SIAM J. COMPUT.
Vol. 42, No. 6, pp. 2114–2131

DETERMINISTIC IDENTITY TESTING OF DEPTH-4 MULTILINEAR CIRCUITS WITH BOUNDED TOP FAN-IN*

ZOHAR S. KARNIN†, PARTHA MUKHOPADHYAY‡, AMIR SHPILKA§, AND ILYA VOLKOVICH§

Abstract. We give the first subexponential time deterministic polynomial identity testing algorithm for depth-4 multilinear circuits with a small top fan-in. More accurately, our algorithm works for depth-4 multilinear circuits with a plus gate at the top (also known as $\Sigma\Pi\Sigma\Pi$ circuits) and has a running time of $\exp(\text{poly}(\log(n), \log(s), k))$ where n is the number of variables, s is the size of the circuit, and k is the fan-in of the top gate. In particular, when the circuit is of polynomial (or quasi-polynomial) size, our algorithm runs in quasi-polynomial time. Prior to this work, subexponential time deterministic algorithms were known for depth-3 circuits with small top fan-in and for very restricted versions of depth-4 circuits. The main ingredient in our proof is a new structural theorem for multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuits. Roughly, this theorem shows that any nonzero multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit contains an "embedded" nonzero multilinear $\Sigma\Pi\Sigma(k)$ circuit. Using ideas from previous works on identity testing of sums of read-once formulas and of depth-3 multilinear circuits, we are able to exploit this structure and obtain an identity testing algorithm for multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuits.

Key words. identity testing, arithmetic circuits, derandomization, bounded depth circuits, multilinear circuits

AMS subject classifications. 68Q25, 12Y05

DOI. 10.1137/110824516

1. Introduction.

Polynomial identity testing (PIT) is one of the central problems in algebraic complexity theory: Given an arithmetic circuit C over a field $\mathbb{F}$ with input variables $x_1, x_2, \ldots, x_n$, can we check efficiently whether C computes the identically zero polynomial in the polynomial ring $\mathbb{F}[x_1, x_2, \ldots, x_n]$? The same question can be asked in the *black-box model* too. There, C is accessed by a black-box where we are allowed to substitute field elements $a_i \in \mathbb{F}$ for x_i and the black-box returns the value of $C(a_1, a_2, \ldots, a_n)$.

A randomized polynomial-time algorithm (more precisely a coRP algorithm) for this problem is known due to the Schwartz–Zippel lemma [Sch80, Zip79]. Over the years PIT has played a significant role in our understanding of important complexity theoretic and algorithmic problems. Well-known examples are the randomized NC algorithms for the matching problem in graphs [Lov79, KUW86, MVV87], and the primality test of Agrawal, Kayal, and Saxena [AKS04], and of Agrawal and Biswass [AB03]. The PIT problem has also played an indirect role in important complexity

*Received by the editors February 14, 2011; accepted for publication (in revised form) June 24, 2013; published electronically November 21, 2013. A preliminary version of this paper appeared in STOC 2010.

http://www.siam.org/journals/sicomp/42-6/82451.html

†Faculty of Computer Science, Technion, Haifa 32000, Israel (zkarnin@cs.technion.ac.il). This author's research was supported by the Israel Science Foundation (grant 339/10).

‡Chennai Mathematical Institute, Siruseri, Padur 603103, India (partham@cmi.ac.in). This author's research was supported in part at Technion by an Aly Kaufman Fellowship and by the Israel Science Foundation (grant 439/06).

§Department of Computer Science, Technion, Haifa 32000, Israel (shpilka@cs.technion.ac.il, ilyav@cs.technion.ac.il). The third and fourth authors' research was partially funded by the European Community's Seventh Framework Programme (FP7/2007-2013) under grant agreement 257575.

results such as IP = PSPACE [LFKN92, Sha92] and the original proof of the PCP theorem [ALM+98].

The main open problem is coming up with a *deterministic* polynomial-time (or at least subexponential-time) algorithm for PIT. In [KI04], Kabanets and Impagliazzo showed that giving a deterministic polynomial-time (even subexponential-time) identity testing algorithm implies that either NEXP $\not\subset$ P/poly or that the integer Permanent has no polynomial size arithmetic circuit. (see [AvM11] for a simpler proof.) Considering the black-box derandomization of PIT, it was shown by Heintz and Schnorr [HS80] that black-box derandomization of PIT implies that an explicit low-degree polynomial (that can be computed in PSPACE) has no subexponential size arithmetic circuit. Later, Agrawal [Agr05] showed that similar implication holds for an explicit multilinear polynomial. These results indicate the difficulty of derandomizing PIT. On the positive side, Kabanets and Impagliazzo also showed that strong lower bounds for general arithmetic circuits imply derandomization of PIT in quasi-polynomial time [KI04]. Currently, such strong lower bounds are known only for restricted models and not for general circuits; however, this result gives hope that derandomizing PIT in restricted models for which lower bounds exist may be possible. In [DSY09], Dvir, Shpilka, and Yehudayoff obtained an almost analogous result to [KI04] for small depth arithmetic circuits. However, even for this case lower boudns are not known. Nevertheless, for multilinear circuits of small depth, Raz and Yehudayoff [RY09] proved exponential lower bounds. Their bounds, in principle, could imply derandomization of PIT, according to [KI04, DSY09]. However, currently we do not have any result that transfers lower bounds for multilinear circuits to PIT algorithms. The question of whether lower bounds for multilinear circuits can be used to derandomize PIT for the same (or a related) multilinear model appears as an open question in [Raz09, SY10].

A depth-4 circuit (also known as a ΣΠΣΠ circuit) is an algebraic circuit of depth-4, where without loss of generality (w.l.o.g.) the gate types are uniform within a gate and alternate.[1] An argument for studying depth-4 circuits was given by Agrawal and Vinay [AV08]. They showed that *black-box* derandomization of PIT of depth-4 ΣΠΣΠ circuits is almost as hard as for *general* arithmetic circuits. Their result is based on a depth reduction technique [VSBR83, AJMV98] that converts any arithmetic circuit C to a depth-4 circuit C' such that C and C' compute the same polynomial. This connection makes the problem of black-box derandomization of PIT for depth-4 circuits an intriguing open problem. In fact, their result also implies that from a polynomial time PIT algorithm for multilinear depth-4 circuits one can get an exponential lower bound for unbounded depth multilinear circuits for an explicit multilinear polynomial. Today the best lower bound for general multilinear circuits is slightly superlinear [RSY08].

To summarize, the motivation for studying depth-4 circuits is two fold. First, by the results of [AV08] solving PIT in this model is as interesting as the general questions. Understanding PIT for multilinear depth-4 circuits is an interesting intermediate goal. Moreover, by the results of [AV08], this could also imply an exponential lower bound for general multilinear circuits. Second, since lower bounds for small-depth multilinear circuits are known [RY09], this seems like a "sane" place to start studying PIT.

So far, most of the derandomization results are known for depth-3 $\Sigma\Pi\Sigma(k,d)$ circuits when the top Σ gate is of bounded fan-in k (d is the fan-in of the Π gates which

[1] For the purpose of PIT, the hard case is where the top gate is a plus gate; hence we assume this is the case throughout this paper.

can be unbounded) [DS06, KS07, KS09, SS10, KS11, SS11a]. These algorithms exploit one common theme: The subspace spanned by the linear forms of an *identically zero* $\Sigma\Pi\Sigma$ circuit (viewing each linear form as a vector in $\mathbb{F}^n$) is of low dimension. More precisely, over a finite field, the current best-known bound for the dimension is $\mathcal{O}(k^3 \log d)$ [SS11a] and over the field $\mathbb{Q}$ of rational numbers, the bound is $\mathcal{O}(k^2 \log k)$, which is a constant for a constant k [SS10]. For multilinear $\Sigma\Pi\Sigma(k,d)$ circuits, over any characteristic, the dimension is bounded by $\mathcal{O}(k^3 \log(k))$ [DS06, SS11a]. Yet, the algorithm with the best running time [SV09] was obtained using a different approach. The authors of [SS11b] gave a new black-box PIT algorithm for $\Sigma\Pi\Sigma(k,d)$ circuits using a slightly different structural theorem, essentially saying that there exists some structured ideal I, generated by linear functions, such that the circuit modulo I is equivalent to a nonzero product of linear functions. The running time of their algorithm is $\text{poly}(nd^k)$.

For depth-4 circuits, the situation is very different. It seems unlikely that the method of black-box derandomization for $\Sigma\Pi\Sigma$ circuits can be adopted/extended for depth-4 circuits. The main difficulty is that there seems to be no notion of a linear space, spanned by the circuit components, that can be used.[2]

In this paper, we study the black-box PIT problem for multilinear depth-4 $\Sigma\Pi\Sigma\Pi(k)$ circuits. We give new techniques and come up with an efficient black-box algorithm that runs in time *quasi-polynomial* in the input size.

Roughly, a multilinear depth-4 $\Sigma\Pi\Sigma\Pi(k)$ circuit computes a polynomial that can be represented as a sum of k products of sparse multilinear polynomials, where the polynomials in each product gate are variable disjoint (which is equivalent to requiring that their product is a multilinear polynomial). Thus, $P(x_1, x_2, \ldots, x_n) = \sum_{i=1}^{k} \prod_{j=1}^{d_i} P_{ij}$, where each P_{ij} is a sparse multilinear polynomial (the sparsity of a polynomials is the number of monomials in it). A trivial but useful observation is that when the circuit is of size s each P_{ij} is s-sparse.

Our main result is a deterministic *black-box* PIT algorithm for this model with a running time $\exp(\text{poly}(\log(n), \log(s), k))$ (s is the size of the circuit) which is *quasi-polynomial* in the input size. More formally, we prove the following theorem.

