2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS) | 978-1-6654-5519-0/22/$31.00 ©2022 IEEE | DOI: 10.1109/FOCS54457.2022.00041

On Matrix Multiplication and Polynomial Identity Testing

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

***Abstract*—We show that lower bounds on the border rank of matrix multiplication can be used to non-trivially derandomize polynomial identity testing for small algebraic circuits. Letting $\underline{\mathrm{R}}(n)$ denote the border rank of $n \times n \times n$ matrix multiplication, we construct a hitting set generator with seed length $O(\sqrt{n} \cdot \underline{\mathrm{R}}^{-1}(s))$ that hits n-variate circuits of multiplicative complexity s. If the matrix multiplication exponent ω is not 2, our generator has seed length $O(n^{1-\varepsilon})$ and hits circuits of size $O(n^{1+\delta})$ for sufficiently small $\varepsilon, \delta > 0$. Surprisingly, the fact that $\underline{\mathrm{R}}(n) \geqslant n^2$ already yields new, non-trivial hitting set generators for circuits of sublinear multiplicative complexity.**

***Keywords*—matrix multiplication; polynomial identity testing**

I. Introduction

Matrix multiplication is a fundamental algorithmic problem in theoretical computer science. Starting with the work of Strassen [1], who gave an algorithm to multiply two $n \times n$ matrices in $O(n^{\log_2 7})$ time, a long line of work [2]–[16] has produced faster algorithms to multiply matrices. Progress on this task is usually measured by ω, the exponent of matrix multiplication, which is the smallest real number such that matrix multiplication can be performed using $O(n^{\omega+\varepsilon})$ arithmetic operations for any positive constant $\varepsilon > 0$. It is evident that $2 \leqslant \omega \leqslant 3$. Strassen's [1] result can be rephrased as a proof that $\omega \leqslant \log_2 7$. The present state-of-the-art algorithm for matrix multiplication is due to Alman and Vassilevska Williams [13], who proved $\omega < 2.37286$. It is a major open question to determine whether or not $\omega = 2$.

The complexity of matrix multiplication governs (and in many cases, is equivalent to) the complexity of numerous problems in linear algebra, including computing the determinant and solving systems of linear equations [1], boolean matrix multiplication [17], QR decomposition [18], LUP decomposition [19], and computing the coefficients of the characteristic polynomial of a matrix [20]. Fast matrix multiplication has also been used to design algorithms for a host of problems in other areas; examples include recognizing context-free languages [21], detecting k-cliques [22], and solving linear programs [23]–[25].

While it is popularly conjectured that $\omega = 2$, progress on obtaining improved upper bounds on ω has slowed over time. In the three decades since Coppersmith and Winograd [9] showed $\omega < 2.3755$, the best-known bound on ω has improved by only ≈ 0.00264. The improvements obtained since then [10]–[13] apply Strassen's so-called laser method [8] to powers of the Coppersmith–Winograd tensor. Recent work [26]–[33] has shown that this slow progress is no coincidence: there are unconditional barriers to obtaining improved bounds on ω using generalizations of this and related techniques.

There is a dual line of work concerned with proving lower bounds on the complexity of matrix multiplication. This usually proceeds by proving lower bounds on the rank or border rank of matrix multiplication, which essentially correspond to the number of scalar multiplications one needs to perform in order to compute a matrix product. It is known that $\omega = 2$ if and only if the rank (or border rank) of matrix multiplication is bounded from above by $n^{2+o(1)}$. The best-known lower bound on the rank of matrix multiplication is $\frac{5}{2}n^2 - 3n$ by Bläser [34], with an improvement over finite fields due to Shpilka [35]. For border rank, an approximate version of rank, the current record is a lower bound of $2n^2 - \log_2(n) - 1$, due to Landsberg and Michałek [36]. In a somewhat different vein, Raz [37] showed that any bounded-coefficient circuit computing $n \times n \times n$ matrix multiplication must be of size $\Omega(n^2 \log n)$.

Naturally, if $\omega = 2$, one obtains extremely fast algorithms for matrix multiplication, leading to improved algorithms for a variety of problems. However, it is not clear if there is a useful algorithmic consequence of the hypothesis $\omega > 2$. The main contribution of this work is an application of the assumption $\omega > 2$ to the design of algorithms. Specifically, we show that if $\omega > 2$, then one can non-trivially derandomize polynomial identity testing for small circuits.

Polynomial identity testing (PIT) is the problem of testing whether an algebraic circuit computes the zero polynomial. There is a simple, fast randomized algorithm for PIT [38], [39], but no non-trivial deterministic algorithm is known. Designing a deterministic polynomial-time algorithm for PIT is a major goal of algebraic complexity. Typically, this is done by constructing a hitting set generator, the analogue of a pseudorandom generator in this setting. There has been considerable success in derandomizing PIT for restricted classes of circuits (see, e.g., Shpilka and Yehudayoff [40] and Saxena [41], [42]). For strong models of computation, like formulas and circuits, only conditional results in the form of hardness-to-pseudorandomness results are known [43]–[48] (see also [49]).

Supported by NSF grant CAREER 20-47310.

978-1-6654-5519-0/22/$31.00 ©2022 IEEE
DOI 10.1109/FOCS54457.2022.00041

A. Our Results

We now describe our result in more detail. We construct a hitting set generator for algebraic circuits that have a small number of multiplication gates.

Theorem (see Theorem IV.1). *Let $\underline{\mathrm{R}}(n)$ denote the border rank of $n \times n \times n$ matrix multiplication. There is an explicit hitting set generator of seed length $O(\sqrt{n}\,\underline{\mathrm{R}}^{-1}(s))$ that hits n-variate circuits with s multiplication gates.*

In terms of the matrix multiplication exponent ω, our generator has seed length $O(\sqrt{n}s^{1/\omega})$. Thus, if $\omega > 2$, we obtain a generator of seed length $O(n^{1-\varepsilon})$ that hits circuits of size $O(n^{1+\delta})$ for sufficiently small $\varepsilon, \delta > 0$. Alternatively, one can phrase this as a win-win result: either $\omega = 2$, giving us fast algorithms for a large collection of problems; or $\omega > 2$, in which case we obtain a non-trivial deterministic algorithm for testing identities given by small circuits.

As $\underline{\mathrm{R}}(n) \geqslant (2 - o(1))n^2$ [36], this also yields an unconditional construction of a hitting set generator with seed length $O(\sqrt{ns})$, which is non-trivial as long as $s \leqslant \varepsilon n$ for sufficiently small $\varepsilon > 0$. It may seem strange to consider circuits of complexity much less than n; for many circuit classes, such circuits are not even capable of reading their entire input. However, circuits with few multiplication gates are capable of computing non-trivial polynomials, mainly through the use of repeated squaring. For example, the polynomial $(x_1 + \cdots + x_n)^d$ can be computed using only $O(\log d)$ multiplication gates.

To the best of our knowledge, nothing is known about derandomizing PIT for circuits with few product gates. For very small s, one can obtain non-trivial algorithms by bounding the sparsity of the computed polynomial and using the Klivans–Spielman generator [50]. This strategy breaks down when $s \geqslant \Omega(\log n)$, as the resulting sparsity bound becomes too large. In contrast, our construction gives a non-trivial algorithm even when $s = \varepsilon n$ for $\varepsilon < \frac{1}{192}$.

Our result comes within a logarithmic factor of converting all known unconditional hardness for algebraic circuits into pseudorandomness. The state-of-the-art in explicit lower bounds on multiplicative complexity dates back to Baur and Strassen [51], who showed that the polynomial $x_1^d + \cdots + x_n^d$ requires $\Omega(n \log d)$ multiplications to compute. If one could construct an explicit generator whose seed length remains non-trivial for multiplicative complexity $s \geqslant \omega(n \log n)$, then this would provide an explicit family of n-variate multilinear polynomials of multiplicative complexity $\omega(n \log n)$. Of course, it remains a possibility that our generator could be improved to hit circuits of multiplicative complexity $O(n \log n)$ without requiring a breakthrough in circuit lower bounds.

As mentioned earlier, there is a collection of works on the hardness-randomness phenomenon in algebraic complexity [43]–[48]. Because the assumption $\omega > 2$ is inherently a circuit lower bound, it seems reasonable to expect that one could instantiate the hardness-randomness connection in order to directly obtain our result. Though this is the spirit of our approach, we remark that the known hardness-randomness framework typically incurs some polynomial overhead in translating a circuit lower bound into a hitting set generator. In particular, for a weak lower bound of the form $\Omega(n^{1+\varepsilon})$ (like what is implied by $\omega > 2$), these techniques fail to imply any kind of non-trivial derandomization. We note that work by Dutta, Saxena, and Thierauf [52] showed that for a particular class of constant-variate circuits, weak lower bounds can in fact be used to derandomize PIT. For more on algebraic hardness versus randomness, see the survey of Kumar and Saptharishi [49].

B. Our Techniques

We briefly describe our generator and the proof of its correctness. Throughout, we consider n-variate circuits as taking as input a matrix X of size $\sqrt{n} \times \sqrt{n}$. Let $\underline{\mathrm{R}}(n)$ be the border rank of $n \times n \times n$ matrix multiplication. To construct our generator, we will show that the set of matrices of rank $O(\underline{\mathrm{R}}^{-1}(s))$ are a hitting set for circuits with s multiplication gates. This will imply that such a circuit cannot vanish on the product of an $\sqrt{n} \times O(\underline{\mathrm{R}}^{-1}(s))$ matrix and an $O(\underline{\mathrm{R}}^{-1}(s)) \times \sqrt{n}$ matrix, which yields our generator. Thus, we are faced with the task of showing that no small circuit can vanish on the set of all matrices of rank $O(\underline{\mathrm{R}}^{-1}(s))$.

