A Note on Polynomial Identity Testing for Depth-3 Circuits

V. Arvind∗ Abhranil Chatterjee † Rajit Datta †
Partha Mukhopadhyay§
May 22, 2018

Abstract

Let $C$ be a depth-3 arithmetic circuit of size at most $s$, computing a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ (where $\mathbb{F} = \mathbb{Q}$ or $\mathbb{C}$) and the fan-in of the product gates of $C$ is bounded by $d$. We give a deterministic polynomial identity testing algorithm to check whether $f \equiv 0$ or not in time $2^d \text{poly}(n, s)$.

Over finite fields, for $\text{Char}(\mathbb{F}) > d$ we give a deterministic algorithm of running time $2^{\gamma d} \text{poly}(n, s)$ where $\gamma \leq 5$.

1 Introduction

Polynomial Identity Testing (PIT) is the following problem: Given an arithmetic circuit $C$ computing a polynomial in $\mathbb{F}[x_1, \ldots, x_n]$, determine whether $C$ computes an identically zero polynomial or not. The problem can be presented either in the white-box model or in the black-box model. In the white-box model, the arithmetic circuit is given explicitly as the input. In the black-box model, the arithmetic circuit is given black-box access, and the circuit can be evaluated over any point in $\mathbb{F}^n$ (or $\mathbb{F}^n$ where $\mathbb{F} \subseteq F$ is an extension field). Over the years, the problem has played pivotal role in many important results in complexity theory and algorithms: Primality Testing [AKS04], the PCP Theorem [ALM+98], IP = PSPACE [Sha90], graph matching algorithms [Lov79, MVV87]. The problem PIT admits a co-RP algorithm

∗Institute of Mathematical Sciences, Chennai, India, email: arvind@imsc.res.in
†Institute of Mathematical Sciences, Chennai, India, email: abhranilc@imsc.res.in
‡Chennai Mathematical Institute, Chennai, India, email: rajit@cmi.ac.in
§Chennai Mathematical Institute, Chennai, India, email: partham@cmi.ac.in
via the Schwartz-Zippel-Lipton-DeMillo Lemma [Sch80,Zip79,DL78], but an efficient deterministic algorithm is known only in some special cases. An important result of Impagliazzo and Kabanets [KI04] (also, see [HS80,Agr05]) shows a connection between the existence of a subexponential time PIT algorithm and arithmetic circuit lower bounds. We refer the reader to the survey of Shpilka and Yehudayoff [SY10] for the exposition of important results in arithmetic circuit complexity, and polynomial identity testing problem. 

In a surprising result, Agrawal and Vinay [AV08] show that an efficient deterministic PIT algorithm only for depth-4 $\Sigma\Pi\Sigma\Pi$ circuits is sufficient for obtaining an efficient deterministic PIT algorithm for the general arithmetic circuits. The main technical ingredient in their proof is an ingenious depth-reduction technique. Over characteristic zero fields, derandomization of PIT even for depth-3 $\Sigma\Pi\Sigma$ circuits suffices [GKKS13].

Motivated by the results of [KI04,Agr05,AV08], a large body of research consider the polynomial identity testing problem for restricted classes of depth-3 and depth-4 circuits. A particularly popular model in depth three arithmetic circuits is $\Sigma\Pi\Sigma(k)$ circuit, where the fan-in of the top $\Sigma$ gate is bounded by $k$. Dvir and Shpilka have shown a white-box quasi-polynomial time deterministic PIT algorithm for $\Sigma\Pi\Sigma(k)$ circuits [DS07]. Kayal and Saxena have given a deterministic $\text{poly}(d^k, n, s)$ white-box algorithm for the same problem [KS07]. Following the result of [KS07] (also see [AM10] for a different analysis), Karnin and Shpilka have given the first black-box quasi-polynomial time algorithm for $\Sigma\Pi\Sigma(k)$ circuits [KST11]. Later, Kayal and Saraf [KS09] have shown a polynomial-time deterministic black-box PIT algorithm for the same class of circuits over $\mathbb{Q}$ or $\mathbb{R}$. Finally, Saxena and Sheshadhri have settled the situation completely by giving a deterministic polynomial-time black-box algorithm for $\Sigma\Pi\Sigma(k)$ circuits [SS12] over any field. Recently, Oliveira et al. have given a sub-exponential PIT-algorithm for depth-3 and depth-4 multilinear formulas [dOSlV16].