THEOREM 1.1. *Let k, n, s be integers. Let $\mathbb{F}$ be a field of size larger than n^{12}. There is an explicit set $\mathcal{H} \subset \mathbb{F}^n$ of size $n^{\mathcal{O}(k^6 \log(k) \log^2 s)}$ that can be constructed in time*[3] $\text{poly}(\log |\mathbb{F}|) \cdot n^{\mathcal{O}(k^6 \log(k) \log^2 s)}$ *such that the following holds. Let $P \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a nonzero polynomial computed by a multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s on n variables. Then there is some $\bar{\alpha} \in \mathcal{H}$ such that $P(\bar{\alpha}) \neq 0$.*

Remark 1. Note that the theorem requires a lower bound on the size of the field. It is clear that in order to have an efficient black-box PIT algorithm, the field cannot be too small. For example, consider the polynomial $\prod_{i=1}^{n}(x_i - a_i)$ over $\mathbb{F}_2$, where $a_i \in \mathbb{F}_2$. This is clearly a nonzero polynomial, but any algorithm, even a randomized one, will require 2^n time to find a nonzero assignment, as the a_i can be arbitrary. See also [CDGK91] for a more thorough discussion. Thus, if the underlying field is too small, then we allow our algorithm to access an extension field of appropriate size.

We also note that we did not make any attempt to reduce the size of the field required for the construction as this is not the main point of this paper.

To the best of our knowledge, prior to our work, efficient deterministic algorithms were known only in the non-black-box setting for very restricted classes of depth-4

[2] Recent papers suggest that perhaps the right notion to study is that of *transcendence degree* rather than the notion of dimension from linear algebra [BMS11, ASSS12].

[3] As a convention, we use $|\mathbb{F}|$ only when $\mathbb{F}$ is finite.

circuits [AM10, Sax08, SV09]. For higher depths, efficient black-box algorithms are known only for read-once formulas [SV08, SV09].

Subsequent work. Following our result, Saraf and Volkovich [SV11] gave a black-box PIT algorithm for the same model (i.e., multilinear ΣΠΣΠ(k) circuits) with running time $n^{\mathcal{O}(k)} \cdot s^{\mathcal{O}(k^3)}$, which is polynomial in the circuit size when k is constant. (For constant k our running time is quasi-polynomial in the circuit size.) Their result employed some of the techniques presented in the current paper. The crux of their work is a new structural theorem stating that in a multilinear ΣΠΣΠ(k) circuit computing the identically zero polynomial, every multiplication gate computes a sparse polynomial. Another progress has been made in a recent work of Anderson, van Melkebeek, and Volkovich [AvMV11]. In their work PIT algorithms for multilinear read-k formulas[4] were given with the running times of $n^{k^{\mathcal{O}(k)}+\mathcal{O}(k \log n)}$ and $n^{k^{\mathcal{O}(k)}}$ in the black-box and the non-black-box settings, respectively. For constant k, this implies quasi-polynomial and polynomial time PIT algorithms, respectively. Both results were obtained via a generalization of the techniques from the current paper and from [SV09]. In fact, they have extended their results to a broader model—multilinear read-k formulas in which each leaf (variable) can be replaced by a sparse polynomial. This model is referred as "sparse-substituted" formulas. We note that the latter model extends the model considered in current paper. Finally, in [ASSS12] Agrawal et al. generalized some previous techniques using the notion of algebraic dependence and obtained a unified approach for obtaining PIT algorithms for ΣΠΣ(k) circuits and multilinear ΣΠΣΠ(k) circuits.

1.1. Overview of our algorithm and proof technique. We begin with a brief definition of *generators* and *hitting sets* for arithmetic circuits; two important notions for our algorithm. Generators were first defined in [SV09], but they are implicit in prior works as well. Intuitively, a generator $\mathcal{G}$ for a class of polynomials $\mathcal{M}$, is a polynomial map from q variables to n variable, i.e., $\mathcal{G} \in \mathbb{F}[y_1, \ldots, y_q]^n$, that can be plugged into any polynomial $P \in \mathcal{M}$ without vanishing it. A set $\mathcal{H} \subseteq \mathbb{F}^n$ is a hitting set for a class of polynomials $\mathcal{M}$, if for every nonzero polynomial $P \in \mathcal{M}$, there exists $\bar{a} \in \mathcal{H}$, such that $P(\bar{a}) \neq 0$. In identity testing, the role of generators and hitting sets are equivalent. Over large enough fields, the image of a generator for a class of circuits always contains a hitting set for the same class of circuits.[5] Conversely, given a hitting set for a class of arithmetic circuits, it is fairly easy to construct a generator for the class (by interpolation).

In our algorithm, we use recursion (by reducing the fan-in of the top Σ gate) to find a generator for ΣΠΣΠ(k) circuits. We now explain how the recursion works.

Recall that the sparsity of the polynomials P_{ij} is bounded by the circuit size s. Thus, for $k = 1$, we need to construct a generator for products of s-sparse polynomials. It is easy to see that a generator for a single s-sparse polynomial is also a generator for a product of s-sparse polynomials (since the product of nonzero polynomials is also a nonzero polynomial). Finally, generators for s-sparse polynomials are well known, e.g., [KS01].

For $k > 1$, we construct the generator via the following procedure: Let P be a nonzero n-variate polynomial computed by a ΣΠΣΠ(k) circuit C of size s and let $\mathcal{G}_{k-1}$ be a generator that works simultaneously for ΣΠΣΠ$(k-1)$ circuits of size s

[4] Read-k formulas are arithmetic formulas in which each variable can appear at most k times.

[5] It is enough that the size of the field is larger than $D \times \delta$, when D and δ are the total formal degrees of the circuit and the generator, respectively.

and for $2s^2$-sparse polynomials. We prove that there exists a set $U \subseteq [n]$ of size $\text{poly}(\log s)$ such that substituting the value of the generator $\mathcal{G}_{k-1}$ to the variables that are indexed by $[n] \setminus U$ leads to a nonzero polynomial. By going over all possible choices of sets for U and all inputs to $\mathcal{G}_{k-1}$, we can produce a small size hitting set for $\Sigma\Pi\Sigma\Pi(k)$ circuits, which in turn is transformed into a generator using the techniques of [SV09]. Notice that the number of choices for U is bounded by $n^{\text{poly}(\log s)}$, which is *quasi-polynomial* in s.

Next, we explain why such U exists, which concludes the outline of our proof. We divide the construction of U into two stages.

Stage I. Assume that there exists some large constant r such that each of the polynomials P_{ij} depend on at most n/r variables. We show that there exists a subset of the variables $V \subseteq [n]$ of size roughly r/k such that every P_{ij} has at most one variable x_ℓ with $\ell \in V$. Now, let $\mathcal{G}_{k-1}$ be a generator for $\Sigma\Pi\Sigma\Pi(k-1)$ circuits. By suitably fixing the variables whose indices are in $[n] \setminus V$ according to a point in the image of $\mathcal{G}_{k-1}$, we obtain a nonzero multilinear depth-3 $\Sigma\Pi\Sigma(k)$ circuit (where each bottom sum gate computes a linear[6] function in one variable). Here we need to use the properties of $\mathcal{G}_{k-1}$: since $\mathcal{G}_{k-1}$ is a generator for sparse polynomials, making the restriction does not introduce new factors to the circuit (in the language of Definition 1 the circuit remains simple). The fact that the generator is good against multilinear $\Sigma\Pi\Sigma\Pi(k-1)$ polynomials guarantees that no subcircuit is set to zero (in the language of Definition 1 the circuit remains minimal). Thus, after making the substitution according to $\mathcal{G}_{k-1}$ we have a simple and minimal $\Sigma\Pi\Sigma(k)$ circuit. Using a structural theorem of identically zero depth-3 circuits from [DS06, SS11a], it follows that the resulting depth-3 circuit computes a nonzero polynomial.

Stage II. For a set $S \subseteq [n]$, let $m(S)$ be the multilinear monomial $\prod_{i \in S} x_i$. For any subset $W \subseteq [n]$ we can express P, the polynomial computed by the circuit, as $P = \sum_{S \subseteq W} m(S) P_S$, where each polynomial P_S is over $x_{[n] \setminus W}$. We prove that for any $\Sigma\Pi\Sigma\Pi(k)$ circuit there exists a set $W \subseteq [n]$ of size $\text{poly}(\log s)$ (recall that s is the size of the given circuit) such that for some subset $S \subseteq W$, P_S can be computed by a $\Sigma\Pi\Sigma\Pi(k)$ circuit C' and each polynomial P'_{ij} that is computed in the third level of C', depends only on a small fraction of the total number of variables. In particular, the circuit C' satisfies the requirement of the first stage discussed above. Since W is small, we can find it by trying all possibilities.

We now explain why such W exists. Let r be some parameter that is chosen to optimize the running time (think of r as $r = \text{poly}(k)$). Let C be the given $\Sigma\Pi\Sigma\Pi(k)$ circuit. Write C as $C = \sum_{i=1}^{k} N_i \cdot A_i$, where $N_i = \prod_j P_{ij}$ is such that each P_{ij} depends on at most n/r variables. Similarly, let A_i be the product of the rest of the polynomials under the ith Π gate. By definition, each P_{ij} in A_i depends on at least n/r variables. Hence, due to multilinearity, each A_i is a product of at most r many P_{ij}-s. Our next step is to remove the A_i-s according to the following process: Consider a variable appearing in some A_i. By either setting the variable to zero or taking a partial derivative with respect to it, we can get rid of at least half of the monomials in A_i. Moreover, we show that such a variable exists with the additional property that both choices (either setting to zero or taking a partial derivative) result in a nonzero polynomial. By iteratively inserting such variables to the set W, we remove all A_i components while remaining with a nonzero polynomial and end up with a set W of size at most $\mathcal{O}(kr \log s)$.