Let $r \in \mathbb{N}$ and let $I_r \subseteq \mathbb{F}[X]$ be the ideal of $\mathbb{F}[X]$ generated by the $r \times r$ minors of the matrix X. It is well-known that when the field $\mathbb{F}$ is algebraically closed, the ideal I_r consists exactly of those polynomials that vanish on matrices of rank less than r. Rephrasing our goal, we need to prove a lower bound of $\Omega(\underline{\mathrm{R}}(r))$ on the number of multiplication gates needed to compute any nonzero polynomial in the ideal I_r.

Using an observation due to Baur and Strassen [51, Corollary 6], a lower bound on the border rank of matrix multiplication lifts to a lower bound on the border multiplicative complexity of the polynomial $\mathrm{tr}(XYZ)$, where X, Y, and Z are $n \times n$ matrices. Results of Andrews and Forbes [48] allow us to further lift this lower bound to the ideal I_r where r is the size of the smallest algebraic branching program computing $\mathrm{tr}(XYZ)$. Because $\mathrm{tr}(XYZ)$ can be computed by an algebraic branching program with $O(n^2)$ vertices, we obtain a lower bound of $\Omega(\underline{\mathrm{R}}(\sqrt{r}))$ on the multiplicative complexity of I_r. This suffices to obtain a hitting set generator of seed length $O(\sqrt{n}\,\underline{\mathrm{R}}^{-1}(s^2))$ for circuits with s product gates. However, such a construction cannot hope to obtain seed length $o(n)$ for circuits with $O(n^{0.6})$ product gates, even if the best-known upper bound on ω is tight.

To improve the dependence on s in the seed length, we instead lift the lower bound to the ideal I_r where r is the size of the smallest *trace* algebraic branching program that computes $\mathrm{tr}(XYZ)$. This polynomial can naturally be computed by a trace ABP of size $O(n)$, which leads to the improved lower bound of $\Omega(\underline{\mathrm{R}}(r))$ on the multiplicative complexity of I_r. This immediately translates into the improved seed length of $O(\sqrt{n}\,\underline{\mathrm{R}}^{-1}(s))$ for our hitting set generator.

To perform this improved lifting step, we essentially need to show that trace ABPs of size s can be expressed as a determinant of size $O(s)$. We do this using the interpretation of the determinant as a sum of weighted cycle covers in an ABP, following Valiant [53].

II. Preliminaries

Throughout this work, we take $\mathbb{F}$ to be a field of characteristic zero. For $n \in \mathbb{N}$ a natural number, we write $[n] := \{1, 2, \ldots, n\}$. We denote by $\overline{x} = (x_1, \ldots, x_n)$ and $X = (x_{i,j})_{i \in [n], j \in [m]}$ a vector of variables and an $n \times m$ matrix of variables, respectively. We write $\mathbb{F}[\overline{x}]$ for the polynomial ring in the variables $\overline{x}$. We use $I^{\det}_{n,m,r}$ to denote the ideal of $\mathbb{F}[X]$ generated by the $r \times r$ minors of a matrix of variables X. For an $n \times m$ matrix A and subsets $R \subseteq [n]$ and $C \subseteq [m]$, we write $A_{R,C}$ for the submatrix of A obtained by selecting the rows indexed by R and the columns indexed by C.

A. Algebraic Circuits

We briefly recall the notions of algebraic circuits, algebraic branching programs, and trace algebraic branching programs. For a more thorough treatment of algebraic circuit complexity, we refer the reader to Shpilka and Yehudayoff [40] and Saptharishi [54]. We begin with the definition of an algebraic circuit.

Definition II.1. An *algebraic circuit* is a directed acyclic graph in which every vertex has in-degree zero or two. Vertices of in-degree zero are called input gates and are labeled by either a field constant or a variable $x_{i,j}$. Vertices of in-degree two are called internal gates and are labeled either as addition or multiplication gates. The gates of the circuit compute polynomials in $\mathbb{F}[X]$ in the natural way. We allow each edge e of the circuit to be labeled by a field constant $\alpha_e \in \mathbb{F}$, which has the effect of multiplying the value carried by that edge by α_e. We measure the *size* of a circuit by the number of gates appearing in the circuit. The *multiplicative complexity* of a circuit is the number of multiplication gates appearing in the circuit. ♢

We will also require the notions of algebraic branching programs (ABPs) and trace algebraic branching programs (trace ABPs).

Definition II.2. A *(single-source, single-sink) algebraic branching program (ABP)* is a layered directed acyclic graph $G = (V, E)$ with a single source vertex s and a single sink vertex t. By layered, we mean that there is a partition $V = V_0 \sqcup V_1 \sqcup \cdots \sqcup V_d$ such that $V_0 = \{s\}$, $V_d = \{t\}$, and every edge in G goes from layer V_{i-1} to V_i for some $i \in [d]$. Every edge e of G is labeled by a linear polynomial $\ell_e(\overline{x}) \in \mathbb{F}[\overline{x}]$. Let $\mathcal{P}_{s,t}$ be the set of s-t paths in G. The ABP computes the polynomial given by

$$\sum_{P \in \mathcal{P}_{s,t}} \prod_{e \in P} \ell_e(\overline{x}).$$

The *size* of the ABP is $|V|$, the number of vertices in G. The *width* of the ABP is $\max_{i \in [d]} |V_i|$.

Equivalently, an ABP is given by a collection of matrices $M_1(\overline{x}), \ldots, M_d(\overline{x})$ whose entries are linear polynomials in $\mathbb{F}[\overline{x}]$. The polynomial computed by the ABP is the $(1,1)$ entry of the matrix product $M_1(\overline{x}) \cdots M_d(\overline{x})$, where the dimensions of the matrices $M_i(\overline{x})$ are such that the resulting product is defined. ♢

A trace ABP endows an ABP with multiple sources $s_1, \ldots, s_m$ and sinks $t_1, \ldots, t_m$. Whereas an ABP computes a sum over all source-to-sink paths, a trace ABP sums over all s_i-t_i paths for all choices of $i \in [m]$, allowing the ABP to reuse intermediate vertices for these different sums. Alternatively, when viewing an ABP as a matrix product, a trace ABP corresponds to taking the trace of the resulting matrix product instead of extracting the $(1,1)$ entry.

Definition II.3. A *trace algebraic branching program (trace ABP)* is a layered directed acyclic graph $G = (V, E)$ with source vertices $s_1, \ldots, s_m$ and sink vertices $t_1, \ldots, t_m$. By layered, we mean that there is a partition $V = V_0 \sqcup V_1 \sqcup \cdots \sqcup V_d$ such that $V_0 = \{s_1, \ldots, s_m\}$, $V_d = \{t_1, \ldots, t_m\}$, and every edge in G goes from layer V_{i-1} to V_i for some $i \in [d]$. Every edge e of G is labeled by a linear polynomial $\ell_e(\overline{x}) \in \mathbb{F}[\overline{x}]$. Let $\mathcal{P}_{s_i,t_i}$ be the set of s_i-t_i paths in G. The trace ABP computes the polynomial given by

$$\sum_{i=1}^{m} \sum_{P \in \mathcal{P}_{s_i,t_i}} \prod_{e \in P} \ell_e(\overline{x}).$$

The *size* of the ABP is $|V|$, the number of vertices in G. The *width* of the ABP is $\max_{i \in [d]} |V_i|$.

Equivalently, a trace ABP is given by a collection of matrices $M_1(\overline{x}), \ldots, M_d(\overline{x})$ whose entries are linear polynomials in $\mathbb{F}[\overline{x}]$. The polynomial computed by the trace ABP is the trace of the matrix product $M_1(\overline{x}) \cdots M_d(\overline{x})$, where the dimensions of the matrices $M_i(\overline{x})$ are such that the resulting product is defined. ♢

It is clear that any polynomial computed by an ABP can be computed by a trace ABP of the same size and width. Conversely, one can transform a trace ABP into a single-source, single-sink ABP by duplicating the trace ABP m times, deleting all but one pair of source and sink vertices in each copy, and identifying the source vertices and sink vertices in the resulting copies. To the best of our knowledge, this is the best-known simulation of trace ABPs by single-source, single-sink ABPs.

Lemma II.4. *Let $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ be a polynomial computed by a trace ABP of size s and width w. Then $f(\overline{x})$ can be computed by a single-source, single-sink ABP of size ws and width w^2*

.

We will make use of the following result of Baur and Strassen [51] that transforms a circuit that computes a polynomial $f(\overline{x})$ into one that computes all first-order partial derivatives of $f(\overline{x})$ while increasing the circuit size by only a constant factor. We state the version of their result for multiplicative complexity, although an analogous statement holds for circuit size. Note that by taking $\mathbb{F} = \mathbb{K}(\varepsilon)$ where $\mathbb{K}$ is a field, this lemma extends to the setting of border complexity (defined in Subsection II-B).