For general depth-3 $\Sigma\Pi\Sigma$ circuits with $\times$-gate fan-in bounded by $d$ no deterministic algorithm with running time better than $\min\{d^n, n^d\} \text{poly}(n, d)$ is known. Our main results are the following.

**Theorem 1.** Let $C$ be a depth-3 $\Sigma\Pi\Sigma$ circuit of size at most $s$, computing a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ (where $\mathbb{F} = \mathbb{Q}$ or $\mathbb{C}$) and the fan-in of the product gates of $C$ is bounded by $d$. We give a white-box deterministic polynomial time identity testing algorithm to check whether $f \equiv 0$ or not in time $2^d \text{poly}(n, s)$.

As an immediate corollary we get the following.
Corollary 1. Let $C$ be a depth-3 $\Sigma\Pi\Sigma$ circuit of size at most $s$, computing a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ (where $\mathbb{F} = \mathbb{Q}$ or $\mathbb{C}$) and the fan-in of the product gates of $C$ is bounded by $O(\log n)$. We give a deterministic $\text{poly}(n, s)$ time identity testing algorithm to check whether $f \equiv 0$ or not.

Over the fields of positive characteristics, we show the following result.

Theorem 2. Let $C$ be a depth-3 $\Sigma\Pi\Sigma$ circuit of size at most $s$, computing a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ and the fan-in of the product gates of $C$ is bounded by $d$. For $\text{Char}(\mathbb{F}) > d$, we give a white-box deterministic polynomial time identity testing algorithm to check whether $f \equiv 0$ or not in time $2^{\gamma d} \text{poly}(n, s)$. The constant $\gamma$ is at most 5.

2 Organization

The paper is organized as follows. Section 3 contains preliminary materials. In Section 4 we prove Theorem 1. Theorem 4 is proved in Section 5.

3 Preliminaries

For a monomial $m$ and a polynomial $f$, let $[m]f$ denote the coefficient of the monomial $m$ in $f$. We denote the field of rational numbers as $\mathbb{Q}$, and the field of complex numbers as $\mathbb{C}$. The depth-3 $\Sigma\Pi\Sigma(s, d)$ circuits compute polynomials of the following form:

$$C(x_1, \ldots, x_n) = \sum_{i=1}^{s} \prod_{j=1}^{d} L_{i,j}(x_1, \ldots, x_n).$$

where $L_{i,j}$’s are affine linear forms over $\mathbb{F}$. The following observation is well-known and it says that for PIT purpose it is sufficient to consider homogeneous circuits.

Observation 1. Let $C(x_1, \ldots, x_n)$ be a $\Sigma\Pi\Sigma(s, d)$ circuit. Then $C \equiv 0$ if and only if $z^d C(x_1/z, \ldots, x_n/z) \equiv 0$ where $z$ is a new variable.

We use the notation $\Sigma[s]\Pi[d]\Sigma$ to denote homogeneous depth-3 circuits of top $\Sigma$ gate fan-in $s$, product gates fan-in bounded by $d$.

We recall the definition of Hadamard Product of two polynomials. The concept of Hadamard product is particularly useful in noncommutative computations [AJ09, AS18].

3
Definition 1. Given two degree $d$ polynomials $f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, the Hadamard Product $f \circ g$ is defined as

$$f \circ g = \sum_{m: \deg(m) \leq d} ([m]f \cdot [m]g) \cdot m.$$ 

For the PIT purpose in the commutative setting, we adapt the notion of Hadamard Product suitably and define a scaled version of Hadamard Product of two polynomials.