[6]The function is in fact an affine linear function. Since earlier versions of this paper used the terminology of linear functions, we shall do the same here.

The final set U that our algorithm considers is defined as $U \triangleq W \cup V$ (the union of the sets found at the first and second stages).

1.2. Organization. We start by giving the required definitions in sections 2 and 3. We prove our main theorem (Theorem 4.11) in section 4, showing a construction of a generator for $\Sigma\Pi\Sigma\Pi(k)$ circuits. In section 4.5 we give as an easy corollary a hitting set for $\Sigma\Pi\Sigma\Pi(k)$ circuits.

2. Preliminaries. For a positive integer n denote $[n] = \{1, \ldots, n\}$. Let $\mathbb{F}$ be a field and let $\bar{\mathbb{F}}$ be its algebraic closure. For a polynomial $P(x_1, \ldots, x_n)$, a variable x_i, and $\alpha \in \mathbb{F}$, $P|_{x_i=\alpha}$ is the polynomial resulting after substituting α to the variable x_i. Let $\mathrm{mon}(A)$ denote the number of monomials in a polynomial A. The following definitions are for polynomials $P, Q \in \mathbb{F}[x_1, x_2, \ldots, x_n]$. We say that P *depends* on x_i if there exist $\bar{a} \in \bar{\mathbb{F}}^n$ and $b \in \bar{\mathbb{F}}$ such that

$$P(a_1, a_2, \ldots, a_{i-1}, a_i, a_{i+1}, \ldots, a_n) \neq P(a_1, a_2, \ldots, a_{i-1}, b, a_{i+1}, \ldots, a_n).$$

We denote $\mathrm{var}(P) \triangleq \{x_i \mid P \text{ depends on } x_i\}$. Intuitively, P depends on x_i if x_i appears when P is listed as a sum of monomials. Given a subset $I \subseteq [n]$ and an assignment $\bar{a} \in \mathbb{F}^n$ we define $P|_{x_I=\bar{a}_I}$ to be the polynomial resulting from substituting a_i to the variable x_i for every $i \in I$. Let P, Q be two nonconstant polynomials. We say that P and Q are *similar* and denote $P \sim Q$ if there exist $\alpha, \beta \in \mathbb{F} \setminus \{0\}$ such that $\alpha \cdot P = \beta \cdot Q$. Let $D_i(P, Q)$ be the polynomial defined as follows:

$$D_i(P,Q)(\bar{x}) \triangleq \left| \begin{pmatrix} P & P|_{x_i=0} \\ Q & Q|_{x_i=0} \end{pmatrix} \right| (\bar{x}) = (P \cdot Q|_{x_i=0} - P|_{x_i=0} \cdot Q)(\bar{x}).$$

The following is an easy observation.

Observation 2.1. Let $P, Q \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be two irreducible multilinear polynomials such that $x_i \in \mathrm{var}(P) \cap \mathrm{var}(Q)$. Then $P \sim Q$ iff $D_i(P, Q) \equiv 0$.

2.1. Mappings and generators for arithmetic circuits. In this section, we formally define the notion of generators (following [SV09]) and hitting sets for polynomials and describe a few useful properties.

A mapping $\mathcal{G} = (\mathcal{G}^1, \ldots, \mathcal{G}^n) : \mathbb{F}^q \to \mathbb{F}^n$ is a *generator* for the circuit class $\mathcal{M}$ if for every nonzero n-variate polynomial $P \in \mathcal{M}$ it holds that $P(\mathcal{G}) \not\equiv 0$. The image of the map $\mathcal{G}$ is denoted as $\mathrm{Im}(\mathcal{G}) = \mathcal{G}(\bar{\mathbb{F}}^q)$. Ideally, q should be very small compared to n. A set $\mathcal{H} \subseteq \mathbb{F}^n$ is a hitting set for a circuit class $\mathcal{M}$ if for every nonzero polynomial $P \in \mathcal{M}$ there exists $\bar{a} \in \mathcal{H}$, such that $P(\bar{a}) \neq 0$. A generator can also be viewed as a mapping whose image is a hitting set for $\mathcal{M}$. That is, for every nonzero $P \in \mathcal{M}$ there exists $\bar{a} \in \mathrm{Im}(\mathcal{G})$ such that $P(\bar{a}) \neq 0$. In [SV09] an efficient method of constructing a generator from a hitting set, for a (relatively) *small* q, is given.

LEMMA 2.2 (see Lemma 4.8 in [SV09]). *Let $|\mathbb{F}| > n$. Given a hitting set $\mathcal{H} \subseteq \mathbb{F}^n$ for a circuit class $\mathcal{M}$ there is an algorithm that in time* $\mathrm{poly}(|\mathcal{H}|, n, \log|\mathbb{F}|)$ *constructs a mapping $L(\bar{y}) : \mathbb{F}^q \to \mathbb{F}^n$, which is a generator for $\mathcal{M}$ with $q \triangleq \lceil \log_n |\mathcal{H}| \rceil$ and the individual degrees of L^i are bounded by $n-1$.*

The following is an immediate yet important property of generators.

Observation 2.3. Let $P = P_1 \cdot P_2, \cdot \cdots \cdot P_k$ be a product of nonzero polynomials $P_i \in \mathcal{M}$ and let $\mathcal{G}$ be a generator for $\mathcal{M}$. Then $P(\mathcal{G}) \not\equiv 0$.

At times we would like to use a partial substitution from a generator to a polynomial. Given a subset $I \subseteq [n]$ we define the mapping $\mathcal{G}^I$ as $(\mathcal{G}^I)_i = \mathcal{G}^i$ when $i \in I$ and $(\mathcal{G}^I)_i = x_i$ when $i \notin I$. In addition, we define $P|_{x_I=\mathcal{G}^I}$ to be the polynomial resulting

from substituting the function $\mathcal{G}^i$ to the variable x_i for each $i \in I$. The following is an immediate observation.

Observation 2.4. Let $\mathcal{M}$ be a class of polynomials and let $\mathcal{G}$ be a generator for n-variate polynomials in $\mathcal{M}$. Let $I \subseteq [n]$ and $P \in \mathcal{M}$ be a nonzero polynomial. Then $P|_{x_I = \mathcal{G}^I} \not\equiv 0$. Moreover, there exists $\bar{a} \in \mathrm{Im}\left(\mathcal{G}^I\right)$ such that $P(\bar{a}) \neq 0$.

2.2. Partial derivatives. Partial derivatives will play an important role in the analysis of our algorithms, when working over finite fields one considers the *discrete partial derivative*. However, we can use the usual notion of a partial derivative since it is the same as the discrete partial derivative for multilinear polynomials, and we will be taking derivatives only for multilinear polynomials.

Thus, we use the notation $\frac{\partial P}{\partial x_i}$ to denote the partial derivative of P according to x_i and for a nonempty subset $I \subseteq [n]$, $I = \left\{i_1, \ldots, i_{|I|}\right\}$, we define the iterated partial derivative with respect to I as

$$\partial_I P \triangleq \frac{\partial^{|I|} P}{\partial x_{i_1} \partial x_{i_2} \partial x_{i_3} \cdots \partial x_{i_{|I|}}}.$$

We state some well-known facts about partial derivatives that we will use in our analysis. Let $P \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a multilinear polynomial. Then, P depends on x_i iff $\frac{\partial P}{\partial x_i} \not\equiv 0$. For every i and j, $\frac{\partial^2 P}{\partial x_i \partial x_j} = \frac{\partial}{\partial x_i}(\frac{\partial P}{\partial x_j}) = \frac{\partial^2 P}{\partial x_j \partial x_i}$. For two different variables x_i, x_j, derivative and substitution commutes: $\frac{\partial P}{\partial x_i}|_{x_j = a} = \frac{\partial}{\partial x_i}(P|_{x_j = a})$.

2.3. Known results. In this section, we recall some known results about sparse polynomials and depth-3 $\Sigma\Pi\Sigma$ circuits which play an important role in the design of our algorithm. An *m-sparse polynomial* is a polynomial with at most m nonzero monomials. Equivalently, it is a polynomial computed by a depth-2 circuit with top fan-in m. Using a result of [KS01], we can construct an efficient generator for sparse polynomials.

LEMMA 2.5 (see Theorem 10 of [KS01]). *Let $\mathbb{F}$ be a field. In time polynomial in m, n, d, and $\log|\mathbb{F}|$, one can output a hitting set $\mathcal{H}$ of cardinality $|\mathcal{H}| = \mathrm{poly}(n, m, d)$ for n-variate m-sparse polynomials of degree d over a field $\mathbb{F}$. If $\mathbb{F} = \mathbb{R}$, then each element of each vector in the set has bit-length at most $\mathcal{O}(\log(nd))$. If $\mathbb{F}$ is a finite field with less than $(nd)^6$ elements, then the elements of the vectors lie in the smallest extension of $\mathbb{F}$ with at least $(nd)^6$ elements; otherwise, the vectors contain just elements of $\mathbb{F}$.*

Using Lemma 2.2, we can construct a generator from the hitting set output by the above result.

