2015 IEEE 56th Annual Symposium on Foundations of Computer Science

Deterministic Divisibility Testing via Shifted Partial Derivatives

(Extended Abstract)

Michael A. Forbes
Princeton University
Princeton, USA
miforbes@csail.mit.edu

Abstract

Kayal [1] has recently introduced the method of *shifted partial derivatives* as a way to give the first exponential lower bound for computing an explicit polynomial as a *sum of powers of constant-degree polynomials*. This method has garnered further attention because of the work of Gupta, Kamath, Kayal and Saptharishi [2] who used this method to obtain lower bounds that approach the "chasm at depth-4" ([3]–[5]).

In this work, we investigate to what extent this method can be used to obtain deterministic *polynomial identity testing (PIT)* algorithms, which are algorithms that decide whether a given algebraic circuit computes the zero polynomial. In particular, we give a $\mathsf{poly}(s)^{\mathcal{O}(\lg s)}$-time deterministic black-box PIT algorithm for a size-s sum of powers of constant-degree polynomials. This is the first sub-exponential deterministic PIT algorithm for this model of computation and the first[1] PIT algorithm based on the method of shifted partial derivatives.

We also study the problem of *divisibility testing*, which when given polynomials $f(\overline{x})$ and $g(\overline{x})$ (as algebraic circuits) asks to decide whether $f(\overline{x})$ divides $g(\overline{x})$. Using Strassen's [6] technique for the elimination of divisions, we show that one can obtain deterministic divisibility testing algorithms via deterministic PIT algorithms, and this reduction does not dramatically increase the complexity of the underlying algebraic circuits.

Using this reduction, we show that deciding divisibility of a constant-degree polynomial f into a sparse polynomial g reduces to PIT of *sums of a monomial multiplied by a power of constant-degree polynomials*. We then extend the method of shifted partial derivatives to give a $\mathsf{poly}(s)^{\mathcal{O}(\lg s)}$-time deterministic black-box PIT algorithm for this model of computation, and thus derive a corresponding deterministic divisibility algorithm. This is the first non-trivial deterministic algorithm for this problem.

Previous work on multivariate divisibility testing due to Saha, Saptharishi and Saxena [7] gave algorithms for when f is linear and g is sparse, and essentially worked via PIT algorithms for *read-once (oblivious) algebraic branching programs (roABPs)*. We give explicit sums of powers of quadratic polynomials that require exponentially-large roABPs in a strong sense, showing that techniques known for roABPs have limited applicability in our regime.

Finally, by combining our results with the algorithm of Forbes, Shpilka and Saptharishi [8] we obtain $\mathsf{poly}(s)^{\mathcal{O}(\lg\lg s)}$-time deterministic black-box PIT algorithms for various models (translations of sparse polynomials, and sums of monomials multiplied by a power of a linear polynomial) where only $\mathsf{poly}(s)^{\Theta(\lg s)}$-time such algorithms were previously known.

Keywords

divisibility testing, polynomial identity testing, shifted partial derivatives, read-once algebraic branching programs, sparse polynomials

This work was performed when the author was a graduate student at MIT CSAIL (supported by Scott Aaronson's Waterman award, NSF CCF-1249349), when the author was a Google Research Fellow at the Simons Institute for the Theory of Computing, and when the author was a member of the School of Mathematics at the Institute for Advanced Study. This material is based upon work supported by the National Science Foundation under agreement No. CCF-1412958. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

[1]Kumar and Saraf (personal communication) have independently obtained PIT algorithms via applying the hardness versus randomness paradigm to lower bounds obtained via shifted partial derivatives. Their results seem related but incomparable to the results obtained here.

0272-5428/15 $31.00 © 2015 IEEE
DOI 10.1109/FOCS.2015.35

I. Introduction

We consider two problems from computational algebra: *polynomial identity testing (PIT)* and *divisibility testing*. In these problems, one receives algebraic circuits as inputs and the problem is to decide whether the multivariate polynomials that these circuits compute satisfy various properties. That is, the input to these problems is a directed acyclic graph whose internal nodes are labeled with algebraic operations (addition and multiplication) and whose leaves are labeled by variables x_i or scalars from a field $\mathbb{F}$. Algebraic circuits naturally compute polynomials in the ring $\mathbb{F}[x_1, \ldots, x_n]$ and are one of the most natural and succinct methods to describe such polynomials. However, the succinctness of this description creates challenges in algorithmically understanding the computed polynomial. In particular, while there are often randomized algorithms for deciding properties of the polynomials computed by algebraic circuits, derandomizing such algorithms is an active area of research, in particular because of its connections to derandomizing other well-known problems such as computing perfects matchings in NC ([9]–[11]).

Polynomial Identity Testing:: The first such problem we consider, polynomial identity testing, asks whether a given algebraic circuit computes the zero polynomial. This problem has a simple randomized algorithm: evaluate the circuit at random point and declare the polynomial non-zero if the evaluation is non-zero. The correctness of this algorithm follows from the Schwartz-Zippel [12]–[14] lemma. This randomized algorithm also has the property of being *black-box* in that it only uses the algebraic circuit as a method to implement an evaluation oracle $\overline{\alpha} \mapsto f(\overline{\alpha})$ for the underlying polynomial $f(\overline{x})$ computed by this circuit. In contrast, a *white-box* algorithm is allowed to use the structure of the computation of this circuit. Thus, as the black-box model allows weaker access, deriving results in this model are correspondingly stronger.

Creating deterministic PIT algorithms is a significant challenge, as it is known to have implications for the existence of explicit polynomials that require large algebraic circuits for their computation ([15]–[17]), which is a long standing open question. As such, attention has focused on designing deterministic PIT algorithms for restricted models of algebraic computation. This focus on restricted models has in particular yielded a long line of work for PIT of bounded top-fan-in depth-3 and depth-4 circuits [7], [18]–[28]. Recently, this focus has been further justified by *depth-reduction* results ([3]–[5], [29], [30]) that in particular show that polynomial-time deterministic black-box PIT algorithms for depth-3 algebraic circuits (of exponential degree) imply a deterministic *quasi*polynomial-time black-box PIT algorithm for general algebraic circuits. These results show that depth-3 circuits essentially capture the full complexity of general algebraic circuits. For more on algebraic circuits, PIT, and depth reduction, see the recent surveys of Shpilka-Yehudayoff [31], Saxena [32], [33] or Saptharishi [34].

Another direction of study has considered *algebraic branching programs (ABPs)*. As this model can simulate formulas (and thus low depth circuits), this direction of study ([8], [35]–[42]) has focused on restricted classes of ABPs such as those that are *non-commutative*, *set-multilinear*, or *read-once (and oblivious)*. These three restrictions are essentially the same, and thus we will focus on read-once (oblivious) ABPs (roABPs). This model is partly interesting because it subsumes various other natural models (such as sums of powers of linear forms ([38], [43])), and because its complexity is exactly characterized by the rank of Nisan's [44] partial derivative matrix (which as a technique is also used for multilinear formula lower bounds ([45]–[47])). Further, developing deterministic polytime black-box PIT has applications to other questions such as derandomizing Noether Normalization ([40], [48]) and can be seen as an algebraic analogue to derandomizing RL (see the discussion in Forbes-Shpilka [38]). For more on read-once ABPs, see the thesis of Forbes [49].

While the inquiry into roABPs remains unfinished (in that we lack deterministic polytime black-box PIT for roABPs), we qualitatively understand this model well. That is, the techniques of Nisan [44] produces explicit polynomials requiring exponentially large roABPs, and we have a deterministic white-box PIT algorithm for roABPs due to Raz and Shpilka [35] [2], as well as another white-box PIT algorithm (in characteristic zero)

[2]These results were originally phrased in terms of non-commutative ABPs. For the explicit translation to roABPs, see the thesis of Forbes [49].

due to Arvind, Joglekar and Srinivasan [50]. To some extent, many of the above papers sought to replicate the Raz-Shpilka [35] algorithm in the black-box model with varying degrees of success [3].

Given the above state of affairs, it is natural to ask what are other restricted models of algebraic computation where the task of designing PIT algorithms is within reach. Historically, our understanding of a restricted model of algebraic computation is first demonstrated by finding a polynomial which is hard to to compute in this model, then with more work an efficient deterministic white-box PIT algorithm is developed, and finally additional work develops a corresponding black-box algorithm. While the hardness versus randomness paradigm (instantiated in the algebraic setting by Kabanets-Impagliazzo [16] and Dvir-Shpilka-Yehudayoff [51]) shows that for general computation one can go directly from lower bounds to black-box PIT, it is not readily applicable for many restricted models of computation. Further, results of Heintz-Schnorr [15], Agrawal [17] and Kabanets-Impagliazzo [16] show that non-trivial PIT algorithms *imply* lower bounds. As such, it seems that to obtain further progress in PIT algorithms we must examine the existing lower bound techniques and attempt to develop corresponding algorithms out of them (or develop new lower bounds).

