Towards blackbox identity testing of log-variate circuits

Michael A. Forbes
University of Illinois at Urbana-Champaign, USA
miforbes@illinois.edu

Sumanta Ghosh
Department of Computer Science, IIT Kanpur, India
sumghosh@cse.iitk.ac.in

Nitin Saxena
Department of Computer Science, IIT Kanpur, India
nitin@cse.iitk.ac.in

Abstract

Derandomization of blackbox identity testing reduces to extremely special circuit models. After a line of work, it is known that focusing on circuits with constant-depth and constantly many variables is enough (Agrawal,Ghosh,Saxena, STOC'18) to get to general hitting-sets and circuit lower bounds. This inspires us to study circuits with few variables, eg. logarithmic in the size s.

We give the first poly(s)-time blackbox identity test for $n = O(\log s)$ variate size-s circuits that have poly(s)-dimensional partial derivative space; eg. depth-3 diagonal circuits (or $\Sigma \wedge \Sigma^n$). The former model is well-studied (Nisan,Wigderson, FOCS'95) but no poly($s2^n$)-time identity test was known before us. We introduce the concept of *cone-closed* basis isolation and prove its usefulness in studying log-variate circuits. It subsumes the previous notions of rank-concentration studied extensively in the context of ROABP models.

2012 ACM Subject Classification Theory of computation– Algebraic complexity theory, Fixed parameter tractability, Pseudorandomness and derandomization; Computing methodologies– Algebraic algorithms; Mathematics of computing– Combinatoric problems.

Keywords and phrases hitting-set, depth-3, diagonal, derandomization, polynomial identity testing, log-variate, concentration, cone closed, basis isolation.

Digital Object Identifier 10.4230/LIPIcs.ICALP.2018.21

Related Version Full Version:
`https://www.cse.iitk.ac.in/users/nitin/papers/log-var-hsg.pdf`

1 Introduction

Polynomial Identity Testing (PIT) problem is to decide whether a multivariate polynomial is *zero*, where the input polynomial is given as an *algebraic circuit*. Algebraic circuits are the algebraic analog of boolean circuits that use ring operations $\{+, \times\}$ and computes polynomials (say) over a field. Since a polynomial computed by a circuit can have exponentially many monomials wrt the circuit size, one cannot solve PIT in polynomial time by explicitly expanding the polynomial. On the other hand, using circuits we can efficiently evaluate polynomials at any point. This helps us to get a polynomial time randomized algorithm for PIT by evaluating the circuit at a random point, since any non-zero polynomial evaluated at a random point outputs a non-zero value with high probability [10, 58, 54]. However, finding a deterministic polynomial time algorithm for PIT is a longstanding open question in

45th International Colloquium on Automata, Languages, and Programming (ICALP 2018).
Editors: Ioannis Chatzigiannakis, Christos Kaklamanis, Dániel Marx, and Don Sannella; Article No. 21; pp. 21:1–21:16
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

algebraic complexity theory. The PIT problem has been studied in two different paradigms: **1)** *whitebox*– allowed to see the internal structure of the circuit, and **2)** *blackbox*– can only use the circuit as an oracle to evaluate at points (from a small field extension). It has deep connections with both circuit lower bounds [29, 31, 1, 2] and many other algorithmic problems [41, 4, 35, 11, 13]. For more details on PIT, see the surveys [51, 52, 55] or review articles [56, 42].

Despite a lot of effort, little progress has been made on the PIT problem in general. However, efficient (deterministic poly-time) PIT algorithms are known for many special circuit models. For example, blackbox PIT for depth-2 circuits (or sparse polynomials) [8, 34, 39], PIT algorithms for subclasses of depth-3 circuits [33, 50, 53], subclasses of depth-4 circuits [5, 7, 46, 15, 36, 37, 45], read-once algebraic branching programs (ROABP) and related models [19, 6, 18, 3, 26, 25], certain types of symbolic determinants [12, 27], as well as non-commutative models [38, 22].

1.1 Our results

In the first result, *we give a polynomial time blackbox PIT algorithm of log-variate depth-*3 *diagonal circuits* $\Sigma\wedge\Sigma$ (i.e. number of variables is logarithmic wrt circuit size). Depth-3 diagonal circuits compute a sum of power of linear polynomials. This model was first introduced by [51] and has since drawn significant attention of PIT research community. Saxena [51] first gave a polynomial time whitebox algorithm and exponential lower bound for this model, by introducing a duality trick. In a subsequent work Kayal [32] gave an alternate polynomial time whitebox algorithm for depth-3 diagonal circuits based on the partial derivative method, which was first introduced by [44] to prove circuit lower bounds; as, $\Sigma\wedge\Sigma$ circuits have a low-dimension partial derivative space. However, one limitation of these approaches was that they depend on the characteristic of the underlying field. Later, [16] gave an alternative proof of duality trick which depends only on the field size (as mentioned in [24, Lem.4.7]) and Saptharishi [48, Chap.3] extended Kayal's idea for large enough field.

Although this model is very weak (it cannot even compute $x_1\cdots x_n$ efficiently), studying this model has proved quite fruitful. Duality trick was crucially used in the work by [23], where they showed that depth-3 circuits, in some sense, capture the complexity of general arithmetic circuits.

Like whitebox PIT, a series of work has been done on *blackbox* PIT for depth-3 diagonal circuits. Both [6] and [19] gave two independent and different quasi-polynomial time blackbox PIT algorithms for this model. Later, [18] gave an $s^{O(\log\log s)}$-time (s is the circuit size) blackbox PIT algorithm for this model. Mulmuley [43, 40] related depth-3 diagonal blackbox PIT to construction of normalization maps for the invariants of the group SL_m for constant m. We can not give the detailed notation here and would like to refer to [40, Sec.9.3]. Despite a lot of effort, no polynomial time blackbox PIT for this model is known. After depth-2 circuits (or sparse polynomials), this can be thought of as the simplest model for which no polynomial time blackbox PIT is known. Because of its simplicity, this model is a good test case for generating new ideas for the PIT problem.

