower bounds on the complexity of all polynomials in the ideal $\langle f \rangle$, not just for those polynomials $g \in \langle f \rangle$ for which f is a factor of low multiplicity. Kaltofen [42] asked in the language of factorization whether this is the case; this question remains open and is now known as the Factor Conjecture. In the setting of approximative algebraic computation, the analogue of the Factor Conjecture was proved by Bürgisser [13]. It is interesting to note that, coincidentally, we also make essential use of approximative computation in our work.

For non-principal ideals, much less is known. What knowledge we do have stems from connections to polynomial identity testing and the Ideal Proof System. We defer our explanation of these connections to Subsection 1.2 and Subsection 1.3, respectively.

Approximate algebraic computation will play a key role in our work, so we briefly discuss it here. For simplicity, we will focus on circuits and polynomials defined over the complex numbers. We say that a polynomial $f(\overline{x})$ can be approximately computed by small algebraic circuits if there is a collection of polynomials $\{f_\varepsilon : \varepsilon > 0\}$ such that (1) for all $\varepsilon > 0$, the polynomial f_ε can be computed by a small circuit, and (2) we have $\lim_{\varepsilon \to 0} f_\varepsilon = f$, where convergence is coefficient-wise. Over the complex numbers, this can be interpreted as saying that f lies in the closure (with respect to the Euclidean topology) of the set of polynomials computable by small circuits. If f can be approximated well by polynomials from a circuit class C, then we say that f is in $\overline{C}$, the closure of C. The circuit complexity of the approximating polynomials f_ε is referred to as the *border complexity* of f. Naturally, one can also consider border complexity with respect to other classes of algebraic circuits, such as formulas or branching programs.

Border complexity appeared as early as the late 1970s, when Bini et al. [9] improved upon the state-of-the-art algorithms for matrix multiplication by considering an approximative version of the problem. The notion of border complexity also plays a prominent role in the geometric complexity theory program of Mulmuley and Sohoni [54]. Roughly speaking, the goal of that program is to prove super-polynomial lower bounds on the border complexity of the permanent using techniques from algebraic geometry and representation theory.

In general, the relationship between exact and border complexity is not well-understood. Forbes [28] (see also Bläser et al. [11]) observed that exact and border complexity are equivalent for read-one oblivious algebraic branching programs. Dutta et al. [23] recently showed that polynomials in the border of depth-three circuits of bounded top fan-in can be computed exactly by small algebraic branching programs. However, for classes like VP and VNP (the algebraic analogues of P and NP), it is not clear how they relate to their closure.

Returning to the complexity of ideals, if we are content to operate in the setting of border complexity, then the work of Bürgisser [13] shows that up to polynomial factors, the complexity of a principal ideal $\langle f \rangle$ is governed by the border complexity of its generator f.

Unfortunately, this seems to be where our understanding of the complexity of ideals stops. Even ideals generated by two polynomials are not well-understood structurally from the viewpoint of complexity theory. There are examples of explicit ideals, coming from polynomial identity testing, that are not principal and for which we can prove lower bounds; see Subsection 1.2 below for more.

1.2 Polynomial Identity Testing

Polynomial identity testing (which we abbreviate as PIT) is the algorithmic problem of testing whether an algebraic circuit computes the zero polynomial. Typically, one assumes that the circuit computes a polynomial of degree at most $n^{O(1)}$, where n is the number of input variables. A simple coRP algorithm for this problem follows from the Schwartz–Zippel lemma [68, 75]. When the input is allowed to be an algebraic circuit without further structural restrictions, no deterministic algorithm is known that improves on the naïve derandomization of this randomized algorithm. In fact, even obtaining a nondeterministic algorithm running in subexponential time is known to imply circuit lower bounds that lie beyond the reach of current techniques [41].

More is known for many restricted classes of circuits, including sparse polynomials [47], depth-three [26, 44–46, 65–67] and depth-four [24, 57, 58, 69] circuits of bounded top fan-in, read-once formulas [53, 70], read-once oblivious algebraic branching programs [1, 4, 10, 29, 30, 35, 37, 38], low-depth multilinear circuits [5, 43, 56, 64], and low-depth circuits [50]. In general, algorithms for PIT are designed by giving an efficient construction of a *hitting set generator*. That is, we construct a low-degree polynomial map $\mathcal{G} : \mathbb{F}^\ell \to \mathbb{F}^n$ with $\ell \ll n$ such that if $f(\overline{x})$ is a nonzero polynomial computable by a small circuit, then $f(\mathcal{G}(\overline{y})) \neq 0$. This reduces the number of variables in the circuit without increasing the degree too much. We then obtain a faster deterministic algorithm by using the brute-force derandomization of the Schwartz–Zippel lemma to test $f(\mathcal{G}(\overline{y}))$.

In fact, constructing such a generator $\mathcal{G}$ corresponds to proving lower bounds against a polynomial ideal. Fix a circuit class C (for example, the class of n^2-size circuits) and let $\mathcal{G}$ be a hitting set generator for C. Let $\mathcal{G}(\overline{y}) = (\mathcal{G}_1(\overline{y}), \ldots, \mathcal{G}_n(\overline{y}))$ and consider the ideal of polynomials $f(\overline{x})$ that vanish on $\mathcal{G}(\overline{y})$, i.e., polynomials such that $f(\mathcal{G}(\overline{y})) = 0$. This ideal can be written as the intersection

$$I_{\mathcal{G}} \coloneqq \langle x_i - \mathcal{G}_i(\overline{y}) : i \in [n] \rangle \cap \mathbb{F}[\overline{x}],$$

and in general is not generated by a single polynomial. Suppose f is a nonzero polynomial in the ideal $I_{\mathcal{G}}$. Because we assumed $\mathcal{G}$ to be a hitting set generator for the circuit class C, this means that f cannot be computed by circuits from C. That is, proving that $\mathcal{G}$ is a generator for C is equivalent to proving that no element of $I_{\mathcal{G}}$ can be computed by a circuit from C. To the best of our knowledge, this connection accounts for almost all known examples of lower bounds for non-principal ideals. We remark that this approach can prove lower bounds against "natural" non-principal ideals. For example, [32, Corollary 6.7] easily generalizes to prove lower bounds against determinantal ideals for weak circuit classes. However, this approach does not necessarily allow one to choose an ideal and subsequently prove a lower bound against that particular ideal.

One can also construct hitting set generators using lower bounds for ideals. Kabanets and Impagliazzo [41] used Kaltofen's factorization result to show that circuit lower bounds for explicit families of polynomials can be used to derandomize PIT. In the analysis of the Kabanets–Impagliazzo generator, what is really needed is a lower bound for all low-degree multiples of a polynomial f, which is exactly what Kaltofen's theorem provides if f is assumed to be hard to compute. Further work on the algebraic hardness-randomness paradigm in the setting of low-depth circuits [17, 27] followed the approach of Kabanets and Impagliazzo [41], proving analogues of Kaltofen's factoring result for bounded-depth circuits.

One can also consider PIT for polynomials of small border complexity. Even in the randomized setting, the complexity of this problem is unclear, as it is not obvious how to evaluate a polynomial $f(\overline{x})$ given only a circuit that approximates $f(\overline{x})$, nor is it clear that such an approximating circuit even has a succinct description. However, one can still try to construct hitting set generators for polynomials of small border complexity. Forbes and Shpilka [31] and Guo et al. [36] gave PSPACE constructions of hitting set generators for polynomials with small border circuit complexity. One of the primary conceptual contributions of Forbes and Shpilka [31] was the definition of a *robust* hitting set generator. Roughly, a generator $\mathcal{G}$ for a class C is robust if for every nonzero polynomial $f \in C$, the composition $f(\mathcal{G}(\overline{y}))$ is "far" from the zero polynomial (after f has been suitably normalized). It is not hard to show that, over a field of characteristic zero, a generator $\mathcal{G}$ for C is robust if and only if $\mathcal{G}$ hits the closure $\overline{C}$ of C. Over an arbitrary field, one can likewise consider the problem of constructing hitting set generators for the closures of circuit classes, although the notion of $f(\mathcal{G}(\overline{y}))$ being far from the zero polynomial is not as clear. In this setting we drop the adjective "robust" and focus simply on hitting sets for the closure of a circuit class. The preceding discussion on the relationship between PIT and the complexity of ideals extends to border complexity.

Designing hitting sets for the closures of circuit classes has been explored as a possible avenue towards resolving grand challenges in polynomial identity testing. Recent work by Medini and Shpilka [52] and Saha and Thankey [61] studied PIT for *orbits* of various classes C. The orbit $\mathrm{orb}(C)$ of a class C corresponds to polynomials of the form $f(A\overline{x} + \overline{b})$, where $f(\overline{x}) \in C$ and A is an invertible $n \times n$ matrix. Studying PIT for orbits is motivated by the fact that for many simple classes C, there is a far richer class $\mathcal{D}$ such that $\overline{\mathrm{orb}(C)} = \overline{\mathcal{D}}$. That is, in order to derandomize PIT for a powerful class $\mathcal{D}$, it suffices to construct hitting set generators for the closure of the much simpler class $\mathrm{orb}(C)$. Unfortunately, this is not always feasible; for example, Medini and Shpilka [52] showed that at least one instantiation of their hitting sets does not extend to the closure of the circuit class it hits.

1.3 The Ideal Proof System

A central question of proof complexity is the following: given an unsatisfiable CNF formula φ, what is the length of the shortest proof of the unsatisfiability of φ? This question can be instantiated with a myriad of different proof systems rooted in logic, algebra, and geometry. Our focus in this work will be on a proof system based in algebra, namely the Ideal Proof System of Grochow and

Pitassi [34]. For a more comprehensive treatment of other proof systems (and proof complexity in general), see the recent book of Krajíček [49].

Let φ be an unsatisfiable 3CNF formula. One way to prove that φ is unsatisfiable is to translate φ into a system of polynomial equations, swapping the roles of 0 and 1, as follows. The literals x and $\neg x$ are translated into the polynomials $1-x$ and x, respectively. A clause $\ell_1 \vee \ell_2 \vee \ell_3$ becomes the polynomial $p_{\ell_1} p_{\ell_2} p_{\ell_3}$, where p_{ℓ_i} is the polynomial corresponding to the literal ℓ_i. Let $f_1, \ldots, f_m$ be the polynomials obtained from the clauses of φ. It is not hard to see that φ is satisfiable if and only if there is a $\{0,1\}$-valued solution to the system of equations $f_1 = \cdots = f_m = 0$; equivalently, φ is satisfiable if and only if there is a solution to the system $f_1 = \cdots = f_m = x_1^2 - x_1 = \cdots = x_n^2 - x_n = 0$.

Thus, to show that φ is unsatisfiable, it suffices to prove that a system of polynomial equations is unsatisfiable. This can be done by finding polynomials $g_1(\overline{x}), \ldots, g_m(\overline{x})$ and $h_1(\overline{x}), \ldots, h_n(\overline{x})$ such that $\sum_{i=1}^m g_i(\overline{x}) f_i(\overline{x}) + \sum_{i=1}^n h_i(\overline{x})(x_i^2 - x_i) = 1$, or more succinctly, by showing that 1 is in the ideal generated by $\{f_1, \ldots, f_m, x_1^2 - x_1, \ldots, x_n^2 - x_n\}$. As a consequence of Hilbert's Nullstellensatz, such a refutation always exists, provided the system is unsatisfiable. These refutations and various notions of their complexity give rise to the Nullstellensatz [7] and Polynomial Calculus [18] proof systems, both of which are well-studied and for which lower bounds are known [7, 15, 40, 60].

The recent Ideal Proof System (abbreviated as IPS) of Grochow and Pitassi [34] measures the complexity of a refutation by the algebraic circuit complexity of the certificate $\sum_i g_i f_i + \sum_i h_i(x_i^2 - x_i)$ when the f_i and $x_i^2 - x_i$ are provided as part of the input to the circuit. Because a refutation in the IPS is written as an algebraic circuit, there are connections between algebraic circuit lower bounds and lower bounds for the IPS. Grochow and Pitassi [34] proved that super-polynomial lower bounds on the size of IPS refutations of a family of CNF formulas imply VP $\neq$ VNP. As a proof system, the IPS is very powerful: Grochow and Pitassi [34] showed that the IPS polynomially simulates Extended Frege, itself a strong logic-based proof system. This simulation also behaves nicely if we consider IPS refutations coming from a restricted circuit class C. For example, over a field of characteristic $p > 0$, the constant-depth version of the IPS polynomially simulates $\mathrm{AC}^0[p]$-Frege, a proof system notorious for its current lack of super-polynomial lower bounds.