Definition 2. Given two degree $d$ polynomials $f, g \in \mathbb{F}[x_1, x_2, \ldots, x_n]$, the scaled version of the Hadamard Product $f \circ^s g$ is defined as

$$f \circ^s g = \sum_{m: \deg(m) \leq d} (m! \cdot [m]f \cdot [m]g) \cdot m$$

where $m = x_1^{e_1} x_2^{e_2} \cdots x_r^{e_r}$ for some $r \leq d$ and by abusing the notation we define $m! = e_1! \cdot e_2! \cdot \cdots \cdot e_r!$.

For the purpose of PIT over $\mathbb{Q}$, it is enough to be able to compute $f \circ^s f(1, 1, \ldots, 1)$. As $f \circ^s f$ has only non-negative coefficients, we will see a non-zero value when we compute $f \circ^s f(1, 1, \ldots, 1)$ if and only if $f \not\equiv 0$. Over $\mathbb{C}$ it is enough to compute $f \circ^s \overline{f}(1, 1, \ldots, 1)$ where $\overline{f}$ denotes the polynomial obtained by conjugating every coefficient of $f$.

We also recall a result of Ryser [Rys63] that gives a $\Sigma^{[2^n]} \Pi^{[n]} \Sigma$ circuit for the Permanent polynomial of $n \times n$ symbolic matrix.

Lemma 1 (Ryser [Rys63]). For a matrix $X$ with variables $x_{ij} : 1 \leq i, j \leq n$ as entries,

$$\text{Perm}(X) = (-1)^n \sum_{S \subseteq [n]} (-1)^{|S|} \prod_{i=1}^n \left( \sum_{j \in S} x_{ij} \right).$$

Lemma 2. For a monomial $m = x_{i_1} x_{i_2} \cdots x_{i_d}$ ($i_1, \ldots, i_d$ need not be distinct) and a homogeneous $\Pi \Sigma$ circuit $C = \prod_{j=1}^d L_j$ we have:

$$[m]C = \frac{1}{m!} \sum_{\sigma \in S_d} \prod_{j=1}^d ([x_{i_j}] L_{\sigma(j)}).$$

Proof. The monomial $m$ can be obtained from $C$ by first fixing a bijection $\sigma : [d] \rightarrow [d]$ and considering the coefficient $[m]C_\sigma = \prod_{j=1}^d [x_{i_{\sigma(j)}}] L_j = \prod_{j=1}^d [x_{i_j}] L_{\sigma^{-1}(j)}$. This is one way of generating this monomial and this
monomial \( m \) can be generated in many different orders. The final \([m]C\) is the sum of all coefficients \([m]C_\sigma\) generated in all distinct orders.

Now if \( m = x_{i_1}^{e_1} x_{i_2}^{e_2} \ldots x_{i_r}^{e_r} \) for some \( r \leq d \) then for a fixed \( \sigma \) one can obtain \( m! \) different bijections that do not change the string \( x_{i_{\sigma(1)}} x_{i_{\sigma(2)}} \ldots x_{i_{\sigma(d)}} \) and these will generate the same coefficient \( \prod_{j=1}^{d} [x_{i_{\sigma(j)}}] L_j \). Thus only the bijections that produce a different string from \( x_{i_{\sigma(1)}} x_{i_{\sigma(2)}} \ldots x_{i_{\sigma(d)}} \) are relevant. To account for the coefficients produced by the extra bijections we divide by \( m! \)

Now we are ready to prove the main theorems.

4 The results over zero characteristics

To prove Theorem 1, the following theorem is sufficient.

**Theorem 3.** Given a homogeneous \( \Sigma^{[s]} \Pi^{[d]} \Sigma \) circuit \( C \) computing a degree \( d \) polynomial in \( \mathbb{F}[x_1, x_2, \ldots, x_n] \) (where \( \mathbb{F} = \mathbb{Q} \) or \( \mathbb{C} \)), we can test whether \( C \equiv 0 \) or not deterministically in \( 2^d \text{poly}(s, n) \) time.