Log-variate models: Now we discuss why studying PIT for log-variate models is so important. The PIT algorithms in current literature always try to achieve a sub-exponential dependence on n, the number of variables. In a recent development, [2] showed that for some constant c a poly(s)-time blackbox PIT for size-s degree-s and $\log^{\circ c} s$-variate[1] circuits is

[1] The function $\log^{\circ c}$ denotes c times composition of the log function. For e.g. $\log^{\circ 2} s = \log\log s$.

sufficient to *completely* solve PIT. Most surprisingly, they also showed that a poly(s)-time blackbox PIT for size-s and $\log^\star s$-variate[2] $\Sigma\wedge\Sigma\Pi$ circuits will 'partially' solve PIT (in quasi-polynomial time) and prove that "either E$\not\subseteq$#P/poly or VP$\neq$VNP" (a weaker version of [2, Thm.21]). For example, even a poly(s)-time blackbox PIT for size-s and $(\log\log s)$-variate depth-4 circuits would be tremendous progress. A similar result also holds for $\Sigma\wedge^a\Sigma\Pi(n)$ circuits, where both a and n are 'arbitrarily small' unbounded functions (i.e. time-complexity may be arbitrary in terms of both a and n), see [2, Thm.21].

The above discussion motivates us to discover techniques and measures that are specialized to this low-variate regime. Many previous works are based on 'support size of a monomial' as a measure for rank-concentration [6, 18, 26]. For a monomial m, its *support* is the set of variables whose exponents are positive. We investigate a 'larger' measure: *cone-size* (see Definition 3) which is the number of monomials that divide m (also see [14]). Using cone-size as a measure for rank-concentration, we give a blackbox PIT algorithm for circuit models with 'low' dimensional partial derivative space.

▶ **Theorem 1.** *Let $\mathbb{F}$ be a field of characteristic 0 or greater than d. Let $\mathcal{P}$ be a set of n-variate d-degree polynomials, over $\mathbb{F}$, computed by circuits of bitsize s such that: $\forall P\in\mathcal{P}$, the dimension of the partial derivative space of P is at most k. Then, blackbox PIT for $\mathcal{P}$ can be solved in $(sdk)^{O(1)}\cdot(3n/\log k)^{O(\log k)}$ time.*

Note that for $n=O(\log k)=O(\log sd)$, the above bound is poly-time and we get a polynomial time blackbox PIT algorithm for log-variate circuits (i.e. number of variables is logarithmic wrt circuit size) with low-dimensional partial derivative space. This was not known before our work. Prior to our work, [18] gave a $(sdk)^{O(\log\log sdk)}$-time algorithm for $\mathcal{P}$, using support size as the measure in the proof. Unlike our algorithm, in the log-variate case their algorithm remains super-polynomial time.

In particular, diagonal depth-3 circuit is a prominent model with low partial derivative space. So, our method gives a polynomial time PIT algorithm for log-variate depth-3 diagonal circuits. No poly-time blackbox PIT for this model was known before our work; again, $s^{O(\log\log s)}$ was the prior best [18].

Structure of log-variate polynomials? In the second result, we investigate a structural property of polynomials over vector spaces. For a polynomial $f(\mathbf{x})$ with coefficients over $\mathbb{F}^k$, let sp(f) be the subspace spanned by its coefficients. Informally, in *rank concentration* we try to concentrate the rank of sp(f) to the coefficients of "few" monomials. It was first introduced by [6]. Many works in PIT achieve rank concentration on low-support monomials, mainly, in the ROABP model [6, 18, 26, 25]. One way of strengthening low-support concentration is through *low-cone concentration*, where rank is concentrated in the low cone-size monomials. This concept was not used before in designing PIT algorithms. Our first result (Theorem 1) can be seen from this point of view. There, we developed a method to get polynomial time blackbox PIT for log-variate models which satisfy 'low-cone concentration property'.

We introduce the concept of cone-closed basis, a much stronger notion of concentration than the previous ones. We say f has a cone-closed basis, if there is a set of monomials B whose coefficients form a basis of sp(f) and B is closed under sub-monomials. This definition is motivated by a special depth-3 diagonal model, which have this property naturally (see Lemma 18). We prove that this notion is a strengthening of both low-support and low-cone

[2] For any positive integer s, $\log^\star s=\min\{i\mid\log^{\circ i}s\leq 1\}$.

concentration ideas (see Lemma 11). Recently, and independently, this notion of closure has also appeared as an 'abstract simplicial complex' in [21].

In the following result, we relate cone-closed basis with 'basis isolating weight assignment' (Defn.12)– another well studied concept in PIT. It was first introduced by [3] and also used in many other subsequent works [26, 12, 28]. Here, we show that a general polynomial f over $\mathbb{F}^k$, when shifted by a basis isolating weight assignment [3], becomes cone-closed. It strengthens some previously proven properties; eg., a polynomial over $\mathbb{F}^k$ when shifted 'randomly' becomes low-support concentrated [17, Cor.3.22] (extended version of [18]) or, when shifted by a basis isolating weight assignment becomes low-support concentrated [26, Lem.5.2].

Notations. For any $n \in \mathbb{N}$, $[n]$ denotes the set of first n positive integers. By $\mathbf{x}$, we denote $(x_1, \ldots, x_n)$, a tuple of n-variables. For any $\mathbf{e} = (e_1, \ldots, e_n) \in \mathbb{N}^n$, $\mathbf{x}^{\mathbf{e}}$ denotes the monomial $\prod_{i=1}^n x_i^{e_i}$. For a polynomial f and a monomial m, $\text{coef}_m(f)$ denotes the coefficient of the monomial m in f. An *weight assignment* $\mathbf{w}$ on the variables $\mathbf{x}$ is an n-tuple $(w_1, \ldots, w_n) \in \mathbb{N}^n$, where w_i is the weight assigned to the variable x_i.

► **Theorem 2.** *Let $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^k$ be an n-variate d-degree polynomial over $\mathbb{F}^k$ and char $\mathbb{F} = 0$ or $> d$. Let $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{N}^n$ be a basis isolating weight assignment of $f(\mathbf{x})$. Then, $f(\mathbf{x} + t^{\mathbf{w}}) := f(x_1 + t^{w_1}, \ldots, x_n + t^{w_n})$ has a cone-closed basis over $\mathbb{F}(t)$.*

1.2 Proof ideas

Proof idea of Theorem 1: The proof of Theorem 1 has two steps. In the first step, we show that with respect to any monomial ordering (say lexicographic monomial ordering), the dimension k of the partial derivative space of a polynomial is lower bounded by the cone-size of its leading monomial. For a polynomial $f \in \mathbb{F}[\mathbf{x}]$, the leading monomial, wrt a monomial ordering, is the largest monomial in the set $\{\mathbf{x}^{\mathbf{e}} \mid \text{coef}_{\mathbf{x}^{\mathbf{e}}}(f) \neq 0\}$. So, for every nonzero $P \in \mathcal{P}$ there is a monomial with nonzero coefficient and cone-size $\leq k$. The second step is to check whether the coefficients of all the monomials in P, with cone-size $\leq k$, are zero. We show that the number of such monomials is small (Lemma 5); the number is quasi-polynomial in general, but, merely polynomial in the log-variate case. Next, we give a new method to efficiently extract a monomial of cone-size$\leq k$, out of a potentially exponential space of monomials (Lemma 4). These facts, combined with the estimates stated in Theorem 1, prove Corollary 6; which gives a polynomial time blackbox PIT algorithm for log-variate circuits with low dimensional partial derivative space.