Lemma II.5 ([51])**.** *Let $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ be a polynomial computed by an algebraic circuit of multiplicative complexity s. Then there is a multi-output algebraic circuit of multiplicative complexity $3s$ that computes $\left\{f(\overline{x}), \frac{\partial f}{\partial x_1}(\overline{x}), \dots \frac{\partial f}{\partial x_n}(\overline{x})\right\}$.*

B. Border Complexity

We will crucially make use of border complexity, which is an approximative version of algebraic computation.

Definition II.6. Let $\mathbb{F}$ be a field and ε be an indeterminate. Let $f(\overline{x}) \in \mathbb{F}[\overline{x}]$ be a polynomial. We say that a circuit Φ over the field $\mathbb{F}(\varepsilon)$ *border computes* $f(\overline{x})$ if Φ computes a polynomial of the form

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

where $g(\overline{x}, \varepsilon) \in \mathbb{F}[\overline{x}, \varepsilon]$. We frequently abbreviate this by saying that Φ computes $f(\overline{x}) + O(\varepsilon)$. $\lozenge$

Over fields of characteristic zero, one can think of border computation as computing a polynomial f up to an arbitrarily small error ε. The definition above extends to fields of positive characteristic, although this will not be relevant for our work. Naturally, one can consider the notion of border complexity for restricted classes of circuits, like formulas or branching programs.

If $\mathcal{C}$ is a class of circuits, we define the *closure* of $\mathcal{C}$ to be the set of polynomials that can be border computed by a $\mathcal{C}$-circuit. For example, if $\mathcal{C}$ is the class of size-s circuits, the closure of $\mathcal{C}$ consists of all polynomials $f(\overline{x})$ such that $f(\overline{x}) + O(\varepsilon)$ can be computed by a size-s circuit over $\mathbb{F}(\varepsilon)$.

In the course of our work, we will prove lower bounds by constructing oracle circuits. The following lemma says that in the setting of border complexity, one can replace an exact oracle with an approximate oracle without incurring an increase in circuit size. This makes our job easier, as we only need to reason about circuits using exact oracles. This lemma is a straightforward consequence of [55, Lemma 2.3(1)]; for a proof, see, e.g., [48, Lemma 2.3].

Lemma II.7. *Let $f(\overline{x}), g(\overline{x}) \in \mathbb{F}[\overline{x}]$ be polynomials. Suppose $f(\overline{x}) + O(\varepsilon)$ can be computed by a circuit of size s with g-oracle gates. Let $h(\overline{x}, \delta) \in \mathbb{F}[\![\delta]\!][\overline{x}]$ be a polynomial such that $h(\overline{x}, \delta) = g(\overline{x}) + O(\delta)$. Then there is some $N \in \mathbb{N}$ such that $f(\overline{x}) + O(\varepsilon)$ can be computed by a circuit of size s with $h(\overline{x}, \varepsilon^N)$-oracle gates.*

C. Polynomial Identity Testing

We will design polynomial identity testing algorithms that operate on circuits in a black-box manner; that is, our algorithms will only evaluate the circuit and will not examine the internal structure of the circuit. This is equivalent to giving an explicit construction of a hitting set for the class of circuits under consideration.

Definition II.8. Let $\mathcal{C} \subseteq \mathbb{F}[\overline{x}]$ be a set of polynomials. A set $\mathcal{H} \subseteq \mathbb{F}^n$ is a *hitting set for* $\mathcal{C}$ if for every nonzero $f \in \mathcal{C}$, there is some $\overline{\alpha} \in \mathcal{H}$ such that $f(\overline{\alpha}) \neq 0$. $\lozenge$

Equivalently, one can attempt to construct a hitting set generator, which is analogous to a pseudorandom generator in this setting.

Definition II.9. Let $\mathcal{C} \subseteq \mathbb{F}[\overline{x}]$ be a set of polynomials. A polynomial map $\mathcal{G} : \mathbb{F}^\ell \to \mathbb{F}^n$ is a *hitting set generator for* $\mathcal{C}$ if for every nonzero $f \in \mathcal{C}$, we have $f(\mathcal{G}(\overline{y})) \neq 0$. We call ℓ the *seed length* of the generator. The *degree* of the generator, denoted by $\deg(\mathcal{G})$, is given by $\max_{i \in [n]} \deg(\mathcal{G}_i)$. $\lozenge$

One can translate between hitting sets and hitting set generators using the Schwartz–Zippel lemma [38], [39] and polynomial interpolation. We note that if $\mathcal{C} \subseteq \mathbb{F}[X]$ is a set of degree-d polynomials and $\mathcal{G} : \mathbb{F}^\ell \to \mathbb{F}^n$ is a hitting set generator for $\mathcal{C}$, one obtains a hitting set of size $(d \cdot \deg(\mathcal{G}) + 1)^\ell$. In contrast, one can always construct a hitting set of size $(d+1)^n$. Note that a generator with $\deg(\mathcal{G}) \leqslant d^{O(1)}$ and $\ell \leqslant o(n)$ corresponds to a hitting set of size $d^{o(n)}$, which is a super-polynomial improvement over the trivial hitting set of size $(d+1)^n$.

D. Determinantal Ideals and Matrix Rank

Let X be an $n \times m$ matrix of variables. 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 make use the following proposition of Andrews and Forbes [48], which reduces the task of proving lower bounds on all polynomials in $I^{\det}_{n,m,r}$ to the task of proving lower bounds on products of minors. We note that the polynomial $(K_\sigma | K_\sigma)(X)$ appearing in the statement of [48, Proposition 3.5] is exactly the same as the product of determinants that appears in the proposition below. However, we give a more direct statement of this proposition to avoid the language of bitableaux and bideterminants, which is unnecessary for the results of this work.

Proposition II.10 ([48, Proposition 3.5])**.** *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 natural numbers $\sigma_1, \dots, \sigma_p$ with $\sigma_1 \geqslant r$ such that*

$$\begin{aligned} &f(\ell_{1,1}(X,\varepsilon), \dots, \ell_{n,m}(X,\varepsilon)) \\ &\quad = \varepsilon^q \alpha \prod_{i=1}^{p} \det\nolimits_{\sigma_i}(X_{[\sigma_i],[\sigma_i]}) + O(\varepsilon^{q+1}). \end{aligned}$$

It is well-known that when the underlying field $\mathbb{F}$ is algebraically closed, the ideal $I^{\det}_{n,m,r}$ consists exactly of those polynomials which vanish on all matrices of rank less than r. In particular, proving a lower bound of s on the complexity of all nonzero polynomials in $I^{\det}_{n,m,r}$ equates to proving that every polynomial of complexity less than s cannot vanish on all matrices of rank less than r. There is a natural hitting set generator whose image contains all low-rank matrices.

Construction II.11. *Let $n, m, r \in \mathbb{N}$ with $r \leqslant \min(n,m)$. Define the map $\mathcal{G}_{n,m,r} : \mathbb{F}^{n \times r} \times \mathbb{F}^{r \times m} \to \mathbb{F}^{n \times m}$ via*

$$\mathcal{G}_{n,m,r}(Y,Z)_{i,j} := (YZ)_{i,j}.$$

It is evident from its definition that the generator of Construction II.11 contains in its image all $n \times m$ matrices of rank at most r. The connection between matrix rank and the ideal $I^{\det}_{n,m,r}$ can be used to prove the following lemma. For the sake of completeness, we provide a proof (the same proof can be found in the discussion preceding [48, Lemma 2.10]).

Lemma II.12. *Let $\mathbb{F}$ be any field and let $n, m, r \in \mathbb{N}$ with $r \leqslant \min(n, m)$. Let $I^{\det}_{n,m,r} \subseteq \mathbb{F}[X]$ denote the ideal of $\mathbb{F}[X]$ generated by the $r \times r$ minors of a generic $n \times m$ matrix X and let $f(X) \in \mathbb{F}[X]$. Then $f(\mathcal{G}_{n,m,r-1}(Y, Z)) = 0$ if and only if $f(X) \in I^{\det}_{n,m,r}$.*

Proof. If $f(X) \in I^{\det}_{n,m,r}$, then we can write f as $f(X) = \sum_{i=1}^{N} g_i(X)h_i(X)$ where the polynomials $\{g_1, \ldots, g_N\}$ are the $r \times r$ minors of X. Because the image of $\mathcal{G}_{n,m,r-1}(Y, Z)$ is necessarily a matrix of rank at most $r - 1$, each $r \times r$ minor of $\mathcal{G}_{n,m,r-1}(Y, Z)$ vanishes, i.e., $g_i(\mathcal{G}_{n,m,r-1}(Y, Z)) = 0$ for all $i \in [N]$. This implies $f(\mathcal{G}_{n,m,r-1}(Y, Z)) = 0$.

To prove the converse direction, we first work under the assumption that the field $\mathbb{F}$ is algebraically closed. Suppose that $f(\mathcal{G}_{n,m,r-1}(Y, Z)) = 0$. Let $J_{n,m,r-1} \subseteq \mathbb{F}[X]$ be the ideal of $\mathbb{F}[X]$ consisting of polynomials that vanish on the set of matrices of rank at most $r - 1$. Because the image of $\mathcal{G}_{n,m,r-1}(Y, Z)$ contains all matrices of rank at most $r - 1$, we have $f \in J_{n,m,r-1}$. To show that $f \in I^{\det}_{n,m,r}$, we will prove the equality $I^{\det}_{n,m,r} = J_{n,m,r-1}$.