In particular, a corresponding deterministic PIT algorithm matching the multilinear formula lower bounds of Raz and Raz-Yehudayoff [45]–[47] was only recently developed by Oliveira, Shpilka, and Volk [52]. While this represents progress in that their algorithm takes $\exp(n^{1-\Omega(1)})$-time, their algorithm requires $\exp(n^{\Omega(1)})$-time even for polynomial size depth-3 multilinear formulas. At present, it is unclear whether this is a limitation of PIT techniques or whether this is even inherent to the lower bound techniques of Raz and Raz-Yehudayoff [45]–[47].

In this work, we study the more recent lower bound technique of *shifted partial derivatives* from the seminal papers of Kayal [1] and Gupta-Kamath-Kayal-Saptharishi [2]. While these works have spawned many follow-up lower bound results ([53]–[60]), there are no PIT algorithms (white-box or black-box) based on these ideas. In this work, we take the first paper of Kayal [1] on this subject and translate his ideas into a black-box PIT algorithm by *scaling down* the lower bound and making it *robust*. That is, Kayal [1] gave an exponential lower bound for the top-fan-in s when the monomial $x_1 \cdots x_n$ is expressed as $\sum_{i=1}^{s} f_i(\overline{x})^{d_i}$ where $\deg f_i \leq \mathcal{O}(1)$ (a sum of powers of constant-degree polynomials, denoted $\sum \bigwedge \sum \prod^{\mathcal{O}(1)}$) and we give a quasipolynomial-time deterministic black-box PIT algorithm for this same model. While in retrospect our new algorithm is a rather simple extension of the methods of Forbes-Shpilka [40] (who did the same for sums of powers of *linear* polynomials (denoted $\sum \bigwedge \sum$), using the partial derivative method of Nisan-Wigderson [61]), there were conceptual reasons to believe this extension was not possible (as we discuss below).

To further demonstrate the novelty of the method, note that sums of powers of linear polynomials are a sub-model of roABPs and as such have efficient deterministic PIT algorithms. It is thus natural to ask how the complexity changes when going from powers of linear polynomials to powers of constant-degree polynomials. To address this question, we give an explicit power of a quadratic polynomial which requires exponentially large roABPs even under partial substitutions (and in any variable order). While weaker versions of this result are folklore, this result more strongly shows that the techniques of roABPs cannot even address sums of powers of quadratic polynomials.

Divisibility Testing:: We now turn to the second computational question we study, that of divisibility testing. In this problem, we are given *two* algebraic circuits computing polynomials $f(\overline{x})$ and $g(\overline{x})$ respectively, and the problem is to decide whether $f(\overline{x})$ divides $g(\overline{x})$ (denoted $f|g$). As in PIT, this question can also be asked in the black-box model where we are only allowed to use the circuits to gain access to the evaluation oracles of f and g. Note that unlike PIT, it is not immediately obvious whether there is *any* efficient randomized algorithm for this question, much less an algorithm in the black-box model. However, such an algorithm can be derived from the works of Kaltofen and Trager [62], [63]. That is, in the black-box model they gave a randomized algorithm that takes an evaluation oracle for a polynomial $h(\overline{x})$ and produces evaluation oracles for the irreducible factors of h. In the white-box model where h is given as an algebraic circuit they showed that they can even compute small

[3]An exception is the paper of Gurjar, Korwar, Saxena, and Thierauf [42], who design a new PIT algorithm for sums of roABPs (in varying variable orders). While this algorithm requires new ideas beyond those of Raz-Shpilka [35], it is to some extent in the same spirit.

algebraic circuits for the irreducible factors of h. Given these algorithms, one can decide whether f divides g by computing their respective irreducible factors and then checking that the multiplicities of each factor is at least as large in g as in f [4].

While the above provides a randomized algorithm (by solving the harder problem of factorization), it leaves open the question of obtaining a deterministic algorithm. However, as PIT efficiently reduces to divisibility testing[5] it is clear that to provide deterministic divisibility algorithms one first needs deterministic PIT algorithms. Conversely, the recent work of Kopparty, Saraf and Shpilka [64] showed that one can derandomize the factorization algorithm of Kaltofen and Trager [62], [63] using a deterministic PIT algorithm. As such, this shows that divisibility testing and PIT are *equivalent* in computational complexity. Indeed, many other works have shown similar results showing that derandomizing PIT suffices for other problems in computational algebra ([7], [65]–[67]).

While the above seemingly suggests that we understand the complexity of divisibility testing, the above equivalence with PIT only works for *general* algebraic circuits. That is, even if one asks for divisibility testing of restricted algebraic circuits then the above reduction yields PIT instances of seemingly difficult complexity (that is, involving determinants). While Shpilka and Volkovich [65] have shown a reduction from factoring multilinear polynomials to PIT that roughly preserves the complexity of the original computation, this reduction is highly tailored to multilinear polynomials. Indeed, even deterministically factoring sparse polynomials is an open question of von zur Gathen and Kaltofen [68] because it is not known whether the factors of sparse polynomials are sparse (in large characteristic).

To the best of our knowledge, it is even an open question to deterministically test whether f divides g when both f, g are sparse[6]. The only deterministic divisibility testing algorithm we are aware of[7] is a polytime algorithm for deciding whether the linear polynomial f divides the sparse polynomial g [8]. This follows from the work of Saha, Saptharishi and Saxena [7] who reduced this question (via long-division) to PIT of expressions of the form $\sum_{i=1}^{s} \overline{x}^{\overline{a}_i} f_i(\overline{x})^{d_i}$, where $\overline{x}^{\overline{a}}$ is the monomial $\prod_{i=1}^{n} x_i^{a_i}$ and $\deg f_i \leq 1$ (which we denote $\sum \mathrm{m} \bigwedge \sum \prod^{1}$). They then (in the reinterpretation of Forbes-Shpilka [38]) reduced this question to PIT of roABPs from which the above mentioned PIT algorithms apply.

In this work, we ask for divisibility testing of when f is constant-degree and g is sparse. Via long-division (or the reduction mentioned below), one can show that this reduces to PIT of expressions of the form $\sum_{i=1}^{s} \overline{x}^{\overline{a}_i} f_i(\overline{x})^{d_i}$ where $\deg f_i \leq \mathcal{O}(1)$ (which we denote $\sum \mathrm{m} \bigwedge \sum \prod^{\mathcal{O}(1)}$). Aside from the monomials $\overline{x}^{\overline{a}_i}$, this is exactly the $\sum \bigwedge \sum \prod^{\mathcal{O}(1)}$ mentioned above. However, the known hard polynomial of Kayal [1] for $\sum \bigwedge \sum \prod^{\mathcal{O}(1)}$ is the monomial $\overline{x}^{\overline{1}}$, whereas in this new $\sum \mathrm{m} \bigwedge \sum \prod^{\mathcal{O}(1)}$ model this monomial is trivial to compute. Thus, we use the *translation* idea of Agrawal, Saha, and Saxena [39] to instead consider the monomial $(\overline{x} + \overline{1})^{\overline{1}} := \prod_i (x_i + 1)$ to be the hard polynomial for $\sum \mathrm{m} \bigwedge \sum \prod^{\mathcal{O}(1)}$ formulas. Using a variant of the shifted partial derivative method we are able to give exponential lower bounds for $(\overline{x} + \overline{1})^{\overline{1}}$ as a $\sum \mathrm{m} \bigwedge \sum \prod^{\mathcal{O}(1)}$ formula, and following the recipe mentioned before (of scaling-down the lower bound and making it robust) we obtain quasipolynomial black-box PIT algorithms for this model. Plugging this into our reduction, we obtain a deterministic quasipolynomial-time black-box algorithm to decide whether the constant-degree f divides the sparse g.

The above reduction from divisibility testing to PIT using long division is highly dependent on the fact that f is low-degree, and does not give a general reduction from divisibility of sparse f into sparse g to a polynomial-size

[4]This can also be viewed as computing the greatest common divisor of f and g, and then checking that this is f.

[5]Note that $0|g(\overline{x})$ iff $g(\overline{x}) = 0$. Perhaps less trivially, $y|(g(\overline{x}) + y)$ iff $g(\overline{x}) = 0$ if $\overline{x}$ and y are variable disjoint.