LEMMA 2.6. *Let n and m be integers and let $\mathbb{F}$ be a field of size larger than n^{12}. In time polynomial in n, m, and $\log|\mathbb{F}|$, we can construct a generator $\mathcal{S}_m \triangleq \left(\mathcal{S}_m^1, \mathcal{S}_m^2, \ldots, \mathcal{S}_m^n\right) : \mathbb{F}^q \to \mathbb{F}^n$ for m-sparse multilinear polynomials, where $q = q(n, m) = \mathcal{O}(\log_n m)$, and such that the individual degrees of each $\mathcal{S}_m^i$ are bounded by $n - 1$.*

Proof. Since the degree of a multilinear polynomial is bounded by n, we apply Lemma 2.2 on the hitting set $\mathcal{H}$ output by Lemma 2.5. Note that as $|\mathcal{H}| = \mathrm{poly}(n, m)$, we obtain that $q = q(n, m) = \mathcal{O}(\log_n m)$. ∎

The proof technique of our main result involves a reduction from identity testing of a class of depth-4 circuits to depth-3 circuits. Here, we define depth-3 circuits formally and recall some of their relevant properties. A depth-3 $\Sigma\Pi\Sigma(k, d)$ circuit C

computes a polynomial of the form

$$C(\bar{x}) = \sum_{i=1}^{k} F_i(\bar{x}) = \sum_{i=1}^{k} \prod_{j=1}^{d_i} L_{ij}(\bar{x}),$$

where the $L_{ij}(\bar{x})$-s are linear functions $L_{ij}(\bar{x}) = \sum_\ell a_{ij\ell} x_\ell + a_{ij0}$ with $a_{ij\ell} \in \mathbb{F}$, and $d_i \leq d$. We refer to the F_m-s as the multiplication gates of the circuit. A *subcircuit* of C is defined as a sum of a subset of the multiplication gates in C. We say that a circuit is *minimal* if no proper subcircuit of C computes the zero polynomial. Let $\gcd(C) \triangleq \gcd(F_1, F_2, \ldots, F_k)$. We say that a circuit is *simple* if $\gcd(C) = 1$. Define the *rank* of C, $\mathrm{rank}(C)$, as the rank of its linear functions when viewed as $(n+1)$-dimensional vectors over $\mathbb{F}^{n+1}$. That is, $\mathrm{rank}(C) \triangleq \dim(\mathrm{span}\{L_{ij}\})$.

A *multilinear* $\Sigma\Pi\Sigma(k,d)$ circuit has the additional property that each F_i is a multilinear polynomial. We shall use an important structural theorem regarding the rank of an identically zero $\Sigma\Pi\Sigma(k,d)$ multilinear circuit. The theorem holds regardless of the underlying field.

THEOREM 2.7 (see [DS06, SS11a]). *There exists an increasing integer function $R(k)$ upper bounded by $\mathcal{O}(k^3 \log(k))$ with the following property: Let C be an n-variate multilinear, simple, and minimal $\Sigma\Pi\Sigma(k,d)$ circuit computing the zero polynomial. Then* $\mathrm{rank}(C) < R(k)$.

We conclude this section with a well-known lemma concerning polynomials, giving a trivial (yet possibly large) hitting set. A proof can be found in [Alo99].

LEMMA 2.8. *Let $P \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial. Suppose that for every $i \in [n]$ the individual degree of x_i is bounded by d_i, and let $S_i \subseteq \mathbb{F}$ be such that $|S_i| > d_i$. We denote $S = S_1 \times S_2 \times \cdots \times S_n$. Then, $P \equiv 0$ iff $P|_S \equiv 0$.*

Notice that Lemma 2.8 is useful in order to obtain a hitting set given a generator. We demonstrate this in section 4.5.

3. Depth-4 multilinear circuits. In this section, we recall the model of depth-4 multilinear circuits and present a simple structural property of such circuits which is useful for our main result.

DEFINITION 3.1. *A multilinear depth-4 $\Sigma\Pi\Sigma\Pi(k)$ circuit C has four layers of alternating Σ and Π gates (the top Σ gate is at level one) and it computes a polynomial of the form*

$$C(\bar{x}) = \sum_{i=1}^{k} F_i(\bar{x}) = \sum_{i=1}^{k} \prod_{j=1}^{d_i} P_{ij}(\bar{x}), \tag{1}$$

where the $P_{ij}(\bar{x})$-s are multilinear polynomials computed by the last two layers of $\Sigma\Pi$ gates of the circuit and are the inputs to the Π gates at the second level. Each multiplication gate F_i computes a multilinear polynomial.

Note that the requirement that the F_i-s compute multilinear polynomials implies that for each i the polynomials $\{P_{ij}\}_{j \in [d_i]}$ are variable-disjoint. It is clear that if the circuit size is s, then the number of monomials in P_{ij} (i.e., its sparsity) is bounded by s. In this paper, we often refer to the polynomials P_{ij} as s-sparse where the sparsity should be understood in terms of the circuit size s. Similarly to the case of depth-3 circuits, a (proper) subcircuit of C is a sum of a (proper) subset of the multiplication gates of C. As before, a $\Sigma\Pi\Sigma\Pi(k)$ circuit is simple when no P_{ij} appears in all the

multiplication gates at the second level. Namely,[7] $\gcd(C) \stackrel{\Delta}{=} \gcd(F_1, \ldots, F_k) = 1$. When C is not simple we define its simplification to be

$$\mathrm{sim}(C) \stackrel{\Delta}{=} C/\gcd(C).$$

Note that $\mathrm{sim}(C)$ is a simple multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit.

Our identity testing algorithm builds on a reduction from identity testing of multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuits to identity testing of a special type of such circuits that we call *r-compressed circuits*.

DEFINITION 3.2. *A multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit, as in* (1), depending *on n variables, is called r-compressed if for every i, j, it holds that $|\mathrm{var}(P_{ij})| \leq n/r$.*

Notice that the definition requires that the fraction of variables in each P_{ij} is at most $1/r$ of the total number of variables in the circuit.

We next prove an easy structural property of r-compressed circuits that will be useful when we design our algorithm.

LEMMA 3.3. *Let n, r, k be integers such that $n \geq r \geq k$. Let $P \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be computed by a multilinear r-compressed $\Sigma\Pi\Sigma\Pi(k)$ circuit $C = \sum_{i=1}^{k} \prod_{j=1}^{d_i} P_{ij}(\bar{x})$. Assume that $|\mathrm{var}(P)| \geq r \geq k$. Then there exists a set $V \subseteq \mathrm{var}(P)$ of size $|V| \geq r/k$ such that for each i, j: $|V \cap \mathrm{var}(P_{ij})| \leq 1$.*

Proof. We select the elements of V one by one in a greedy iterative process. The first element of V can be arbitrarily set to any variable appearing in P. Assume w.l.o.g. that $i_1 = 1$. Let $T_1 \subseteq [n]$ be the set of variables that appear in some P_{ij} along with x_1. As $|\mathrm{var}(P_{ij})| \leq |\mathrm{var}(P)|/r$, we get that $|T_1| \leq k \cdot (\frac{|\mathrm{var}(P)|}{r} - 1)$. Hence, the set $W = \mathrm{var}(P) \setminus (T_1 \cup \{x_1\})$ is nonempty. We now pick an index $i_2 \in W$ arbitrarily (say, the minimal index in W) and construct a set T_2 analogous to T_1 according to i_2. We now continue this process with the set $W' = W \setminus (T_2 \cup \{x_{i_2}\})$. Due to the size restriction of T_1 (and the other T_i's), we can continue this process at least r/k times. The set V is the set of these (at least) r/k chosen indices. ∎

4. Black-box PIT. In this section we give an efficient black-box PIT algorithm for multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuits. We do so by constructing a generator for such circuits. We start by describing the construction of a polynomial map which we eventually use as our generator.

4.1. The construction and some easy properties. In this section we give a construction of a map from $\mathbb{F}^{2t}$ to $\mathbb{F}^n$, due to [SV09], whose image contains all vectors $\bar{a} \in \mathbb{F}^n$ with at most t nonzero entries.

We assume from now on that $|\mathbb{F}| > n$ as we are allowed to use elements from an appropriate extension field. Throughout the entire section we fix a set $A = \{\alpha_1, \alpha_2, \ldots, \alpha_n\} \subseteq \mathbb{F}$ of n distinct elements.

The following construction is taken from [SV09].

DEFINITION 4.1. *For every $i \in [n]$ let $u_i(w) : \mathbb{F} \to \mathbb{F}$ be the ith Lagrange Interpolation polynomial for the set A. That is, each $u_i(w)$ is the polynomial of degree $n-1$ satisfying $u_i(\alpha_j) = 1$ if $i = j$ and zero otherwise. For every $i \in [n]$ and*

[7] Notice that it may be the case that no P_{ij} appears in any of the multiplication gates at the second level but $\gcd(C) \neq 1$ when the P_{ij}'s are reducible. Since the circuit is given to us in a black-box model, we can assume w.l.o.g. that the P_{ij}'s are irreducible. This does not violate the sparsity requirement since the circuit is multilinear.

$t \geq 1$ we define $G_t^i(y_1, \ldots, y_t, z_1, \ldots, z_t) : \mathbb{F}^{2t} \to \mathbb{F}$ as

$$G_t^i(y_1, \ldots, y_t, z_1, \ldots, z_t) \triangleq \sum_{j=1}^{t} u_i(y_j) \cdot z_j.$$

Finally, let $G_t(y_1, \ldots, y_t, z_1, \ldots, z_t) : \mathbb{F}^{2t} \to \mathbb{F}^n$ be defined as

$$G_t(y_1, \ldots, y_t, z_1, \ldots, z_t) \triangleq \left(G_t^1, G_t^2, \ldots, G_t^n\right)$$
$$= \left(\sum_{j=1}^{t} u_1(y_j) \cdot z_j, \sum_{j=1}^{t} u_2(y_j) \cdot z_j, \ldots, \sum_{j=1}^{t} u_n(y_j) \cdot z_j\right).$$

We will use the following immediate observations.