**Proof.** For simplicity, we present the proof only over \( \mathbb{Q} \). Over \( \mathbb{C} \), we need a minor modification as explained in Remark 1. Given the circuit \( C \) we compute \( C \circ^s C \) and evaluate at \((1, 1, \ldots, 1)\) point. Notice that over rationals, \( C \circ^s C \) has non-negative coefficients. This also implies that \( C \equiv 0 \) if and only if \( C \circ^s C(1, 1, \ldots, 1) = 0 \). So it is sufficient to show that \( C \circ^s C(1, \ldots, 1) \) can be computed deterministically in time \( 2^d \text{poly}(s, n) \). Since the scaled Hadamard Product distributes over addition, we only need to show that the scaled Hadamard Product of two \( \Pi \Sigma \) circuits can be computed efficiently.

**Lemma 3.** Given two homogeneous \( \Pi^{[d]} \Sigma \) circuits \( C_1 = \prod_{i=1}^{d} L_i \) and \( C_2 = \prod_{i=1}^{d} L'_i \); we have:

\[
C_1 \circ^s C_2 = \sum_{\sigma \in S_d} \prod_{i=1}^{d} (L_i \circ^s L'_{\sigma(i)}).
\]

**Proof.** We prove the formula monomial by monomial. Let \( m = x_{i_1} x_{i_2} \ldots x_{i_d} \) be a monomial in \( C_1 \) (Note that \( i_1, i_2, \ldots, i_d \) need not be distinct).
Now let \( m \) be a monomial that appears in both \( C_1 \) and \( C_2 \). From Lemma \( \ref{lemma2} \) the coefficients are

\[
[m] C_1 = \alpha_1 = \frac{1}{m!} \left( \sum_{\sigma \in S_d} \prod_{j=1}^{d} \left[ x_{i_j} \right] L_{\sigma(j)} \right)
\]

and

\[
[m] C_2 = \alpha_2 = \frac{1}{m!} \left( \sum_{\pi \in S_d} \prod_{j=1}^{d} \left[ x_{i_j} \right] L'_{\pi(j)} \right)
\]

respectively.

From the definition \( \ref{definition2} \) we have

\[
[m](C_1 \circ^s C_2) = m! \cdot \alpha_1 \cdot \alpha_2.
\]

Now let us consider the matrix \( T \) where \( T_{ij} = L_i \circ^s L_j' : 1 \leq i, j \leq d \) and

\[
\text{Perm}(T) = \sum_{\sigma \in S_d} \prod_{i=1}^{d} L_i \circ^s L'_{\sigma(i)}.
\]

The coefficient of \( m \) in \( \text{Perm}(T) \) is

\[
[m] \text{Perm}(T) = \sum_{\sigma \in S_d} [m] \left( \prod_{j=1}^{d} L_j \circ^s L'_{\sigma(j)} \right).
\]

Similar to Lemma \( \ref{lemma2} \) we notice the following.

\[
[m] \text{Perm}(T) = \sum_{\sigma \in S_d} \frac{1}{m!} \sum_{\pi \in S_d} \prod_{j=1}^{d} \left[ x_{i_j} \right] (L_{\pi(j)} \circ^s L'_{\sigma(\pi(j))})
\]

\[
= \frac{1}{m!} \sum_{\sigma \in S_d} \sum_{\pi \in S_d} \prod_{j=1}^{d} \left( \left[ x_{i_j} \right] L_{\pi(j)} \right) \cdot \left( \left[ x_{i_j} \right] L'_{\sigma(\pi(j))} \right)
\]

\[
= \frac{1}{m!} \sum_{\sigma \in S_d} \sum_{\pi \in S_d} \prod_{j=1}^{d} \left( \left[ x_{i_j} \right] L_{\pi(j)} \right) \cdot \prod_{j=1}^{d} \left( \left[ x_{i_j} \right] L'_{\sigma(\pi(j))} \right)
\]