[6]Dvir and Oliviera [69] showed that if factors of sparse polynomials are sparse then one can efficiently (within quasipolynomial time) do divisibility testing when f, g are sparse. They also claimed to prove this conjecture about the sparsity of factors of sparse polynomials, but withdrew their claims due to an error.

[7]There are also several works ([70]–[73] and references there-in) giving deterministic algorithms for determining low-degree factors of sparse multivariate polynomials of *exponential* degree (known as *lacunary* polynomials), thus implying similar divisibility testing algorithms. However, these deterministic algorithms have a runtime which has an exponential dependence on the number of variables. As such, their results seem incomparable to the ones of this paper.

[8]One can actually show that this algorithm works if f is a sum of univariates and g is computable by small roABP.

PIT problem. However, we remedy this situation by providing such a general reduction from divisibility testing to PIT that avoids the need for factorization and also roughly preserves the complexity of the original algebraic circuits. Instead of using long-division, we consider Strassen's [6] procedure for removing division gates from algebraic circuits. That is, if h, f, g are polynomials where $h = g/f$, Strassen [6] showed how to derive an algebraic circuit for h from circuits for f and g. We observe here that this procedure is still well-defined even if $f \nmid g$, in which case it produces some polynomial $\widetilde{g/f}$. Thus, one can check whether $f|g$ by checking that $g - f \cdot \widetilde{g/f} = 0$. By a careful inspection of Strassen's [6] procedure this shows that the complexity of this computation is not too far above that of f and g. Unfortunately, the PIT problems that arise when f and g are sparse are still beyond what is known how to perform deterministically.

A. Our Results and Techniques

We now more formally discuss our results and techniques.

PIT for $\sum\bigwedge\sum\prod^t$ formulas:: We begin by defining $\sum\bigwedge\sum\prod^t$ formulas (in general we are most interested in $t = \mathcal{O}(1)$), one of the main classes of algebraic computation of interest in this paper.

Definition I.1. *A polynomial $f(\overline{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ is **computable by a $\sum\bigwedge\sum\prod^t$ formula** if*

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

*where $\deg f_i \leq t$. The **size** of the formula is $\sum_i (d_i + 1) \cdot \binom{n+t}{t}$. When $t = 1$ we write this as $\sum\bigwedge\sum$.* $\Diamond$

We use the notation '$\bigwedge$' to denote *powering*, following Gupta, Kamath, Kayal, and Saptharishi [30]. This notation is meant to be suggestive of the exponentiation character '^'. Note that one could also consider constants α_i in front of the $f_i^{d_i}$ and modify our results essentially without change. However since our algorithms will mostly be black-box, we can think of these constants as being absorbed into the $f_i^{d_i}$ by taking d_i-th roots.

The notion of a $\sum\bigwedge\sum$ formula is well-studied but not completely understood. This notion has been previously studied under the name of a "*depth-3 diagonal formula*", as named by Saxena [43]. These formulas are also classically studied in mathematics, where the size of these formulas is known as *symmetric tensor rank* or *Waring rank* of a polynomial, see for example Landsberg [74]. These formulas are in some sense the weakest complete model of algebraic computation for which we do not have a complete understanding; in particular, we lack polynomial-time deterministic black-box PIT algorithms. We now briefly summarize what is known about these formulas, in particular how this class lies in the intersection of two techniques in algebraic complexity theory (roABPs and the partial derivative method of Nisan-Wigderson [61]).

We begin by reviewing the definition of roABPs. While there are equivalent definitions that are more flexible (see for example Forbes [49, Section 4.4]), the definition we use is a normal form for this type of computation.

Definition I.2. *Let $\mathbb{F}$ be a field, $n \geq 1$, and let $\pi : [n] \to [n]$ be a permutation. The polynomial $f(\overline{x}) \in \mathbb{F}[x_1, \ldots . x_n]$ is computable by a **read-once (oblivious) algebraic branching program (roABP) with variable order** π, with depth-n, individual degree-$(\leq d)$ and width-$(\leq w)$, if there exist matrices $M_i(x_{\pi(i)}) \in \mathbb{F}[x_{\pi(i)}]^{(\leq w)\times(\leq w)}$ of (individual) degree $\leq d$ such that*

$$f(\overline{x}) = \left(\prod_{i=1}^{n} M_i\left(x_{\pi(i)}\right) \right)_{1,1} ,$$

*where the sizes of the matrices is such that this product is well-defined. The **size** of such a roABP is defined to be nw^2d. The roABP computes in a **known order** if π is a fixed known permutation (such as the identity permutation), and computes in an **unknown order** if π exists but is unknown. We say that $f(\overline{x})$ is computed by a **commutative roABP** if it is computed by an roABP in* every *variable order π.* $\Diamond$

One can also force (by padding) the M_i to *all* be width w.

Saxena [43] (using his *duality trick*) implicitly (made explicit in Forbes-Shpilka [38]) showed that $\sum\bigwedge\sum$ formulas are computable by small *commutative* roABPs, and this is stated more generally in the following result.

Lemma I.3 (Saxena [43], Forbes-Shpilka [38])**.** *Let $\mathbb{F}$ be any field. Let $f(x_1, \ldots, x_n)$ be computed by a $\sum\bigwedge\sum\bigwedge$ formula (diagonal depth-4 formula), so that*

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

where each f_i is a sum of univariate polynomials, so that $f_i(\overline{x}) = \sum_{j\in[n]} f_{i,j}(x_j)$, and where $\deg f_i \leq d$. Then for any variable order $\pi : [n] \to [n]$, f is computed by a $\mathsf{poly}(\sum_{i\in[s]} a_i, n)$-explicit [9] width-$\left(\sum_{i\in[s]}(a_i+1)\right)$ roABP, of individual degree $\leq d \max_{i\in[s]} a_i$, in the variable order π. □

Nisan [44] essentially gave exponential lower bounds for computing the determinant as a roABP, so this lower bound similarly extends to $\sum\bigwedge\sum$. Further, as the above reduction is explicit and Raz-Shpilka [35] gave a polytime white-box PIT algorithm for roABPs, these results combine to give such a corresponding algorithm for $\sum\bigwedge\sum$ (reinterpreting the paper of Saxena [43]). Initially, black-box PIT algorithms for $\sum\bigwedge\sum$ formulas all essentially worked also for commutative roABPs ([37]–[39]) and had complexity $\mathsf{poly}(nwd)^{\log n}$ for width-w, n-variate, degree $\leq d$ commutative roABPs. Obtaining better algorithms for roABPs (or even commutative roABPs) seems challenging.

However, $\sum\bigwedge\sum$ formulas are also simple with respect to the *partial derivative* measure of Nisan and Wigderson [61]. That is, to a polynomial f one considers the space $\boldsymbol{\partial}_{\overline{x}^{<\infty}}(f) := \{\partial_{\overline{x}^{\overline{a}}} f\}_{\overline{a}}$, where $\partial_{\overline{x}^{\overline{a}}} f$ is the derivative $\partial_{x_1^{a_1}\cdots x_n^{a_n}}$ of f, and the exponent vector $\overline{a}$ ranges over all derivatives. In particular, one looks at the *dimension* of $\boldsymbol{\partial}_{\overline{x}^{<\infty}}(f)$ as a vector space. Kayal (see Saxena [43]) observed that polynomial-size $\sum\bigwedge\sum$ formulas have a partial derivative space of low-dimension, while the monomial $x_1\cdots x_n$ has $\dim \boldsymbol{\partial}_{\overline{x}^{<\infty}}(x_1\cdots x_n) = 2^n$ showing that $x_1\cdots x_n$ requires exponential size as a $\sum\bigwedge\sum$ formula (which is tight by Fischer [75]).