The inclusion $I^{\det}_{n,m,r} \subseteq J_{n,m,r-1}$ is immediate, as the $r \times r$ minors vanish on matrices of rank less than r. For the inclusion in the reverse direction, we use the correspondence between ideals and varieties. Recall that for an ideal $I \subseteq F[X]$, we denote by $V(I) \subseteq \mathbb{F}^{n\times m}$ the *variety* of I, defined as

$$V(I) := \{A \in \mathbb{F}^{n\times m} : \forall h(X) \in I, h(A) = 0\}.$$

Let $V(I^{\det}_{n,m,r})$ be the variety over $\mathbb{F}$ defined by the ideal $I^{\det}_{n,m,r}$ and let $A \in V(I^{\det}_{n,m,r})$ be a point in this variety. By definition, each $r \times r$ minor of A vanishes, so $\operatorname{rank}(A) \leqslant r - 1$, which implies $A \in V(J_{n,m,r-1})$. This shows $V(I^{\det}_{n,m,r}) \subseteq V(J_{n,m,r-1})$. By Hilbert's Nullstellensatz, this implies $\sqrt{J_{n,m,r-1}} \subseteq \sqrt{I^{\det}_{n,m,r}}$, where $\sqrt{I}$ denotes the *radical* of an ideal I. The ideal $I^{\det}_{n,m,r}$ is radical (see, e.g., [56, Theorem 2.10 and Remark 2.12]), so we have the desired inclusion

$$J_{n,m,r-1} \subseteq \sqrt{J_{n,m,r-1}} \subseteq \sqrt{I^{\det}_{n,m,r}} = I^{\det}_{n,m,r}.$$

This proves $J_{n,m,r-1} = I^{\det}_{n,m,r}$, hence $f(X) \in I^{\det}_{n,m,r}$ as claimed.

If $\mathbb{F}$ is not algebraically closed, we can still consider $f(X)$ as a polynomial over the algebraic closure $\overline{\mathbb{F}}$. If $f(\mathcal{G}_{n,m,r-1}(Y, Z)) = 0$, the previous argument implies that $f \in I^{\det}_{n,m,r}$ when $I^{\det}_{n,m,r}$ is considered as an ideal over $\overline{\mathbb{F}}$. Letting $I_{\mathbb{F}}$ and $I_{\overline{\mathbb{F}}}$ denote $I^{\det}_{n,m,r}$ when considered as an ideal over $\mathbb{F}$ and $\overline{\mathbb{F}}$, respectively, we have $f \in I_{\overline{\mathbb{F}}} \cap \mathbb{F}[\overline{x}]$. Lemma II.13 below shows that $I_{\overline{\mathbb{F}}} \cap \mathbb{F}[\overline{x}] = I_{\mathbb{F}}$, so we in fact have $f \in I_{\mathbb{F}}$ as desired. □

The following is an elementary lemma used in the proof of Lemma II.12 in the case where $\mathbb{F}$ is not algebraically closed. In the spirit of keeping this work self-contained, we provide a proof.

Lemma II.13. *Let $\mathbb{F}$ be a field and let $\mathbb{K} \supseteq \mathbb{F}$ be an extension of $\mathbb{F}$. Let $\{g_1, \ldots, g_m\} \subseteq \mathbb{F}[\overline{x}]$ be a set of polynomials. Let $I_{\mathbb{F}}$ and $I_{\mathbb{K}}$ be the ideals generated by $\{g_1, \ldots, g_m\}$ over $\mathbb{F}[\overline{x}]$ and $\mathbb{K}[\overline{x}]$, respectively. Then $I_{\mathbb{F}} = I_{\mathbb{K}} \cap \mathbb{F}[\overline{x}]$.*

Proof. The inclusion $I_{\mathbb{F}} \subseteq I_{\mathbb{K}} \cap \mathbb{F}[X]$ is immediate. For the other direction, let $\{v_1, v_2, \ldots\}$ be a basis of $\mathbb{K}$ as a vector space over $\mathbb{F}$ with the additional property that v_1 spans $\mathbb{F}$. Consider the linear projection $\pi : \mathbb{K} \to \mathbb{F}$ that sends v_1 to itself and v_i to zero for $i \geqslant 2$. We extend π to a projection $\pi : \mathbb{K}[X] \to \mathbb{F}[X]$ by applying the projection from $\mathbb{K}$ to $\mathbb{F}$ coefficient-wise. Let $f(X) \in I_{\mathbb{K}} \cap \mathbb{F}[X]$ be given by $f(X) = \sum_{i=1}^{N} g_i(X)h_i(X)$. We claim that $f(X) = \sum_{i=1}^{N} g_i(X)\pi(h_i(X))$, which proves $f(X) \in I_{\mathbb{F}}$ as desired.

To see this, let m be a monomial and consider the coefficient $\operatorname{Coeff}_m(f)$ of m in f. Because $\operatorname{Coeff}_m(f) \in \mathbb{F}$, we have $\pi(\operatorname{Coeff}_m(f)) = \operatorname{Coeff}_m(f)$. Using the fact that π is $\mathbb{F}$-linear, this implies

$$\begin{aligned}\operatorname{Coeff}_m(f) &= \pi(\operatorname{Coeff}_m(f)) \\ &= \sum_{i=1}^{N} \pi(\operatorname{Coeff}_m(g_i h_i)) \\ &= \sum_{i=1}^{N} \sum_{m=m_1 m_2} \pi(\operatorname{Coeff}_{m_1}(g_i) \operatorname{Coeff}_{m_2}(h_i)) \\ &= \sum_{i=1}^{N} \sum_{m=m_1 m_2} \operatorname{Coeff}_{m_1}(g_i)\pi(\operatorname{Coeff}_{m_2}(h_i)) \\ &= \sum_{i=1}^{N} \operatorname{Coeff}_m(g_i \pi(h_i)),\end{aligned}$$

where the inner sum is over all monomials m_1 and m_2 whose product is m. The equality

$$\pi(\operatorname{Coeff}_{m_1}(g_i) \operatorname{Coeff}_{m_2}(h_i)) = \operatorname{Coeff}_{m_1}(g_i)\pi(\operatorname{Coeff}_{m_2}(h_i))$$

follows from the fact that π is $\mathbb{F}$-linear and $\operatorname{Coeff}_{m_1}(g_i) \in \mathbb{F}$. Thus, $f(X) = \sum_{i=1}^{N} g_i(X)\pi(h_i(X))$. □

One can use Lemma II.12 to design PIT algorithms for circuit classes $\mathcal{C}$ that are too weak to efficiently compute a nonzero element of $I^{\det}_{n,m,r}$. If every small $\mathcal{C}$-circuit cannot compute a nonzero element of $I^{\det}_{n,m,r}$, then Lemma II.12 implies that the map $\mathcal{G}_{n,m,r-1}(Y, Z)$ of Construction II.11 is a hitting set generator for the class of small $\mathcal{C}$-circuits.

E. Complexity of Matrix Multiplication

This subsection introduces the language of tensors and their relationship with the complexity of matrix multiplication. For a more thorough treatment of tensors and matrix multiplication, we refer the reader to Bürgisser, Clausen, and Shokrollahi [57, Chapters 14 and 15] and Bläser [58].

For our purposes, a tensor T of order d is a set-multilinear polynomial in d disjoint sets of variables $X^{(1)}, \dots, X^{(d)}$. The fact that T is set-multilinear means that every monomial appearing in T is a product of d variables, where exactly one of these variables is taken from each of the sets $X^{(1)}, \dots, X^{(d)}$. That is, we can write T as

$$T(X^{(1)}, \dots, X^{(d)}) = \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} t_{i_1,\dots,i_d} x^{(1)}_{i_1} \cdots x^{(d)}_{i_d}.$$

We say that a tensor is rank-one if there are linear forms $\ell_1(X^{(1)}), \dots, \ell_d(X^{(d)})$ such that

$$T(X^{(1)}, \dots, X^{(d)}) = \ell_1(X^{(1)}) \cdots \ell_d(X^{(d)}).$$

The rank of T, written as $\mathrm{R}(T)$, is the minimal r such that T can be written as a sum of rank-one tensors. The border rank of T, denoted by $\underline{\mathrm{R}}(T)$, is the minimal r such that T can be obtained as a limit of rank-r tensors. More explicitly, a tensor T has border rank r if there are linear forms $\ell_{i,j}(X^{(i)}, \varepsilon) \in \mathbb{F}(\varepsilon)[X^{(i)}]$ such that

$$\sum_{j=1}^{r} \prod_{i=1}^{d} \ell_{i,j}(X^{(i)}, \varepsilon) = T(X^{(1)}, \dots, X^{(d)}) + O(\varepsilon)$$

and there is no such expression for $T + O(\varepsilon)$ involving fewer than r rank-one tensors.

We denote by $\langle n, m, p \rangle$ the order-3 tensor

$$\langle n, m, p \rangle \coloneqq \sum_{i=1}^{n} \sum_{j=1}^{m} \sum_{k=1}^{p} x_{i,j} y_{j,k} z_{i,k},$$

which corresponds to the multiplication of an $n \times m$ matrix with an $m \times p$ matrix. Note that $\langle n, m, p \rangle = \mathrm{tr}(XYZ^\top)$, a fact that we will use later on.

The complexity of $n \times n \times n$ matrix multiplication is captured by the rank of the tensor $\langle n, n, n \rangle$ (see, e.g., [57, Proposition 15.1]). We now define ω, the exponent of matrix multiplication.