Observation 4.2. For every $t \geq 1$, it holds that $G_t(\bar{y}, \bar{0}) \equiv \bar{0}$.

Thus, the zero vector is in the image of G_t.

Observation 4.3. Denote with $\bar{e}_i \in \{0,1\}^n$ the vector that has 1 in the ith coordinate and 0 elsewhere. Then

$$G_{t+1} = G_t + \sum_{i=1}^{n} u_i(y_{t+1}) \cdot z_{t+1} \cdot \bar{e}_i.$$

Hence, for every $t \geq 1$ and $\alpha_m \in A$ we have that

$$G_{t+1}|_{y_{t+1} = \alpha_m} = G_t + z_{t+1} \cdot \bar{e}_m.$$

We now state a simple but crucial property of the generator G that follows from Observation 4.3. (Recall the notation before Observation 2.4.)

Observation 4.4. Let $\ell, t \in \mathbb{N}$, $I \subseteq [n]$, and $|I| \leq \ell$. Then, it holds that

$$\mathrm{Im}\left(G_t^{[n]\setminus I}\right) \subseteq \mathrm{Im}\left(G_{\ell+t}\right).$$

4.2. Stage I: r-compressed $\Sigma\Pi\Sigma\Pi(k)$ circuits. In this section we consider a restricted class of $\Sigma\Pi\Sigma\Pi(k)$ circuits: For a fixed r, we assume that the polynomial is computed by a simple, r-compressed, multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s. Using a generator that works for sparse polynomials as well as for $\Sigma\Pi\Sigma\Pi(k-1)$ circuits of size s, we construct a generator for r-compressed $\Sigma\Pi\Sigma\Pi(k)$ circuits of size s. The key idea is to use the set $V \subseteq [n]$ that we obtain from Lemma 3.3. Recall that V has the following property: The size of V is at least r/k and for every P_{ij} in C, $|V \cap \mathrm{var}(P_{ij})| \leq 1$. Let $\mathcal{G}_{k-1}$ be a generator for $\Sigma\Pi\Sigma\Pi(k-1)$ circuits and for sparse polynomials of suitable sparsity. In the following lemma we show that if $r = R(k) \cdot k$ (recall the definition of $R(k)$ from Theorem 2.7), then when we restrict the variables in $[n] \setminus V$ to $\mathcal{G}_{k-1}$, we obtain a nonzero polynomial.

LEMMA 4.5. *Let n and k be integers such that $k \geq 2$ and $n \geq k \cdot R(k)$. Let $0 \not\equiv P \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial with $|\mathrm{var}(P)| \geq k \cdot R(k)$. Assume that P is computed by a simple, multilinear, $(k \cdot R(k))$-compressed $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s. Let $\mathcal{G}_{k-1}$ be a generator for $\Sigma\Pi\Sigma\Pi(k-1)$ circuits of size s and $(2s^2)$-sparse polynomials.*[8] *Then there exists a subset $V \subseteq \mathrm{var}(P)$ of size $|V| \leq R(k)$ such that*

$$P|_{x_{[n]\setminus V} = \mathcal{G}_{k-1}^{[n]\setminus V}} = P \circ \mathcal{G}_{k-1}^{[n]\setminus V} \not\equiv 0.$$

[8] Namely, $\mathcal{G}_{k-1}$ is a generator for both models.

Proof. Let $C = \sum_{i=1}^{k} \prod_{j=1}^{d_i} P_{ij}(\bar{x})$ be a simple, multilinear, and $(k \cdot R(k))$-compressed $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s, computing P. Assume w.l.o.g. that the P_{ij}'s are irreducible polynomials. If C is not minimal, then P can be computed by a $\Sigma\Pi\Sigma\Pi(k-1)$ circuit of size s and we are done (set $V = \emptyset$). So assume that C is minimal. Let $V \subseteq \mathrm{var}(P)$ be a set promised by Lemma 3.3. We can assume w.l.o.g. that $|V| = k \cdot R(k)/k = R(k)$ by keeping $R(k)$ arbitrary indices from V. Define the set T as $T = [n] \setminus V$.

We now describe a way to find an assignment for x_T such that the resulting polynomial is nonzero. We do so via a reduction to depth-3 circuits. Let $C_1, \ldots, C_{2^k-2}$ be the proper subcircuits of C (excluding the empty circuit). Clearly they are all $\Sigma\Pi\Sigma\Pi(k-1)$ circuits of size s. For any $P_{i_1 j_1}$ and $P_{i_2 j_2}$ appearing in C, and a variable x_ℓ such that $\mathrm{var}(P_{i_1 j_1}) \cap \mathrm{var}(P_{i_2 j_2}) \cap V = \{x_\ell\}$, define the polynomial $Q_{i_1,j_1,i_2,j_2,\ell} = D_\ell(P_{i_1 j_1}, P_{i_2 j_2})$, where D_ℓ is the same polynomial from Observation 2.1. Let $\mathcal{Q}$ be the set of all nonzero such $Q_{i_1,j_1,i_2,j_2,\ell}$'s. The following lemma gives a sufficient condition that a given partial assignment for x_T results in a simple, minimal, and nonzero depth-3 circuit.

LEMMA 4.6. *Let*

$$\varphi = \prod_{i=1}^{2^k-1} C_i \cdot \prod_{Q \in \mathcal{Q}} Q.$$

Let $\bar{a} \in \bar{\mathbb{F}}^n$ be such that $\varphi|_{x_T = \bar{a}_T} \not\equiv 0$. Then $C|_{x_T=\bar{a}_T}$ is a simple, minimal, multilinear $\Sigma\Pi\Sigma(k)$ circuit.

Proof. The minimality of $C|_{x_T=\bar{a}_T}$ is clear since all of the subcircuits of C are factors of φ. If one of them is zero, then so is $\varphi|_{x_T=\bar{a}_T}$. Notice that due to the same reason, no P_{ij} is reduced to zero. In order to prove that $C|_{x_T=\bar{a}_T}$ is simple, notice the following simple facts. First, by the definition of V, for every i, j it holds that $|\mathrm{var}(P_{ij}|_{x_T=\bar{a}_T})| \leq 1$. Second, consider $i_1 \neq i_2$ and j_1, j_2 such that $P_{i_1 j_{i_1}} \nsim P_{i_2 j_2}$. If $\mathrm{var}(P_{i_1 j_1}|_{x_T=\bar{a}_T}) \neq \mathrm{var}(P_{i_2 j_2}|_{x_T=\bar{a}_T})$, then we still have $P_{i_1 j_1}|_{x_T=\bar{a}_T} \nsim P_{i_2 j_2}|_{x_T=\bar{a}_T}$. If, on the other hand, $\mathrm{var}(P_{i_1 j_1}|_{x_T=\bar{a}_T}) = \mathrm{var}(P_{i_2 j_2}|_{x_T=\bar{a}_T}) = \{x_\ell\}$, and $P_{i_1 j_1} \nsim P_{i_2 j_2}$, then by Observation 2.1 $D_\ell(P_{i_1 j_1}, P_{i_2 j_2})$ is a factor of φ and so

$$D_\ell(P_{i_1 j_1}, P_{i_2 j_2})|_{x_T=\bar{a}_T} = D_\ell(P_{i_1 j_1}|_{x_T=\bar{a}_T}, P_{i_2 j_2}|_{x_T=\bar{a}_T}) \not\equiv 0.$$

Thus, $P_{i_1 j_1}|_{x_T=\bar{a}_T} \nsim P_{i_2 j_2}|_{x_T=\bar{a}_T}$ (applying Observation 2.1 again). Since C is itself a simple circuit, the claim follows from these two facts. ∎

We now return to the proof of Lemma 4.5. The polynomial φ is a product of $(2s^2)$-sparse polynomials and $\Sigma\Pi\Sigma\Pi(k-1)$ multilinear circuits of size s. By Observations 2.3 and 2.4 we get that $\varphi|_{x_T=\mathcal{G}_{k-1}^T} \not\equiv 0$. It follows that there exists some $\bar{a} \in \mathrm{Im}\,(\mathcal{G}_{k-1})$ for which $C|_{x_T=\bar{a}_T}$ is a simple, minimal, and multilinear $\Sigma\Pi\Sigma(k)$ circuit. Notice now that $C|_{x_T=\bar{a}_T}$ contains $R(k)$ variables (the previous proof shows that all the variables in V "survived") and any linear function appearing in it contains only one variable. Hence, the rank of $C|_{x_T=\bar{a}_T}$ is $R(k)$. By the definition of $R(k)$ (Theorem 2.7) $C|_{x_T=\bar{a}_T}$ cannot be a zero circuit. We thus proved that $P|_{x_T=\mathcal{G}_{k-1}^T} \not\equiv 0$. ∎

We remark that although the proof above used the full fledged rank bound of Theorem 2.7, we needed only apply it for the special case where each linear function in the circuit contains one variable. It is quite possible that a better result could be proved in this case (see, e.g., [SV09]). However, we did not make an attempt to improve the rank bound for this case as it will not change our results in a significant way.

4.3. A reduction to r-compressed multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuits. In this section we prove a structural theorem for multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuits. This theorem enables us to reduce identity testing of multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuits to identity testing of r-compressed multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuits for any $r > 0$. Roughly, the theorem says that there exists a small set of variables W with the following property. Let $P = \sum_{T\subseteq W} m(T)F_T$, where the F_T are polynomials defined over the variables $[n]\setminus W$ and $m(T) = \prod_{i\in T} x_i$. Then, there exists a subset $T \subseteq W$ such that F_T can be computed by an r-compressed multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s.