In contrast, roABPs very easily compute the monomial $x_1\cdots x_n$ and thus the partial derivative method seems to give more insight into $\sum\bigwedge\sum$ formulas. Based on this insight, and using the *hardness of representation* idea of Shpilka and Volkovich [76], Forbes-Shpilka [40] gave a $\mathsf{poly}(s)^{\log s}$-time black-box PIT algorithm for size-s $\sum\bigwedge\sum$ formulas. As our techniques are a generalization of theirs, we briefly describe their approach. We begin by *scaling down* the above lower bound. That is, observe that *if* a size-s $\sum\bigwedge\sum$ computes a monomial $\overline{x}^{\overline{a}}$, which involves $\|\overline{a}\|_0 := \{i \mid a_i \neq 0\})$ many variables, then it must be (by the above lower bound) that $\|\overline{a}\|_0 \leq \mathcal{O}(\lg s)$. The next observation is that the above lower bound is *robust*. That is, if $f(\overline{x}) = \overline{x}^{\overline{a}} + o(\overline{x}^{\overline{a}})$ in that we mean "$o(\overline{x}^{\overline{a}})$" to consist of lower order terms (say, with respect to the lexicographic ordering), then it follows that the measure of f is at least that of its leading monomial $\overline{x}^{\overline{a}}$, that is, $\dim \boldsymbol{\partial}_{\overline{x}^{<\infty}}(f) \geq \dim \boldsymbol{\partial}_{\overline{x}^{<\infty}}(\overline{x}^{\overline{a}})$. The measure $\dim \boldsymbol{\partial}_{\overline{x}^{<\infty}}$ is thus *robust* in that it ignores lower order terms. From these facts, Forbes-Shpilka [40] deduced that any size-s $\sum\bigwedge\sum$ formula must compute a monomial (in fact, its leading monomial) that involves $\mathcal{O}(\lg s)$ variables. That is, any such $\sum\bigwedge\sum$ formula computing a non-zero polynomial must compute a polynomial with a *small-support monomial*. A brute force algorithm on this small-support monomial then yields the desired PIT algorithm.

Thus, the above two methodologies offer different ways to obtain quasipolynomial-time PIT algorithms for $\sum\bigwedge\sum$ formulas. However, somewhat surprisingly, Forbes, Shpilka and Saptharishi [8] showed how to *combine* these two approaches to obtain $\mathsf{poly}(s)^{\mathcal{O}(\lg\lg s)}$-time black-box PIT for size-s $\sum\bigwedge\sum$ formulas. We will further discuss this approach below, as our variant of shifted partial derivatives along with this method allows us to derive $\mathsf{poly}(s)^{\mathcal{O}(\lg\lg s)}$-time black-box PIT for size-s *translations of sparse polynomials*, which improves on the previous best runtime of $\mathsf{poly}(s)^{\mathcal{O}(\lg s)}$ ([37]–[39]) which comes from viewing such polynomials as a subclass of commutative roABPs.

We now return to discuss $\sum\bigwedge\sum\prod^t$ formulas for $t > 1$. One may ask to what extent the above two approaches (roABPs and the partial derivative measure) generalize to this case. We address these techniques in order. It seems

[9] In this paper, *explicit* means computable in a Turing machine model where there are special registers holding elements of $\mathbb{F}$ and all $\mathbb{F}$-operations on these registers are unit cost.

to be a folklore result that the $\sum\bigwedge\sum\prod^2$ formula $(\sum_{i=1}^n x_i y_i)^n$ requires an exponentially large roABP in any variable order where $\overline{x}$ precedes $\overline{y}$. However, this result is only for these special variable orders, and in fact this formula *is* computable by a small roABP in the variable order $x_1 < y_1 < \cdots < x_n < y_n$. Thus, while black-box PIT algorithms for roABPS in a fixed order (such as Forbes-Shpilka [38]) would not work on this formula (without knowing the order), the black-box PIT algorithms for roABPs that work in an unknown order (such as Agrawal, Gurjar, Korwar, and Saxena [41]) would succeed, and thus this formula does not give a convincing example that roABP methods cannot succeed for PIT of $\sum\bigwedge\sum\prod^2$. Thus, we provide such an example.

Theorem. *Let $\mathbb{F}$ be a field of characteristic $\geq \mathsf{poly}(n)$. Then there is an explicit n-variate $\sum\bigwedge\sum\prod^2$ formula of degree n, such that under any partial substitution that leaves $\Omega(n)$ variables untouched, the resulting polynomial requires $\exp(\Omega(n))$-size roABPs in any variable order.* □

The explicit polynomial is of the form $(\overline{x}^t A \overline{x})^n$, where $A \in \mathbb{F}^{n\times n}$ is a totally non-singular matrix. Thus, this polynomial essentially embeds the hard polynomial $(\sum_{i=1}^n x_i y_i)^n$ under any partition of the variables, but more work is required to formalize this intuition.

We now turn to the partial derivative method. As it is folklore that $\sum\bigwedge\sum\prod^2$ formulas (such as $(\sum_{i=1}^n x_i^2)^n$) can have exponentially large partial derivative space, it follows that this method alone will not generalize from $\sum\bigwedge\sum$ to $\sum\bigwedge\sum\prod^2$ or $\sum\bigwedge\sum\prod^{\mathcal{O}(1)}$. However, Kayal's [1] *shifted partial derivatives* were discovered exactly for this purpose. This operator maps a polynomial f to the space $\overline{\boldsymbol{x}}^{\leq\ell}\boldsymbol{\partial}_{\overline{x}^{\leq k}}(f) := \{\overline{x}^{\overline{b}}\partial_{\overline{x}^{\overline{a}}}(f)\}_{\overline{b},\overline{a}}$ where the exponent vectors $\overline{a},\overline{b}$ are chosen so that $\deg\overline{x}^{\overline{b}} \leq \ell$ and $\deg\overline{x}^{\overline{a}} \leq k$. For $\ell = 0$ one recovers the partial derivative method of Nisan and Wigderson [61]. By carefully choosing parameters via the analysis of Gupta-Kamath-Kayal-Saptharishi [2], Kayal [1] obtained the following theorem (which as Kayal [1] notes, is tight).

Theorem (Kayal [1], Gupta-Kamath-Kayal-Saptharishi [2])**.** *For any field $\mathbb{F}$, computing $x_1\cdots x_n$ as a $\sum\bigwedge\sum\prod^t$ formula requires top-fan-in $\geq \exp(\Omega(n/t))$.* □

While the above seems very similar to the analogous lower bound for $\sum\bigwedge\sum$ formulas, there is a very tangible difference. That is, for polynomial-size $\sum\bigwedge\sum$ formulas the $\dim\boldsymbol{\partial}_{\overline{x}^{<\infty}}$ measure is polynomially-bounded. In contrast, for the values of k,ℓ chosen for the above lower bound, the dimension $\dim\overline{\boldsymbol{x}}^{\leq\ell}\boldsymbol{\partial}_{\overline{x}^{\leq k}}(\sum\bigwedge\sum\prod^t)$ is *exponentially* large. That is, to obtain the above lower bound Kayal [1], [2] had to show that this exponential is exponentially smaller than the corresponding measure of the monomial $x_1\cdots x_n$.

The largeness of this measure seems very problematic for designing PIT algorithms. That is, PIT algorithms often seek to reduce PIT of n-variate polynomials to polynomials on m-variate polynomials for $m \ll n$. Sometimes one arrives at $m = 1$ directly in which case univariate interpolation is then applied, and other times one gets $m = n/2$ to which recursion is applied. However, in order for these reductions to succeed one needs to argue that non-zero polynomials *remain non-zero*. For inductive purposes this often requires preserving more than just non-zeroness alone, and the amount of information to preserve is often quantified by the measure of complexity of the computation (for example, in roABPs one needs to preserve $\approx w$ amount of information in width-w roABPs). However, in shifted partial derivatives this measure is exponentially large and thus there is no efficient way to preserve this amount of information while reducing the amount of variables.

However, despite this obstacle we show how to extend the methods of Forbes-Shpilka [40] for the partial derivative space to shifted partials, in particular by scaling down Kayal's [1], [2] bound and making it robust. As such, we arrive at the following theorem.

Theorem. *Let $\mathbb{F}$ be a field of characteristic $> d$. Then the "leading monomial" of a size-s $\sum\bigwedge\sum\prod^t$ formula involves $\mathcal{O}(t\lg s)$ variables. In particular, there is a deterministic $\mathsf{poly}(s)^{\mathcal{O}(t\lg s)}$-time black-box PIT algorithm for size-s $\sum\bigwedge\sum\prod^t$ n-variate, degree-$(\leq d)$ formulas.* □

This is the first non-trivial (white-box or black-box) PIT algorithm for this class of computations. In particular, the above algorithm is black-box and no better white-box algorithm is known.