Definition II.14. $\omega \coloneqq \inf\{\tau \in \mathbb{R} : \mathrm{R}(\langle n, n, n \rangle) \leqslant O(n^\tau)\}$. ♢

Bini [59] showed that one can equivalently define ω in terms of the border rank of $\langle n, n, n \rangle$.

Lemma II.15 ([59]). $\omega = \inf\{\tau \in \mathbb{R} : \underline{\mathrm{R}}(\langle n, n, n \rangle) \leqslant O(n^\tau)\}$.

As mentioned in the introduction, the obvious bounds on ω are $2 \leqslant \omega \leqslant 3$. The best-known upper bound on ω is due to Alman and Vassilevska Williams [13].

Theorem II.16 ([13]). $\omega < 2.37286$.

It is popularly conjectured that $\omega = 2$. There has been some progress on lower bounds for $\underline{\mathrm{R}}(\langle n, n, n \rangle)$, with the best-known lower bound due to Landsberg and Michałek [36].

Theorem II.17 ([36]). $\underline{\mathrm{R}}(\langle n, n, n \rangle) \geqslant 2n^2 - \log_2 n - 1$.

One can also consider the multiplicative complexity of matrix multiplication, where we do not restrict ourselves to computing variable-disjoint products of the form $\ell_1(X)\ell_2(Y)$, but instead consider products $\ell_1(X, Y)\ell_2(X, Y)$ of arbitrary linear polynomials. The following lemma shows that for matrix multiplication, border rank and border multiplicative complexity differ by at most a factor of 2. In the case of (exact) rank and multiplicative complexity, this is a well-known fact (see, e.g., [57, Eqn. 14.8] and the discussion preceding it). The proof for the case of border computation is nearly identical.

Lemma II.18 (cf. [57, Eqn. 14.8])**.** *Let $\underline{\mathrm{L}}(n)$ denote the border multiplicative complexity of $n \times n \times n$ matrix multiplication. Then $\underline{\mathrm{L}}(n) \leqslant \underline{\mathrm{R}}(\langle n, n, n \rangle) \leqslant 2\,\underline{\mathrm{L}}(n)$.*

III. Lifting Border Rank Lower Bounds to Determinantal Ideals

In this section, we will show that lower bounds on the border rank of matrix multiplication can be lifted to lower bounds on the border multiplicative complexity of any nonzero polynomial in the ideal $I^{\det}_{n,m,r} \subseteq \mathbb{F}[X]$. Letting $\underline{\mathrm{R}}(n) \coloneqq \underline{\mathrm{R}}(\langle n, n, n \rangle)$ be the border rank of $n \times n \times n$ matrix multiplication, our goal will be to prove a lower bound of order $\underline{\mathrm{R}}(r)$ on the border multiplicative complexity of the ideal $I^{\det}_{n,m,r}$. To do this, we make use of tools recently developed by Andrews and Forbes [48] to prove lower bounds on the complexity of polynomials in this ideal.

We now state and prove our main technical lemma, which is an analogue of [48, Lemma 3.6] for trace ABPs.

Lemma III.1. *Let $\mathbb{F}$ be a field of characteristic zero. Let $X^{(1)}, \dots, X^{(m)}$ be matrices of variables, where $X^{(i)}$ is an $n_i \times n_{i+1}$ matrix and $n_1 = n_{m+1}$. Let $N \coloneqq \sum_{i=1}^{m+1} n_i$. Let $\sigma = (\sigma_1, \dots, \sigma_p)$ be a non-increasing sequence of natural numbers with $\sigma_1 \geqslant N$. Then there is a matrix $M \in \mathbb{F}(\varepsilon)[X^{(1)}, \dots, X^{(m)}]^{\sigma_1 \times \sigma_1}$ where each entry $M_{i,j}$ is either a constant or a scalar multiple of a variable and we have*

$$\prod_{i=1}^{p} \det(M_{[\sigma_i],[\sigma_i]}) = 1 + \varepsilon \,\mathrm{tr}(X^{(1)} \cdots X^{(m)}) + O(\varepsilon^2).$$

Proof. Without loss of generality, it suffices to consider the case where $\sigma_1 = N$. If instead $\sigma_1 > N$, we extend the matrix M to a $\sigma_1 \times \sigma_1$ matrix by placing ones along the main diagonal and zeroes elsewhere.

Let G be the underlying directed graph of the trace ABP that computes $\mathrm{tr}(X^{(1)} \cdots X^{(m)})$. We modify G as follows:

- Add a self-loop of weight 1 to every vertex of G.
- Let $s_1, \dots, s_{n_1}$ denote the sources of G and $t_1, \dots, t_{n_1}$ the corresponding sinks. Add an edge of weight ε from t_i to s_i for every $i \in [n_1]$.

Let G' denote the resulting graph and let M' be the adjacency matrix of G', i.e.,

$$M' := \begin{pmatrix} I_{n_1} & X^{(1)} & 0 & 0 & \cdots & 0 & 0 \\ 0 & I_{n_2} & X^{(2)} & 0 & \cdots & 0 & 0 \\ 0 & 0 & I_{n_3} & X^{(3)} & \cdots & 0 & 0 \\ 0 & 0 & 0 & I_{n_4} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & I_{n_{m-1}} & X^{(m)} \\ \varepsilon I_{n_1} & 0 & 0 & 0 & \cdots & 0 & I_{n_m} \end{pmatrix}.$$

We will first determine $\prod_{i=1}^{p} \det(M'_{[\sigma_i],[\sigma_i]})$, after which we will modify M' to obtain the desired matrix M.

Fix some $k \in [N]$. If $k \leqslant \sum_{i=1}^{m-1} n_i$, then it is clear that $\det(M'_{[k],[k]}) = 1$, as $M'_{[k],[k]}$ is an upper triangular matrix with ones along the diagonal. For k in the range $N - n_m < k \leqslant N$, we compute $\det(M'_{[k],[k]})$ using the cycle cover interpretation of the determinant.

Let G'_k denote the graph whose adjacency matrix is $M'_{[k],[k]}$. Recall that $\det(M'_{[k],[k]})$ can be computed as

$$\det(M'_{[k],[k]}) = \sum_{C \in \mathscr{C}(G'_k)} (-1)^{\mathrm{even}(C)} \pi(C),$$

where $\mathscr{C}(G'_k)$ is the set of all cycle covers in G'_k, $\mathrm{even}(C)$ is the number of even cycles in C, and $\pi(C)$ is the product of the weights on the edges appearing in the cycle cover C. We partition the set of cycle covers of G'_k into three sets: those containing no edges of weight ε, those containing exactly one edge of weight ε, and those containing two or more edges of weight ε. In each case, we determine the contribution of these cycle covers to $\det(M'_{[k],[k]})$.

- Suppose C is a cycle cover with no edges of weight ε. The construction of G'_k implies that C must be the cycle cover consisting entirely of self-loops. This cycle cover contributes 1 to $\det(M'_{[k],[k]})$.
- Let C be a cycle cover containing exactly one edge labeled ε. By the construction of G'_k, the cycle in C containing the edge labeled ε must correspond to a path from s_i to t_i in G for some $i \in [k - (N - n_m)]$ together with the ε edge from t_i to s_i. Because every non-trivial cycle in G'_k must use an edge labeled ε, the remaining cycles in C consist of self-loops. Thus, C contributes a term of the form
$$(-1)^{m+1} \varepsilon X^{(1)}_{i,i_2} X^{(2)}_{i_2,i_3} \cdots X^{(m-1)}_{i_{m-1},i_m} X^{(m)}_{i_m,i}.$$
to $\det(M'_{[k],[k]})$, where the factor of $(-1)^{m+1}$ accounts for the parity of the length of the non-trivial cycle. There is exactly one such cycle cover for every $i \in [k - (N - n_m)]$ and every path from s_i to t_i in G_k. This implies that the set of all cycle covers containing exactly one edge of weight ε contributes
$$\begin{aligned} &(-1)^{m+1} \varepsilon \sum_{i_1=1}^{N-n_m+k} \sum_{i_2,\ldots,i_m} X^{(1)}_{i_1,i_2} X^{(2)}_{i_2,i_3} \cdots X^{(m)}_{i_m,i_1} \\ &= (-1)^{m+1} \varepsilon \operatorname{tr}((X^{(1)} \cdots X^{(m)})_{[k-(N-n_m)],[k-(N-n_m)]}) \end{aligned}$$
to $\det(M'_{[k],[k]})$.
- Finally, consider the case when C is a cycle cover containing two or more edges labeled by ε. By definition, this cycle cover contributes an $O(\varepsilon^2)$ term to $\det(M'_{[k],[k]})$, which we consider negligible.