Lower bounds, both conditional and unconditional, are known for the IPS. Conditionally, Alekseev et al. [3] showed that the Shub–Smale hypothesis implies super-polynomial lower bounds on the size of IPS refutations of a particular instance of subset sum. Later work by Santhanam and Tzameret [62] showed that over finite fields, if there is an explicit family of polynomials that cannot be computed by polynomial-size algebraic circuits, then a particular family of CNF formulas cannot be refuted by polynomial-size IPS refutations. Combined with earlier work by Grochow and Pitassi [34], this establishes that over finite fields, proving super-polynomial lower bounds for the IPS is equivalent to proving super-polynomial lower bounds for algebraic circuits. Forbes et al. [32] used techniques from algebraic circuit complexity to prove unconditional lower bounds for restricted subsystems of the IPS, including those computed by depth-three powering formulas, read-once algebraic branching programs, and multilinear formulas.

The Ideal Proof System is defined in terms of algebraic circuits, so it is natural to expect progress on IPS lower bounds to mirror progress on lower bounds for algebraic circuits. Empirically, this has been the case, although additional effort is required to translate circuit lower bounds into IPS lower bounds. To prove circuit lower bounds, one only needs to show that a single polynomial cannot be computed by small circuits. In contrast, to prove lower bounds on the circuit size of IPS refutations of a system of polynomials, it is necessary to show that small circuits cannot compute any valid refutation.

Luckily, the set of IPS refutations of a fixed system of equations exhibits some algebraic structure: all refutations of a fixed system of polynomials lie in a coset of a particular ideal, as observed by Grochow and Pitassi [34, Section 6]. Thus, one can try to prove lower bounds for the IPS by proving circuit lower bounds for nonzero cosets of ideals. To the best of our knowledge, the only known lower bounds for nonzero cosets of ideals are those that follow from previously-mentioned lower bounds on the IPS. Notably, these proofs do not directly establish lower bounds for cosets of ideal, but rather reduce the task of proving IPS lower bounds to the more-tractable task of proving algebraic circuit lower bounds. One could hope that by better understanding the complexity of (cosets of) ideals, this progress could be used to prove lower bounds for IPS and restricted variants thereof. We refer the interested reader to Grochow and Pitassi [34] and Grochow [33] for further details.

For more on the Ideal Proof System, see the recent survey of Pitassi and Tzameret [59].

1.4 Our Results

We now describe our results in more detail. Throughout this subsection, we let X denote an $n \times m$ matrix of variables and $I^{\det}_{n,m,r} \subseteq \mathbb{F}[X]$ the ideal generated by the $r \times r$ minors of X. For simplicity, we state our results over fields of characteristic zero (such as the rational or complex numbers).

1.4.1 Complexity of Determinantal Ideals. Our main theorem constructs, for any nonzero polynomial $f(X) \in I^{\det}_{n,m,r}$, a small f-oracle circuit that approximately computes the $s \times s$ determinant for $s = \Theta(r^{1/3})$. This answers a question of Grochow [33, Conjecture 6.3] in the setting of border complexity.

Theorem 1.1 (Informal version of Theorem 3.15 and Corollary 3.16)**.** *Let $\mathbb{F}$ be a field of characteristic zero. Let X be an $n \times m$ matrix of variables and let $I^{\det}_{n,m,r} \subseteq \mathbb{F}[X]$ be the ideal generated by the $r \times r$ minors of X. Let $f(X) \in I^{\det}_{n,m,r}$ be a nonzero polynomial. Then there is a depth-three f-oracle circuit of size $O(n^2 m^2)$ that approximately computes the $s \times s$ determinant for $s = \Theta(r^{1/3})$.*

More generally, the conclusion of Theorem 1.1 holds if the determinant is replaced by any polynomial g that can be approximately computed by an algebraic branching program with r vertices. The conclusion of Theorem 1.1 also holds if we have oracle gates that approximately compute f instead of oracles that compute f exactly.

An immediate consequence of Theorem 1.1 is that for formulas and low-depth circuits, the border complexity of any nonzero

polynomial in $I^{\det}_{n,m,r}$ is at least as large as the border complexity of the $\Theta(r^{1/3}) \times \Theta(r^{1/3})$ determinant, up to polynomial factors. To the best of our knowledge, the only complexity lower bounds for the ideal $I^{\det}_{n,m,r}$ known prior to this work are due to Wiersig [74] and Forbes et al. [32, Corollary 6.7], who showed that every nonzero polynomial in $I^{\det}_{n,m,r}$ is $\exp(\Omega(r))$-hard for several weak circuit classes.

To prove Theorem 1.1, we have to reason about arbitrary polynomials in $I^{\det}_{n,m,r}$. That is, if $\{g_1, \ldots, g_N\}$ are the $r \times r$ minors of X, we have to consider all nonzero polynomials of the form $\sum_{i=1}^{N} f_i g_i$, where the f_i are arbitrary polynomials. This is difficult in part because if we apply a linear change of variables $X \mapsto L(X)$, it is not clear how to control the behavior of the f_i. To circumvent this, we use an alternate basis for $\mathbb{F}[X]$ instead of the monomial basis. This alternate basis consists of products of minors (of possibly different sizes) of X that satisfy a particular combinatorial condition; these products are known as *standard bideterminants*. Working in this basis, we gain a better understanding of how the multiplicands f_i behave under a change of variables.

The proof of Theorem 1.1 then proceeds in two steps. First, we find a change of variables that takes a polynomial $f \in I^{\det}_{n,m,r}$ to an approximation (in the border complexity sense) of a standard bideterminant $h(X)$ in the support of f. The analysis of this step crucially relies on the use of the standard bideterminant basis and its properties, which we describe in Subsection 3.1. Because f lies in the ideal $I^{\det}_{n,m,r}$, one can show that $h(X)$ is divisible by a $t \times t$ minor of X for some $t \geqslant r$. The second step is to find a projection of $h(X)$ to the $\Theta(r^{1/3}) \times \Theta(r^{1/3})$ determinant. Since h may be a product of minors of varying sizes, we need to find a projection that (1) behaves nicely on small minors of X and (2) allows us to deal with the possibility that h may be a large power of a minor. We accomplish this by modifying an argument of Valiant [73].

1.4.2 Pfaffian Ideals. Let Y be a $2n \times 2n$ skew-symmetric matrix. It is well-known that the determinant of Y is the square of another polynomial, the *Pfaffian* $\mathrm{Pf}(Y)$ of Y. Let $I^{\mathrm{pfaff}}_{2n,2n} \subseteq \mathbb{F}[Y]$ be the ideal generated by the Pfaffians of the $2r \times 2r$ principal submatrices of Y. Our next result is an analogue of Theorem 1.1 for the ideal $I^{\mathrm{pfaff}}_{2n,2r}$.

Theorem 1.2. *Let $\mathbb{F}$ be a field of characteristic zero. Let Y be a $2n \times 2n$ skew-symmetric matrix of variables and let $I^{\mathrm{pfaff}}_{2n,2r} \subseteq \mathbb{F}[Y]$ be the ideal generated by the Pfaffians of the $2r \times 2r$ principal submatrices of Y. Let $f(Y) \in I^{\mathrm{pfaff}}_{2n,2r}$ be a nonzero polynomial. Then there is a depth-three f-oracle circuit of size $O(n^4)$ that approximately computes the $s \times s$ Pfaffian for $s = \Theta(r^{1/3})$.*

The proof of Theorem 1.2 is similar to that of Theorem 1.1. The primary difference is that we now express polynomials in $I^{\mathrm{pfaff}}_{2n,2r}$ in an alternate basis consisting of products of Pfaffians of principal submatrices of Y. Along the way, we modify some of the technical details of the construction to accommodate for Pfaffians instead of determinants.

We remark that because the Pfaffian is the square root of the skew-symmetric determinant (in the sense that $\mathrm{Pf}(Y)^2 = \det(Y)$), it is natural to attempt proving Theorem 1.2 using Theorem 1.1. For any polynomial $f(\overline{x})$, one can use the Taylor series expansion of $\sqrt{1 + x^2}$ to construct a small $f(\overline{x})^2$-oracle circuit that computes $f(\overline{x})$. Combining this with Theorem 1.1, one obtains an analogue of Theorem 1.1 for the ideal generated by the squares of sub-Pfaffians of Y, which is weaker than Theorem 1.2 above.

1.4.3 The Space of Partial Derivatives in Determinantal Ideals. The remainder of our work consists of three applications of Theorem 1.1 and its proof, the first of which is to algebraic circuit complexity. For a polynomial $f \in \mathbb{F}[X]$, let $\partial_{<\infty}(f)$ denote the span of the partial (Hasse) derivatives of f. The dimension of $\partial_{<\infty}(f)$ and related spaces has been used successfully as a complexity measure in proving lower bounds for restricted classes of algebraic circuits (see the survey of Saptharishi [63] for more on this). While Theorem 1.1 shows that computing a polynomial in $I^{\det}_{n,m,r}$ is not much harder than computing the $\Theta(r^{1/3}) \times \Theta(r^{1/3})$ determinant, it is natural to ask if there are polynomials in $I^{\det}_{n,m,r}$ that are "simpler" than the $r \times r$ determinant with respect to complexity measures like $\dim(\partial_{<\infty}(\bullet))$. Our next result shows that among nonzero polynomials in the ideal $I^{\det}_{n,m,r}$, the $r \times r$ determinant in fact minimizes the value of $\dim(\partial_{<\infty}(\bullet))$.

Theorem 1.3. *For every nonzero polynomial $f(X) \in I^{\det}_{n,m,r}$, we have $\dim(\partial_{<\infty}(f)) \geqslant \dim(\partial_{<\infty}(\det_r)) = \binom{2r}{r}$.*

Using tools developed in the proof of Theorem 1.1, we can easily reduce the task of proving Theorem 1.3 to the case where $f(X)$ is a product of minors of X. As f is in the ideal $I^{\det}_{n,m,r}$, at least one factor of f must be an $s \times s$ minor of X for some $s \geqslant r$. We can then directly bound $\dim(\partial_{<\infty}(f))$ from below by a slight generalization of the argument used to bound $\dim(\partial_{<\infty}(\det_s))$.

We note that one can easily prove a lower bound of 2^r on $\dim(\partial_{<\infty}(f))$ using observations due to Forbes et al. [32]. Our result improves on this, obtaining an optimal bound of $\binom{2r}{r} = \Theta(4^r/\sqrt{r})$.

1.4.4 Polynomial Identity Testing for Low-Depth Circuits and Formulas. Next, we use Theorem 1.1 to derandomize special cases of polynomial identity testing. It is a straightforward consequence of Theorem 1.1 that for circuit classes like low-depth circuits and formulas, computing any nonzero element of $I^{\det}_{n,m,r}$ is effectively as hard as computing the $\Theta(r^{1/3}) \times \Theta(r^{1/3})$ determinant. Over an algebraically closed field, the ideal $I^{\det}_{n,m,r}$ can be equivalently described as the ideal of polynomials that vanish on matrices of rank less than r. Using this alternate description, we construct hitting set generators that unconditionally hit the closure of small low-depth circuits and conditionally hit the closure of small formulas.

Theorem 1.4. *Let $\mathbb{F}$ be a field of characteristic zero. For every $k \in \mathbb{N}$, there is a hitting set generator $\mathcal{G}_k$ with seed length $n^{1/2^k + o(1)}$ and degree 2^k that hits the closure of polynomial-size low-depth algebraic circuits. The generator $\mathcal{G}_k$ can be computed by either (1) a circuit of product-depth k and size $n^{1+o(1)}$, or (2) a formula of size $n^{1+o(1)}$. Assuming the border formula complexity of the determinant is super-polynomial, the generator $\mathcal{G}_k$ is also a hitting set generator for the closure of polynomial-size algebraic formulas.*

Our hitting set generators are very simple to describe. For $k = 1$, our generator takes as input two matrices of variables Y and Z, where Y is a $\sqrt{n} \times n^{o(1)}$ matrix and Z is an $n^{o(1)} \times \sqrt{n}$ matrix, and outputs the matrix product YZ. For $k \geqslant 2$, we construct the

generator $\mathcal{G}_k$ by arranging the input variables of $\mathcal{G}_{k-1}$ into a square matrix and replacing them with the product of an $n^{1/2^k+o(1)} \times n^{o(1)}$ matrix and an $n^{o(1)} \times n^{1/2^k+o(1)}$ matrix.

To prove that our generators correctly hit polynomial-size low-depth circuits, we must show that every small low-depth circuit does not vanish on the output of our generator. Using the description of $I^{\det}_{n,m,r}$ as the ideal of polynomials vanishing on matrices of rank at most r, establishing the correctness of our generators equates to proving that no small low-depth circuit can compute a polynomial in the ideal $I^{\det}_{\sqrt{n},\sqrt{n},n^{o(1)}}$. Such a lower bound follows in a straightforward manner by combining our Theorem 1.1 with the recent breakthrough lower bounds of Limaye et al. [50].

In the regime of $n^{\Theta(1)}$ seed length, our generators attain a near-optimal tradeoff between seed length and degree. It is not hard to show that a generator of seed length $n^{1/2^k+o(1)}$ must be of degree at least 2^k, and conversely that any generator of degree 2^k must have seed length at least $\Omega(n^{1/2^k})$. We also note that the circuit complexity of our generators is near-optimal, as any function with n outputs necessarily requires size $\Omega(n)$ to compute.