THEOREM 4.7. *Let P be an n-variate polynomial computable by a multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s. Let $r > 0$ be a parameter. Then there exists a set W of size $|W| \leq 2\log n \cdot \log s \cdot kr$ for which the following holds. Let*

$$P = \sum_{T\subseteq W} m(T)F_T,$$

where the F_T's are multilinear polynomials independent of the variables in W. Then there exists at least one set $T \subseteq W$ for which $F_T = Q \cdot H$, where Q is a product of s-sparse polynomials and H is computable by a simple, r-compressed multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s.

An alternative view of the theorem states that there exist two sets I, J having the following properties: If we set the variables of J to zero and take a partial derivative with respect to the variable set I (i.e., compute $\partial_I P|_{x_J=\bar{0}_J}$), then we get an r-compressed multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s, multiplied by several s-sparse polynomials. The set I corresponds to the variables in the monomial m_T (i.e., $I = T$) and J to the variables of W outside the monomial (i.e., $J = W \setminus T$). This alternative view will be more convenient for proving the theorem.

LEMMA 4.8. *Let $n, s, r, k > 1$ be integers. Let $P \not\equiv 0$ be an n-variate polynomial computable by a multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit C of size s. Then there exist disjoint subsets $I, J \subseteq [n]$ such that $|I| + |J| \leq 2kr\log(s)$ and $\partial_I P|_{x_J=\bar{0}_J}$ is a nonzero polynomial computed by a multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit C_{IJ} with at least one of the following properties:*

- $|\mathrm{var}(\mathrm{sim}(C_{IJ}))| < |\mathrm{var}(\mathrm{sim}(C))|/2$ *(recall the definition of* $\mathrm{sim}(C)$ *from section 3).*
- $\mathrm{sim}(C_{IJ})$ *is an r-compressed $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s.*

Proof. If C itself satisfies one of the two properties, then we are done. (Clearly this can happen only if C satisfies the second property.) If this is not the case, then let $\mathrm{sim}(C) = \sum_{j=1}^{k} M_j$, where M_j divides the jth multiplication gate of C. We can express each M_j as $M_j = N_j \cdot A_j$, where N_j is a product of s-sparse polynomials each containing at most $|\mathrm{var}(\mathrm{sim}(C))|/2r$ variables and $A_j = M_j/N_j$. Clearly, A_j is a product of s-sparse polynomials, such that each of them contains at least $|\mathrm{var}(\mathrm{sim}(C))|/2r$ many variables. Since M_j is a multilinear polynomial this implies that A_j has at most $2r$ factors. It follows that A_j must be s^{2r}-sparse. Let[9]

$$\Phi(C) \triangleq \sum_{j=1}^{k} \log\left(\mathrm{mon}\left(A_j\right)\right)$$

be a potential function that will be used in the proof. We assume w.l.o.g. that $\Phi(C)$ is minimal with respect to all possible $\Sigma\Pi\Sigma\Pi(k)$ circuits of size s computing P. Notice that $\Phi(C) \leq 2kr\log(s)$.

[9]Recall that $\mathrm{mon}(A)$ is the number of monomials in a polynomial A.

Let $I_0 = J_0 = \emptyset$, $P_0 = P$, and $A_{j,0} = A_j$ for each $j \in [k]$. We now describe an algorithm that produces the required sets I and J. At each step of the algorithm we add a single element either to I or J. Denote by I_ℓ and J_ℓ the sets at the end of step ℓ. Correspondingly, define $P_\ell = \partial_{I_\ell} P|_{x_{J_\ell}=\bar{0}_{J_\ell}}$. Let C_ℓ be a multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size at most s computing P_ℓ. Notice that such a circuit exists as $\partial_{I_\ell} C|_{x_{J_\ell}=\bar{0}_{J_\ell}}$ is a circuit of the mentioned properties. Let

$$C_\ell = \gcd(C_\ell) \cdot \mathrm{sim}(C_\ell) = \gcd(C_\ell) \cdot \sum_{j=1}^{k} N_{j,\ell} A_{j,\ell},$$

where each $N_{j,\ell}$ is an s-sparse polynomial that depends on at most $|\mathrm{var}(\mathrm{sim}(C))|/2r$ many variables (notice that we use $|\mathrm{var}(\mathrm{sim}(C))|$ and not $|\mathrm{var}(\mathrm{sim}(C_\ell))|$). Let $\Phi_C(C_\ell) \triangleq \sum_{j=1}^{k} \log(\mathrm{mon}(A_{j,\ell}))$.[10] We choose C_ℓ as a circuit achieving the minimal Φ_C among all size s $\Sigma\Pi\Sigma\Pi(k)$ circuits computing P_ℓ. Also, among the circuits achieving minimal Φ_C measure, we choose C_ℓ such that $|\mathrm{var}(\mathrm{sim}(C_\ell))|$ is minimal.

We now describe the process of adding a single variable to I or J. The idea is to take a variable appearing in some $A_{j,\ell}$ and add it to a set that will result in a maximal reduction to the monomials of $A_{j,\ell}$. One of the choices must reduce the number of monomials by a factor of 2 and thus reduce the Φ_C function by at least 1. This is since adding the variable to I means keeping only monomials in which it appears and adding it to J means keeping only the monomials in which it does not appear. The problem is to ensure that the resulting circuit computes a nonzero polynomial. The following lemma guarantees the existence of a variable for which neither action would result in a zero polynomial.

LEMMA 4.9. *Let $\ell \geq 0$. Assume that $P_\ell \not\equiv 0$ and that C_ℓ does not satisfy the two conditions of Lemma 4.8. Then there exist some $i \in [n]$ and $j \in [k]$ such that $A_{j,\ell}$ and P_ℓ depend on x_i and x_i is not a factor of P_ℓ.*

Proof. Assume that the claim is false. We have one of the following cases.

Case 1. For some i and j, $A_{j,\ell}$ depends on x_i and P_ℓ does not. In this case we can replace $A_{j,\ell}$ with $A_{j,\ell}|_{x_i=0}$ and get a circuit C' computing the same polynomial P_ℓ with a $\Phi_C(C') < \Phi_C(C_\ell)$, in contradiction to the minimality of C_ℓ.

Case 2. All the $A_{j,\ell}$'s are constant. That is: $C_\ell = \gcd(C_\ell) \cdot \sum_{j=1}^{k} N_{j,\ell}$. In this case, either $\mathrm{var}(\mathrm{sim}(C_\ell)) < \mathrm{var}(\mathrm{sim}(C))/2$ or $\mathrm{sim}(C_\ell)$ is an r-compressed $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s. Thus, C_ℓ satisfies one of the two properties in Lemma 4.8, and the claim is not false.

Case 3. There exists a variable x_i in $A_{j,\ell}$ (for some j) that divides P_ℓ. Let $C' \triangleq C_\ell|_{x_i=1} \cdot x_i$. Then $P_\ell \equiv C'$, $\Phi_C(C') \leq \Phi_C(C_\ell)$, and $|\mathrm{var}(\mathrm{sim}(C'))| < |\mathrm{var}(\mathrm{sim}(C_\ell))|$. This is a contradiction to the minimality of C_ℓ with respect to (w.r.t.) Φ_C. ∎

We return to the proof of lemma 4.8. By Lemma 4.9, there exists some x_i that appears in some $A_{j,\ell}$ where both $\frac{\partial P_\ell}{\partial x_i}$ and $P_\ell|_{x_i=0}$ are nonzero polynomials. Clearly, one of these choices for $P_{\ell+1}$ results in a nonzero polynomial for which $\Phi_C(C_{\ell+1}) \leq \Phi_C(C_\ell) - 1$. Since Φ_C is always nonnegative, after at most $2kr\log(s)$ (the initial value for Φ) steps, we get the required circuit. ∎

By repeating Lemma 4.8 at most $\log n$ times, we get Theorem 4.7 (indeed, in each step if we do not have the conclusion of Theorem 4.7, then $|\mathrm{var}(\mathrm{sim}(C_\ell))|$ is reduced by a factor of 2).

[10] We use a different notation (Φ_C and not Φ) as the $A_{j,\ell}$'s depend on $|\mathrm{var}(\mathrm{sim}(C))|$.

4.4. Completing the proof. In the previous sections we found that, if $|\mathrm{var}(P)|$ is not too small, then there exists a small set of variables W and an additional, disjoint, small set of variables V with the following properties: When substituting the variables outside of W and V according to a generator for $\Sigma\Pi\Sigma\Pi(k-1)$ circuits and for $(2s^2)$-sparse polynomials, we get a nonzero polynomial. The following is the formal claim.