Next, we discuss the idea to get a polynomial time blackbox PIT algorithm for depth-3 diagonal circuits where *rank* of the linear polynomials is logarithmic wrt the circuit size (see Definition 7 & Theorem 9). Here, the proof has two steps. First, in Lemma 8, we show how to efficiently reduce a low-rank depth-3 diagonal circuit to a low-*variate* depth-3 diagonal circuit while preserving nonzeroness. This we do by a Vandermonde based linear map on the variables. Since a depth-3 diagonal circuit has low-dimensional partial derivative space (i.e. polynomial wrt circuit size), we apply Corollary 6 on the low-variate depth-3 diagonal circuits and get Theorem 9.

Proof idea of Theorem 2: First, wrt the weight assignment $\mathbf{w}$, we define an ordering among the set of bases (see Section 3). Then, we show that wrt the basis isolating weight assignment $\mathbf{w}$, there exists a *unique minimum basis* and its weight is strictly less than the weight of every other basis (Lemma 13). Let B be the set of monomials whose coefficients form the least basis, wrt $\mathbf{w}$, of f.

Now, we consider the set of all sub-monomials of those in B and identify a subset A that is cone-closed. We define A in an algorithmic way (see Algorithm 1). Besides the cone-closed property, A also satisfies an algebraic property (Lemma 17)— In the *transfer matrix* T, that captures the variable-shift transformation (Equation 3), the sub-matrix $T_{A,B}$ is *full* rank. We prove that A is exactly a basis of the shifted f by studying the action of the shift on the coefficient vectors. The properties proved above and Cauchy-Binet Formula [57] are crucially used in the study of the coefficient vectors after the variable-shift.

Theorem 2 has an immediate consequence that any polynomial f over $\mathbb{F}^k$, when shifted by formal (or random) variables, becomes cone-closed; since the weight induced by the formal variables on the monomials is a basis isolating weight assignment. This seems quite a nontrivial and an interesting property of general polynomials (over vector spaces).

2 Low-cone concentration and hitting-sets– Proof of Theorem 1

In this section we initiate a study of properties that are relevant for low-variate circuits (or the log-variate regime).

Notations. For a circuit C, $|C|$ denotes the size of C. For a monomial m, by $\mathrm{coef}_m(C)$, we denote the coefficient of monomial m in the polynomial computed by C. For a circuit C, we also use C to denote the polynomial computed by C.

▶ **Definition 3** (Cone of a monomial). A monomial $\mathbf{x}^{\mathbf{e}}$ is called a *sub-monomial* of $\mathbf{x}^{\mathbf{f}}$, if $\mathbf{e} \leq \mathbf{f}$ (i.e. coordinate-wise). We say that $\mathbf{x}^{\mathbf{e}}$ is a *proper sub-monomial* of $\mathbf{x}^{\mathbf{f}}$, if $\mathbf{e} \leq \mathbf{f}$ and $\mathbf{e} \neq \mathbf{f}$.

For a monomial $\mathbf{x}^{\mathbf{e}}$, the *cone of* $\mathbf{x}^{\mathbf{e}}$ is the set of all sub-monomials of $\mathbf{x}^{\mathbf{e}}$. The cardinality of this set is called *cone-size of* $\mathbf{x}^{\mathbf{e}}$. It equals $\prod(\mathbf{e}+\mathbf{1}) := \prod_{i\in[n]}(e_i+1)$, where $\mathbf{e} = (e_1, \ldots, e_n)$.

A set S of monomials is called *cone-closed* if for every monomial in S all its sub-monomials are also in S.

▶ **Lemma 4** (Coef. extraction). *Let C be a blackbox circuit which computes an n-variate and degree-d polynomial over a field of size greater than d. Then for any monomial $m = \prod_{i\in[n]} x_i^{e_i}$, we have a $\mathrm{poly}(|C|d, \mathrm{cs}(m))$-time algorithm to compute the coefficient of m in C, where $\mathrm{cs}(m)$ denotes the cone-size of m.*

Proof. Our proof is in two steps. First, we inductively build a circuit computing a polynomial which has two parts; one is $\mathrm{coef}_m(C) \cdot m$ and the other one is a "junk" polynomial where every monomial is a proper super-monomial of m. Second, we construct a circuit which extracts the coefficient of m. In both these steps the key is a classic interpolation trick.

We induct on the variables. For each $i \in [n]$, let $m_{[i]}$ denote $\prod_{j\in[i]} x_j^{e_j}$. We will construct a circuit $C^{(i)}$ which computes a polynomial of the form,

$$C^{(i)}(\mathbf{x}) \;=\; \mathrm{coef}_{m_{[i]}}(C) \cdot m_{[i]} \;+\; C^{(i)}_{\text{junk}} \tag{1}$$

where, for every monomial m' in the support of $C^{(i)}_{\text{junk}}$, $m_{[i]}$ is a proper submonomial of $m'_{[i]}$.

Base case: Since $C =: C^{(0)}$ computes an n-variate degree-d polynomial, $C(\mathbf{x})$ can be written as $C(\mathbf{x}) = \sum_{j=0}^{d} c_j x_1^j$ where, $c_j \in \mathbb{F}[x_2, \ldots, x_n]$. Let $\alpha_0, \ldots, \alpha_{e_1}$ be some e_1+1 distinct elements in $\mathbb{F}$. For every α_j, let $C_{\alpha_j x_1}$ denote the circuit $C(\alpha_j x_1, x_2, \ldots, x_n)$ which computes $c_0 + c_1\alpha_j x_1 + \ldots + c_{e_1}\alpha_j^{e_1} x_1^{e_1} + \cdots + c_d \alpha_j^d x_1^d$. Since

$$M = \begin{bmatrix} 1 & \alpha_0 & \ldots & \alpha_0^{e_1} \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_{e_1} & \ldots & \alpha_{e_1}^{e_1} \end{bmatrix}$$

is an invertible Vandermonde matrix, one can find an $\mathbf{a} = [a_0, \dots, a_{e_1}] \in \mathbb{F}^{e_1+1}$, $\mathbf{a} \cdot M = [\ 0,\ 0,\ \dots,\ 1]$. Using this $\mathbf{a}$, we get the circuit $C^{(1)} := \sum_{j=0}^{e_1} a_j C^{(0)}_{\alpha_j x_1}$. Its least monomial wrt x_1 has $\deg_{x_1} \geq e_1$, which is the property that we wanted.