Prior to this, the best-known hitting set generator for low-depth circuits was given by Limaye et al. [50], using the hardness-to-randomness results of Chou et al. [17]. They obtained, for every fixed $\varepsilon > 0$, a generator with seed length $O(n^\varepsilon)$ and degree $O(\log n/\log\log n)$. Our construction attains the same seed length, but improves on the degree (as remarked above) and the circuit complexity of the generator. When instantiated to hit circuits of size s, the generator of Limaye et al. [50] necessarily has circuit complexity $\Omega(s)$. In contrast, our generator can be computed by a constant-depth circuit or formula of size $n^{1+o(1)}$, even when hitting low-depth circuits of size $O(n^{10^{100}})$.

For formulas, the best-known (conditional) constructions of hitting set generators prior to our work are due to Dvir et al. [27] and Chou et al. [17]. Both works yield generators with parameters similar to the low-depth generator of Limaye et al. [50] mentioned above (although the generator of [27] can only hit formulas of small individual degree). While our construction has better parameters, we use a stronger hardness assumption than what is needed by prior work. The constructions of Dvir et al. [27] and Chou et al. [17] can be instantiated with any explicit family of polynomials that requires formulas of super-polynomial size. In contrast, our construction depends crucially on super-polynomial lower bounds on the border formula complexity of the determinant. This is a stronger assumption, as the determinant is computable by polynomial-size branching programs and circuits, a fact which likely does not hold for all explicit families of polynomials.

1.4.5 Lower Bounds for the Ideal Proof System. Finally, we use Theorem 1.1 to prove lower bounds for the Ideal Proof System. Let X and Y be $n \times n$ matrices of variables and let I_n be the $n \times n$ identity matrix. Consider the system of polynomial equations given by $\{\det_n(X) = 0, XY - I_n = 0\}$. This system is unsatisfiable, as $\det_n(X) = 0$ if and only if X is non-invertible, while $XY - I_n = 0$ implies that X is invertible with inverse Y. We show that the constant-depth version of the Ideal Proof System cannot efficiently refute this system. Assuming lower bounds on the border formula complexity of the determinant, we also show that formula-IPS cannot efficiently refute this system. We remark that our lower bounds also hold when the boolean axioms $x^2_{i,j} - x_{i,j} = 0$ are included in the system of equations, but we suppress these here for brevity.

Theorem 1.5. *Let $\mathbb{F}$ be a field of characteristic zero. Let X and Y be $n \times n$ matrices of variables and let I_n be the $n \times n$ identity matrix. Then any IPS refutation of the system $\{\det(X) = 0, XY - I_n = 0\}$ cannot be approximately computed by a constant-depth circuit of polynomial size. Assuming the border formula complexity of the determinant is super-polynomial, then any IPS refutation of this system cannot be approximately computed by a formula of polynomial size.*

We do this by following the approach of Forbes et al. [32], who showed that lower bounds for the IPS can be derived from circuit lower bounds for multiples of a polynomial. Our choice of the system $\{\det_n(X) = 0, XY - I_n = 0\}$ is motivated by the fact that, using the techniques of [32], the desired IPS lower bounds follow from circuit lower bounds for multiples of the determinant. We can obtain the necessary lower bounds by combining our Theorem 1.1 with lower bounds against the determinant. In the case of low-depth circuits, our IPS lower bounds are unconditional thanks to the recent breakthrough circuit lower bounds of Limaye et al. [50]. For formula-IPS, our lower bounds remain conditional.

We also show that computing an IPS refutation of our hard instance $\{\det_n(X) = 0, XY - I_n = 0\}$ reduces to computing the determinant. Namely, we give a small depth-three circuit with $\det_n$-oracle gates that computes an IPS refutation of our hard instance. Passing to border complexity, this shows that the approximative complexity of the smallest IPS refutation of $\{\det_n(X) = 0, XY - I_n = 0\}$ is sandwiched between the approximative complexity of the $\Theta(n^{1/3}) \times \Theta(n^{1/3})$ and $n \times n$ determinants.

The strongest unconditional lower bounds for the IPS prior to our work are due to Forbes et al. [32], who proved lower bounds for subsystems of the IPS computed by restricted classes of circuits, including read-once oblivious algebraic branching programs and multilinear formulas. Impagliazzo et al. [39] showed that the constant-depth version of Polynomial Calculus (PC) over finite fields is surprisingly strong. The size of a constant-depth IPS refutation is essentially the number of lines in a constant-depth PC refutation, so lower bounds for constant-depth IPS over finite fields imply comparable lower bounds for constant-depth PC. However, our lower bounds do not extend to finite fields, nor do our lower bounds hold for refutations of an unsatisfiable CNF, so we are unable to conclude lower bounds for constant-depth PC and related proof systems.

We also mention a recent work of Alekseev [2], who proved lower bounds on the bit-size of refutations in a version of PC augmented with an extension rule. This is somewhat incomparable to our result: Alekseev's proof system allows for proofs of arbitrary depth, but must pay to use constants of large bit complexity; on the other hand, we work with a low-depth proof system that can use arbitrary rational numbers (or even arbitrary complex numbers) for free. Our lower bound is on circuit size, which is analogous to the number of lines in PC, whereas Alekseev's lower bound is on the number of bits needed to write down a refutation, which does not necessarily imply a lower bound on the number of proof lines.

1.5 Organization

In the remainder of this paper, we give a proof of Theorem 1.1. Proofs of the other results can be found in the full version of this work available on arXiv.[1]

2 PRELIMINARIES

For a natural number $n \in \mathbb{N}$, we write $[n] \coloneqq \{1, 2, \ldots, n\}$. We use $\overline{x} = (x_1, \ldots, x_n)$ to denote a vector of variables and $X = (x_{i,j})_{i\in[n], j\in[m]}$ to denote a matrix of variables. Given a field $\mathbb{F}$ and an indeterminate x, we write $\mathbb{F}[x]$ for the ring of polynomials in x with coefficients from $\mathbb{F}$ and $\mathbb{F}(x)$ for the field of rational functions in x with $\mathbb{F}$-coefficients. For a matrix $A \in \mathbb{F}^{n\times m}$ and sets $R \subseteq [n]$, $C \subseteq [m]$, we denote by $A_{R,C}$ the submatrix of A whose rows and columns are taken from the sets R and C, respectively. If X is an $n \times m$ matrix of variables, then for $r \leqslant \min(n, m)$ we denote by $I^{\det}_{n,m,r} \subseteq \mathbb{F}[X]$ the ideal of $\mathbb{F}[X]$ generated by the $r \times r$ minors of X.

We endow $\mathbb{F}[X]$ with a $(\mathbb{N}^n \oplus \mathbb{N}^m)$-grading in the following way. Let $\overline{e}_i \in \mathbb{N}^n$ denote the element of $\mathbb{N}^n$ with 1 in the i^{th} position and zeroes elsewhere. By abuse of notation, we also use $\overline{e}_i$ to denote the corresponding element of $\mathbb{N}^m$. We assign degree $\overline{e}_i \oplus \overline{e}_j$ to the variable $x_{i,j}$ and extend this to $\mathbb{F}[X]$ in the natural way. The degree of an element $f \in \mathbb{F}[X]$ with respect to this grading is called the *multidegree* of f, written multideg(f). We say an element of $\mathbb{F}[X]$ is *multihomogeneous* if it is homogeneous with respect to this grading.

We assume familiarity with the basic notion of an algebraic circuit and restricted classes thereof, including formulas, branching programs, and bounded-depth circuits. The interested reader may consult the surveys of Shpilka and Yehudayoff [71] and Saptharishi [63] or the text of Bürgisser et al. [14] for more on algebraic circuits. We also use the notion of monomial orders; for definitions, we refer the reader to Cox et al. [19, Chapter 2].

A key notion in this work is border complexity, a modification of the standard definition of algebraic computation. Briefly, a circuit C *border computes* a polynomial $f(\overline{x})$ if C is defined over $\mathbb{F}(\varepsilon)$ and computes a polynomial such that

$$C(\overline{x}) = f(\overline{x}) + \varepsilon \cdot g(\overline{x}, \varepsilon),$$

where $g \in \mathbb{F}[\overline{x}, \varepsilon]$, i.e., there are no negative powers of ε appearing in g. We abbreviate this as $C(\overline{x}) = f(\overline{x}) + O(\varepsilon)$. Over the complex numbers (or more generally, over a field of characteristic zero), one can think of C as computing f in the limit as $\varepsilon \to 0$.

3 HARDNESS OF DETERMINANTAL IDEALS

Recall that X denotes an $n\times m$ matrix of variables and $I^{\det}_{n,m,r} \subseteq \mathbb{F}[X]$ is the ideal generated by the $r \times r$ minors of X. In this section, we study the minimum possible border complexity of a nonzero polynomial in $I^{\det}_{n,m,r}$. Our main result is that, up to polynomial factors, there is no polynomial $f \in I^{\det}_{n,m,r}$ that is easier to compute than the $r \times r$ determinant. We do this by constructing, for every nonzero $f \in I^{\det}_{n,m,r}$, a depth-three f-oracle circuit that border computes the $\Theta(r^{1/3}) \times \Theta(r^{1/3})$ determinant.

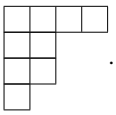
The argument proceeds in two steps. First, we show that for every $f(X) \in I^{\det}_{n,m,r}$, there is a linear change of variables that takes $f(X)$ to $(S|T)(X) + O(\varepsilon)$ for some bideterminant $(S|T)$ of width at least r. The analysis of this step crucially relies on the so-called straightening law, which we describe in Subsection 3.1. Second, for any $g(\overline{y})$ computed by an ABP of size at most r and any bideterminant $(S|T)(X)$ of width r, we construct a depth-three $(S|T)$-oracle circuit computing $g(\overline{y}) + O(\varepsilon)$. As the determinant can be efficiently computed by ABPs, composing these steps yields an f-oracle circuit for $\det_{\Theta(r^{1/3})}(X) + O(\varepsilon)$.

3.1 Bideterminants and the Straightening Law

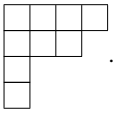
The proof of Theorem 3.15 relies on understanding how a polynomial $f \in I^{\det}_{n,m,r}$ behaves under the map $X \mapsto AXB$ for invertible matrices A and B. For example, it is easy to see that $f(AXB)$ also lies in $I^{\det}_{n,m,r}$. However, it is not clear if there is other structure we may take advantage of. By working in a different basis of $\mathbb{F}[X]$, we can better understand how $f(AXB)$ relates to $f(X)$. Before describing this basis, we recall the notions of a Young diagram and Young tableau.

Definition 3.1. A *partition* $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_k)$ is a non-increasing sequence of natural numbers. If $\sum_{i=1}^{k} \sigma_i = n$, we write $\sigma \vdash n$. The *transpose* of σ, denoted $\hat{\sigma}$, is the partition given by $\hat{\sigma}_i = \left|\{j : \sigma_j \geqslant i\}\right|$. Associated with a partition σ is its *Young diagram* $D_\sigma \subseteq \mathbb{N} \times \mathbb{N}$, given by $D_\sigma = \{(i, j) : j \leqslant \sigma_i\}$. ◊

Note that $\hat{\sigma}_1$ counts the number of rows in the Young diagram of σ. We graphically depict the Young diagram of a partition as a collection of boxes. For example, the Young diagram of the partition $(4, 2, 2, 1)$ is



This partition has transpose $(4, 3, 1, 1)$, with Young diagram given by



The lexicographic ordering on integer sequences induces an ordering on partitions, which we denote by $<_{\text{lex}}$.

We now define Young tableaux, which can be obtained by writing a number in each cell of the Young diagram of some partition σ.

Definition 3.2. Given a partition σ, a *Young tableau* T *of shape* σ is a map $T : D_\sigma \to \mathbb{N}$ assigning a natural number to each cell of the Young diagram of σ. We denote the i^{th} row of T by $T(i, \bullet)$, which we will view as either a set or a one-row Young tableau depending on context. A Young tableau is *standard* if its entries are strictly increasing along each column and along each row. A Young tableau is *semistandard* if its entries are strictly increasing along each column and are nondecreasing along each row. If $T : D_\sigma \to \mathbb{N}$ is a Young tableau, its *conjugate tableau* $\hat{T} : D_{\hat{\sigma}} \to \mathbb{N}$ is given by $\hat{T}(i, j) = T(j, i)$. ◊

[1] https://arxiv.org/abs/2112.00792

Continuing the example above, one Young tableau (of many) of shape $(4,2,2,1)$ is given by

$$\begin{array}{|c|c|c|c|}\hline 1&2&4&3\\\hline 1&2\\\cline{1-2} 4&1\\\cline{1-2} 3\\\cline{1-1}\end{array}\,.$$

Next, we introduce bitableaux and bideterminants. A bitableau is simply a pair of Young tableau of the same shape, while a bideterminant is a natural polynomial associated to this pair of tableaux.