\[
= \sum_{\pi \in S_d} \left( \prod_{j=1}^{d} \left( \left[ x_{i_j} \right] L_{\pi(j)} \right) \cdot \frac{1}{m!} \sum_{\sigma \in S_d} \prod_{j=1}^{d} \left( \left[ x_{i_j} \right] L'_{\sigma(\pi(j))} \right) \right)
\]

\[
= m! \cdot \frac{1}{m!} \sum_{\pi \in S_d} \left( \prod_{j=1}^{d} \left( \left[ x_{i_j} \right] L_{\pi(j)} \right) \cdot \frac{1}{m!} \sum_{\sigma \in S_d} \prod_{j=1}^{d} \left( \left[ x_{i_j} \right] L'_{\sigma(\pi(j))} \right) \right).
\]

Clearly, for any fixed \( \pi \in S_d \), we have that \( \sum_{\sigma \in S_d} \prod_{j=1}^{d} \left[ x_{i_j} \right] L'_{\sigma(\pi(j))} = m! \alpha_2 \). Hence, \([m] \text{Perm}(T) = m! \cdot \alpha_1 \cdot \alpha_2 \) and the lemma follows.

\( \square \)
Lemma 4. Given two $\Pi[d]\Sigma$ circuits $C_1$ and $C_2$ we can compute a $\Sigma[2^d]\Pi[d]\Sigma$ for $C_1 \circ^s C_2$ in time $2^d \text{poly}(n, d)$.

Proof. From Lemma 3 we observe that $\text{Perm}(T)$ gives a circuit for $C_1 \circ^s C_2$. A $\Sigma[2^d]\Pi[d]\Sigma$ circuit for $\text{Perm}(T)$ can be computed in $2^d \text{poly}(n, d)$ time using Lemma 1.

Now we show how to take the scaled Hadamard Product of two $\Sigma\Pi\Sigma$ circuits.

Lemma 5. Given two $\Sigma\Pi[d]\Sigma$ circuits $C = \sum_{i=1}^{s} P_i$ and $\tilde{C} = \sum_{i=1}^{\tilde{s}} \tilde{P}_i$ We can compute a $\Sigma[2^d\tilde{s}]\Pi[d]\Sigma$ circuit for $C \circ^s \tilde{C}$ in time $2^d \text{poly}(s, \tilde{s}, d, n)$.

Proof. We first note that by distributivity,

$$C \circ^s \tilde{C} = \sum_{i=1}^{s} \sum_{j=1}^{\tilde{s}} P_i \circ^s \tilde{P}_j.$$

Using Lemma 4 for each pair $P_i \circ^s \tilde{P}_j$ we get a $\Sigma[2^d]\Pi[d]\Sigma$ circuit $P_{ij}$. Now the formula $\sum_{i=1}^{s} \sum_{j=1}^{\tilde{s}} P_{ij}$ is a $\Sigma[2^d\tilde{s}]\Pi[d]\Sigma$ formula which can be computed in $2^d \text{poly}(s, \tilde{s}, d, n)$ time.

Remark 1. To adapt the algorithm over $\mathbb{C}$, we need to just compute $C \circ^s \tilde{C}$ where $\tilde{C}$ is the polynomial obtained from $C$ by conjugating each coefficient. Note that a circuit computing $\tilde{C}$ can be obtained from $C$ by just conjugating the scalars that appear in the linear forms of $C$. This follows from the fact that the conjugation operation distributes over addition and multiplication. Now we have $[m](C \circ^s \tilde{C}) = [m](C)^2$, so the coefficients are all positive and thus evaluating $C \circ^s \tilde{C}(1, 1, \ldots, 1)$ is sufficient for the PIT algorithm.
5 The results over finite fields

In this section we extend the PIT results over the finite fields. Now we state the main theorem of the section.