Induction step $(i \to i+1)$*:* From induction hypothesis, we have the circuit $C^{(i)}$ with the properties mentioned in Eqn.1. The polynomial can also be written as $b_0 + b_1 x_{i+1} + \dots + b_{e_{i+1}} x_{i+1}^{e_{i+1}} + \dots b_d x_{i+1}^d$, where every b_j is in $\mathbb{F}[x_1, \dots, x_i, x_{i+2}, \dots, x_n]$. Like the proof of the base case, for $e_{i+1} + 1$ distinct elements $\alpha_0, \dots, \alpha_{e_{i+1}} \in \mathbb{F}$, we get $C^{(i+1)} = \sum_{j=0}^{e_{i+1}} a_j C^{(i)}_{\alpha_j x_{i+1}}$, for some $\mathbf{a} = [a_0, \dots, a_{e_{i+1}}] \in \mathbb{F}^{e_{i+1}+1}$ and the structural constraint of $C^{(i+1)}$ is easy to verify, completing the induction.

Now we describe the second step of the proof. After first step, we get

$$C^{(n)}(\mathbf{x}) \;=\; \mathrm{coef}_m(C) \cdot m \;+\; C^{(n)}_{\mathrm{junk}}\,,$$

where for every monomial m' in the support of $C^{(n)}_{\mathrm{junk}}$, m is a proper submonomial of m'. Consider the polynomial $C^{(n)}(x_1 t, \dots, x_n t)$ for a fresh variable t. Then, using interpolation wrt t we can construct a $O(|C^{(n)}| \cdot d)$-size circuit for $\mathrm{coef}_m(C) \cdot m$, by extracting the coefficient of $t^{\deg(m)}$, since the degree of every monomial appearing in $C^{(n)}_{\mathrm{junk}}$ is $> \deg(m)$. Now evaluating at $\mathbf{1}$, we get $\mathrm{coef}_m(C)$. The size, or time, constraint of the final circuit clearly depends polynomially on $|C|, d$ and $\mathrm{cs}(m)$. ◀

But, how many low-cone monomials can there be? Fortunately, in the log-variate regime they are not too many [47]. Though, in general, they are quasi-polynomially many.

▶ **Lemma 5** (Counting low-cones). *The number of n-variate monomials with cone-size at most k is $O(rk^2)$, where $r := (3n/\log k)^{\log k}$.*

Proof. First, we prove that for any fixed support set, the number of cone-size $\leq k$ monomials is less than k^2. Next, we multiply by the number of possible support sets to get the estimate.

Let $T(k, \ell)$ denote the number of cone-size$\leq k$ monomials m with support set, say, exactly $\{x_1, \dots, x_\ell\}$. Since the exponent of x_ℓ in such an m is at least 1 and at most $k-1$, we have the following by the disjoint-sum rule: $T(k, \ell) \leq \sum_{i=2}^{k} T(k/i, \ell - 1)$. This recurrence affords an easy inductive proof as, $T(k, \ell) < \sum_{i=2}^{k} (k/i)^2 < k^2 \cdot \sum_{i=2}^{k} \left(\frac{1}{i-1} - \frac{1}{i}\right) < k^2$.

From the definition of cone, a cone-size $\leq k$ monomial can have support size at most $\ell := \lfloor \log k \rfloor$. The number of possible support sets, thus, is $\sum_{i=0}^{\ell} \binom{n}{i}$. Using the binomial estimates [30, Chapter 1], we get $\sum_{i=0}^{\ell} \binom{n}{i} \leq (3n/\ell)^{\ell}$. ◀

The partial derivative space of polynomials was first used by Nisan and Wigderson [44] to prove circuit lower bounds. Later, it was used in many other works. For more details see the following surveys [9, 49]. Here, using cone-size as a measure, we describe a blackbox PIT algorithm for circuits models with low dimensional partial derivative space. This algorithm runs in polynomial time when we are in log-variate regime. For a polynomial $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$, by $\partial_{\mathbf{x}^{<\infty}}(f)$ we denote the space generated all partial derivatives of f.

Proof of Theorem 1. The proof has two steps. First, we show that with respect to any monomial ordering $\prec$ (say lexicographic monomial ordering), for all nonzero $P \in \mathcal{P}$, the dimension of the partial derivative space of P is lower bounded by the cone-size of the leading monomial in P. Using this, we can get a blackbox PIT algorithm for $\mathcal{P}$ by testing the coefficients of all the monomials of P of cone-size $\leq k$ for zeroness. Next, we analyze the time complexity to do this.

The first part is the same as the proof of [14, Corollary 4.14] (with origins in [20]). Here, we give a brief outline. Let $LM(\cdot)$ be the *leading monomial* operator wrt the monomial ordering $\prec$. It can be shown that for any polynomial $f(\mathbf{x})$, the dimension of its partial derivative space $\partial_{\mathbf{x}^{<\infty}}(f)$ is the same as $D := \#\{LM(g) \mid g \in \partial_{\mathbf{x}^{<\infty}}(f)\}$ (see [14, Lemma 8.4.12]). This means that $\dim \partial_{\mathbf{x}^{<\infty}}(f)$ is lower-bounded by the cone-size of $LM(f)$ [14, Corollary 8.4.13], which completes the proof of our first part.

Next, we apply Lemma 4, on the circuit of P and a monomial m of cone-size $\le k$, to get the coefficient of m in C in poly(sdk)-time. Finally, Lemma 5 tells that we have to access at most $k^2 \cdot (3n/\log k)^{\log k}$ many monomials m. Multiplying these two expressions gives us the time bound. ◀

This gives us immediately,

▶ **Corollary 6.** *Let $\mathbb{F}$ be a field of characteristic 0 or $> d$. Let $\mathcal{P}$ be a set of n-variate d-degree polynomials, over $\mathbb{F}$, computable by circuits of bitsize s; with $n = O(\log sd)$. Suppose that, for all $P \in \mathcal{P}$, the dimension of the partial derivative space of P is poly(sd). Then, blackbox PIT for $\mathcal{P}$ can be solved in poly(sd)-time.*

Now we discuss our result regarding depth-3 diagonal circuits $\Sigma \wedge \Sigma$.