LEMMA 4.10. *Let $k \geq 2$ and $P \not\equiv 0 \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be computed by a multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s. Assume that $|\mathrm{var}(P)| \geq k \cdot R(k)$. Let $\mathcal{G}_{k-1}$ be a generator for $\Sigma\Pi\Sigma\Pi(k-1)$ circuits of size s and $(2s^2)$-sparse polynomials. Then there exists a subset $U \subseteq [n]$, depending only on P (i.e., U does not depend on the generator), of size $|U| \leq 3k^2 R(k)\log(s)\log(n)$ such that $P|_{x_{[n]\setminus U}=\mathcal{G}_{k-1}^{[n]\setminus U}} \not\equiv 0$ for any such generator $\mathcal{G}_{k-1}$.*

Proof. Let W and $T_0 \subseteq W$ be the sets guaranteed by Theorem 4.7. Namely, when writing $P = \sum_{T \subseteq W} m(T)F_T$, we have that $F_{T_0} = Q \cdot H$, where Q is a product of s-sparse polynomials and H can be computed by a simple, multilinear $(k \cdot R(k))$-compressed $\Sigma\Pi\Sigma\Pi(k)$ circuit. Note that Q and H are defined on $[n] \setminus W$ (that is $\mathrm{var}(Q) \cup \mathrm{var}(H) \subseteq [n] \setminus W$). By Lemma 4.5 there exists a subset $V \subseteq \mathrm{var}(H)$ of size $|V| \leq R(k)$ such that

$$H|_{x_{[n]\setminus V}=\mathcal{G}_{k-1}^{[n]\setminus V}} \not\equiv 0.$$

Let $U \triangleq V \cup W$. It holds that

$$H|_{x_{[n]\setminus U}=\mathcal{G}_{k-1}^{[n]\setminus U}} \not\equiv 0.$$

As Q is a product of s-sparse polynomials, we get, by Observations 2.3 and 2.4, that

$$Q|_{x_{[n]\setminus U}=\mathcal{G}_{k-1}^{[n]\setminus U}} \not\equiv 0.$$

It follows that under the restriction $x_{[n]\setminus U} = \mathcal{G}_{k-1}^{[n]\setminus U}$, F_{T_0} is a nonzero polynomial. The claim follows since we did not substitute anything to the variables in W. ∎

We now establish the generator for $\Sigma\Pi\Sigma\Pi(k)$ circuits. This is our main theorem and it guarantees that we get the required black-box algorithm. In particular, Theorem 1.1 is an immediate corollary.

THEOREM 4.11 (main theorem). *Let $P \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a nonzero polynomial computed by a multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s. For every $\ell \geq 3k^3 R(k)\log(s)\log(n)$ it holds that $P\left(G_\ell(\bar{y}, \bar{z}) + \mathcal{S}_{2s^2}(\bar{w})\right) \not\equiv 0$, where $\bar{y}, \bar{z}$, and $\bar{w}$ are new sets of variables and $R(k)$ is the rank bound of Theorem 2.7.*

Proof. We prove the claim by induction on k. For $k = 1$ we note that P is a product of $(2s^2)$-sparse polynomials. By definition of $\mathcal{S}_{2s^2}$ (recall Lemma 2.6), and Observations 2.3 and 4.2, we get that $P(G_\ell + \mathcal{S}_{2s^2}) \not\equiv 0$ and the claim follows.[11]

Assume that $k \geq 2$. We consider two cases. If $|\mathrm{var}(P)| < k \cdot R(k)$, then clearly $\ell > |\mathrm{var}(P)|$. Observation 4.4 implies that for any assignment ρ to the variables in $\mathrm{var}(P)$, $\mathrm{Im}\,(G_\ell + \mathcal{S}_{2s^2})$ contains a vector that agrees with that assignment on $\mathrm{var}(P)$. In particular, $P(G_\ell + \mathcal{S}_{2s^2}) \not\equiv 0$.

We now consider the case where $|\mathrm{var}(P)| \geq k \cdot R(k)$. Let $U \subseteq [n]$ be the subset guaranteed by Lemma 4.10. By the induction hypothesis, we get that for $v = \left\lceil 3(k-1)^3 R(k-1)\log(s)\log(n)\right\rceil$ the mapping $G_v(\bar{y}, \bar{z}) + \mathcal{S}_{2s^2}(\bar{w})$ is a generator

[11] Since $\vec{0} \in \mathrm{Im}\,(G_\ell)$, we have that $\mathrm{Im}\,(\mathcal{S}_{2s^2}) \subseteq \mathrm{Im}\,(G_\ell + \mathcal{S}_{2s^2})$.

for both $\Sigma\Pi\Sigma\Pi(k-1)$ circuits and for $(2s^2)$-sparse polynomials. From Lemma 4.10 and Observation 2.4 it follows that $\mathrm{Im}(G_v^{[n]\setminus U} + \mathcal{S}_{2s^2}^{[n]\setminus U})$ contains a point $\bar{a}$ for which $P(\bar{a}) \neq 0$. Since $|U| \leq 3k^2R(k)\log(s)\log(n) \leq \ell - v$, Observation 4.4 gives

$$\mathrm{Im}\left(G_v^{[n]\setminus U} + \mathcal{S}_{2s^2}^{[n]\setminus U}\right) \subseteq \mathrm{Im}\left(G_v^{[n]\setminus U} + \mathcal{S}_{2s^2}\right) \subseteq \mathrm{Im}\left(G_\ell + \mathcal{S}_{2s^2}\right),$$

and thus $\bar{a} \in \mathrm{Im}\left(G_\ell(\bar{y},\bar{z}) + \mathcal{S}_{2s^2}(\bar{w})\right)$, and the claim holds. ∎

4.5. An explicit hitting set. The hitting set for $\Sigma\Pi\Sigma\Pi(k)$ circuit is an immediate corollary of Theorem 4.11. Basically, as $G_\ell(\bar{y},\bar{z}) + \mathcal{S}_{2s^2}(\bar{w})$ are (relatively) low degree polynomials defined on $m = \mathcal{O}(k^3R(k)\log(s)\log(n))$ many variables, we can simply evaluate $P \circ (G_\ell(\bar{y},\bar{z}) + \mathcal{S}_{2s^2}(\bar{w}))$ on all inputs from E^m where $E \subseteq \mathbb{F}$ is a set of size $\mathrm{poly}(n)$. Algorithm 1 follows exactly this intuition and produces the hitting set. As before we assume that the field is large enough, i.e., $|\mathbb{F}| \geq n^2$, otherwise we construct the hitting set over an extension field of the relevant size.

Input: $n, k, s \in \mathbb{N}$.
Output: A set $\mathcal{H}$
Let $E \subseteq \mathbb{F}$ be of size $|E| = n^2$.
Let $\ell \triangleq \lceil 3k^3R(k)\log(s)\log(n)\rceil$, where $R(k)$ is defined in Theorem 2.7.
Let $q \triangleq q(n, 2s^2)$ as defined in Lemma 2.6.
Initialize $\mathcal{H} = \emptyset$.
foreach $\bar{a}, \bar{b} \in E^\ell$ *and* $\bar{c} \in E^q$ **do**
  Evaluate $G_\ell(\bar{a},\bar{b}) + \mathcal{S}_{2s^2}(\bar{c})$ and add it to $\mathcal{H}$.
end

Algorithm 1. Construction of a hitting set for $\Sigma\Pi\Sigma\Pi(k)$ circuits

THEOREM 4.12. *Let n, s, and k be positive integers. Let $\mathbb{F}$ be a field of size at least n^{12}. Algorithm 1, given n, s, k as input, runs in time* $\mathrm{poly}(\log|\mathbb{F}|) \cdot n^{\mathcal{O}(k^3R(k)\log^2 s)} = \mathrm{poly}(\log|\mathbb{F}|) \cdot n^{\mathcal{O}(k^6\log(k)\log^2 s)}$ *and outputs a set $\mathcal{H}$ of size* $n^{\mathcal{O}(k^3R(k)\log^2 s)} = n^{\mathcal{O}(k^6\log(k)\log^2 s)}$. *The set $\mathcal{H}$ is a hitting set for n-variate polynomials that can be computed by a $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s.*

Proof. Let $P \in \mathbb{F}[x_1, x_2, \ldots, x_n]$ be a polynomial computed by a multilinear $\Sigma\Pi\Sigma\Pi(k)$ circuit of size s. Let $\mathcal{H}$ be the set given by Algorithm 1. We claim that $P \equiv 0$ iff $P|_{\mathcal{H}} \equiv 0$. If $P \equiv 0$, then the claim is trivial. If $P \not\equiv 0$, by Theorem 4.11 we get that $P(G_\ell + \mathcal{S}_{2s^2}) \not\equiv 0$. According to their definition, the degrees of each entry of the mapping of G_ℓ and $\mathcal{S}_{2s^2}$ is at most $n-1$. Therefore, the individual degrees in $P(G_\ell + \mathcal{S}_{2s^2})$ are bounded by $(n-1)n < n^2$. Since $P(G_\ell + \mathcal{S}_{2s^2}) \not\equiv 0$, Lemma 2.8 implies that $P|_{\mathcal{H}} \not\equiv 0$.

We now bound the size of $\mathcal{H}$ and the time required to construct it. By definition, G_ℓ depends on 2ℓ variables and $\mathcal{S}_{2s^2}$ depends on $\mathcal{O}(\log_n s)$ variables. Hence,

$$|\mathcal{H}| \leq n^{4\ell + 2q(n,2s^2)} = n^{\mathcal{O}(k^3R(k)\log s \log n \cdot \log_n s)} = n^{\mathcal{O}(k^3R(k)\log^2 s)}.$$

By Lemma 2.6, the time required to construct $\mathcal{S}_{2s^2}$ is polynomial in n, s, and $\log|\mathbb{F}|$. Similarly, from Definition 4.1, it is clear that evaluating G_ℓ can be done in time polynomial in n, ℓ, and $\log|\mathbb{F}|$. Hence, the time to construct $\mathcal{H}$ is

$$|\mathcal{H}| \cdot \mathrm{poly}(\log|\mathbb{F}|) \cdot (ns)^{\mathcal{O}(1)} = \mathrm{poly}(\log|\mathbb{F}|) \cdot n^{\mathcal{O}(k^3R(k)\log^2 s)}. \quad ∎$$

5. Conclusion. Derandomizing the polynomial identity testing problem for depth-4 arithmetic circuits is an outstanding open problem in complexity theory [AV08]. Any efficient derandomized algorithm for depth-4 circuits will imply strong lower bounds [HS80, KI04, Agr05, AvM11]. So far, the progress in depth-4 identity testing is very limited [AM10, Sax08, SV09]. In this paper, we improve the situation by giving a quasi-polynomial time *black-box* identity testing algorithm for depth-4 multilinear circuits with bounded fan in top gate. Our algorithm is based on new structural theorems about such circuits.