Definition 3.3. Let $X = (x_{1,1}, \ldots, x_{n,n})$ be an $n \times n$ matrix of variables. A *bitableau* (S, T) is a pair of Young tableaux of the same shape σ. If the entries of S and T are from $[n]$, we associate to (S, T) the *bideterminant* $(S|T)(X)$, defined as

$$(S|T)(X) := \prod_{i=1}^{\hat{\sigma}_1} \det \begin{pmatrix} x_{S(i,1),T(i,1)} & x_{S(i,1),T(i,2)} & \cdots & x_{S(i,1),T(i,\sigma_i)} \\ x_{S(i,2),T(i,1)} & x_{S(i,2),T(i,2)} & \cdots & x_{S(i,2),T(i,\sigma_i)} \\ \vdots & \vdots & \ddots & \vdots \\ x_{S(i,\sigma_i),T(i,1)} & x_{S(i,\sigma_i),T(i,2)} & \cdots & x_{S(i,\sigma_i),T(i,\sigma_i)} \end{pmatrix}.$$

The i^{th} term in this product is the determinant of the submatrix whose rows and columns are listed in the i^{th} row of the tableaux S and T, respectively. The *width* of the bideterminant $(S|T)$ is given by σ_1. We say that the bitableau (S, T) and bideterminant $(S|T)$ are *standard* if, as tableaux, both S and T are increasing along each row and nondecreasing along each column (equivalently, that S and T are both the conjugate of a semistandard Young tableau). ◊

For example, associated to the bitableau

$$\left(\begin{array}{|c|c|c|}\hline 1&2&3\\\cline{1-3} 1&3\\\cline{1-2} 4\\\cline{1-1}\end{array}\,,\ \begin{array}{|c|c|c|}\hline 1&3&4\\\cline{1-3} 2&4\\\cline{1-2} 3\\\cline{1-1}\end{array}\right)$$

is the bideterminant

$$\det\begin{pmatrix} x_{1,1} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,3} & x_{3,4} \end{pmatrix} \det\begin{pmatrix} x_{1,2} & x_{1,4} \\ x_{3,2} & x_{3,4} \end{pmatrix} \det\begin{pmatrix} x_{4,3} \end{pmatrix}.$$

Note that a bideterminant $(S|T)$ is multihomogeneous of degree $(s_1\overline{e}_1 + \cdots + s_n\overline{e}_n) \oplus (t_1\overline{e}_1 + \cdots + t_n\overline{e}_n)$, where s_i and t_i count the number of occurrences of i in S and T, respectively.

It is easy to see that the bideterminants span $\mathbb{F}[X]$, since a monomial $\prod_{i=1}^{d} x_{r_i,c_i}$ is the bideterminant corresponding to the bitableau

$$\left(\begin{array}{|c|}\hline r_1\\\hline r_2\\\hline \cdots\\\hline r_d\\\hline\end{array}\,,\ \begin{array}{|c|}\hline c_1\\\hline c_2\\\hline \cdots\\\hline c_d\\\hline\end{array}\right).$$

Perhaps surprisingly, there is a natural subset of the bideterminants which form a basis of $\mathbb{F}[X]$.

Theorem 3.4 ([22]). *The standard bideterminants form a basis of* $\mathbb{F}[X]$.

To show $\mathbb{F}[X]$ is spanned by standard bideterminants, it suffices to express non-standard bideterminants as linear combinations of standard bideterminants. The fact that this can be done, along with some additional structural information, is known as the straightening law. For more on the straightening law, including its history and its applications to invariant theory, see the introduction of Désarménien et al. [21].

Theorem 3.5 ([22], see also [20, 21]). *Let $(S|T)(X)$ be a bideterminant of shape σ. Then $(S|T)(X)$ can be expressed as a linear combination*

$$(S|T)(X) = \sum_{(A,B)} c_{A,B}(A|B)(X),$$

where the $c_{A,B}$ are integers and the sum ranges over all standard bitableaux (A, B) of shape τ such that $\tau \geqslant_{\text{lex}} \sigma$.

One immediate corollary of this is a characterization of polynomials in the ideal $I^{\det}_{n,m,r}$ by their support in the standard bideterminant basis.

Corollary 3.6. *A polynomial $f \in \mathbb{F}[X]$ is an element of the ideal $I^{\det}_{n,m,r}$ if and only if f is supported on bideterminants of width at least r.*

3.2 Transforming to a Single Bideterminant

For $i, j \in [n]$ with $i \neq j$, we define the *substitution operator* $\text{Sub}_{i\to j}$ acting on a conjugate semistandard Young tableau T as follows: for every row in T containing i but not j, substitute i with j and re-order the row to be in increasing order. Let $h^j_i(T)$ denote the number of rows of T changed by applying $\text{Sub}_{i\to j}$ to T. In general, the map $T \mapsto (\text{Sub}_{i\to j}(T), h^j_i(T))$ may not be injective. However, the following lemma shows that mapping is injective when restricted to tableaux satisfying a particular property.

Lemma 3.7 ([20, Proposition 1.6]). *Let $i, j \in [n]$. Suppose T is a conjugate semistandard tableau with entries in $[n]$ with the property that if a row of T contains an integer $k \leqslant i$, then that row contains all integers in $\{i, i+1, \ldots, j-1\}$. Then $\text{Sub}_{i\to j}(T)$ is also a conjugate semistandard tableau and T is determined by $\text{Sub}_{i\to j}(T)$ and $h^j_i(T)$.*

While the condition in the above lemma seems strange at first, it arises in a natural way when one repeatedly applies the $\text{Sub}_{i\to j}$ operators as described by the next claim.

Claim 3.8 (implicit in proof of [20, Corollary 1.7]). *Let T be a conjugate semistandard tableau with entries in $[n]$. Let*

$$(1,2) \prec (1,3) \prec \cdots \prec (1,n) \prec (2,3) \prec \cdots \\ \prec (n-2,n-1) \prec (n-2,n) \prec (n-1,n)$$

be a partial order on $[n]^2$. Let $i, j \in [n]$ be such that $i < j$ and let (i', j') be the immediate predecessor of (i, j) in the $\prec$ order. Then the tableau

$$T' := \text{Sub}_{i'\to j'} \circ \cdots \circ \text{Sub}_{1\to 3} \circ \text{Sub}_{1\to 2}(T)$$

satisfies the hypothesis of Lemma 3.7 for (i, j). In other words, if a row of T' contains an integer $k \leqslant i$, then that row contains all integers in $\{i, i+1, \ldots, j-1\}$.

For a partition σ and natural number $n \in \mathbb{N}$, we let K_σ and $\overline{K}_\sigma$ denote the conjugate semistandard tableaux whose i^{th} row has entries $(1, \ldots, \sigma_i)$ and $(n - \sigma_i + 1, n - \sigma_i + 2, \ldots, n)$, respectively. For example, if $\sigma = (4,3,1)$ and $n = 5$, we have

$$K_{(4,3,1)} = \begin{array}{|c|c|c|c|}\hline 1&2&3&4\\\cline{1-4} 1&2&3\\\cline{1-3} 1\\\cline{1-1}\end{array} \qquad \overline{K}_{(4,3,1)} = \begin{array}{|c|c|c|c|}\hline 2&3&4&5\\\cline{1-4} 3&4&5\\\cline{1-3} 5\\\cline{1-1}\end{array}\,.$$

The operators $\mathrm{Sub}_{i\to j}$ provide a convenient way to transform an arbitrary conjugate semistandard tableau into $\overline{K}_\sigma$.

Lemma 3.9 ([20, Corollary 1.7]). *Let T be a conjugate semistandard tableau of shape σ. Then*

$$(\mathrm{Sub}_{n-1\to n}\circ\cdots\circ\mathrm{Sub}_{2\to 3}\circ\mathrm{Sub}_{1\to n}\circ\cdots\circ\mathrm{Sub}_{1\to 2})(T)=\overline{K}_\sigma.$$

Moreover, if we denote by h_i^j the number of times i is replaced by j in the application of $\mathrm{Sub}_{i\to j}$ above, then T is determined by σ and the h_i^j.

We are now ready to progress towards the main result of this section. Namely, for any nonzero $f\in I^{\det}_{n,m,r}$, we will find a linear change of variables that sends f to $(K_\sigma|K_\sigma)+O(\varepsilon)$ where σ is the shape of some standard bideterminant in the support of f when f is written in the standard bideterminant basis. For comparison, it is easy to do something similar in the monomial basis: given a polynomial $f(\overline{x})$ of degree d, there is some $m\in\mathbb{N}$ such that

$$\begin{aligned}&\varepsilon^m f(\varepsilon^{-(d+1)}x_1,\varepsilon^{-(d+1)^2}x_2,\dots,\varepsilon^{-(d+1)^n}x_n)\\&\quad=\mathrm{LC}_{\mathrm{lex}}(f)\,\mathrm{LM}_{\mathrm{lex}}(f)+O(\varepsilon)\end{aligned}$$

where $\mathrm{LC}_{\mathrm{lex}}(f)$ and $\mathrm{LM}_{\mathrm{lex}}(f)$ are the leading coefficient and leading monomial, respectively, of f in the lexicographic monomial order induced by $x_1\succ x_2\succ\cdots\succ x_n$, and $O(\varepsilon)$ denotes a polynomial in $\mathbb{F}[\varepsilon,\overline{x}]$ divisible by ε. To some extent, we are constructing an analogous change of variables in the bideterminant basis.

The main difficulty lies in finding a useful change of variables. In the monomial basis, individual terms can be distinguished by their degree, so it suffices to use a change of variables that only involves multiplying each x_i by some power of ε. However, in the bideterminant basis, multidegree is too coarse a notion to distinguish between bideterminants, so it seems that finding a clever substitution $x_{i,j}\mapsto\varepsilon^{d_{i,j}}x_{i,j}$ will not be enough.

We start by working in a larger polynomial ring $\mathbb{F}[X,\Lambda,\Xi]$. We will give two changes of variables: one that enforces structure on the tableaux encoding the rows of the bideterminants in the support of a polynomial f, and another that handles the tableaux encoding the columns of the bideterminants. The proof of this lemma is inspired by and borrows ideas from the proof of [20, Theorem 3.3].

Lemma 3.10. *Let $\Lambda=(\lambda_{i,j})$ be an $n\times n$ matrix of variables and let $\prec_\Lambda$ be the lexicographic monomial order on $\mathbb{F}[\Lambda]$ induced by the order $\lambda_{i,j}\succ\lambda_{k,\ell}$ if $i<k$ or $i=k$ and $j<\ell$. Likewise, let $\Xi=(\xi_{i,j})$ be an $m\times m$ matrix of variables and let $\prec_\Xi$ be the corresponding lexicographic monomial order on $\mathbb{F}[\Xi]$. Then there are matrices $M\in\mathbb{F}[\Lambda]^{n\times n}$ and $N\in\mathbb{F}[\Xi]^{m\times m}$ with $\det(M)=\pm1$ and $\det(N)=\pm1$ such that the following holds.*

Let $f(X)\in I^{\det}_{n,m,r}$ be a nonzero polynomial and let $f(X)=\sum_{k\in[s]}\alpha_k(S_k|T_k)(X)$ be the expansion of f in the standard bideterminant basis. For $k\in[s]$, let σ_k be the shape of the bideterminant $(S_k|T_k)$. Then there are nonempty sets $A,B\subseteq[s]$ such that

$$\begin{aligned}\mathrm{LC}_{\prec_\Lambda}(f(MX))&=\sum_{k\in A}\alpha_k(K_{\sigma_k}|T_k)(X)\\\mathrm{LC}_{\prec_\Xi}(f(XN))&=\sum_{k\in B}\alpha_k(S_k|K_{\sigma_k})(X),\end{aligned}$$

where we take leading coefficients in the rings $\mathbb{F}[X][\Lambda]$ and $\mathbb{F}[X][\Xi]$, respectively.

Proof. We first construct the matrix M and prove the corresponding claim. For $i,j\in[n]$ with $i\neq j$, let $E_{i,j}(z)$ be the $n\times n$ matrix with ones on the diagonal and z in the (i,j) entry. Let J_n be the $n\times n$ matrix whose (i,j) entry is 1 if $i+j=n+1$ and zero otherwise. We define the matrix M as

$$M\coloneqq E_{1,2}(\lambda_{1,2})\cdots E_{1,n}(\lambda_{1,n})E_{2,3}(\lambda_{2,3})\cdots E_{n-1,n}(\lambda_{n-1,n})J_n.$$

Since $\det(J_n)=\pm1$ and $\det(E_{i,j}(z))=1$ for $i\neq j$, it follows that $\det(M)=\pm1$.