▶ **Definition 7** (Depth-3 diagonal circuit and its rank)**.** A *depth-*3 *diagonal circuit* is of the form $\Sigma \wedge \Sigma$ (sum-power-sum). It computes a polynomial presented as $C(\mathbf{x}) = \sum_{i\in[k]} c_i \ell_i^{d_i}$, where ℓ_i's are linear polynomials over $\mathbb{F}$ and c_i's in $\mathbb{F}$.

By $\mathrm{rk}(C)$ we denote the linear rank of the polynomials $\{\ell_i\}_{i\in[k]}$.

The next lemma introduces an efficient *nonzeroness preserving variable reduction* map $(n \mapsto \mathrm{rk}(C))$ for depth-3 diagonal circuits. For a set of n-variate circuits $\mathcal{C}$ over $\mathbb{F}$, a *polynomial map* $\Psi : \mathbb{F}^m \to \mathbb{F}^n$ is called nonzeroness preserving variable reduction map for $\mathcal{C}$, if $m < n$ and for all $C \in \mathcal{C}$, $C \neq 0$ if and only if $\Psi(C) \neq 0$.

▶ **Lemma 8** (Variable reduction)**.** *Let $P(\mathbf{x})$ be an n-variate d-degree polynomial computed by a size-s depth-*3 *diagonal circuit over some sufficiently large field $\mathbb{F}$. Then, there exists a* poly(nds)*-time computable nonzeroness preserving variable reduction map which converts P to another rk(P)-variate degree-d polynomial computed by* poly(s)*-size depth-3 diagonal circuit.*

For proof, see the full version linked on the first page.

▶ **Theorem 9** (Log-rank $\Sigma \wedge \Sigma$)**.** *Let $\mathbb{F}$ be a field of characteristic 0 or $> d$. Let $\mathcal{P}$ be the set of n-variate d-degree polynomials P, computable by depth-*3 *diagonal circuits of bitsize s, with rk$(P) = O(\log sd)$. Then, blackbox PIT for $\mathcal{P}$ can be solved in poly(sd)-time.*

Proof. The above description gives us a non-zeroness preserving variable reduction $(n \mapsto \mathrm{rk}(P))$ method that reduces P to an $O(\log(sd))$-variate and degree-d polynomial P' computed by poly(s)-size depth-3 diagonal circuit.

Since the dimension of the partial derivative space of P' is poly(sd) [14, Lem.8.4.8], Corollary 6 gives us a poly(sd)-time hitting-set for P'. ◀

3 Cone-closed basis after shifting– Proof of Theorem 2

In this section we will consider polynomials over a vector space, say $\mathbb{F}^k$. This viewpoint has been useful in studying algebraic branching programs (ABP), eg. [6, 18, 3, 26]. Let $D \in \mathbb{F}^k[\mathbf{x}]$ and let sp(D) be the vector space spanned by its coefficients. Now, we formally define various kinds of rank concentrations of D.

▶ **Definition 10** (Rank Concentration)**.** We say that D has a

1. *cone-closed basis* if there is a cone-closed set of monomials B (see Definition 3) whose coefficients in D form a basis of $\mathrm{sp}(D)$.
2. *ℓ-support concentration*, if there is a set of monomials B with support size less than ℓ whose coefficients form a basis of $\mathrm{sp}(D)$.
3. *ℓ-cone concentration*, if there is a set of monomials B with cone size less than ℓ (see Definition 3) whose coefficients form a basis of $\mathrm{sp}(D)$.

In the next lemma, we show that cone-closed basis notion subsumes the other two notions.

▶ **Lemma 11.** *Let $D(\mathbf{x})$ be a polynomial in $\mathbb{F}^k[\mathbf{x}]$. Suppose that $D(\mathbf{x})$ has a cone-closed basis. Then, $D(\mathbf{x})$ has $(k+1)$-cone concentration and $(\lg 2k)$-support concentration.*

Proof. Let B be a cone-closed set of monomials forming the basis of $\mathrm{sp}(D)$. Clearly, $|B| \leq k$. Thus, each $m \in B$ has cone-size $\leq k$. In other words, D is $(k+1)$-cone concentrated.

Moreover, each $m \in B$ has support-size $\leq \lg k$. In other words, D is $(\lg 2k)$-support concentrated. ◀

Next, we define the notions which will be used in the proof of Theorem 2.

Basis & weights. Consider a weight assignment $\mathbf{w} = (w_1, \ldots, w_n) \in \mathbb{N}^n$ on the variables $\mathbf{x} = (x_1, \ldots, x_n)$. It extends to monomials $m = \mathbf{x}^{\mathbf{e}}$ as $\mathbf{w}(m) := \langle \mathbf{e}, \mathbf{w} \rangle = \sum_{i=1}^n e_i w_i$. Sometimes, we also use $\mathbf{w}(\mathbf{e})$ to denote $\mathbf{w}(m)$. Similarly, for a set of monomials B, the weight of B is $\mathbf{w}(B) := \sum_{m \in B} \mathbf{w}(m)$.

Let $B = \{m_1, \ldots, m_\ell\}$ resp. $B' = \{m'_1, \ldots, m'_\ell\}$ be an ordered set of monomials (non-decreasing wrt $\mathbf{w}$) that forms a basis of the span of coefficients of $f \in \mathbb{F}^k[\mathbf{x}]$. Let $\mathbf{w}$ be a weight assignment on the variables. We say that $B < B'$ wrt $\mathbf{w}$, if there exists $i \in [\ell]$ such that $\forall j < i$, $\mathbf{w}(m_j) = \mathbf{w}(m'_j)$ but $\mathbf{w}(m_i) < \mathbf{w}(m'_i)$.

We say that $B \leq B'$ if either $B < B'$ or if $\forall i \in [\ell]$, $\mathbf{w}(m_i) \leq \mathbf{w}(m'_i)$. A basis B is called a *least basis*, if for any other basis B', $B \leq B'$. Next, we describe a condition on $\mathbf{w}$ such that least basis will be unique.

▶ **Definition 12.** (Basis Isolating Weight Assignment [3, Defn.5])**.** A weight assignment $\mathbf{w}$ is called a *basis isolating weight assignment* for a polynomial $f(\mathbf{x}) \in \mathbb{F}^k[\mathbf{x}]$ if there exists a set of monomials B such that:

1. the coefficients of the monomials in B form a basis for $\mathrm{sp}(f)$,
2. weights of all monomials in B are distinct, and
3. the coefficient of every $m \in \mathrm{supp}(f) \setminus B$ is in the linear span of $\{\mathrm{coef}_{m'}(f) \mid m' \in B, \mathbf{w}(m') < \mathbf{w}(m)\}$.