PIT of $\sum\mathrm{m}\bigwedge\sum\prod^t$:: Given the above results on $\sum\bigwedge\sum\prod^t$, we now seek to generalize this to a larger class of formulas, called $\sum\mathrm{m}\bigwedge\sum\prod^t$, which we now define. We motivate this class in two ways. The first

way is that this class naturally contains both $\sum\bigwedge\sum\prod^t$ and sparse polynomials. While the monomial $x_1\cdots x_n$ is hard for $\sum\bigwedge\sum\prod^t$, it is very easy to compute as a sparse polynomial. As such, obtaining lower bounds simultaneously against *both* $\sum\bigwedge\sum\prod^t$ and sparse polynomials seems to be a challenge. While for $t=1$ both of these models are subsumed by roABPs so that the methods from that literature apply, our results show these methods are not relevant for $t>1$. The second motivation comes from the mentioned connections with divisibility testing. That is, we will describe in the next section how testing whether a degree-t polynomial divides a sparse polynomial reduces to PIT of this $\sum\mathrm{m}\bigwedge\sum\prod^t$ class. Given this motivation, we now define this class.

Definition I.4. *A polynomial $f(\overline{x})\in\mathbb{F}[x_1,\ldots,x_n]$ is **computable by a** $\sum\mathrm{m}\bigwedge\sum\prod^t$ **formula** if*

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

*where $\deg f_i\le t$. The **size** of the formula is $\sum_i(\deg\overline{x}^{\overline{a}_i}+(d_i+1)\binom{n+t}{t})$. When $t=1$ we write this as $\sum\mathrm{m}\bigwedge\sum$.* ◊

When $t=1$, the notion of a $\sum\mathrm{m}\bigwedge\sum$ formula has been previously studied under the name of a *semi-diagonal depth-3 formula* by Saha, Saptharishi, and Saxena [7]. We rename this class here to mimic the above notation for a $\sum\bigwedge\sum$ formula. For arbitrary t, a more globally consistent name for this class would be "$\sum(\prod)\cdot(\bigwedge\sum^t)$", as the "$(\prod)$" here indicates a monomial and "$(\bigwedge\sum^t)$" indicates the power of a degree-$(\le t)$ polynomial. However, this notation seems cumbersome, and thus we instead opt for "m" to indicate the presence of the extra monomial added to the $\sum\bigwedge\sum\prod^t$ formula.

We begin by discussing the $t=0$ case, that is, sparse polynomials. The partial derivative method does not yield good results for these polynomials because the monomial $\overline{x}^{\overline{1}}:=x_1\cdots x_n$ has a large space of partial derivatives. However, observe that sparse polynomials are not closed under translation, so that $(\overline{x}+\overline{1})^{\overline{1}}:=\prod_i(x_i+1)$ is very *non-sparse*. Thus, even though the monomial $\overline{x}^{\overline{1}}$ is easy for sparse polynomials to compute, the *translated* monomial $(\overline{x}+\overline{1})^{\overline{1}}$ is hard to compute. While this lower bound is trivial to obtain, we now seek to scale it down and make it robust as done for $\sum\bigwedge\sum\prod^t$ formulas. That is, consider a s-sparse polynomial $f(\overline{x})=\sum_{i=1}^{s}\alpha_i\overline{x}^{\overline{a}_i}$. Suppose the translation $f(\overline{x}+\overline{1})$ computes the monomial $\overline{x}^{\overline{a}}$. Scaling down the previous argument it follows that $\overline{x}^{\overline{a}}$ must involve $\mathcal{O}(\lg s)$ variables, as $f(\overline{x})=(\overline{x}-\overline{1})^{\overline{a}}$ must have sparsity $\le s$. Now consider robustness. For size-s $\sum\bigwedge\sum\prod^t$ formulas we argued that in some sense the *first* monomial must involve few variables as this monomial is the "dominant term". However, for sparse polynomials this is false as the first monomial of $(\overline{x}+\overline{1})^{\overline{1}}$ is $\overline{x}^{\overline{1}}$, which involves many variables. However, the key insight here is that the *last* monomial (which in the previous example is 1) must involve few variables. Indeed, this is what we can show.

Theorem. *Let $f(\overline{x})\in\mathbb{F}[\overline{x}]$ be $(\le s)$-sparse. Then the "last" monomial of $f(\overline{x}+\overline{1})$ involves $\le\lg s$ variables.* □

We prove the above using a variant of the partial derivative method. That is, consider the differential operator $(x+1)\partial_x$ which maps $f\mapsto(x+1)\cdot\partial_x(f)$. Note that for a sparse polynomial in $x+1$ basis such as $(x+1)^2$, this operator leaves the polynomial unchanged (up to a constant). However, for a polynomial sparse in the x-basis such as x^2, it gets mapped to the "different" $2x(x+1)$. In general, we will consider the space $((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}(f):=\{(\overline{x}+\overline{1})^{\overline{b}}\cdot\partial_{\overline{x}^{\overline{b}}}(f)\}_{\overline{b}}$ where the exponent $\overline{b}$ ranges over monomials where $\deg\overline{x}^{\overline{b}}\le k$. Note that while this superficially seems similar to the space of shifted partial derivatives, there are some tangible differences. In particular, this space of differential operators *correlates* the "shift" $(\overline{x}+\overline{1})^{\overline{b}}$ with the derivative $\partial_{\overline{x}^{\overline{b}}}$. In shifted partial derivatives the shift and derivatives are uncorrelated. In particular, the dimension of shifted partial derivatives is translation invariant and as such *cannot* separate polynomials sparse in the $\overline{x}$ basis from polynomials sparse in the translated $\overline{x}+\overline{1}$ basis.

By the above arguments, if $f(\overline{x})$ is s-sparse (in the $\overline{x}$ basis) then $\dim((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}(f(\overline{x}+\overline{1}))$ is at most s. Further, we can show the following robustness property, that $\dim((\overline{\boldsymbol{x}}+\overline{\boldsymbol{\alpha}})\circ\boldsymbol{\partial}_{\overline{x}})^{\le k}(f(\overline{x}))\ge\dim\boldsymbol{\partial}_{\overline{x}^{\le k}}(\mathrm{TM}(f))$, where $\mathrm{TM}(f)$ is the "trailing monomial" or "last monomial" of f. As the usual partial derivative method shows that $\dim\boldsymbol{\partial}_{\overline{x}^{\le k}}(\overline{x}^{\overline{a}})$ is at least $2^{\|\overline{a}\|_0}$ where $\|\overline{a}\|_0$ is the number of variables in the monomial $\overline{x}^{\overline{a}}$, this gives the above theorem.

While polynomial-time black-box PIT is already known for sparse polynomials (for example, see Klivans and Spielman [77]), those results crucially exploit sparsity as a *combinatorial* criteria instead of an *algebraic* one. In particular, these results do not work for the class of translated sparse polynomials, that is, the class of $\{f(\overline{x}+\overline{\alpha})|f(\overline{x})\ s\text{-sparse}\ ,\overline{\alpha}\in\mathbb{F}^n\}$. However, as the above methods are algebraic they are somewhat insensitive to translations. As such, we can obtain the following black-box PIT algorithm by combining our results with those for roABPs of Forbes, Shpilka, Saptharishi [8].

Theorem. *Let $|\mathbb{F}|\geq \mathsf{poly}(n,d,s)$. There is a $\mathsf{poly}(n,d,s)^{\mathcal{O}(\lg\lg s)}$-time black-box PIT algorithm for the class of polynomials $f(\overline{x})\in\mathbb{F}[x_1,\ldots,x_n]$ that are translations of some s-sparse polynomial.* □

Prior to this work the best such black-box algorithm came from PIT of roABPs and thus took $\mathsf{poly}(n,d,s)^{\Theta(\lg n)}$ time.

Given that the above variant of the partial derivative method sufficed to understand $\sum\mathrm{m}\bigwedge\sum\prod^t$ for $t=0$, it seems logical that applying this variant to the *shifted* partial derivative method would yield similar results for $t>0$ when combined with our results for $\sum\bigwedge\sum\prod^t$ formulas. Indeed, we show the following result.

Theorem. *Let $\mathbb{F}$ be a field of characteristic $>d$. If $f(\overline{x})$ is a size-s $\sum\mathrm{m}\bigwedge\sum\prod^t$ formula then the "trailing/last monomial" of $f(\overline{x}+\overline{1})$ involves $\mathcal{O}(t\lg s)$ variables. In particular, there is a deterministic $\mathsf{poly}(s)^{\mathcal{O}(t\lg s)}$-time black-box PIT algorithm for size-s $\sum\mathrm{m}\bigwedge\sum\prod^t$ n-variate, degree-$(\leq d)$ formulas.* □

One can also obtain such a result for $\sum\mathrm{m}\bigwedge\sum\prod^t$ under translation[10]. Now, noting that $\sum\mathrm{m}\bigwedge\sum$ formulas are computable as commutative roABPs we can obtain the following black-box PIT algorithm for $\sum\mathrm{m}\bigwedge\sum$ by combining our results with those for roABPs of Forbes, Shpilka, Saptharishi [8].