We now analyze the polynomial $f(MX)$. Recall that for a tableau S, we denote by $h_i^j(S)$ the number of entries changed from i to j when we apply the operator $\mathrm{Sub}_{i\to j}$ to S. Observe that for a bideterminant $(S|T)$, it follows from properties of the determinant that

$$(S|T)(E_{i,j}(z)X)=z^{h_i^j(S)}(\mathrm{Sub}_{i\to j}(S)|T)(X)+O(z^{h_i^j(S)-1}),$$

where $O(z^{h_i^j(S)-1})$ denotes a polynomial in $\mathbb{F}[X][z]$ of degree at most $h_i^j(S)-1$. For $i,j\in[n]$ with $i\neq j$, define

$$f_{i,j}(X,\Lambda)\coloneqq f(E_{1,2}(\lambda_{1,2})\cdots E_{1,n}(\lambda_{1,n})E_{2,3}(\lambda_{2,3})\cdots E_{i,j}(\lambda_{i,j})X).$$

Note that $f(MX)=f_{n-1,n}(J_nX,\Lambda)$.

We claim that for every $i,j\in[n]$ with $i<j$, there is a non-empty set $A_{i,j}\subseteq[s]$ such that

$$\begin{aligned}&\mathrm{LC}_{\prec_\Lambda}(f_{i,j}(X,\Lambda))\\&\quad=\sum_{k\in A_{i,j}}\alpha_k(\mathrm{Sub}_{i\to j}\circ\cdots\circ\mathrm{Sub}_{2\to 3}\\&\qquad\qquad\circ\mathrm{Sub}_{1\to n}\circ\cdots\circ\mathrm{Sub}_{1\to 2}(S_k)|T_k)(X).\end{aligned}$$

By Lemma 3.9, this implies

$$\mathrm{LC}_{\prec_\Lambda}(f_{n-1,n}(X,\Lambda))=\sum_{k\in A_{n-1,n}}\alpha_k(\overline{K}_{\sigma_k}|T_k)(X).$$

Using the fact that $(\overline{K}_{\sigma_k}|T)(J_nX)=(K_{\sigma_k}|T_k)(X)$, this yields

$$\mathrm{LC}_{\prec_\Lambda}(f(MX))=\mathrm{LC}_{\prec_\Lambda}(f_{n-1,n}(J_nX,\Lambda))=\sum_{k\in A_{n-1,n}}\alpha_k(K_{\sigma_k}|T_k)(X)$$

as desired.

We now prove the claim by induction on (i,j) in the order $(1,2)\prec(1,3)\prec\cdots\prec(1,n)\prec(2,3)\prec\cdots\prec(n-1,n)$. Let (i',j') be the predecessor of (i,j) in the $\prec$ order. In the case that $(i,j)=(1,2)$, we abuse notation and set $f_{i',j'}\coloneqq f$ and $A_{i',j'}\coloneqq[s]$. Let

$$H_i^j\coloneqq\max_{k\in A_{i',j'}}h_i^j(\mathrm{Sub}_{i'\to j'}\circ\cdots\circ\mathrm{Sub}_{1\to 2}(S_k))$$

and

$$A_{i,j}=\{k\in A_{i',j'}:h_i^j(\mathrm{Sub}_{i'\to j'}\circ\cdots\circ\mathrm{Sub}_{1\to 2}(S_k))=H_i^j\}.$$

Note that $A_{i,j}$ is necessarily non-empty, as H_i^j is a maximum over a finite nonempty set. By induction, there is some $\overline{e}\in\mathbb{N}^{n\times n}$ such that

$$\begin{aligned}&f_{i',j'}(X,\Lambda)\\&\quad=\Lambda^{\overline{e}}\sum_{k\in A_{i',j'}}\alpha_k(\mathrm{Sub}_{i'\to j'}\circ\cdots\circ\mathrm{Sub}_{1\to 2}(S_k)|T_k)(X)+g(X,\Lambda),\end{aligned}$$

where $g(X,\Lambda)\in\mathbb{F}[X][\Lambda]$ is a polynomial in which every monomial is smaller than $\Lambda^{\overline{e}}$ in the $\prec_\Lambda$ order. Since $f_{i',j'}$ only depends

on $\lambda_{1,2}, \dots, \lambda_{i',j'}$, it follows that $\Lambda^{\overline{e}}$ is a monomial in only these variables. We then apply the definition of $f_{i,j}$ to obtain

$$\begin{aligned} &f_{i,j}(X,\Lambda) \\ &= f_{i',j'}(E_{i,j}(\lambda_{i,j})X,\Lambda) \\ &= \Lambda^{\overline{e}} \sum_{k\in A_{i',j'}} \alpha_k(\mathrm{Sub}_{i'\to j'} \circ \cdots \circ \mathrm{Sub}_{1\to 2}(S_k)|T_k)(E_{i,j}(\lambda_{i,j})X) \\ &\quad + g(E_{i,j}(\lambda_{i,j})X,\Lambda) \\ &= \Lambda^{\overline{e}}\lambda_{i,j}^{H_i^j} \sum_{k\in A_{i,j}} \alpha_k(\mathrm{Sub}_{i\to j} \circ \cdots \circ \mathrm{Sub}_{1\to 2}(S_k)|T_k)(X) \\ &\quad + \Lambda^{\overline{e}}p(X,\lambda_{i,j}) + g(E_{i,j}(\lambda_{i,j})X,\Lambda), \end{aligned}$$

where $p(X,\lambda_{i,j}) \in \mathbb{F}[X][\Lambda]$ is a polynomial of degree at most $H_i^j - 1$ in $\lambda_{i,j}$. This implies that every monomial of $\Lambda^{\overline{e}}p(X,\Lambda)$ is smaller than $\Lambda^{\overline{e}}\lambda_{i,j}^{H_i^j}$ in the $\prec_\Lambda$ order. Observe that the substitution $X \mapsto E_{i,j}(\lambda_{i,j})X$ only changes the $\lambda_{i,j}$-degree of any Λ-monomial in $g(X,\Lambda)$. In particular, because every monomial of $g(X,\Lambda)$ is smaller than $\Lambda^{\overline{e}}$ in the $\prec_\Lambda$ order, the same holds true for every Λ-monomial of $g(E_{i,j}(\lambda_{i,j})X,\Lambda)$. This implies that

$$\mathrm{LC}_{\prec_\Lambda}(f_{i,j}) = \sum_{k\in A_{i,j}} \alpha_k(\mathrm{Sub}_{i\to j} \circ \cdots \circ \mathrm{Sub}_{1\to 2}(S_k)|T_k)(X)$$

as claimed. This establishes the claimed properties of M.

To construct the matrix N, we overload notation and let $E_{i,j}(z)$ be the $m\times m$ matrix with ones on the diagonal and z in the (i,j) entry. Just as the matrix M consisted of a sequence of row operations, the matrix N will be composed of a sequence of column operations. We define N as

$$N := J_m E_{m-1,m}(\xi_{m-1,m}) \cdots E_{2,3}(\xi_{2,3}) E_{1,m}(\xi_{1,m}) \cdots E_{1,2}(\xi_{1,2}).$$

Since $\det(J_m) = \pm 1$ and $\det(E_{i,j}(z)) = 1$ for $i < j$, we get that $\det(N) = \pm 1$.

As in the previous case, it follows from properties of the determinant that for a bideterminant $(S|T)$, we have

$$(S|T)(XE_{i,j}(z)) = z^{h_i^j(T)}(S|\mathrm{Sub}_{i\to j}(T))(X) + O(z^{h_i^j(T)-1}).$$

Using this, the analysis of the leading coefficient of $f(XN) \in \mathbb{F}[X][\Xi]$ proceeds in a manner analogous to the case of $f(MX)$, so we omit the details. □

We will need the following lemma, which says that given a polynomial $f(\overline{x}) \in R[\overline{x}]$ where R is a commutative ring, we can substitute the $\overline{x}$ variables by powers of a new indeterminate ε to isolate the leading coefficient of f. The proof is similar to the process described following the statement of Lemma 3.9, but we omit it due to space constraints.

Lemma 3.11. *Let R be a commutative ring. Let $f(\overline{x}) \in R[\overline{x}]$ and let $\prec$ be a lexicographic monomial order on $\overline{x}$. Then there are natural numbers $m, d_1, \dots, d_n \in \mathbb{N}$ such that*

$$\varepsilon^m f(\varepsilon^{-d_1}, \dots, \varepsilon^{-d_n}) = \mathrm{LC}(f) + O(\varepsilon),$$

where $\mathrm{LC}(f) \in R$ is the leading coefficient of f with respect to $\prec$ and $O(\varepsilon^{m+1})$ denotes a polynomial in $R[\varepsilon]$ divisible by ε.

We now come to the main result of this subsection: a change of variables that sends a polynomial $f(X)$ to $(K_\sigma|K_\sigma)(X) + O(\varepsilon)$ where σ is the shape of some standard bideterminant in the support of f.

Proposition 3.12. *Let $f(X) \in I^{\det}_{n,m,r}$ be nonzero. There is a collection of nm linearly independent linear functions $\ell_{i,j}(X,\varepsilon) \in \mathbb{F}(\varepsilon)[X]$ indexed by $(i,j) \in [n]\times[m]$, an integer $q \in \mathbb{Z}$, a nonzero $\alpha \in \mathbb{F}$, and a partition σ with $\sigma_1 \geqslant r$ such that*

$$f(\ell_{1,1}(X,\varepsilon), \dots, \ell_{n,m}(X,\varepsilon)) = \varepsilon^q \alpha(K_\sigma|K_\sigma)(X) + O(\varepsilon^{q+1}).$$

Proof. Let $f = \sum_{k\in[s]} \alpha_k(S_k|T_k)$ be the expansion of f in the standard bideterminant basis. Let M and N be the matrices constructed in Lemma 3.10. Let $\prec$ denote the lexicographic order on $\mathbb{F}[X][\Lambda,\Xi]$ induced by $\lambda_{1,2} \succ \lambda_{1,3} \succ \cdots \succ \lambda_{n-1,n} \succ \xi_{1,2} \succ \cdots \succ \xi_{m-1,m}$. Lemma 3.10 implies that there is a non-empty set $A \subseteq [s]$ such that

$$g(X) := \mathrm{LC}_\prec(f(MX)) = \sum_{k\in A} \alpha_k(K_{\sigma_k}|T_k)(X),$$

and likewise that there is a non-empty set $B \subseteq A$ such that

$$\mathrm{LC}_\prec(g(XN)) = \sum_{k\in B} \alpha_k(K_{\sigma_k}|K_{\sigma_k})(X).$$

This implies that

$$\mathrm{LC}_\prec(f(MXN)) = \sum_{k\in B} \alpha_k(K_{\sigma_k}|K_{\sigma_k})(X),$$

where σ_k denotes the shape of the bideterminant $(S_k|T_k)$. By Corollary 3.6, each bideterminant in the above sum has width at least r, so we have $(\sigma_k)_1 \geqslant r$ for all $k \in A$.

Let y and z be new indeterminates and let $D := \deg(f(X))$. Consider the change of variables

$$x_{i,j} \mapsto y^{(D+1)^i} z^{(D+1)^j} x_{i,j}.$$

Let $h(X,\Lambda,\Xi,y,z)$ be the image of $f(MXN)$ under this map. By construction, an X-monomial of multidegree $(\sum_i a_i\overline{e}_i) \oplus (\sum_i b_i\overline{e}_i)$ is multiplied by a factor of $y^{\sum_i a_i(D+1)^i} z^{\sum_j b_j(D+1)^j}$. In particular, since $\max_i a_i \leqslant D$ and $\max_i b_i \leqslant D$, X-monomials of distinct multidegree have distinct (y,z)-degree under this mapping. Observe that $\mathrm{multideg}((K_\sigma|K_\sigma)(X)) \neq \mathrm{multideg}((K_\tau|K_\tau)(X))$ for distinct partitions $\sigma \neq \tau$. Since each bideterminant $(K_\sigma|K_\sigma)(X)$ is mapped to a unique (y,z)-degree under this substitution, we get that the polynomial

$$p(X) = \mathrm{LC}_{(y,z)}(\mathrm{LC}_{(\Lambda,\Xi)}(h(X,\Lambda,\Xi,y,z)))$$

is a nonzero multiple of the bideterminant $(K_{\sigma_k}|K_{\sigma_k})(X)$ for some $k \in B$. If we augment the monomial order $\prec$ by setting $\Lambda \succ \Xi \succ y \succ z$ and taking the corresponding lexicographic order, we then have

$$\mathrm{LC}_\prec(h(X,\Lambda,\Xi,y,z)) = \alpha_k(K_{\sigma_k}|K_{\sigma_k})(X)$$

for some $k \in B$.