▶ **Lemma 13.** *If $\mathbf{w}$ is a basis isolating weight assignment for $f \in \mathbb{F}^k[\mathbf{x}]$, then f has a unique least basis B wrt $\mathbf{w}$. In particular, for any other basis B' of f, we have $\mathbf{w}(B) < \mathbf{w}(B')$.*

For proof, see the full version linked on the first page. Next, we want to study the effect of shifting f by a basis isolating weight assignment. To do that we require an elaborate notation. As before $f(\mathbf{x})$ is a n-variate and degree-d polynomial over $\mathbb{F}^k$. For a weight assignment $\mathbf{w}$, by $f(\mathbf{x} + t^{\mathbf{w}})$ we denote the polynomial $f(x_1 + t^{w_1}, \ldots, x_n + t^{w_n})$. For $\mathbf{a} = (a_1, \ldots, a_n)$ and $\mathbf{b} = (b_1, \ldots, b_n)$ in $\mathbb{N}^n$, $\binom{\mathbf{a}}{\mathbf{b}}$ denotes $\prod_{i=1}^n \binom{a_i}{b_i}$, where $\binom{a_i}{b_i} = 1$ for $b_i = 0$ and $\binom{a_i}{b_i} = 0$ for $a_i < b_i$. Let $M_{n,d} = \{\mathbf{a} \in \mathbb{N}^n : |\mathbf{a}|_1 \leq d\}$ corresponds to the set of all n-variate d-degree monomials. For every $\mathbf{a} \in M_{n,d}$, $\mathrm{coef}_{\mathbf{x}^{\mathbf{a}}}(f(\mathbf{x} + t^{\mathbf{w}}))$ can be expanded using the binomial expansion, and we get:

$$\sum_{\mathbf{b} \in M_{n,d}} \binom{\mathbf{b}}{\mathbf{a}} \cdot t^{\mathbf{w}(\mathbf{b}) - \mathbf{w}(\mathbf{a})} \cdot \mathrm{coef}_{\mathbf{x}^{\mathbf{b}}}(f(\mathbf{x}))\,. \tag{2}$$

We express this data in matrix form as

$$F' = D^{-1}TD \cdot F, \tag{3}$$

where the matrices involved are,

1. F and F': rows are indexed by the elements of $M_{n,d}$ and columns are indexed by $[k]$. In F resp. F' the $\mathbf{a}$-th row is $\text{coef}_{\mathbf{x}^{\mathbf{a}}}(f(\mathbf{x}))$ resp. $\text{coef}_{\mathbf{x}^{\mathbf{a}}}(f(\mathbf{x}+t^{\mathbf{w}}))$.
2. D: is a diagonal matrix with both the rows and columns indexed by $M_{n,d}$. For $\mathbf{a} \in M_{n,d}$, $D_{\mathbf{a},\mathbf{a}} := t^{\mathbf{w}(\mathbf{x}^{\mathbf{a}})}$.
3. T: both the rows and columns are indexed by $M_{n,d}$. For $\mathbf{a}, \mathbf{b} \in M_{n,d}$, $T_{\mathbf{a},\mathbf{b}} := \binom{\mathbf{b}}{\mathbf{a}}$. It is known as *transfer matrix.*

We will prove the following combinatorial property of T: For any $B \subseteq M_{n,d}$, there is a *cone-closed* $A \subseteq M_{n,d}$ such that the submatrix $T_{A,B}$ has full rank. Our proof is an involved double-induction, so we describe the construction of A as Algorithm 1.

Algorithm 1 Finding cone-closed set

Input: A subset B of the n-tuples M.
Output: A cone-closed $A \subseteq M$ with full rank $T_{A,B}$.
function FIND-CONE-CLOSED(B, n)
  if $n = 1$ **then**
    $s \leftarrow |B|$;
   return $\{0. \dots, s-1\}$;
  else
    Let π_n be the map which projects the set of monomials B on the first $n-1$ variables;
    Let ℓ be the maximum number of preimages under π_n;
    $\forall i \in [\ell]$, F_i collects those elements in $\text{Img}(\pi_n)$ whose preimage size$\geq i$;
    $A_0 \leftarrow \emptyset$;
    for $i \leftarrow 1$ to ℓ **do**
      $S_i \leftarrow$ FIND-CONE-CLOSED$(F_i, n-1)$;
      $A_i \leftarrow A_{i-1} \bigcup \left(S_i \times \{i-1\}\right)$;
    end for
   return A;
  end if
end function

▶ **Lemma 14** (Comparison). *Let B and B' be two nonempty subsets of M such that $B \subseteq B'$. Let $A =$ FIND-CONE-CLOSED(B, n) and $A' =$ FIND-CONE-CLOSED(B', n) in Algorithm 1. Then $A \subseteq A'$.*

▶ **Lemma 15** (Closure). *Let B be a nonempty subset of M. If $A =$ FIND-CONE-CLOSED(B, n) in Algorithm 1, then A is cone-closed. Moreover, $|A| = |B|$.*

For proofs of the above two lemmas, see the full version linked on the first page. Next, we recall a fact that has been used for ROABP PIT.

▶ **Lemma 16.** *[25, Claim 3.3] Let $a_1, \dots, a_n$ be distinct non-negative integers and char $\mathbb{F} = 0$ or greater than the maximum of all a_is. Let A be an $n \times n$ matrix with, $i, j \in [n]$, $A_{i,j} := \binom{a_j}{i-1}$. Then, A is full rank.*

In the following lemma, we prove that the sub-matrix $T_{A,B}$ has full rank, where $B \subseteq M_{n,d}$ and A is the output of Algorithm 1 on input A. It requires char $\mathbb{F} = 0$ or greater than d.

▶ **Lemma 17** (Full rank). *If $A = $ FIND-CONE-CLOSED(B, n), then $T_{A,B}$ has full rank.*

Proof. The proof will be by double-induction– outer induction on n and an inner induction on iteration i of the 'for' loop (Algorithm 1).

Base case: For $n = 1$, the claim is true due to Lemma 16.

Induction step $(n-1 \to n)$*:* To show $T_{A,B}$ full rank, we prove that for any vector $\mathbf{b} \in \mathbb{F}^{|B|}$: if $T_{A,B} \cdot \mathbf{b} = 0$ then $\mathbf{b} = 0$. For this we show that the following invariant holds at the end of each iteration i of the 'for' loop (Algorithm 1). Here, we assume the coordinates of $\mathbf{b}$ are indexed by the elements of B and for all $\mathbf{f} \in B$, $\mathbf{b_f}$ denotes the value of $\mathbf{b}$ at coordinate $\mathbf{f}$.