In identity testing and explicit lower bound proofs, multilinear circuits have already received significant attention from the community [DS06, SV08, SV09, Raz06, RSY08, Raz09, RY09, KS11]. In [Raz09], Raz asked whether one could design efficient identity testing algorithms for multilinear formulas. The best-known algorithms prior to this paper were for sums of read-once formulas [SV09] and for set-multilinear depth-3 formulas [RS05]. The latter algorithm is non-black-box.[12] By generalizing several ideas and techniques presented in the current paper, as well as in [SV09], an efficient identity testing algorithm for constant-read multilinear formulas was obtained in [AvMV11].

For depth-4 multilinear circuits with bounded fan in top gate, our result provided the first efficient identity testing algorithm. This result was improved in [SV11]. It will be very interesting to generalize our result for non-multilinear circuits with bounded fan in top gate.

Acknowledgments. We thank Salil Vadhan and the anonymous referees for helpful comments that improved the presentation of our results.

REFERENCES

[AB03] M. Agrawal and S. Biswas, *Primality and identity testing via Chinese remaindering*, J. ACM, 50 (2003), pp. 429–443.

[Agr05] M. Agrawal, *Proving lower bounds via pseudo-random generators*, in Proceedings of the 25th Foundations of Software Technology and Theoretical Computer Science, Lecture Notes in Comput. Sci. 3821, Springer, Berlin, 2005, pp. 92–105.

[AJMV98] E. Allender, J. Jiao, M. Mahajan, and V. Vinay, *Non-commutative arithmetic circuits: Depth reduction and size lower bounds*, Theoret. Comput. Sci., 209 (1998), pp. 47–86.

[AKS04] M. Agrawal, N. Kayal, and N. Saxena, *Primes is in P*, Ann. Math. (2), 160 (2004), pp. 781–793.

[ALM+98] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, *Proof verification and the hardness of approximation problems*, J. ACM, 45 (1998), pp. 501–555.

[Alo99] N. Alon, *Combinatorial Nullstellensatz*, Combin. Probab. Comput., 8 (1999), pp. 7–29.

[AM10] V. Arvind and P. Mukhopadhyay, *The ideal membership problem and polynomial identity testing*, Inform. Comput., 208 (2010), pp. 351–363.

[ASS13] M. Agrawal, C. Saha, and N. Saxena, *Quasi-polynomial hitting-set for set-depth-formulas*, in Symposium on Theory of Computing, ACM, New York, 2013, pp. 321–330.

[ASSS12] M. Agrawal, C. Saha, R. Saptharishi, and N. Saxena, *Jacobian hits circuits: Hitting-sets, lower bounds for depth-d occur-k formulas and depth-3 transcendence degree-k circuits*, in Proceedings of the 44th Annual Symposium on Theory of Computing, ACM, New York, 2012, pp. 599–614.

[AV08] M. Agrawal and V. Vinay, *Arithmetic circuits: A chasm at depth four*, in Proceedings of the 49th Annual Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2008, pp. 67–75.

[12] It was only recently that an efficient black-box algorithm for set-multilinear depth-3 formulas was given [ASS13].

[AvM11] S. Aaronson and D. van Melkebeek, *On circuit lower bounds from derandomization*, Theory Comput., 7 (2011), pp. 177–184.

[AvMV11] M. Anderson, D. van Melkebeek, and I. Volkovich, *Derandomizing polynomial identity testing for multilinear constant-read formulae*, in Proceedings of the 26th Annual Conference on Computational Complexity, IEEE Computer Society, Los Alamitos, CA, 2011, pp. 273–282.

[BMS11] M. Beecken, J. Mittmann, and N. Saxena, *Algebraic independence and blackbox identity testing*, in Automata, Languages, and Programming, Lecture Notes in Comput. Sci., Springer, Heidelberg, 2011, pp. 137–148.

[CDGK91] M. Clausen, A. W. M. Dress, J. Grabmeier, and M. Karpinski, *On zero-testing and interpolation of k-sparse multivariate polynomials over finite fields*, Theoret. Comput. Sci., 84 (1991), pp. 151–164.

[DS06] Z. Dvir and A. Shpilka, *Locally decodable codes with* 2 *queries and polynomial identity testing for depth* 3 *circuits*, SIAM J. Comput., 36 (2006), pp. 1404–1434.

[DSY09] Z. Dvir, A. Shpilka, and A. Yehudayoff, *Hardness-randomness tradeoffs for bounded depth arithmetic circuits*, SIAM J. Comput., 39 (2009), pp. 1279–1293.

[HS80] J. Heintz and C. P. Schnorr, *Testing polynomials which are easy to compute (extended abstract)*, in Proceedings of the 12th Annual ACM Symposium on Theory of Computing, ACM, New York, 1980, pp. 262–272.

[KI04] V. Kabanets and R. Impagliazzo, *Derandomizing polynomial identity tests means proving circuit lower bounds*, Comput. Complexity, 13 (2004), pp. 1–46.

[KS01] A. Klivans and D. Spielman, *Randomness efficient identity testing of multivariate polynomials*, in Proceedings of the 33rd Annual ACM Symposium on Theory of Computing, ACM, New York, 2001, pp. 216–223.

[KS07] N. Kayal and N. Saxena, *Polynomial identity testing for depth* 3 *circuits*, Comput. Complexity, 16 (2007), pp. 115–138.

[KS09] N. Kayal and S. Saraf, *Blackbox polynomial identity testing for depth* 3 *circuits*, in Proceedings of the 50th Annual IEEE Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2009, pp. 198–207.

[KS11] Z. S. Karnin and A. Shpilka, *Black box polynomial identity testing of depth-*3 *arithmetic circuits with bounded top fan-in*, Combinatorica, 31 (2011), pp. 333–364.

[KUW86] R. M. Karp, E. Upfal, and A. Wigderson, *Constructing a perfect matching is in random nc*, Combinatorica, 6 (1986), pp. 35–48.

[LFKN92] C. Lund, L. Fortnow, H. Karloff, and N. Nisan, *Algebraic methods for interactive proof systems*, J. ACM, 39 (1992), pp. 859–868.

[Lov79] L. Lovasz, *On Determinants, Matchings, and Random Algorithms*, in Fundamentals of Computing Theory, L. Budach, ed., Akademia-Verlag, Berlin, 1979.

[MVV87] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani, *Matching is as easy as matrix inversion*, Combinatorica, 7 (1987), pp. 105–113.

[Raz06] R. Raz, *Separation of multilinear circuit and formula size*, Theory Comput., 2 (2006), pp. 121–135.

[Raz09] R. Raz, *Multi-linear formulas for permanent and determinant are of super-polynomial size*, J. ACM, 56 (2009), 8.

[RS05] R. Raz and A. Shpilka, *Deterministic polynomial identity testing in non-commutative models*, Comput. Complexity, 14 (2005), pp. 1–19.

[RSY08] R. Raz, A. Shpilka, and A. Yehudayoff, *A lower bound for the size of syntactically multilinear arithmetic circuits*, SIAM J. Comput., 38 (2008), pp. 1624–1647.

[RY09] R. Raz and A. Yehudayoff, *Lower bounds and separations for constant depth multilinear circuits*, Comput. Complexity, 18 (2009), pp. 171–207.

[Sax08] N. Saxena, *Diagonal circuit identity testing and lower bounds*, in Automata, Languages, and Programming, Springer, Berlin, 2008, pp. 60–71.

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

[Sha92] A. Shamir, *IP = PSPACE*, J. ACM, 39 (1992), pp. 869–877.

[SS10] N. Saxena and C. Seshadhri, *From Sylvester-Gallai configurations to rank bounds: Improved black-box identity test for depth-*3 *circuits*, in Proceedings of the 51st Annual Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2010, pp. 21–29.

[SS11a] N. Saxena and C. Seshadhri, *An almost optimal rank bound for depth-*3 *identities*, SIAM J. Comput., 40 (2011), pp. 200–224.

[SS11b] N. Saxena and C. Seshadhri, *Blackbox identity testing for bounded top fanin depth-*3 *circuits: The field doesn't matter*, in Proceedings of the 43th Annual ACM Symposium on Theory of Computing, ACM, New York, 2011, pp. 431–439.

[SV08] A. Shpilka and I. Volkovich, *Read-once polynomial identity testing*, in Proceedings of the 40th Annual Symposium on Theory of Computing, ACM, New York, 2008, pp. 507–516.

[SV09] A. Shpilka and I. Volkovich, *Improved polynomial identity testing for read-once formulas*, in APPROX-RANDOM, 2009, pp. 700–713; full version available online at http://www.cs.technion.ac.il/∼shpilka/publications/PROF-journal.pdf.

[SV11] S. Saraf and I. Volkovich, *Blackbox identity testing for depth-4 multilinear circuits*, in Proceedings of the 43rd Annual Symposium on Foundations of Computer Science, IEEE Computer Society, Los Alamitos, CA, 2011, pp. 421–430.

[SY10] A. Shpilka and A. Yehudayoff, *Arithmetic circuits: A survey of recent results and open questions*, Found. Trends Theor. Comput. Sci., 5 (2009), pp. 207–388.

[VSBR83] L. G. Valiant, S. Skyum, S. Berkowitz, and C. Rackoff, *Fast parallel computation of polynomials using few processors*, SIAM J. Comput., 12 (1983), pp. 641–644.

[Zip79] R. Zippel, *Probabilistic algorithms for sparse polynomials*, in Symbolic and Algebraic Computation, Lecture Notes in Comput. Sci. 72, Springer, Berlin, 1979, pp. 216–226.