Applying Lemma 3.11 to $h(X,\Lambda,\Xi,y,z)$ viewed as an element of $\mathbb{F}[X][\Lambda,\Xi,y,z]$, we get a map $\varphi : (\Lambda \cup \Xi \cup \{y,z\}) \to \{\varepsilon^{-d} : d \in \mathbb{N}\}$ such that

$$\varphi(h(X,\Lambda,\Xi,y,z)) = \varepsilon^q \alpha_k(K_{\sigma_k}|K_{\sigma_k})(X) + O(\varepsilon^{q+1})$$

for some integer q.

Note that $h(X, \Lambda, \Xi, y, z)$ was obtained from $f(X)$ by an invertible linear transformation of the X variables. That is, there are nm linearly independent linear polynomials $\ell'_{1,1}(X), \ldots, \ell'_{n,m}(X) \in \mathbb{F}[\Lambda, \Xi, y, z][X]$ such that

$$h(X, \Lambda, \Xi, y, z) = f(\ell'_{1,1}(X), \ldots, \ell'_{n,m}(X)).$$

Set $\ell_{i,j}(X, \varepsilon) := \varphi(\ell'_{i,j}(X)) \in \mathbb{F}(\varepsilon)[X]$ for each $(i, j) \in [n] \times [m]$. Since the transformation $x_{i,j} \mapsto \ell'_{i,j}(X)$ is invertible as long as $y \neq 0$ and $z \neq 0$, the transformation $x_{i,j} \mapsto \ell_{i,j}(X, \varepsilon)$ remains invertible under φ. Finally, it follows from the definition of φ that

$$\begin{aligned} f(\ell_{1,1}(X, \varepsilon), \ldots, \ell_{n,m}(X, \varepsilon)) &= f(\varphi(\ell'_{1,1}(X)), \ldots, \varphi(\ell'_{n,m}(X))) \\ &= \varphi(f(\ell'_{1,1}(X), \ldots, \ell'_{n,m}(X))) \\ &= \varphi(h(X, \Lambda, \Xi, y, z)) \\ &= \varepsilon^q \alpha_k(K_{\sigma_k}|K_{\sigma_k})(X) + O(\varepsilon^{q+1}). \quad \square \end{aligned}$$

3.3 Projecting to the Determinant

So far, we have constructed a linear change of variables taking a polynomial $f \in I^{\det}_{n,m,r}$ to $(K_\sigma|K_\sigma) + O(\varepsilon)$ for a bideterminant $(K_\sigma|K_\sigma)$ of width at least r. Next, we show that a $(K_\sigma|K_\sigma)$-oracle can be used to compute $g(\overline{y}) + O(\varepsilon)$, where g is any polynomial computable by an algebraic branching program on r vertices. Ideally, one would like to appeal to the VBP-completeness of the determinant, which gives a projection from $\det_r(X)$ to $g(\overline{y})$, to prove such a result. The difficulty lies in the fact that a bideterminant may be a product of multiple determinants of varying sizes. Because of this, we need a projection that behaves well on proper minors of X and also allows us to deal with the possibility that we may be projecting from a power of the determinant as opposed to the determinant itself. We almost construct such a projection, but we will need some post-processing in the form of an extra addition gate in order to handle powers of the determinant.

Let $g(\overline{y})$ be computable by a small algebraic branching program. We begin by describing a projection $\varphi : X \to \mathbb{F}[\overline{y}]$ of a generic matrix X where each $\varphi(x_{i,j})$ is a linear polynomial in $\overline{y}$ such that $\det(\varphi(X)) = 1 + g(\overline{y})$ and the leading principal minors of $\varphi(X)$ have determinant 1. This is a small modification of an argument due to Valiant [73, Theorem 1]; we include a proof for the sake of completeness.

Lemma 3.13. *Let $g(\overline{y}) \in \mathbb{F}[\overline{y}]$ and suppose g can be computed by a layered algebraic branching program on m vertices. Then there is an $m \times m$ matrix $A \in \mathbb{F}[\overline{y}]^{m \times m}$ whose entries are linear polynomials in $\overline{y}$ such that*

(1) $\det(A) = 1 + g(\overline{y})$, and
(2) for every $k \in [m-1]$, we have $\det(A_{[k],[k]}) = 1$.

Proof. We first recall the correspondence between cycle covers in graphs and the determinant. Let G be a weighted directed graph on m vertices and denote the weight of the edge (i, j) by $w(i, j)$. Let $A(G) = (a_{i,j})$ be the $m \times m$ matrix given by

$$a_{i,j} = \begin{cases} w(i,j) & (i,j) \in E(G) \\ 0 & (i,j) \notin E(G). \end{cases}$$

Recall that a *cycle cover* C of G is a collection of vertex-disjoint cycles in G which span the vertices of G. Let $CC(G)$ denote the collection of all cycle covers of G. Given a cycle cover C of G, let $\pi(C)$ denote the product of the edge weights in C. If every cycle cover of G consists of odd-length cycles, then the definitions of $A(G)$ and the determinant imply that

$$\det(A(G)) = \sum_{C \in CC(G)} \pi(C).$$

We now proceed with the proof of Lemma 3.13. Suppose $g(\overline{y})$ can be computed by a layered algebraic branching program on m nodes. Let s and t be the start and end nodes of this branching program, respectively. Since the program is layered, every s-t path has the same length. If the length of each s-t path is even, we add an edge of weight 1 from t to s and a self-loop of weight 1 to every vertex (including s and t); if the length of each s-t path is odd, we identify the vertices s and t with one another (resulting in a graph on $m-1$ nodes), add an isolated vertex r, and then add a self-loop to every vertex. Denote the resulting graph by G. In both cases, G has one cycle cover for every s-t path in the branching program, as well as a single cycle cover corresponding to the set of self-loops in the graph. Moreover, every cycle cover in G consists solely of odd-length cycles.

For a cycle cover C corresponding to an s-t path P in the branching program, it follows from the definition of G that $\pi(C) = \pi(P)$, where $\pi(P)$ is the product of the weights on the edges of P. If C is the all-self-loops cycle cover, then $\pi(C) = 1$. Since every cycle cover in G consists of odd-length cycles, we have

$$\det(A(G)) = \sum_{C \in CC(G)} \pi(C) = 1 + \sum_{P} \pi(P) = 1 + g(\overline{y}),$$

where the second summation is over all s-t paths P in the branching program. This proves the first part of the lemma.

To prove the second part, let $v_1, \ldots, v_m$ be a topological ordering of the vertices in the algebraic branching program. Note that $v_1 = s$ and $v_m = t$. If every s-t path in the branching program has even length, we order the rows and columns of $A(G)$ such that $A(G)_{i,j} = w(v_i, v_j)$. If instead every s-t path in the branching program has odd length, we set

$$A(G)_{i,j} = \begin{cases} w(r, v_j) & i = 1 \\ w(v_i, r) & j = 1 \\ w(v_i, v_j) & \text{otherwise,} \end{cases}$$

where r is the isolated vertex with a self-loop. In either case, note that if $i > j$ and $A(G)_{i,j} \neq 0$, then we must have $i = m$. This implies that for every $k \in [m-1]$, the matrix $A(G)_{[k],[k]}$ is upper-triangular with ones along the diagonal. Thus $\det(A(G)_{[k],[k]}) = 1$ as desired. $\square$

Although we want to construct an $(K_\sigma|K_\sigma)$-oracle circuit that computes any polynomial $g(\overline{y})$ that is computable by a small layered algebraic branching program, it will be convenient for us to assume that g is homogeneous. This is not restrictive, as one can always introduce a new variable z and consider the homogeneous polynomial $\hat{g}(\overline{y}, z) := z^{\deg(g)} g(y_1/z, \ldots, y_n/z)$, which specializes to $g(\overline{y})$ under the map $z \mapsto 1$. One needs to show that $\hat{g}(\overline{y}, z)$ is as easy to compute as $g(\overline{y})$. This can be done for layered ABPs by relabeling the edges of the ABP: if $\ell_e(\overline{y}) = \alpha_0 + \sum_{i=1}^n \alpha_i y_i$ is the polynomial

labeling edge e, replacing it with $\hat{\ell}_e(\overline{y}, z) = \alpha_0 z + \sum_{i=1}^n \alpha_i y_i$ results in a layered ABP that computes $z^d g(y_1/z, \ldots, y_n/z)$ for some $d \geqslant \deg(g)$. We record this as a lemma.

Lemma 3.14. *Let $g(\overline{y}) \in \mathbb{F}[\overline{y}]$ be a polynomial and suppose that g can be computed by a layered algebraic branching program on m vertices. Let z be a new variable. Then there is a homogeneous polynomial $\hat{g}(\overline{y}, z) \in \mathbb{F}[\overline{y}, z]$ such that $\hat{g}$ can be computed by a layered algebraic branching program on m vertices and that $\hat{g}(\overline{y}, 1) = g(\overline{y})$.*

Given a nonzero $f(X) \in I^{\det}_{n,m,r}$, we will use the preceding lemmas together with Proposition 3.12 to construct a depth-three f-oracle circuit computing $\det_{\Theta(r^{1/3})}(X) + O(\varepsilon)$. In fact, for any polynomial $g(\overline{y})$ computable by a layered algebraic branching program on r vertices, we can construct an f-oracle circuit computing g.

Theorem 3.15. *Let $f(X) \in I^{\det}_{n,m,r}$ be a nonzero polynomial.Let $g(\overline{y}) \in \mathbb{F}[\overline{y}]$ be a polynomial computable by a layered algebraic branching programs with at most r vertices. Then there is a depth-three f-oracle circuit Φ defined over $\mathbb{F}(\varepsilon)$ such that the following hold.*

1. *Φ has nm addition gates at the bottom layer, a single f-oracle gate in the middle layer, and a single addition gate at the top layer.*
2. *If $\mathrm{char}(\mathbb{F}) = 0$, then Φ computes $g(\overline{y}) + O(\varepsilon)$.*
3. *If $\mathrm{char}(\mathbb{F}) = p > 0$, then Φ computes $g(\overline{y})^{p^k} + O(\varepsilon)$ for some $k \in \mathbb{N}$.*

Proof. Applying Proposition 3.12 to $f(X)$, we obtain linear functions $\ell_{1,1}(X, \varepsilon), \ldots, \ell_{n,m}(X, \varepsilon)$, a nonzero $\alpha \in \mathbb{F}$, and some $q \in \mathbb{Z}$ such that

$$f(\ell_{1,1}(X, \varepsilon), \ldots, \ell_{n,m}(X, \varepsilon)) = \varepsilon^q \alpha (K_\sigma | K_\sigma)(X) + O(\varepsilon^{q+1})$$

for some partition σ of width at least r.

Lemma 3.14 implies that there is a homogeneous polynomial $\hat{g}(\overline{y}, z) \in \mathbb{F}[\overline{y}, z]$ computable by a layered algebraic branching program on at most r vertices such that $\hat{g}(\overline{y}, 1) = g(\overline{y})$. Since $\hat{g}(\overline{y}, z)$ can be computed by a layered algebraic branching program on at most r vertices, we can obtain a layered ABP on exactly r vertices computing $\hat{g}(\overline{y}, z)$ by adding isolated vertices. Let $A(\overline{y}, z) \in \mathbb{F}[\overline{y}, z]^{r \times r}$ be the matrix obtained by applying Lemma 3.13 to $\hat{g}(\overline{y}, z)$. Extend $A(\overline{y}, z)$ to an $n \times m$ matrix by adding ones along the main diagonal and zeroes elsewhere. Then we have

$$\begin{aligned}
&f(\ell_{1,1}(A(\overline{y}, z), \varepsilon), \ldots, \ell_{n,m}(A(\overline{y}, z), \varepsilon)) \\
&\quad = \varepsilon^q \alpha (K_\sigma | K_\sigma)(A(\overline{y}, z)) + O(\varepsilon^{q+1}) \\
&\quad = \varepsilon^q \alpha \prod_{i=1}^{\hat{o}_1} \det_{\sigma_i}(A(\overline{y}, z)_{[\sigma_i],[\sigma_i]}) + O(\varepsilon^{q+1}) \\
&\quad = \varepsilon^q \alpha \prod_{i:\sigma_i \geqslant r} \det_{\sigma_i}(A(\overline{y}, z)_{[\sigma_i],[\sigma_i]}) \cdot \prod_{i:\sigma_i < r} \det_{\sigma_i}(A(\overline{y}, z)_{[\sigma_i],[\sigma_i]}) \\
&\qquad + O(\varepsilon^{q+1}) \\
&\quad = \varepsilon^q \alpha \prod_{i:\sigma_i \geqslant r} (1 + \hat{g}(\overline{y}, z)) + O(\varepsilon^{q+1}).
\end{aligned}$$

Let $h(\overline{y}, \varepsilon, z) := f(\ell_{1,1}(A(\overline{y}, z), \varepsilon), \ldots, \ell_{n,m}(A(\overline{y}, z), \varepsilon))$ and let $t = |\{i : \sigma_i \geqslant r\}|$. The above establishes $h(\overline{y}, \varepsilon, z) = \varepsilon^q \alpha (1 + \hat{g}(\overline{y}, z))^t + O(\varepsilon^{q+1})$.