Invariant (n-variate & i-th iteration): For each $\mathbf{f} \in B$ such that the preimage size of $\pi_n(\mathbf{f})$ is at most i, the product $T_{A_i,B} \cdot \mathbf{b} = 0$ implies that $\mathbf{b_f} = 0$. Here,

At the end of iteration $i = 1$, we have the vector $T_{A_1,B} \cdot \mathbf{b}$. Recall that $A_1 = S_1 \times \{0\}$ and $F_1 = \pi_n(B)$. So $T_{A_1,B} \cdot \mathbf{b} = T_{S_1,F_1} \cdot \mathbf{c}$, where $\mathbf{c} \in \mathbb{F}^{|F_1|}$ and for $\mathbf{e} \in F_1$, $\mathbf{c_e} := \sum_{(\mathbf{e},k) \in \pi_n^{-1}(\mathbf{e})} \binom{k}{0} \mathbf{b}_{(\mathbf{e},k)}$. Thus, $T_{A_1,B} \cdot \mathbf{b} = 0$ implies $T_{S_1,F_1} \cdot \mathbf{c} = 0$. Since $S_1 =$ FIND-CONE-CLOSED$(F_1, n-1)$, using induction hypothesis, we get that $\mathbf{c} = 0$. This means that for $\mathbf{e} \in B$ such that the preimage size of $\pi_n(\mathbf{e})$ is at most 1, we have $\mathbf{c_e} = 0$. This proves our invariant at the end of the iteration $i = 1$.

$(i-1 \to i)$: Suppose that at the end of $(i-1)$-th iteration, the invariant holds. We show that it also holds at the end of the i-th iteration. For each $j \in [i]$, let $\mathbf{v}_j$ denote the projection of $T_{A_i,B} \cdot \mathbf{b}$ on the coordinates indexed by $S_j \times \{j-1\}$. By focusing on the rows of $T_{A_j,B}$, we can see that $\mathbf{v}_j = T_{S_j,F_1} \cdot \mathbf{c}_j$ where the vector $\mathbf{c}_j \in \mathbb{F}^{|F_1|}$ is defined as, for $\mathbf{e} \in F_1$,

$$\mathbf{c}_{j\mathbf{e}} := \sum_{(\mathbf{e},k) \in \pi_n^{-1}(\mathbf{e})} \binom{k}{j-1} \cdot \mathbf{b}_{(\mathbf{e},k)} \,. \tag{4}$$

Suppose that $T_{A_i,B} \cdot \mathbf{b} = 0$. Because of the invariant at $i-1$th round, for all $\mathbf{f} \in B$ with preimage size of $\pi_n(\mathbf{f})$ is less than i, $\mathbf{b_f} = 0$. So all we have to argue is that for every $\mathbf{f} \in B$ such that the preimage size of $\mathbf{e} := \pi_n(\mathbf{f})$ is i, the coordinate $\mathbf{b_f} = 0$.

To prove our goal, first we show that each $\mathbf{c}_j$ is a zero vector. Since $T_{A_i,B} \cdot \mathbf{b} = 0$, its projection $\mathbf{v}_j = T_{S_j,F_1} \cdot \mathbf{c}_j$ is zero too. By induction hypothesis (on $i-1$), for each $\mathbf{e} \in F_1$ with preimage size $< i$, the coordinate $\mathbf{c}_{j\mathbf{e}} = 0$. Thus, the vector $T_{S_j,F_1} \cdot \mathbf{c}_j = T_{S_j,F_j} \cdot \mathbf{c}'_j$ where the vector $\mathbf{c}'_j \in \mathbb{F}^{|F_j|}$ is defined as, for $\mathbf{e} \in F_j$, $\mathbf{c}'_{j\mathbf{e}} := \mathbf{c}_{j\mathbf{e}}$. Consequently, $T_{S_j,F_j} \cdot \mathbf{c}'_j = 0$, for $j \in [i]$. By induction hypothesis (on $n-1$), we know that T_{S_j,F_j} is full rank. So $\mathbf{c}'_j = 0$, which tells us that $\mathbf{c}_j = 0$, for $j \in [i]$.

Fix an $\mathbf{e} \in F_1$, with preimage size $= i$, and let the preimages be $\{(\mathbf{e}, k_1), \ldots, (\mathbf{e}, k_i)\}$ where k_j's are distinct nonnegative integers. From Equation 4, we can write

$$\begin{bmatrix} \mathbf{c}_{1\mathbf{e}} \\ \mathbf{c}_{2\mathbf{e}} \\ \vdots \\ \mathbf{c}_{i\mathbf{e}} \end{bmatrix} = \begin{bmatrix} \binom{k_1}{0} & \binom{k_2}{0} & \cdots & \binom{k_i}{0} \\ \binom{k_1}{1} & \binom{k_2}{1} & \cdots & \binom{k_i}{1} \\ \vdots & \vdots & \cdots & \vdots \\ \binom{k_1}{i-1} & \binom{k_2}{i-1} & \cdots & \binom{k_i}{i-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{b}_{(\mathbf{e},k_1)} \\ \mathbf{b}_{(\mathbf{e},k_2)} \\ \vdots \\ \mathbf{b}_{(\mathbf{e},k_i)} \end{bmatrix}.$$

Since for each $j \in [i]$, $\mathbf{c}_j$ is a zero vector, from the above equation we get

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \begin{bmatrix} \binom{k_1}{0} & \binom{k_2}{0} & \cdots & \binom{k_i}{0} \\ \binom{k_1}{1} & \binom{k_2}{1} & \cdots & \binom{k_i}{1} \\ \vdots & \vdots & \cdots & \vdots \\ \binom{k_1}{i-1} & \binom{k_2}{i-1} & \cdots & \binom{k_i}{i-1} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{b}_{(\mathbf{e},k_1)} \\ \mathbf{b}_{(\mathbf{e},k_2)} \\ \vdots \\ \mathbf{b}_{(\mathbf{e},k_i)} \end{bmatrix}.$$

Now invoking Lemma 16, we get $\mathbf{b}_{(\mathbf{e},k_j)} = 0$ for all $j \in [i]$. In other words, for any $\mathbf{f} \in B$ such that the preimage size of $\pi_n(\mathbf{f})$ is i, the coordinate $\mathbf{b_f} = 0$.