In summary, we have

$$\begin{aligned} &\det(M'_{[k],[k]}) \\ &= 1 + (-1)^{m+1} \varepsilon \operatorname{tr}((X^{(1)} \cdots X^{(m)})_{[k-(N-n_m)],[k-(N-n_m)]}) \\ &\qquad + O(\varepsilon^2). \end{aligned}$$

Using this, we now determine $\prod_{i=1}^{p} \det(M'_{[\sigma_i],[\sigma_i]})$. Let $a_i := |\{j \in [p] : \sigma_j = i\}|$ count the number of elements of σ equal to i. The analysis above implies

$$\begin{aligned} &\prod_{i=1}^{p} \det(M'_{[\sigma_i],[\sigma_i]}) \\ &= \prod_{k=1}^{N} \det(M'_{[k],[k]})^{a_k} \\ &= \prod_{\ell=1}^{n_m} \det(M'_{[N-n_m+\ell],[N-n_m+\ell]})^{a_{N-n_m+\ell}} \\ &= \prod_{\ell=1}^{n_m} \Big(1 + (-1)^{m+1} \varepsilon \operatorname{tr}((X^{(1)} \cdots X^{(m)})_{[\ell],[\ell]}) \\ &\qquad\qquad + O(\varepsilon^2)\Big)^{a_{N-n_m+\ell}} \\ &= \prod_{\ell=1}^{n_m} \Big(1 + (-1)^{m+1} \varepsilon a_{N-n_m+\ell} \operatorname{tr}((X^{(1)} \cdots X^{(m)})_{[\ell],[\ell]}) \\ &\qquad\qquad + O(\varepsilon^2)\Big) \\ &= 1 + (-1)^{m+1} \varepsilon \sum_{\ell=1}^{n_m} a_{N-n_m+\ell} \operatorname{tr}((X^{(1)} \cdots X^{(m)})_{[\ell],[\ell]}) \\ &\quad + O(\varepsilon^2) \\ &= 1 + (-1)^{m+1} \varepsilon \sum_{i=1}^{n_m} \left[\left(\sum_{\ell=i}^{n_m} a_{N-n_m+\ell} \right) (X^{(1)} \cdots X^{(m)})_{i,i} \right] \\ &\quad + O(\varepsilon^2). \end{aligned}$$

We now perform a change of variables to transform the matrix M' into the desired matrix M. Let A be the diagonal matrix given by

$$A_{i,i} = \frac{1}{\sum_{\ell=i}^{n_m} a_{N-n_m+\ell}}.$$

(Note that the entries of A are well-defined, since $a_N \geqslant 1$ and $a_i \geqslant 0$ for all $i \in [N]$.) Let M be the image of M' under the change of variables $X^{(1)} \mapsto (-1)^{m+1} A X^{(1)}$. Then we have

$$\begin{aligned} &\left(\sum_{\ell=i}^{n_m} a_{N-n_m+\ell} \right) ((-1)^{m+1} A X^{(1)} \cdots X^{(m)})_{i,i} \\ &= (-1)^{m+1} A_{i,i}^{-1} A_{i,i} (X^{(1)} \cdots X^{(m)})_{i,i} \\ &= (-1)^{m+1} (X^{(1)} \cdots X^{(m)})_{i,i}, \end{aligned}$$

so

$$\prod_{i=1}^{p} \det(M_{[\sigma_i],[\sigma_i]}) = 1 + \varepsilon \sum_{i=1}^{n_m} (X^{(1)} \cdots X^{(m)})_{i,i} + O(\varepsilon^2)$$
$$= 1 + \varepsilon \operatorname{tr}(X^{(1)} \cdots X^{(m)}) + O(\varepsilon^2). \qquad \square$$

Remark III.2. In the proof of the preceding lemma, suppose we were to add edges of weight 1 from t_i to s_i for each i and add self-loops of weight 1 to all vertices. To compute $\operatorname{tr}(X^{(1)} \cdots X^{(d)})$ using the cycle cover interpretation of the determinant, we want to restrict ourselves to only count cycle covers containing a single edge t_i-s_i edge. We accomplish this by multiplying the weight of each such edge by a factor of ε, which guarantees that the linear term of the determinant of the adjacency matrix corresponds to cycle covers using exactly one t_i-s_i edge. In fact, we get more: the coefficient of ε^k in the determinant of the adjacency matrix corresponds to cycle covers using exactly k such edges.

A similar idea is used in algorithms for "exact" problems in combinatorial optimization. For example, the algorithms of Barahona and Pulleyblank [60] for counting exact arborescences and exact perfect matchings in planar graphs modify the edge weights of the graph in a manner similar to what we do in the proof of Lemma III.1. By exploiting the notion of border complexity, we avoid an interpolation step used in these combinatorial algorithms. ◊

Using the preceding lemma, we establish an analogue of [48, Theorem 3.8] for trace ABPs.

Proposition III.3. *Let $\mathbb{F}$ be a field of characteristic zero. 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 $g(\overline{y}) \in \mathbb{F}[\overline{y}]$ be a polynomial in the border of layered trace algebraic branching programs with at most r vertices. Then there is a depth-three h-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 h-oracle gate in the middle layer, and a single addition gate at the top layer.*
2) *Φ computes $g(\overline{y}) + O(\varepsilon)$.*

Proof. By Lemma II.7, it is sufficient to consider the case where the oracle gates compute $f(X)$ exactly. Using Proposition II.10, there are nm linear functions $\{\ell_{i,j}(X, \varepsilon) \in \mathbb{F}(\varepsilon)[X] : (i, j) \in [n] \times [m]\}$, an integer $q \in \mathbb{Z}$, a nonzero $\alpha \in \mathbb{F}$, and a sequence $\sigma = (\sigma_1, \ldots, \sigma_p)$ of natural numbers with $\sigma_1 \geqslant r$ such that

$$f(\ell_{1,1}(X, \varepsilon), \ldots, \ell_{n,m}(X, \varepsilon))$$
$$= \varepsilon^q \alpha \prod_{i=1}^{p} \det(X_{[\sigma_i],[\sigma_i]}) + O(\varepsilon^{q+1}).$$

By assumption, there is a polynomial $\widetilde{g}(\overline{y}, \varepsilon) \in \mathbb{F}(\varepsilon)[\overline{y}]$ such that $\widetilde{g}(\overline{y}, \varepsilon) = g(\overline{y}) + O(\varepsilon)$ and that $\widetilde{g}(\overline{y}, \varepsilon)$ can be computed by a layered trace ABP on s vertices for some $s \leqslant r$. That is, there are matrices of variables $Z^{(1)}, \ldots, Z^{(m)}$, where $Z^{(i)}$ is an $n_i \times n_{i+1}$ matrix, we have $n_1 = n_{m+1}$, and $\sum_{i=1}^{m+1} = s$, along with a projection $\varphi : Z^{(1)} \cup \cdots \cup Z^{(m)} \to \overline{y} \cup \mathbb{F}(\varepsilon)$ such that $\operatorname{tr}(\varphi(Z^{(1)}) \cdots \varphi(Z^{(m)})) = \widetilde{g}(\overline{y}, \varepsilon)$.

Applying Lemma III.1 to the matrices $Z^{(1)}, \ldots, Z^{(m)}$ and the sequence $(\sigma_1, \ldots, \sigma_p)$, we obtain a matrix $M(Z, \varepsilon) \in \mathbb{F}(\varepsilon)[Z^{(1)}, \ldots, Z^{(m)}]^{r \times r}$ such that

$$\prod_{i=1}^{p} \det(M(Z, \varepsilon)_{[\sigma_i],[\sigma_i]}) = 1 + \varepsilon \operatorname{tr}(Z^{(1)} \cdots Z^{(m)}) + O(\varepsilon^2).$$

We now compose $f(X)$, the linear functions $\ell_{i,j}(X, \varepsilon)$, the matrix $M(Z, \varepsilon)$, and the projection $\varphi : Z \to \overline{y} \cup \mathbb{F}(\varepsilon)$. Let

$$h(\overline{y}, \varepsilon, \delta)$$
$$:= f(\ell_{1,1}(M(\varphi(Z), \delta), \varepsilon), \ldots, \ell_{n,m}(M(\varphi(Z), \delta), \varepsilon)).$$

The preceding discussion implies

$$\begin{aligned} &h(\overline{y}, \varepsilon, \delta) \\ &= \varepsilon^q \alpha \cdot \prod_{i=1}^{p} \det(M(\varphi(Z), \delta)_{[\sigma_i],[\sigma_i]}) + O(\varepsilon^{q+1}) \\ &= \varepsilon^q \alpha \cdot \Big(1 + \delta \operatorname{tr}(\varphi(Z^{(1)}) \cdots \varphi(Z^{(m)})) + O(\delta^2)\Big) \\ &\quad + O(\varepsilon^{q+1}) \\ &= \varepsilon^q \alpha + \varepsilon^q \delta \alpha \widetilde{g}(\overline{y}, \varepsilon) + O(\varepsilon^q \delta^2) + O(\varepsilon^{q+1}). \end{aligned}$$

Performing the substitution $\varepsilon \mapsto \varepsilon^2$ and $\delta \mapsto \varepsilon$, we obtain

$$\begin{aligned} h(\overline{y}, \varepsilon^2, \varepsilon) &= \varepsilon^{2q} \alpha + \varepsilon^{2q+1} \alpha \widetilde{g}(\overline{y}, \varepsilon^2) + O(\varepsilon^{2q+2}) \\ &= \varepsilon^{2q} \alpha + \varepsilon^{2q+1} \alpha g(\overline{y}) + O(\varepsilon^{2q+2}). \end{aligned}$$

The desired f-oracle circuit is then given by

$$\Phi(\overline{y}) := \frac{h(\overline{y}, \varepsilon^2, \varepsilon) - \varepsilon^{2q} \alpha}{\varepsilon^{2q+1} \alpha} = g(\overline{y}) + O(\varepsilon). \qquad \square$$

We now use Proposition III.3 to lift lower bounds on the border rank of matrix multiplication to lower bounds on the border multiplicative complexity of the ideal $I^{\det}_{n,m,r}$.