Suppose $\mathrm{char}(\mathbb{F}) = 0$. Under the substitution $y_i \mapsto \delta \cdot y_i$ and $z \mapsto \delta$, we have

$$\begin{aligned}
h(\delta \cdot \overline{y}, \varepsilon, \delta) &= \varepsilon^q \alpha (1 + \hat{g}(\delta \cdot \overline{y}, \delta))^t + O(\varepsilon^{q+1}) \\
&= \varepsilon^q \alpha (1 + \delta^{\deg(\hat{g})} \hat{g}(\overline{y}, 1))^t + O(\varepsilon^{q+1}) \\
&= \varepsilon^q \alpha (1 + \delta^{\deg(\hat{g})} g(\overline{y}))^t + O(\varepsilon^{q+1}) \\
&= \varepsilon^q \alpha \sum_{i=0}^{t} \binom{t}{i} \delta^{i \cdot \deg(\hat{g})} g(\overline{y})^i + O(\varepsilon^{q+1}) \\
&= \varepsilon^q \alpha + \varepsilon^q \delta^{\deg(\hat{g})} \alpha t g(\overline{y}) + O(\varepsilon^q \delta^{2\deg(\hat{g})}) + O(\varepsilon^{q+1}).
\end{aligned}$$

Performing the substitution

$$\varepsilon \mapsto \varepsilon^N \qquad \delta \mapsto \varepsilon$$

for N sufficiently large yields

$$h(\varepsilon \cdot \overline{y}, \varepsilon^N, \varepsilon) = \varepsilon^{qN} \alpha + \varepsilon^{qN + \deg(\hat{g})} \alpha t g(\overline{y}) + O(\varepsilon^{qN + \deg(\hat{g}) + 1}).$$

The desired f-oracle circuit for g is then given by

$$\Phi(\overline{y}) := \frac{h(\varepsilon \cdot \overline{y}, \varepsilon^N, \varepsilon) - \varepsilon^{qN} \alpha}{\varepsilon^{qN + \deg(\hat{g})} \alpha t} = g(\overline{y}) + O(\varepsilon).$$

If instead $\mathrm{char}(\mathbb{F}) = p > 0$, the above proof only needs to be modified in the case that p divides t. Let $k \in \mathbb{N}$ be the largest natural number such that p^k divides t and write $t = p^k b$. In this case, we instead get

$$h(\delta \cdot \overline{y}, \varepsilon, \delta) = \varepsilon^q \alpha + \varepsilon^q \delta^{\deg(\hat{g}) p^k} \alpha b g(\overline{y})^{p^k} + O(\varepsilon^q \delta^{2\deg(\hat{g}) p^k}) + O(\varepsilon^{q+1}).$$

Again, for N sufficiently large, we obtain an f-oracle circuit for g via

$$\Phi(\overline{y}) := \frac{h(\varepsilon \cdot \overline{y}, \varepsilon^N, \varepsilon) - \varepsilon^{qN} \alpha}{\varepsilon^{qN + \deg(\hat{g}) p^k} \alpha b} = g(\overline{y})^{p^k} + O(\varepsilon). \qquad \square$$

We now instantiate Theorem 3.15 with the determinant and iterated matrix multiplication polynomials. These corollaries are essentially obvious, but seem interesting in their own right.

Corollary 3.16. *Let $f(X) \in I^{\det}_{n,m,r}$ be a nonzero polynomial and let $h(X, \varepsilon) \in \mathbb{F}[\![\varepsilon]\!][X]$ be any polynomial such that $h(X, \varepsilon) = f(X) + O(\varepsilon)$. Let $t \leqslant O(r^{1/3})$. Then there is a depth-three h-oracle circuit Φ defined over $\mathbb{F}(\varepsilon)$ with the following properties.*

1. *The bottom layer of Φ consists of nm addition gates, the middle layer has a single h-oracle gate, and the top layer has a single addition gate.*
2. *If $\mathrm{char}(\mathbb{F}) = 0$, then Φ computes $\det_t(Y) + O(\varepsilon)$.*
3. *If $\mathrm{char}(\mathbb{F}) = p > 0$, then Φ computes $\det_t(Y)^{p^k} + O(\varepsilon)$ for some $k \in \mathbb{N}$.*

Proof. Mahajan and Vinay [51, Theorem 2] constructed a layered ABP on $O(t^3) \leqslant r$ vertices that computes $\det_t(Y)$. The corollary then follows from Theorem 3.15. $\square$

Corollary 3.17. *Let $f(X) \in I^{\det}_{n,m,r}$ be a nonzero polynomial and let $h(X, \varepsilon) \in \mathbb{F}[\![\varepsilon]\!][X]$ be any polynomial such that $h(X, \varepsilon) = f(X) + O(\varepsilon)$. Let $w, d \in \mathbb{N}$ satisfy $w(d-1) + 2 \leqslant r$. Then there is a depth-three h-oracle circuit Φ defined over $\mathbb{F}(\varepsilon)$ with the following properties.*

1. *The bottom layer of Φ consists of nm addition gates, the middle layer has a single h-oracle gate, and the top layer has a single addition gate.*
2. *If $\mathrm{char}(\mathbb{F}) = 0$, then Φ computes $\mathrm{IMM}_{w,d}(\overline{y}) + O(\varepsilon)$.*

(3) If $\mathrm{char}(\mathbb{F}) = p > 0$*, then* Φ *computes* $\mathrm{IMM}_{w,d}(\overline{y})^{p^k} + O(\varepsilon)$ *for some* $k \in \mathbb{N}$.

Proof. It is clear that $\mathrm{IMM}_{w,d}(\overline{y})$ is computable by a layered algebraic branching program on $w(d-1)+2 \leqslant r$ vertices. Theorem 3.15 completes the proof. □

We conclude this section with a remark on the fact that in characteristic $p > 0$, we only obtain an oracle circuit for a p^{th} power of the target polynomial $g(\overline{y})$.

Remark 3.18. Let $\mathbb{F}$ be a field of characteristic $p > 0$. If we interpret Theorem 3.15 as a result on "factoring" a polynomial $I^{\det}_{n,m,r}$, then the appearance of p^{th} powers in the "factors" is not too surprising. Most results on polynomial factorization [17, 27, 42, 48] only guarantee a circuit that computes a p^{th} power of a factor if the multiplicity of this factor is a multiple of p^k for some $k > 0$. In fact, if $f(\overline{x})^p$ can be computed by a size s circuit, it is open whether $f(\overline{x})$ can be computed by a circuit of size $\mathrm{poly}(n, \deg(f), s)$, although some results are known when n is small compared to s [6]. ◊

ACKNOWLEDGMENTS

This work is supported by the National Science Foundation under grants CCF-1755921, CCF-1814788, and CAREER award 2047310.

REFERENCES

[1] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. 2015. Hitting-Sets for ROABP and Sum of Set-Multilinear Circuits. *SIAM J. Comput.* 44, 3 (2015), 669–697. https://doi.org/10.1137/140975103

[2] Yaroslav Alekseev. 2021. A Lower Bound for Polynomial Calculus with Extension Rule. In *36th Computational Complexity Conference (CCC 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 200)*, Valentine Kabanets (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 21:1–21:18. https://doi.org/10.4230/LIPIcs.CCC.2021.21

[3] Yaroslav Alekseev, Dima Grigoriev, Edward A. Hirsch, and Iddo Tzameret. 2020. Semi-Algebraic Proofs, IPS Lower Bounds, and the τ-Conjecture: Can a Natural Number Be Negative?. In *Proceedings of the 52nd Annual ACM Symposium on Theory of Computing (STOC 2020)* (Chicago, IL, USA). Association for Computing Machinery, New York, NY, USA, 54–67. https://doi.org/10.1145/3357713.3384245

[4] Matthew Anderson, Michael A. Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. 2018. Identity Testing and Lower Bounds for Read-k Oblivious Algebraic Branching Programs. *ACM Trans. Comput. Theory* 10, 1, Article 3 (2018), 30 pages. https://doi.org/10.1145/3170709

[5] Matthew Anderson, Dieter van Melkebeek, and Ilya Volkovich. 2015. Deterministic polynomial identity tests for multilinear bounded-read formulae. *Computational Complexity* 24 (2015), 695–776. https://doi.org/10.1007/s00037-015-0097-4

[6] Robert Andrews. 2020. Algebraic Hardness Versus Randomness in Low Characteristic. In *35th Computational Complexity Conference (CCC 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 169)*, Shubhangi Saraf (Ed.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 37:1–37:32. https://doi.org/10.4230/LIPIcs.CCC.2020.37

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

[8] Vishwas Bhargava, Shubhangi Saraf, and Ilya Volkovich. 2020. Deterministic Factorization of Sparse Polynomials with Bounded Individual Degree. *J. ACM* 67, 2 (2020), 8:1–8:28. https://doi.org/10.1145/3365667 Preliminary version in the *59th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2018)*.

[9] Dario Bini, Milvio Capovani, Francesco Romani, and Grazia Lotti. 1979. $O(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication. *Inform. Process. Lett.* 8, 5 (1979), 234–235. https://doi.org/10.1016/0020-0190(79)90113-3

[10] Pranav Bisht and Nitin Saxena. 2021. Blackbox identity testing for sum of speacial ROABPs and its border class. *Computational Complexity* 30, 8 (2021), 1–48. https://doi.org/10.1007/s00037-021-00209-y

[11] Markus Bläser, Julian Dörfler, and Christian Ikenmeyer. 2021. On the Complexity of Evaluating Highest Weight Vectors. In *36th Computational Complexity Conference (CCC 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 200)*, Valentine Kabanets (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 29:1–29:36. https://doi.org/10.4230/LIPIcs.CCC.2021.29

[12] Peter Bürgisser. 2000. *Completeness and Reduction in Algebraic Complexity Theory*. Springer-Verlag Berlin Heidelberg. https://doi.org/10.1007/978-3-662-04179-6

[13] Peter Bürgisser. 2004. The complexity of factors of multivariate polynomials. *Foundations of Computational Mathematics* 4, 4 (2004), 369–396. https://doi.org/10.1007/s10208-002-0059-5

[14] Peter Bürgisser, Michael Clausen, and M. Amin Shokrollahi. 1997. *Algebraic complexity theory*. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences], Vol. 315. Springer-Verlag, Berlin. xxiv+618 pages. https://doi.org/10.1007/978-3-662-03338-8 With the collaboration of Thomas Lickteig.

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

[16] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. 2019. Closure of VP under taking factors: a short and simple proof. https://doi.org/10.48550/arXiv.1903.02366 arXiv:1903.02366

[17] Chi-Ning Chou, Mrinal Kumar, and Noam Solomon. 2019. Closure Results for Polynomial Factorization. *Theory of Computing* 15, 13 (2019), 1–34. https://doi.org/10.4086/toc.2019.v015a013 Preliminary version in the *33rd Annual Computational Complexity Conference (CCC 2018)*.

[18] Matthew Clegg, Jeffery Edmonds, and Russell Impagliazzo. 1996. Using the Groebner Basis Algorithm to Find Proofs of Unsatisfiability. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC 1996)* (Philadelphia, Pennsylvania, USA). Association for Computing Machinery, New York, NY, USA, 174–183. https://doi.org/10.1145/237814.237860

[19] David A. Cox, John Little, and Donal O'Shea. 2015. *Ideals, varieties, and algorithms - an introduction to computational algebraic geometry and commutative algebra* (4 ed.). Springer.

[20] Corrado de Concini, David Eisenbud, and Claudio Procesi. 1980. Young diagrams and determinantal varieties. *Invent. Math.* 56, 2 (1980), 129–165. https://doi.org/10.1007/BF01392548

[21] Jacques Désarménien, Joseph P. S. Kung, and Gian-Carlo Rota. 1978. Invariant theory, Young bitableaux, and combinatorics. *Advances in Math.* 27, 1 (1978), 63–92. https://doi.org/10.1016/0001-8708(78)90077-4

[22] Peter Doubilet, Gian-Carlo Rota, and Joel Stein. 1974. On the foundations of combinatorial theory. IX. Combinatorial methods in invariant theory. *Studies in Applied Mathematics* 53 (1974), 185–216. https://doi.org/10.1002/sapm1974533185

[23] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. 2021. Demystifying the border of depth-3 algebraic circuits. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2021)*. 92–103. https://doi.org/10.1109/FOCS52979.2021.00018

[24] Pranjal Dutta, Prateek Dwivedi, and Nitin Saxena. 2021. Deterministic Identity Testing Paradigms for Bounded Top-Fanin Depth-4 Circuits. In *36th Computational Complexity Conference (CCC 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 200)*, Valentine Kabanets (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 11:1–11:27. https://doi.org/10.4230/LIPIcs.CCC.2021.11

[25] Pranjal Dutta, Nitin Saxena, and Amit Sinhababu. 2018. Discovering the roots: uniform closure results for algebraic classes under factoring. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)*. 1152–1165. https://doi.org/10.1145/3188745.3188760

[26] Zeev Dvir and Amir Shpilka. 2007. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM J. Comput.* 36, 5 (2007), 1404–1434. https://doi.org/10.1137/05063605X

[27] Zeev Dvir, Amir Shpilka, and Amir Yehudayoff. 2009. Hardness-Randomness Tradeoffs for Bounded Depth Arithmetic Circuits. *SIAM J. Comput.* 39, 4 (2009), 1279–1293. https://doi.org/10.1137/080735850

[28] Michael A. Forbes. 2016. Some concrete questions on the border complexity of polynomials. Talk presented at the Workshop on Algebraic Complexity Theory (WACT), Tel Aviv.