Theorem. *Let $\mathbb{F}$ be a field of characteristic $>d$. There is a deterministic $\mathsf{poly}(s)^{\mathcal{O}(\lg\lg s)}$-time black-box PIT algorithm for size-s $\sum\mathrm{m}\bigwedge\sum$ n-variate, degree-$(\leq d)$ formulas.* □

As with translations of sparse polynomials, the previous best black-box PIT algorithm for this class required $\mathsf{poly}(s)^{\Theta(\lg n)}$ time.

Reducing Divisibility Testing to PIT:: As sketched above, we use Strassen's [6] elimination of divisions to give a deterministic reduction in the black-box model from testing whether $f(\overline{x})$ divides $g(\overline{x})$ to a PIT problem of a polynomial $h(\overline{x})$ which is not too much more complicated than f and g. As an example of such a reduction, consider the following lemma.

Lemma I.5. *Let $f(\overline{x})\in\mathbb{F}[\overline{x}]$ and $g(\overline{x},y)\in\mathbb{F}[\overline{x},y]$. Then $(y-f(\overline{x}))|g(\overline{x},y)$ iff $g(\overline{x},f(\overline{x}))=0$.* □

This lemma can be proven via long-division of multivariate polynomials (see for example Cox-Little-O'Shea [78]). In particular, if h is a linear polynomial then without loss of generality it is of the form $y-f(\overline{x})$ for some distinguished variable y. If $g(\overline{x},y)$ is sparse, then $g(\overline{x},f(\overline{x}))$ is then a $\sum\mathrm{m}\bigwedge\sum$ formula as noted by Saha, Saptharishi and Saxena [7]. With more work, one can push[11] the long-division method to reduce divisibility of a degree-t f into a sparse g to PIT of $\sum\mathrm{m}\bigwedge\sum\prod^t$. However, the resulting formula has size exponential in t and thus this approach seems limited to $t=\mathcal{O}(1)$.

Instead, as sketched above, we reduce "$f|g$?" to PIT of a polynomial h, where h is constructed using Strassen's [6] elimination of divisions. By carefully inspecting this procedure we can bound the complexity of h, yielding the following result.

Theorem. *Let $\mathbb{F}$ be a field with $|\mathbb{F}|\geq\mathsf{poly}(d)$. Let $\mathcal{C}_1,\mathcal{C}_2\subseteq\mathbb{F}[x_1,\ldots,x_n]$ be two classes of n-variate degree-$(\leq d)$ polynomials which are closed under the operations $f(\overline{x})\mapsto f(\alpha\cdot\overline{x}+\overline{\beta})$ for any $\alpha\in\mathbb{F}$ and $\overline{\beta}\in\mathbb{F}^n$. Then testing divisibility of $\mathcal{C}_1$ polynomials into $\mathcal{C}_2$ polynomials is efficiently reducible to polynomial identity testing of $\sum^{\mathsf{poly}(d)}\mathcal{C}_1\cdot\mathcal{C}_2\cdot\bigwedge^{\mathsf{poly}(d)}\mathcal{C}_1$ polynomials in both the black-box and white-box models of computation, where $\sum^{\mathsf{poly}(d)}\mathcal{C}_1\cdot\mathcal{C}_2\cdot$*

[10] In a future version of this work, we will show that these results also hold for a sum of a constant number of $\sum\mathrm{m}\bigwedge\sum\prod^t$ formula under *different* translations.

[11] An exposition of this will be contained in a future version of this work, but this technique is essentially subsumed by the use of Strassen's [6] elimination of divisions.

$\bigwedge^{\mathsf{poly}(d)} \mathcal{C}_1$ *is the class of polynomials* $\left\{\sum_{i=1}^{s} \alpha_i \cdot f_i(\overline{x}) \cdot g_i(\overline{x}) \cdot h_i(\overline{x})^{d_i} \mid f_i, h_i \in \mathcal{C}_1, g_i \in \mathcal{C}_2, s, d_i \leq \mathsf{poly}(d)\right\}$. *In particular, there is an efficient randomized algorithm for divisibility testing.* □

However, the above theorem as stated is limited in that it needs the closure of $\mathcal{C}_1$ and $\mathcal{C}_2$ under the maps $f(\overline{x}) \mapsto f(\alpha \cdot \overline{x} + \overline{\beta})$. In particular, sparse polynomials are not closed under this operation. However, by tailoring our reduction to the specific divisibility testing problem of whether a constant-degree polynomial divides a sparse polynomial and applying our PIT results we obtain the following.

Theorem (Main Result)**.** *Let* $\mathbb{F}$ *be a field* $\mathrm{char}(\mathbb{F}) \geq \mathsf{poly}(d)$. *Let* $f \in \mathbb{F}[x_1, \ldots, x_n]$ *where* $\deg f \leq \mathcal{O}(1)$. *Let* $g \in \mathbb{F}[\overline{x}]$ *be of degree* $\leq d$ *and be computable by a size-*s $\sum \mathrm{m} \bigwedge \sum \prod^{\mathcal{O}(1)}$ *formula. Then computing the multiplicity of* f *as a factor of* g *can be deterministically computed in* $\mathsf{poly}(s, n, d)^{\mathcal{O}(\lg s)}$*-time in the black-box model. In particular, one can take* g *to be a* s*-sparse polynomial.* □

ACKNOWLEDGMENTS

We would like to thank Pritish Kamath, Ramprasad Saptharishi and Amir Shpilka for many discussions about shifted partial derivatives. We would also like to thank Amir Shpilka in particular for the question of how to deterministically test whether a quadratic polynomial divides a sparse polynomial, Chandan Saha for questions that led to the lower bound results on roABPs, and Jakob Nordström for inviting him to KTH to give lectures on algebraic complexity, where the ideas on sparse polynomials were then conceived. We would also like to thank Peter Bürgisser, Yuval Filmus, Neeraj Kayal, Pascal Koiran, Swastik Kopparty, Mrinal Kumar, Rafael Oliveira, Shubhangi Saraf, Avi Wigderson, and anonymous reviewers for various comments and questions.

REFERENCES

[1] N. Kayal, "An exponential lower bound for the sum of powers of bounded degree polynomials," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 19, no. 81, 2012. [Online]. Available: http://eccc.hpi-web.de/report/2012/081

[2] A. Gupta, P. Kamath, N. Kayal, and R. Saptharishi, "Approaching the chasm at depth four," *J. ACM*, vol. 61, no. 6, pp. 33:1–33:16, Dec. 2014, Preliminary version in the *28th Annual IEEE Conference on Computational Complexity (CCC 2013)*.

[3] M. Agrawal and V. Vinay, "Arithmetic circuits: A chasm at depth four," in *Proceedings of the 49th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2008)*, 2008, pp. 67–75, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR08-062.

[4] P. Koiran, "Arithmetic circuits: The chasm at depth four gets wider," *Theor. Comput. Sci.*, vol. 448, pp. 56–65, 2012, Preliminary version at arXiv:1006.4700.

[5] S. Tavenas, "Improved bounds for reduction to depth 4 and depth 3," in *Proceedings of the 38th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 2013)*, 2013, pp. 813–824, Full version at arXiv:1304.5777.

[6] V. Strassen, "Vermeidung von divisionen," *J. Reine Angew. Math.*, vol. 264, pp. 184–202, 1973.

[7] C. Saha, R. Saptharishi, and N. Saxena, "A case of depth-3 identity testing, sparse factorization and duality," *Computational Complexity*, vol. 22, no. 1, pp. 39–69, 2013, Preliminary version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR11-021.

[8] M. A. Forbes, R. Saptharishi, and A. Shpilka, "Hitting sets for multilinear read-once algebraic branching programs, in any order," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, 2014, pp. 867–875, Full version at arXiv:1309.5668. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591816

[9] L. Lovász, "On determinants, matchings, and random algorithms," in *Fundamentals of computation theory (Proc. Conf. Algebraic, Arith. and Categorical Methods in Comput. Theory, Berlin/Wendisch-Rietz, 1979)*, ser. Math. Res. Akademie-Verlag, Berlin, 1979, vol. 2, pp. 565–574.

[10] R. M. Karp, E. Upfal, and A. Wigderson, "Constructing a perfect matching is in random NC," *Combinatorica*, vol. 6, no. 1, pp. 35–48, 1986, Preliminary version in the *17th Annual ACM Symposium on Theory of Computing (STOC 1985)*.