**Theorem 4.** Let $C$ be a depth-3 $\Sigma\Pi\Sigma$ circuit of size at most $s$, computing a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ and the fan-in of the product gates of $C$ is bounded by $d$. For $\text{Char}(\mathbb{F}) > d$, we give a white-box deterministic polynomial time identity testing algorithm to check whether $f \equiv 0$ or not in time $2^{\gamma d} \text{poly}(n, s)$. The constant $\gamma$ is at most $5$.

**Proof.** Consider first the case when $p = \text{Char}(\mathbb{F}) > d$. From Lemma 2, notice that for any $\Pi[d] \Sigma$ circuit $P$, 

$$[m]P = \frac{1}{m!} \sum_{\sigma \in S_d} \prod_{j=1}^{d} ([x_{i_j}] L_{\sigma(j)}).$$

and $m! \neq 0 \mod p$. Now define the $d \times d$ matrix $T_P$ such that each row of $T_P$ is just the linear forms $L_1 L_2 \ldots L_d$ appearing in $P$. Clearly the following is true.

$$\text{Perm}(T_P) = \sum_{\sigma \in S_d} \prod_{j=1}^{d} L_{\sigma(j)}.$$

Use Ryser’s formula given by Lemma 1 to express $\text{Perm}(T_P)$ as a depth-3 $\Sigma[2^d] \Pi[d] \Sigma$ circuit. If $C = P_1 + \ldots + P_s$, consider the polynomial $f_C = \sum_{i=1}^{s} \text{Perm}(T_{P_i})$. Notice that $f_C$ can be expressed as $\Sigma[2^d, s] \Pi[d] \Sigma$ circuit. Consider the noncommutative version of the polynomial $f_C$ which we denote as $f_C^{nc}$. Clearly we have a noncommutative ABP for $f_C^{nc}$ of width $w = 2^d \cdot s$ and $d$ many layers.

Now we make an important observation from the proof of Lemma 2. Suppose $\mathcal{M}$ be the set of all monomials of degree $d$ over $x_1, \ldots, x_n$. For a fixed monomial $m \in \mathcal{M}$ of form $x_{i_1} x_{i_2} \ldots x_{i_d}$ where $i_1 \leq i_2 \leq \ldots \leq i_d$ and $\sigma \in S_d$, define $m^\sigma = x_{i_{\sigma(1)}} x_{i_{\sigma(2)}} \ldots x_{i_{\sigma(d)}}$. The monomial $m$ can be present in $f_C^{nc}$ in different orders $m^\sigma$. We claim that $f \equiv 0$ if and only if $f_C^{nc} \equiv 0$. To see the claim, the following simple lemma suffices.

**Lemma 6.** Let $f = \sum_{m \in \mathcal{M}} [m] f \cdot m$ where $[m] f \in \mathbb{F}$ for all monomials $m \in \mathcal{M}$. Then 

$$f_C^{nc} = \sum_{m \in \mathcal{M}} \sum_{\sigma \in S_d} m! \cdot [m] f \cdot m^\sigma.$$

\[\text{Again, we identify the linear forms as } L_1, L_2, \ldots, L_d \text{ where } L_1, \ldots, L_{e_1} \text{ are the same, } L_{e_1+1}, \ldots, L_{e_1+e_2} \text{ are the same and so on.}\]
Proof. Let $x_{i_1} \ldots x_{i_d}$ be a fixed ordering of a monomial $m$ appearing in $f_C^{nc}$. The coefficient of $x_{i_1} \ldots x_{i_d}$ in $\text{Perm}(T_p) = \sum_{\sigma \in S_d} \prod_{j=1}^{d} L_{\sigma(j)}$ is simply $\sum_{\sigma \in S_d} \prod_{j=1}^{d} [x_{i_j}] L_{\sigma(j)}$. But from Lemma 2, $\sum_{\sigma \in S_d} \prod_{j=1}^{d} [x_{i_j}] L_{\sigma(j)}$ is exactly $m! \cdot [m] P$. Since $[m] f = \sum_{i=1}^{s} [m] P_i$, the lemma follows.