[29] Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. 2014. 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)*. 867–875. https://doi.org/10.1145/2591796.2591816

[30] Michael A. Forbes and Amir Shpilka. 2013. 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)*. 243–252. https://doi.org/10.1109/FOCS.2013.34

[31] Michael A. Forbes and Amir Shpilka. 2018. A PSPACE Construction of a Hitting Set for the Closure of Small Algebraic Circuits. In *Proceedings of the 50th Annual ACM Symposium on Theory of Computing (STOC 2018)* (Los Angeles, CA, USA). Association for Computing Machinery, New York, NY, USA, 1180–1192. https://doi.org/10.1145/3188745.3188792

[32] Michael A. Forbes, Amir Shpilka, Iddo Tzameret, and Avi Wigderson. 2016. Proof Complexity Lower Bounds from Algebraic Circuit Complexity. In *Proceedings*

of the 31st Annual Computational Complexity Conference (CCC 2016). 32:1–32:17. https://doi.org/10.4230/LIPIcs.CCC.2016.32

[33] Joshua A. Grochow. 2020. Complexity in ideals of polynomials: questions on algebraic complexity of circuits and proofs. *Bull. EATCS* 130 (2020).

[34] Joshua A. Grochow and Toniann Pitassi. 2018. Circuit Complexity, Proof Complexity, and Polynomial Identity Testing: The Ideal Proof System. *J. ACM* 65, 6, Article 37 (Nov. 2018), 37:1–37:59 pages. https://doi.org/10.1145/3230742

[35] Zeyu Guo and Rohit Gurjar. 2020. Improved Explicit Hitting-Sets for ROABPs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 176)*, Jarosław Byrka and Raghu Meka (Eds.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 4:1–4:16. https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2020.4

[36] Zeyu Guo, Nitin Saxena, and Amit Sinhababu. 2019. Algebraic Dependencies and PSPACE Algorithms in Approximative Complexity over Any Field. *Theory of Computing* 15, 16 (2019), 1–30. https://doi.org/10.4086/toc.2019.v015a016

[37] Rohit Gurjar, Arpita Korwar, and Nitin Saxena. 2017. Identity Testing for Constant-Width, and Any-Order, Read-Once Oblivious Arithmetic Branching Programs. *Theory of Computing* 13, 1 (2017), 1–21. https://doi.org/10.4086/toc.2017.v013a002

[38] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. 2017. Deterministic Identity Testing for Sum of Read-Once Oblivious Arithmetic Branching Programs. *Computational Complexity* 26, 4 (2017), 835–880. https://doi.org/10.1007/s00037-016-0141-z

[39] Russell Impagliazzo, Sasank Mouli, and Toniann Pitassi. 2020. The Surprising Power of Constant Depth Algebraic Proofs. In *Proceedings of the Thirty fifth Annual IEEE Symposium on Logic in Computer Science (LICS 2020)* (Saarbrucken, Germany). IEEE Computer Society Press, 591–603. https://doi.org/10.1145/3373718.3394754

[40] Russell Impagliazzo, Pavel Pudlák, and Jiří Sgall. 1999. Lower bounds for the polynomial calculus and the Gröbner basis algorithm. *Computational Complexity* 8 (1999), 127–144. https://doi.org/10.1007/s000370050024

[41] Valentine Kabanets and Russell Impagliazzo. 2004. Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds. *Computational Complexity* 13, 1-2 (2004), 1–46. https://doi.org/10.1007/s00037-004-0182-6

[42] Erich Kaltofen. 1987. Single-Factor Hensel Lifting and its Application to the Straight-Line Complexity of Certain Polynomials. In *Proceedings of the 19th Annual ACM Symposium on Theory of Computing, 1987, New York, New York, USA*. 443–452. https://doi.org/10.1145/28395.28443

[43] Zohar S. Karnin, Partha Mukhopadhyay, Amir Shpilka, and Ilya Volkovich. 2013. Deterministic Identity Testing of Depth-4 Multilinear Circuits with Bounded Top Fan-in. *SIAM J. Comput.* 42, 6 (2013), 2114–2131. https://doi.org/10.1137/110824516

[44] Zohar S. Karnin and Amir Shpilka. 2011. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica* 31, 3 (2011), 333–364. https://doi.org/10.1007/s00493-011-2537-3

[45] Neeraj Kayal and Shubhangi Saraf. 2009. Blackbox polynomial identity testing for depth 3 circuits. In *Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2009)*. IEEE Computer Soc., Los Alamitos, CA, 198–207. https://doi.org/10.1109/FOCS.2009.67

[46] Neeraj Kayal and Nitin Saxena. 2007. Polynomial identity testing for depth 3 circuits. *Comput. Complexity* 16, 2 (2007), 115–138. https://doi.org/10.1007/s00037-007-0226-9

[47] Adam R. Klivans and Daniel Spielman. 2001. Randomness Efficient Identity Testing of Multivariate Polynomials. In *Proceedings of the 33rd Annual ACM Symposium on Theory of Computing (STOC 2001)* (Hersonissos, Greece). Association for Computing Machinery, New York, NY, USA, 216–223. https://doi.org/10.1145/380752.380801

[48] Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. 2015. Equivalence of Polynomial Identity Testing and Polynomial Factorization. *Computational Complexity* 24, 2 (2015), 295–331. https://doi.org/10.1007/s00037-015-0102-y

[49] Jan Krajíček. 2019. *Proof Complexity*. Cambridge University Press.

[50] Nutan Limaye, Srikanth Srinivasan, and Sébastien Tavenas. 2021. Superpolynomial Lower Bounds Against Low-Depth Algebraic Circuits. In *Proceedings of the 62nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2021)*. 804–814. https://doi.org/10.1109/FOCS52979.2021.00083

[51] Meena Mahajan and V. Vinay. 1997. Determinant: Combinatorics, Algorithms, and Complexity. *Chicago Journal of Theoretical Computer Science* 1997, 5 (1997). https://doi.org/10.4086/cjtcs.1997.005

[52] Dori Medini and Amir Shpilka. 2021. Hitting Sets and Reconstruction for Dense Orbits in VP_e and $\Sigma\Pi\Sigma$ Circuits. In *36th Computational Complexity Conference (CCC 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 200)*, Valentine Kabanets (Ed.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 19:1–19:27. https://doi.org/10.4230/LIPIcs.CCC.2021.19

[53] Daniel Minahan and Ilya Volkovich. 2018. Complete Derandomization of Identity Testing and Reconstruction of Read-Once Formulas. *ACM Trans. Comput. Theory* 10, 3, Article 10 (2018), 11 pages. https://doi.org/10.1145/3196836

[54] Ketan Mulmuley and Milind A. Sohoni. 2001. Geometric Complexity Theory I: An Approach to the P vs. NP and Related Problems. *SIAM J. Comput.* 31, 2 (2001), 496–526. https://doi.org/10.1137/S009753970038715X

[55] Rafael Oliveira. 2016. Factors of low individual degree polynomials. *Computational Complexity* 25, 2 (2016), 507–561. https://doi.org/10.1007/s00037-016-0130-2

[56] Rafael Oliveira, Amir Shpilka, and Ben Lee Volk. 2016. Subexponential Size Hitting Sets for Bounded Depth Multilinear Formulas. *Computational Complexity* 25 (2016), 455–505. https://doi.org/10.1007/s00037-016-0131-1

[57] Shir Peleg and Amir Shpilka. 2020. A Generalized Sylvester-Gallai Type Theorem for Quadratic Polynomials. In *35th Computational Complexity Conference (CCC 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 169)*, Shubhangi Saraf (Ed.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 8:1–8:33. https://doi.org/10.4230/LIPIcs.CCC.2020.8

[58] Shir Peleg and Amir Shpilka. 2021. Polynomial Time Deterministic Identity Testing Algorithm for $\Sigma[3]\Pi\Sigma\Pi[2]$ Circuits via Edelstein–Kelly Type Theorem for Quadratic Polynomials. In *Proceedings of the 53rd Annual ACM Symposium on Theory of Computing (STOC 2021)*. Association for Computing Machinery, New York, NY, USA, 259–271. https://doi.org/10.1145/3406325.3451013

[59] Toniann Pitassi and Iddo Tzameret. 2016. Algebraic Proof Complexity: Progress, Frontiers and Challenges. *ACM SIGLOG News* 3, 3 (Aug. 2016), 21–43. https://doi.org/10.1145/2984450.2984455

[60] Alexander A. Razborov. 1998. Lower bounds for the polynomial calculus. *Computational Complexity* 7 (1998), 291–324. https://doi.org/10.1007/s000370050013

[61] Chandan Saha and Bhargav Thankey. 2021. Hitting Sets for Orbits of Circuit Classes and Polynomial Families. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2021) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 207)*, Mary Wootters and Laura Sanità (Eds.). Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 50:1–50:26. https://doi.org/10.4230/LIPIcs.APPROX/RANDOM.2021.50

[62] Rahul Santhanam and Iddo Tzameret. 2021. Iterated Lower Bound Formulas: A Diagonalization-Based Approach to Proof Complexity. In *Proceedings of the 53rd Annual ACM Symposium on Theory of Computing (STOC 2021)*. Association for Computing Machinery, New York, NY, USA, 234–247. https://doi.org/10.1145/3406325.3451010

[63] Ramprasad Saptharishi. 2019. A survey of lower bounds in arithmetic circuit complexity. https://github.com/dasarpmar/lowerbounds-survey

[64] Shubhangi Saraf and Ilya Volkovich. 2018. Black-Box Identity Testing of Depth-4 Multilinear Circuits. *Combinatorica* 38 (2018), 1205–1238. https://doi.org/10.1007/s00493-016-3460-4

[65] Nitin Saxena and C. Seshadhri. 2011. An almost optimal rank bound for depth-3 identities. *SIAM J. Comput.* 40, 1 (2011), 200–224. https://doi.org/10.1137/090770679

[66] Nitin Saxena and C. Seshadhri. 2012. Blackbox identity testing for bounded top-fanin depth-3 circuits: the field doesn't matter. *SIAM J. Comput.* 41, 5 (2012), 1285–1298. https://doi.org/10.1137/10848232

[67] Nitin Saxena and C. Seshadhri. 2013. From Sylvester-Gallai configurations to rank bounds: improved blackbox identity test for depth-3 circuits. *J. ACM* 60, 5 (2013), 33:1–33:33. https://doi.org/10.1145/2528403

[68] Jacob T. Schwartz. 1980. Fast Probabilistic Algorithms for Verification of Polynomial Identities. *J. ACM* 27, 4 (1980), 701–717. https://doi.org/10.1145/322217.322225

[69] Amir Shpilka. 2019. Sylvester-Gallai Type Theorems for Quadratic Polynomials. In *Proceedings of the 51st Annual ACM Symposium on Theory of Computing (STOC 2019)* (Phoenix, AZ, USA). Association for Computing Machinery, New York, NY, USA, 1203–1214. https://doi.org/10.1145/3313276.3316341

[70] Amir Shpilka and Ilya Volkovich. 2015. Read-once polynomial identity testing. *Computational Complexity* 27 (2015), 477–532. https://doi.org/10.1007/s00037-015-0105-8

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

[72] Amit Sinhababu and Thomas Thierauf. 2020. Factorization of Polynomials Given By Arithmetic Branching Programs. In *Proceedings of the 35th Annual Computational Complexity Conference (CCC 2020) (Leibniz International Proceedings in Informatics (LIPIcs), Vol. 169)*, Shubhangi Saraf (Ed.). Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, 33:1–33:19. https://doi.org/10.4230/LIPIcs.CCC.2020.33

[73] Leslie G. Valiant. 1979. Completeness Classes in Algebra. In *Proceedings of the 11th Annual ACM Symposium on Theory of Computing (STOC 1979)* (Atlanta, Georgia, USA). Association for Computing Machinery, New York, NY, USA, 249–261. https://doi.org/10.1145/800135.804419

[74] Finn Wiersig. 2020. Sparse Polynomials in Polynomial Ideals.

[75] Richard Zippel. 1979. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation, EUROSAM 1979*. 216–226. https://doi.org/10.1007/3-540-09519-5_73