[11] K. Mulmuley, U. V. Vazirani, and V. V. Vazirani, "Matching is as easy as matrix inversion," *Combinatorica*, vol. 7, no. 1, pp. 105–113, 1987, Preliminary version in the *19th Annual ACM Symposium on Theory of Computing (STOC 1987)*.

[12] J. T. Schwartz, "Fast probabilistic algorithms for verification of polynomial identities," *J. ACM*, vol. 27, no. 4, pp. 701–717, Oct. 1980, Preliminary version in the *International Symposium on Symbolic and Algebraic Computation (EUROSAM 1979)*.

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

[14] R. A. DeMillo and R. J. Lipton, "A probabilistic remark on algebraic program testing," *Inf. Process. Lett.*, vol. 7, no. 4, pp. 193–195, 1978.

[15] 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 (STOC 1980)*, 1980, pp. 262–272.

[16] V. Kabanets and R. Impagliazzo, "Derandomizing polynomial identity tests means proving circuit lower bounds," *Computational Complexity*, vol. 13, no. 1-2, pp. 1–46, 2004, Preliminary version in the *35th Annual ACM Symposium on Theory of Computing (STOC 2003)*.

[17] M. Agrawal, "Proving lower bounds via pseudo-random generators," in *Proceedings of the 25th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2005)*, 2005, pp. 92–105.

[18] Z. Dvir and A. Shpilka, "Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits," *SIAM J. Comput.*, vol. 36, no. 5, pp. 1404–1434, 2007, Preliminary version in the *37th Annual ACM Symposium on Theory of Computing (STOC 2005)*.

[19] N. Kayal and N. Saxena, "Polynomial identity testing for depth 3 circuits," *Computational Complexity*, vol. 16, no. 2, pp. 115–138, 2007, Preliminary version in the *21st Annual IEEE Conference on Computational Complexity (CCC 2006)*.

[20] Z. S. Karnin and A. Shpilka, "Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in," *Combinatorica*, vol. 31, no. 3, pp. 333–364, 2011, Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*.

[21] 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 (FOCS 2009)*, 2009, pp. 198–207, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR09-032.

[22] N. Saxena and C. Seshadhri, "An almost optimal rank bound for depth-3 identities," *SIAM J. Comput.*, vol. 40, no. 1, pp. 200–224, 2011, Preliminary version in the *24th Annual IEEE Conference on Computational Complexity (CCC 2009)*.

[23] Z. S. Karnin, P. Mukhopadhyay, A. Shpilka, and I. Volkovich, "Deterministic identity testing of depth-4 multilinear circuits with bounded top fan-in," *SIAM J. Comput.*, vol. 42, no. 6, pp. 2114–2131, 2013, Preliminary version in the *42nd Annual ACM Symposium on Theory of Computing (STOC 2010)*.

[24] S. Saraf and I. Volkovich, "Black-box identity testing of depth-4 multilinear circuits," in *Proceedings of the 43rd Annual ACM Symposium on Theory of Computing (STOC 2011)*, 2011, pp. 421–430, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR11-046.

[25] M. Agrawal, C. Saha, R. Saptharishi, and N. Saxena, "Jacobian hits circuits: hitting-sets, lower bounds for depth-D occur-k formulas & depth-3 transcendence degree-k circuits," in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, 2012, pp. 599–614, Full version at arXiv:1111.0582.

[26] N. Saxena and C. Seshadhri, "Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter," *SIAM J. Comput.*, vol. 41, no. 5, pp. 1285–1298, 2012, Preliminary version in the *43rd Annual ACM Symposium on Theory of Computing (STOC 2011)*.

[27] ——, "From Sylvester-Gallai configurations to rank bounds: Improved blackbox identity test for depth-3 circuits," *J. ACM*, vol. 60, no. 5, p. 33, 2013, Preliminary version in the *51st Annual IEEE Symposium on Foundations of Computer Science (FOCS 2010)*.

[28] A. Gupta, "Algebraic geometric techniques for depth-4 PIT & Sylvester-Gallai conjectures for varieties," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 21, p. 130, 2014. [Online]. Available: http://eccc.hpi-web.de/report/2014/130

[29] L. G. Valiant, S. Skyum, S. J. Berkowitz, and C. Rackoff, "Fast parallel computation of polynomials using few processors," *SIAM J. Comput.*, vol. 12, no. 4, pp. 641–644, 1983, Preliminary version in the *6th Internationl Symposium on the Mathematical Foundations of Computer Science (MFCS 1981)*.

[30] A. Gupta, P. Kamath, N. Kayal, and R. Saptharishi, "Arithmetic circuits: A chasm at depth three," in *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, 2013, pp. 578–587, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR13-026.

[31] A. Shpilka and A. Yehudayoff, "Arithmetic circuits: A survey of recent results and open questions," *Foundations and Trends in Theoretical Computer Science*, vol. 5, no. 3-4, pp. 2070–388, 2010.

[32] N. Saxena, "Progress on polynomial identity testing," *Bulletin of the EATCS*, vol. 99, pp. 49–79, 2009. [Online]. Available: https://www.eatcs.org/images/bulletin/beatcs99.pdf

[33] ——, "Progress on polynomial identity testing - II," *arXiv*, vol. 1401.0976, 2014. [Online]. Available: http://arxiv.org/abs/1401.0976

[34] R. Saptharishi, "Recent progress on arithmetic circuit lower bounds," *Bulletin of the EATCS*, vol. 114, pp. 76–118, 2014. [Online]. Available: http://eatcs.org/beatcs/index.php/beatcs/article/view/293

[35] R. Raz and A. Shpilka, "Deterministic polynomial identity testing in non-commutative models," *Comput. Complex.*, vol. 14, no. 1, pp. 1–19, Apr. 2005, Preliminary version in the *19th Annual IEEE Conference on Computational Complexity (CCC 2004)*.

[36] V. Arvind, P. Mukhopadhyay, and S. Srinivasan, "New results on noncommutative and commutative polynomial identity testing," *Computational Complexity*, vol. 19, no. 4, pp. 521–558, 2010, Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*.

[37] M. A. Forbes and A. Shpilka, "On identity testing of tensors, low-rank recovery and compressed sensing," in *Proceedings of the 44th Annual ACM Symposium on Theory of Computing (STOC 2012)*, 2012, pp. 163–172, Full version at arXiv:1111.0663.

[38] ——, "Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs," in *Proceedings of the 54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, 2013, pp. 243–252, Full version at arXiv:1209.2408.

[39] M. Agrawal, C. Saha, and N. Saxena, "Quasi-polynomial hitting-set for set-depth-Δ formulas," in *Proceedings of the 45th Annual ACM Symposium on Theory of Computing (STOC 2013)*, 2013, pp. 321–330, Full version at arXiv:1209.2333.

[40] M. A. Forbes and A. Shpilka, "Explicit Noether Normalization for simultaneous conjugation via polynomial identity testing," in *Proceedings of the 17th International Workshop on Randomization and Computation (RANDOM 2013)*, 2013, pp. 527–542, Full version at arXiv:1303.0084.

[41] M. Agrawal, R. Gurjar, A. Korwar, and N. Saxena, "Hitting-sets for ROABP and sum of set-multilinear circuits," *arXiv*, vol. 1406.7535, 2014. [Online]. Available: http://arxiv.org/abs/1406.7535

[42] R. Gurjar, A. Korwar, N. Saxena, and T. Thierauf, "Deterministic identity testing for sum of read once ABPs," in *Proceedings of the 30th Annual IEEE Conference on Computational Complexity (CCC 2015)*, 2015, Full version at arXiv:1411.7341.

[43] N. Saxena, "Diagonal circuit identity testing and lower bounds," in *Proceedings of the 35th International Colloquium on Automata, Languages and Programming (ICALP 2008)*, 2008, pp. 60–71, Preliminary version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR07-124.

[44] N. Nisan, "Lower bounds for non-commutative computation," in *Proceedings of the 23rd Annual ACM Symposium on Theory of Computing (STOC 1991)*, 1991, pp. 410–418.

[45] R. Raz, "Separation of multilinear circuit and formula size," *Theory of Computing*, vol. 2, no. 6, pp. 121–135, 2006, Preliminary version in the *45th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2004)*.

[46] ——, "Multi-linear formulas for permanent and determinant are of super-polynomial size," *J. ACM*, vol. 56, no. 2, 2009, Preliminary version in the *36th Annual ACM Symposium on Theory of Computing (STOC 2004)*.

[47] R. Raz and A. Yehudayoff, "Lower bounds and separations for constant depth multilinear circuits," *Computational Complexity*, vol. 18, no. 2, pp. 171–207, 2009, Preliminary version in the *23rd Annual IEEE Conference on Computational Complexity (CCC 2008)*.