Theorem III.4. *Let $\mathbb{F}$ be a field of characteristic zero. The border multiplicative complexity of any nonzero polynomial in $I^{\det}_{n,m,r}$ is bounded from below by $\frac{1}{6} \underline{\mathrm{R}}(r/4)$, where $\underline{\mathrm{R}}(n) := \underline{\mathrm{R}}(\langle n, n, n \rangle)$ is the border rank of $n \times n \times n$ matrix multiplication.*

Proof. Let Φ be a circuit of border multiplicative complexity s computing a nonzero polynomial in $I^{\det}_{n,m,r}$. Let X, Y, and Z be $r/4 \times r/4$ matrices of variables. The polynomial $\operatorname{tr}(XYZ)$ can naturally be computed by a layered trace ABP on r vertices. Applying Proposition III.3 to the circuit Φ yields a circuit Ψ of multiplicative complexity s that computes $\operatorname{tr}(XYZ) + O(\varepsilon)$. We then apply Lemma II.5 to Ψ to obtain a circuit of multiplicative complexity $3s$ that simultaneously computes all first-order partial derivatives of $\operatorname{tr}(XYZ) + O(\varepsilon)$.

Observe that the partial derivative of $\operatorname{tr}(XYZ)$ with respect to $z_{j,i}$ is, up to the $O(\varepsilon)$ error term, the (i, j) entry of the matrix product XY. Thus, we have a circuit of multiplicative complexity $3s$ that approximates the product of two $r/4 \times r/4$ matrices. By Lemma II.18, this implies that the border rank of $r/4 \times r/4 \times r/4$ matrix multiplication is bounded from above by $6s$. That is, we have $\underline{\mathrm{R}}(r/4) \leqslant 6s$. This yields the claimed lower bound on the multiplicative complexity of any nonzero polynomial in $I^{\det}_{n,m,r}$. □

Combining Theorem II.17 with Theorem III.4 yields the following unconditional lower bound on the border multiplicative complexity of all nonzero polynomials in the ideal $I^{\det}_{n,m,r}$.

Corollary III.5. *The border multiplicative complexity of any nonzero polynomial in $I^{\det}_{n,m,r}$ is bounded from below by $\frac{1}{48}r^2 - \frac{1}{6}\log_2 r + \frac{1}{6}$.*

IV. Constructing a Hitting Set Generator

In this section, we use Theorem III.4 to design hitting set generators for the closure of circuits of small multiplicative complexity. Letting $\underline{\mathrm{R}}(n) := \underline{\mathrm{R}}(\langle n,n,n \rangle)$ be the border rank of $n \times n \times n$ matrix multiplication, we will construct a generator with seed length $O(\sqrt{n}\,\underline{\mathrm{R}}^{-1}(s))$ for n-variate circuits of multiplicative complexity s. We stress that the correctness of this generator is unconditional.

Theorem IV.1. *Let $\mathbb{F}$ be a field of characteristic zero. Let $\underline{\mathrm{R}}(n) := \underline{\mathrm{R}}(\langle n,n,n \rangle)$ be the border rank of $n \times n \times n$ matrix multiplication. Then there is an explicit degree-two hitting set generator of seed length $8\sqrt{n}\,\underline{\mathrm{R}}^{-1}(6s+1)$ that hits the closure of n-variate circuits of multiplicative complexity s.*

Proof. Let Φ be an n-variate circuit of multiplicative complexity s that computes $\Phi(\overline{x}) + O(\varepsilon)$ for some nonzero polynomial $\Phi(\overline{x})$. Let $r := 4\,\underline{\mathrm{R}}^{-1}(6s+1)$. Arrange the input variables of $\Phi(\overline{x})$ into a $\sqrt{n} \times \sqrt{n}$ matrix. Let $\mathcal{G}_{n,m,r}(Y,Z)$ be the generator of Construction II.11. We claim that the generator $\mathcal{G}_{\sqrt{n},\sqrt{n},r-1}(Y,Z)$ hits $\Phi(\overline{x})$, i.e., that $\Phi(\mathcal{G}_{\sqrt{n},\sqrt{n},r-1}(Y,Z)) \neq 0$.

To see this, suppose instead that $\Phi(\mathcal{G}_{\sqrt{n},\sqrt{n},r-1}(Y,Z)) = 0$. Lemma II.12 implies that $\Phi(\overline{x}) \in I^{\det}_{\sqrt{n},\sqrt{n},r} \setminus \{0\}$. As $\Phi(\overline{x})$ has border multiplicative complexity s, it follows from Theorem III.4 that $6s \geqslant \underline{\mathrm{R}}(r/4)$. However, our choice of r implies $\underline{\mathrm{R}}(r/4) = 6s+1 > 6s$, a contradiction. Thus, it must be the case that in fact $\mathcal{G}_{\sqrt{n},\sqrt{n},r-1}(Y,Z)$ hits $\Phi(\overline{x})$. Since Φ was an arbitrary n-variate circuit of multiplicative complexity s, we conclude that $\mathcal{G}_{\sqrt{n},\sqrt{n},r-1}(Y,Z)$ hits all polynomials in the closure of n-variate circuits of multiplicative complexity s. Finally, note that the definition of $\mathcal{G}_{\sqrt{n},\sqrt{n},r-1}(Y,Z)$ immediately implies the claimed bounds on the seed length and degree of the generator. □

Combining Theorem IV.1 with Theorem II.17, we obtain the following corollary. To the best of our knowledge, this is the first non-trivial hitting set generator for circuits of multiplicative complexity $s \leqslant o(n)$.

Corollary IV.2. *There is an explicit hitting set generator of seed length $(8\sqrt{3}+o(1))\sqrt{ns}$ that hits the closure of n-variate circuits of multiplicative complexity s.*

One can also state Theorem IV.1 as a win-win result: either there are extremely fast algorithms for $n \times n \times n$ matrix multiplication, or there is a non-trivial deterministic algorithm for testing polynomial identities given by small circuits.

Corollary IV.3. *Let $\mathbb{F}$ be a field of characteristic zero and let ω denote the exponent of matrix multiplication over $\mathbb{F}$. At least one of the following is true.*

1) *$\omega = 2$.*
2) *For any positive constants $\varepsilon, \delta > 0$ that satisfy $2\omega\varepsilon + 2\delta < \omega - 2$, there is an explicit hitting set generator of seed length $O(n^{1-\varepsilon})$ that hits n-variate algebraic circuits of multiplicative complexity $O(n^{1+\delta})$. If these circuits are also restricted to have degree $n^{O(1)}$ and size $n^{O(1)}$, then this yields a deterministic algorithm to test identities given by such circuits that runs in $\exp(O(n^{1-\varepsilon}\log n))$ time.*

Acknowledgments

We thank Shubhang Kulkarni for telling us about the work of Barahona and Pulleyblank [60]. We also thank the anonymous reviewers for comments that helped improve the presentation of this work.

References

[1] V. Strassen, "Gaussian elimination is not optimal," *Numerische Mathematik*, vol. 13, pp. 354–356, 1969.

[2] V. Y. Pan, "Strassen's algorithm is not optimal. Trilinear technique of aggregating, uniting and canceling for constructing fast algorithms for matrix operations," in *Proceedings of the 19th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1978)*. IEEE, Long Beach, Calif., 1978, pp. 166–176.

[3] D. Bini, M. Capovani, F. Romani, and G. Lotti, "$o(n^{2.7799})$ complexity for $n \times n$ approximate matrix multiplication," *Information Processing Letters*, vol. 8, no. 5, pp. 234–235, 1979.

[4] V. Y. Pan, "New fast algorithms for matrix operations," *SIAM J. Comput.*, vol. 9, no. 2, pp. 321–342, 1980.

[5] A. Schönhage, "Partial and total matrix multiplication," *SIAM J. Comput.*, vol. 10, no. 3, pp. 434–455, 1981.

[6] F. Romani, "Some properties of disjoint sums of tensors related to matrix multiplication," *SIAM J. Comput.*, vol. 11, no. 2, pp. 263–267, 1982.

[7] D. Coppersmith and S. Winograd, "On the asymptotic complexity of matrix multiplication," *SIAM J. Comput.*, vol. 11, no. 3, pp. 472–492, 1982.

[8] V. Strassen, "Relative bilinear complexity and matrix multiplication," *J. Reine Angew. Math.*, vol. 375/376, pp. 406–443, 1987.

[9] D. Coppersmith and S. Winograd, "Matrix multiplication via arithmetic progressions," *J. Symbolic Comput.*, vol. 9, no. 3, pp. 251–280, 1990.

[10] A. M. Davie and A. J. Stothers, "Improved bound for complexity of matrix multiplication," *Proc. Roy. Soc. Edinburgh Sect. A*, vol. 143, no. 2, pp. 351–369, 2013.

[11] V. Vassilevska Williams, "Multiplying matrices faster than Coppersmith-Winograd," in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*. ACM, New York, 2012, pp. 887–898.

[12] F. Le Gall, "Powers of tensors and fast matrix multiplication," in *Proceedings of the 2014 International Symposium on Symbolic and Algebraic Computation (ISSAC 2014)*. ACM, New York, 2014, pp. 296–303.

[13] J. Alman and V. Vassilevska Williams, "A refined laser method and faster matrix multiplication," in *Proceedings of the 32nd Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2021)*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2021, pp. 522–539.