Now we apply the identity testing algorithm of Raz and Shpilka for noncommutative ABPs on the ABP of $f_C^{nc}$ to get the desired result [RS05]. The bound on $\gamma$ comes from Theorem 4 of their paper [RS05].

As an immediate application of Theorem 4, we state the following corollary.

**Corollary 2.** Let $C$ be a depth-$3$ $\Sigma\Pi\Sigma$ circuit of size at most $s$, computing a polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ and the fan-in of the product gates of $C$ is bounded by $d$. Suppose that $\text{Char}(\mathbb{F}) > d$. For $d = O(\log n)$, we give a deterministic $\text{poly}(n, s)$ time identity testing algorithm to check whether $f \equiv 0$ or not.

**References**

[Agr05] Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science, 25th International Conference*, pages 92–105, 2005.

[AJ09] Vikraman Arvind and Pushkar S. Joglekar. Arithmetic circuits, monomial algebras and finite automata. In *Mathematical Foundations of Computer Science 2009, 34th International Symposium, MFCS 2009, Novy Smokovec, High Tatras, Slovakia, August 24-28, 2009. Proceedings*, pages 78–89, 2009.

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Ann. of Math.*, 160(2):781–793, 2004.

[ALM+98] Sanjeev Arora, Carsten Lund, Rajeev Motwani, Madhu Sudan, and Mario Szegedy. Proof verification and the hardness of approximation problems. *J. ACM*, 45(3):501–555, 1998.
[AM10] Vikraman Arvind and Partha Mukhopadhyay. The ideal membership problem and polynomial identity testing. *Inf. Comput.*, 208(4):351–363, 2010.

[AS18] Vikraman Arvind and Srikanth Srinivasan. On the hardness of the noncommutative determinant. *Computational Complexity*, 27(1):1–29, 2018.

[AV08] Manindra Agrawal and V Vinay. Arithmetic circuits: A chasm at depth four. In *Proceedings-Annual Symposium on Foundations of Computer Science*, pages 67–75. IEEE, 2008.

[DL78] Richard A. DeMillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Inf. Process. Lett.*, 7:193–195, 1978.

[dOSlV16] Rafael Mendes de Oliveira, Amir Shpilka, and Ben lee Volk. Subexponential size hitting sets for bounded depth multilinear formulas. *Computational Complexity*, 25(2):455–505, 2016.

[DS07] Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM J. Comput.*, 36(5):1404–1434, 2007.

[GKKS13] Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth three. In *FOCS*, pages 578–587, 2013.

[HS80] Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing*, 1980, pages 262–272, 1980.

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004.

[KS07] Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007.

[KS09] Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *50th Annual IEEE Symposium on Foundations of Computer Science*, pages 501–510, 2009.
sium on Foundations of Computer Science, FOCS 2009, pages 198–207, 2009.

[KS11] Zohar Shay Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333–364, 2011.

[Lov79] László Lovász. On determinants, matchings, and random algorithms. In *FCT*, pages 565–574, 1979.

[MVV87] Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. *Combinatorica*, 7(1):105–113, 1987.

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005.

[Rys63] H.J. Ryser. *Combinatorial Mathematics*. Carus mathematical monographs. Mathematical Association of America, 1963.

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980.

[Sha90] Adi Shamir. IP=PSPACE. In *31st Annual Symposium on Foundations of Computer Science, St. Louis, Missouri, USA, October 22-24, 1990, Volume I*, pages 11–15, 1990.

[SS12] Nitin Saxena and C. Seshadhri. Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn’t matter. *SIAM J. Comput.*, 41(5):1285–1298, 2012.

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010.

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Symbolic and Algebraic Computation, EUROSAM ’79, An International Symposium on Symbolic and Algebraic Computation*, pages 216–226, 1979.