[48] K. Mulmuley, "Geometric complexity theory V: Equivalence between blackbox derandomization of polynomial identity testing and derandomization of Noether's normalization lemma," in *Proceedings of the 53rd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2012)*, 2012, pp. 629–638, Full version at arXiv:1209.5993.

[49] M. Forbes, "Polynomial identity testing of read-once oblivious algebraic branching programs," Ph.D. dissertation, Massachusetts Institute of Technology, Jun. 2014. [Online]. Available: http://hdl.handle.net/1721.1/89843

[50] V. Arvind, P. S. Joglekar, and S. Srinivasan, "Arithmetic circuits and the hadamard product of polynomials," in *Proceedings of the 29th International Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2009)*, 2009, pp. 25–36, Full version at arXiv:0907.4006.

[51] Z. Dvir, A. Shpilka, and A. Yehudayoff, "Hardness-randomness tradeoffs for bounded depth arithmetic circuits," *SIAM J. Comput.*, vol. 39, no. 4, pp. 1279–1293, 2009, Preliminary version in the *40th Annual ACM Symposium on Theory of Computing (STOC 2008)*.

[52] R. Oliveira, A. Shpilka, and B. L. Volk, "Subexponential size hitting sets for bounded depth multilinear formulas," in *Proceedings of the 30th Annual IEEE Conference on Computational Complexity (CCC 2015)*, 2015, Full version at arXiv:1411.7492.

[53] N. Kayal, C. Saha, and R. Saptharishi, "A super-polynomial lower bound for regular arithmetic formulas," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, 2014, pp. 146–153, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR13-091. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591847

[54] H. Fournier, N. Limaye, G. Malod, and S. Srinivasan, "Lower bounds for depth 4 formulas computing iterated matrix multiplication," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, 2014, pp. 128–135, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR13-100. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591824

[55] N. Kayal, N. Limaye, C. Saha, and S. Srinivasan, " Super-polynomial lower bounds for depth-4 homogeneous arithmetic formulas," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, 2014, pp. 119–127, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR14-005. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591823

[56] M. Kumar and S. Saraf, " The limits of depth reduction for arithmetic formulas: it's all about the top fan-in," in *Proceedings of the 46th Annual ACM Symposium on Theory of Computing (STOC 2014)*, 2014, pp. 136–145, Full version at arXiv:1311.6716. [Online]. Available: http://doi.acm.org/10.1145/2591796.2591827

[57] ——, " Superpolynomial lower bounds for general homogeneous depth 4 arithmetic circuits," in *Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP 2014)*, 2014, pp. 751–762, Full version at arXiv:1312.5978.

[58] N. Kayal, N. Limaye, C. Saha, and S. Srinivasan, "An exponential lower bound for homogeneous depth four arithmetic formulas," in *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*, 2014, pp. 61–70, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR14-005.

[59] M. Kumar and S. Saraf, " On the power of homogeneous depth 4 arithmetic circuits," in *Proceedings of the 55th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2014)*, 2014, pp. 364–373, Full version at arXiv:1404.1950.

[60] N. Kayal and C. Saha, "Lower bounds for depth three arithmetic circuits with small bottom fanin," in *Proceedings of the 30th Annual IEEE Conference on Computational Complexity (CCC 2015)*, 2015, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR14-089.

[61] N. Nisan and A. Wigderson, "Lower bounds on arithmetic circuits via partial derivatives," *Computational Complexity*, vol. 6, no. 3, pp. 217–234, 1996, Preliminary version in the *36th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1995)*.

[62] E. L. Kaltofen, "Factorization of polynomials given by straight-line programs," in *Randomness and Computation*, ser. Advances in Computing Research, S. Micali, Ed. Greenwich, CT, USA: JAI Press, Inc., 1989, vol. 5, pp. 375–412. [Online]. Available: http://www.math.ncsu.edu/~kaltofen/bibliography/89/Ka89_slpfac.pdf

[63] E. L. Kaltofen and B. M. Trager, "Computing with polynomials given by black boxes for their evaluations: greatest common divisors, factorization, separation of numerators and denominators," *J. Symb. Comput.*, vol. 9, no. 3, pp. 301–320, 1990, Preliminary version in the *29th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1988)*.

[64] S. Kopparty, S. Saraf, and A. Shpilka, "Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization," in *Proceedings of the 29th Annual IEEE Conference on Computational Complexity (CCC 2014)*, 2014, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR14-001.

[65] A. Shpilka and I. Volkovich, "On the relation between polynomial identity testing and finding variable disjoint factors," in *Proceedings of the 37th International Colloquium on Automata, Languages and Programming (ICALP 2010)*, 2010, pp. 408–419, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR10-036.

[66] Z. Dvir, R. Oliveira, and A. Shpilka, "Testing equivalence of polynomials under shifts," in *Proceedings of the 41st International Colloquium on Automata, Languages and Programming (ICALP 2014)*, 2014, pp. 417–428, Full version at arXiv:1401.3714. [Online]. Available: http://dx.doi.org/10.1007/978-3-662-43948-7_35

[67] I. Volkovich, "Deterministically factoring sparse polynomials into multilinear factors," *Electronic Colloquium on Computational Complexity (ECCC)*, vol. 21, p. 168, 2014. [Online]. Available: http://eccc.hpi-web.de/report/2014/168

[68] J. von zur Gathen and E. L. Kaltofen, "Factoring sparse multivariate polynomials," *J. Comput. Syst. Sci.*, vol. 31, no. 2, pp. 265–287, 1985, Preliminary version in the *24th Annual IEEE Symposium on Foundations of Computer Science (FOCS 1983)*.

[69] Z. Dvir and R. Oliveira, "Factors of sparse polynomials are sparse," *arXiv*, vol. 1404.4834, 2014, this manuscript has been withdrawn. [Online]. Available: http://arxiv.org/abs/1404.4834

[70] E. L. Kaltofen and P. Koiran, "On the complexity of factoring bivariate supersparse (lacunary) polynomials," in *Proceedings of the 2005 International Symposium on Symbolic and Algebraic Computation (ISSAC 2005)*, 2005, pp. 208–215.

[71] ——, "Finding small degree factors of multivariate supersparse (lacunary) polynomials over algebraic number fields," in *Proceedings of the 2006 International Symposium on Symbolic and Algebraic Computation (ISSAC 2006)*, 2006, pp. 162–168.

[72] A. Chattopadhyay, B. Grenet, P. Koiran, N. Portier, and Y. Strozecki, " Factoring bivariate lacunary polynomials without heights," in *Proceedings of the 40th International Colloquium on Automata, Languages and Programming (ICALP 2013)*, 2013, pp. 141–148, the full version of this work ([79]) contains multivariate generalizations of the original (bivariate) full version (arXiv:1206.4224).

[73] B. Grenet, " Computing low-degree factors of lacunary polynomials: a Newton-Puiseux approach," in *Proceedings of the 2014 International Symposium on Symbolic and Algebraic Computation (ISSAC 2014)*, 2014, pp. 224–231, the full version of this work ([80]) contains simplified proofs of the original full version (arXiv:1401.4720).

[74] J. M. Landsberg, *Tensors: geometry and applications*, ser. Graduate Studies in Mathematics. American Mathematical Society, Providence, RI, 2012, vol. 128.

[75] I. Fischer, "Sums of like powers of multivariate linear forms," *Mathematics Magazine*, vol. 67, no. 1, pp. 59–61, 1994. [Online]. Available: http://www.jstor.org/stable/2690560

[76] A. Shpilka and I. Volkovich, "Improved polynomial identity testing for read-once formulas," in *Proceedings of the 13th International Workshop on Randomization and Computation (RANDOM 2009)*, 2009, pp. 700–713, Full version in the Electronic Colloquium on Computational Complexity (ECCC), Technical Report TR10-011.

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

[78] D. Cox, J. Little, and D. O'Shea, *Ideals, varieties, and algorithms*, 3rd ed., ser. Undergraduate Texts in Mathematics. Springer, New York, 2007, an introduction to computational algebraic geometry and commutative algebra.

[79] A. Chattopadhyay, B. Grenet, P. Koiran, N. Portier, and Y. Strozecki, "Computing the multilinear factors of lacunary polynomials without heights," *arXiv*, vol. 1311.5694, 2013. [Online]. Available: http://arxiv.org/abs/1311.5694

[80] B. Grenet, "Bounded-degree factors of lacunary multivariate polynomials," *arXiv*, vol. 1412.3570, 2014. [Online]. Available: http://arxiv.org/abs/1412.3570