[14] H. Cohn and C. Umans, "A group-theoretic approach to fast matrix multiplication," in *Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2003)*, 2003, pp. 438–449.

[15] H. Cohn, R. Kleinberg, B. Szegedy, and C. Umans, "Group-theoretic algorithms for matrix multiplication," in *Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2005)*, 2005, pp. 379–388.

[16] H. Cohn and C. Umans, "Fast matrix multiplication using coherent configurations," in *Proceedings of the 24th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2013)*. SIAM, Philadelphia, PA, 2013, pp. 1074–1087.

[17] M. J. Fischer and A. R. Meyer, "Boolean matrix multiplication and transitive closure," in *12th Annual Symposium on Switching and Automata Theory (SWAT 1971)*, 1971, pp. 129–131.

[18] A. Schönhage, "Unitäre Transformationen grosser Matrizen," *Numer. Math.*, vol. 20, pp. 409–417, 1973.

[19] J. R. Bunch and J. E. Hopcroft, "Triangular factorization and inversion by fast matrix multiplication," *Math. Comp.*, vol. 28, pp. 231–236, 1974.

[20] W. Keller-Gehrig, "Fast algorithms for the characteristic polynomial," *Theoret. Comput. Sci.*, vol. 36, no. 2-3, pp. 309–317, 1985.

[21] L. G. Valiant, "General context-free recognition in less than cubic time," *J. Comput. System Sci.*, vol. 10, pp. 308–315, 1975.

[22] J. Nešetřil and S. Poljak, "On the complexity of the subgraph problem," *Comment. Math. Univ. Carolin.*, vol. 26, no. 2, pp. 415–419, 1985.

[23] M. B. Cohen, Y. T. Lee, and Z. Song, "Solving linear programs in the current matrix multiplication time," *J. ACM*, vol. 68, no. 1, jan 2021.

[24] J. van den Brand, "A deterministic linear program solver in current matrix multiplication time," in *Proceedings of the 31st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2020)*. USA: Society for Industrial and Applied Mathematics, 2020, pp. 259–278.

[25] S. Jiang, Z. Song, O. Weinstein, and H. Zhang, "A faster algorithm for solving general LPs," in *Proceedings of the 53rd Annual ACM Symposium on Theory of Computing (STOC 2021)*. New York, NY, USA: Association for Computing Machinery, 2021, pp. 823–832.

[26] A. Ambainis, Y. Filmus, and F. Le Gall, "Fast matrix multiplication: Limitations of the coppersmith-winograd method," in *Proceedings of the 47th Annual ACM Symposium on Theory of Computing (STOC 2015)*. New York, NY, USA: Association for Computing Machinery, 2015, pp. 585–593.

[27] J. Blasiak, T. Church, H. Cohn, J. A. Grochow, E. Naslund, W. F. Sawin, and C. Umans, "On cap sets and the group-theoretic approach to matrix multiplication," *Discrete Anal.*, no. 3, 2017.

[28] J. Blasiak, T. Church, H. Cohn, J. A. Grochow, and C. Umans, "Which groups are amenable to proving exponent two for matrix multiplication?" 2017.

[29] J. Alman and V. Vassilevska Williams, "Further Limitations of the Known Approaches for Matrix Multiplication," in *9th Innovations in Theoretical Computer Science Conference (ITCS 2018)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), A. R. Karlin, Ed., vol. 94. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2018, pp. 25:1–25:15.

[30] ——, "Limits on all known (and some unknown) approaches to matrix multiplication," in *Proceedings of the 59th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2018)*, 2018, pp. 580–591.

[31] J. Alman, "Limits on the universal method for matrix multiplication," *Theory of Computing*, vol. 17, no. 1, pp. 1–30, 2021.

[32] M. Christandl, P. Vrana, and J. Zuiddam, "Barriers for Fast Matrix Multiplication from Irreversibility," in *Proceedings of the 34th Annual Computational Complexity Conference (CCC 2019)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), A. Shpilka, Ed., vol. 137. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2019, pp. 26:1–26:17.

[33] M. Christandl, F. Le Gall, V. Lysikov, and J. Zuiddam, "Barriers for rectangular matrix multiplication," *CoRR*, vol. abs/2003.03019, 2020.

[34] M. Bläser, "A $\frac{5}{2}n^2$-lower bound for the rank of $n \times n$-matrix multiplication over arbitrary fields," in *Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1999)*, 1999, pp. 45–50.

[35] A. Shpilka, "Lower bounds for matrix product," *SIAM J. Comput.*, vol. 32, no. 5, pp. 1185–1200, 2003.

[36] J. M. Landsberg and M. Michałek, "A $2n^2 - \log_2(n) - 1$ lower bound for the border rank of matrix multiplication," *Int. Math. Res. Not. IMRN*, no. 15, pp. 4722–4733, 2018.

[37] R. Raz, "On the complexity of matrix product," *SIAM Journal on Computing*, vol. 32, no. 5, pp. 1356–1369, 2003.

[38] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM*, vol. 27, no. 4, pp. 701–717, 1980.

[39] R. Zippel, "Probabilistic algorithms for sparse polynomials," in *Proceedings of the International Symposium on Symbolic and Algebraic Computation, EUROSAM 1979*, 1979, pp. 216–226.

[40] A. Shpilka and A. Yehudayoff, "Arithmetic circuits: A survey of recent results and open questions," *Foundations and Trends in Theoretical Computer Science*, vol. 5, no. 3-4, pp. 207–388, 2010.

[41] N. Saxena, "Progress on polynomial identity testing," *Bulletin of the EATCS*, vol. 99, pp. 49–79, 2009.

[42] ——, "Progress on polynomial identity testing ii," in *Proceedings of the Workshop celebrating Somenath Biswas' 60th Birthday*, 2014, pp. 131–146.

[43] V. Kabanets and R. Impagliazzo, "Derandomizing polynomial identity tests means proving circuit lower bounds," *Computational Complexity*, vol. 13, no. 1-2, pp. 1–46, 2004.

[44] Z. Dvir, A. Shpilka, and A. Yehudayoff, "Hardness-randomness tradeoffs for bounded depth arithmetic circuits," *SIAM J. Comput.*, vol. 39, no. 4, pp. 1279–1293, 2009.

[45] C.-N. Chou, M. Kumar, and N. Solomon, "Closure results for polynomial factorization," *Theory of Computing*, vol. 15, no. 13, pp. 1–34, 2019.

[46] Z. Guo, M. Kumar, R. Saptharishi, and N. Solomon, "Derandomization from algebraic hardness," *SIAM Journal on Computing*, vol. 51, no. 2, pp. 315–335, 2022.

[47] R. Andrews, "Algebraic Hardness Versus Randomness in Low Characteristic," in *35th Computational Complexity Conference (CCC 2020)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), S. Saraf, Ed., vol. 169. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2020, pp. 37:1–37:32.

[48] R. Andrews and M. A. Forbes, "Ideals, determinants, and straightening: Proving and using lower bounds for polynomial ideals," in *Proceedings of the 54th Annual ACM Symposium on Theory of Computing (STOC 2022)*, 2022, pp. 389–402.

[49] M. Kumar and R. Saptharishi, "Hardness-randomness tradeoffs for algebraic computation," *Bull. Eur. Assoc. Theor. Comput. Sci.*, vol. 129, pp. 56–87, 2019.

[50] A. R. Klivans and D. Spielman, "Randomness efficient identity testing of multivariate polynomials." New York, NY, USA: Association for Computing Machinery, 2001.

[51] W. Baur and V. Strassen, "The complexity of partial derivatives," *Theoretical Computer Science*, vol. 22, no. 3, pp. 317–330, 1983.

[52] P. Dutta, N. Saxena, and T. Thierauf, "A Largish Sum-Of-Squares Implies Circuit Hardness and Derandomization," in *Proceedings of the 12th Annual Conference on Innovations in Theoretical Computer Science (ICS 2021)*, ser. Leibniz International Proceedings in Informatics (LIPIcs), vol. 185. Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2021, pp. 23:1–23:21.

[53] L. G. Valiant, "Completeness classes in algebra." New York, NY, USA: Association for Computing Machinery, 1979.

[54] R. Saptharishi, "A survey of lower bounds in arithmetic circuit complexity," 2019.

[55] P. Bürgisser, "The complexity of factors of multivariate polynomials," *Foundations of Computational Mathematics*, vol. 4, no. 4, pp. 369–396, 2004.

[56] W. Bruns and U. Vetter, *Determinantal rings*, ser. Lecture Notes in Mathematics. Springer-Verlag, Berlin, 1988, vol. 1327.

[57] P. Bürgisser, M. Clausen, and M. A. Shokrollahi, *Algebraic complexity theory*, ser. Grundlehren der Mathematischen Wissenschaften [Fundamental Principles of Mathematical Sciences]. Springer-Verlag, Berlin, 1997, vol. 315, with the collaboration of Thomas Lickteig.

[58] M. Bläser, *Fast Matrix Multiplication*, ser. Graduate Surveys. Theory of Computing Library, 2013, no. 5.

[59] D. Bini, "Relations between exact and approximate bilinear algorithms. applications," *Calcolo*, vol. 17, pp. 87–97, 1980.

[60] F. Barahona and W. R. Pulleyblank, "Exact arborescences, matchings and cycles," *Discrete Applied Mathematics*, vol. 16, no. 2, pp. 91–99, 1987.