($i = \ell$): Since $A = A_\ell$, the output of FIND-CONE-CLOSED(B, n), using our invariant at the end of ℓ-th iteration we deduce that $T_{A,B} \cdot \mathbf{b} = 0$ implies $\mathbf{b} = 0$. Thus, $T_{A,B}$ has full rank. ◀

Now we are ready to prove our main theorem using the transfer matrix equation.

Proof of Theorem 2. As we mentioned in Equation 2, the shifted polynomial $f(\mathbf{x} + t^{\mathbf{w}})$ yields a matrix equation $F' = D^{-1}TD \cdot F$. Let k' be the rank of F. We consider the following two cases.

Case 1 ($k' < k$)*:* We reduce this case to the other one where $k' = k$. Let S be a subset of k' columns such that $F_{M,S}$ has rank k'. The matrix $F_{M,S}$ denotes the polynomial $f_S(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]^{k'}$, where $f_S(\mathbf{x})$ is the projection of the 'vector' $f(\mathbf{x})$ on the coordinates indexed by S. So, any linear dependence relation among the coefficients of $f(\mathbf{x})$ is also valid for $f_S(\mathbf{x})$. So $\mathbf{w}$ is also a basis isolating weight assignment for $f_S(\mathbf{x})$. Now from our Case 2, we can claim that $f_S(\mathbf{x}+t^{\mathbf{w}})$ has a cone-closed basis A. Thus, coefficients of the monomials, corresponding to A, in $f(\mathbf{x})$ form a basis of $\mathrm{sp}(f)$. This implies that $f(\mathbf{x} + t^{\mathbf{w}})$ has a cone-closed basis A.

Case 2 ($k' = k$)*:* Let B be the least basis of $f(\mathbf{x})$ wrt $\mathbf{w}$ and A = FIND-CONE-CLOSED(B, n). We prove that the coefficients of monomials in A form a basis of the coefficient space of $f(\mathbf{x} + t^{\mathbf{w}})$. To prove this, we show that $\det(F'_{A,[k]}) \neq 0$. Define $T' := TDF$ so that $F' = D^{-1}T'$. Using Cauchy-Binet formula [57], we get that

$$\det(F'_{A,[k]}) \;=\; \sum_{C \in \binom{M}{k}} \det(D^{-1}_{A,C}) \cdot \det(T'_{C,[k]}) \,.$$

Since for all $C \in \binom{M}{k} \setminus \{A\}$, the matrix $D^{-1}_{A,C}$ is singular, we have $\det(F'_{A,[k]}) = \det(D^{-1}_{A,A}) \cdot \det(T'_{A,[k]})$. Again applying Cauchy-Binet formula for $\det(T'_{A,[k]})$, we get

$$\det(F'_{A,[k]}) \;=\; \det(D^{-1}_{A,A}) \cdot \sum_{C \in \binom{M}{k}} t^{\mathbf{w}(C)} \det(T_{A,C}) \cdot \det(F_{C,[k]}) \,.$$

From Lemma 13, we have that for all basis $C \in \binom{M}{k} \setminus \{B\}$, $\mathbf{w}(C) > \mathbf{w}(B)$. The matrix $T_{A,B}$ is nonsingular by Lemma 17, and the other one $F_{B,[k]}$ is nonsingular since B is a basis. Hence, the sum is a nonzero polynomial in t. In particular, $\det(F'_{A,[k]}) \neq 0$, which ensures that the coefficients of the monomials corresponding to A form a basis of $\mathrm{sp}_{\mathbb{F}(t)}(f(\mathbf{x} + t^{\mathbf{w}}))$. Since Lemma 15 says that A is also cone-closed, we get that $f(\mathbf{x} + t^{\mathbf{w}})$ has a cone-closed basis. ◀

3.1 Models with a cone-closed basis

We give a simple proof showing that a typical diagonal depth-3 circuit is already cone-closed. Consider the polynomial $D(\mathbf{x}) = (\mathbf{1} + \mathbf{a}_1 x_1 + \ldots + \mathbf{a}_n x_n)^d$ in $\mathbb{F}^k[\mathbf{x}]$, where $\mathbb{F}^k$ is seen as an $\mathbb{F}$-algebra with coordinate-wise multiplication.

▶ **Lemma 18.** *$D(\mathbf{x})$ has a cone-closed basis.*

Proof. Consider the n-tuple $L := (\mathbf{a}_1, \ldots, \mathbf{a}_n)$. Then for every monomial $\mathbf{x}^{\mathbf{e}}$, the coefficient of $\mathbf{x}^{\mathbf{e}}$ in D is $L^{\mathbf{e}} := \prod_{i=1}^{n} \mathbf{a}_i^{e_i}$, with some nonzero scalar factor (note: here we seem to need char($\mathbb{F}$) zero or large). We ignore this constant factor, since it does not affect linear dependence relations. Consider *deg-lex* monomial ordering, i.e. first order the monomials by lower to higher total degree, then within each degree arrange them according to a lexicographic order. Now we prove that the 'least basis' of $D(\mathbf{x})$ with respect to this monomial ordering is cone-closed.

We incrementally devise a monomial set B as follows: Arrange all the monomials in ascending order. Starting from least monomial, put a monomial in B if its coefficient cannot be written as a linear combination of its previous (thus, smaller) monomials. From construction, the coefficients of monomials in B form the least basis for the coefficient space of $D(\mathbf{x})$. Now we show that B is cone-closed. We prove it by contradiction.

Let $\mathbf{x}^{\mathbf{f}} \in B$ and let $\mathbf{x}^{\mathbf{e}}$ be its submonomial that is not in B. Then we can write

$$L^{\mathbf{e}} = \sum_{\mathbf{x}^{\mathbf{b}} \prec \mathbf{x}^{\mathbf{e}}} c_{\mathbf{b}} L^{\mathbf{b}} \text{ with } c_{\mathbf{b}}\text{'s in } \mathbb{F}\,.$$

Multiplying by $L^{\mathbf{f}-\mathbf{e}}$ on both sides, we get

$$L^{\mathbf{f}} = \sum_{\mathbf{x}^{\mathbf{b}} \prec \mathbf{x}^{\mathbf{e}}} c_{\mathbf{b}} L^{\mathbf{b}+\mathbf{f}-\mathbf{e}} = \sum_{\mathbf{x}^{\mathbf{b}'} \prec \mathbf{x}^{\mathbf{f}}} c'_{\mathbf{b}'} L^{\mathbf{b}'}\,.$$

Note that $\mathbf{x}^{\mathbf{b}'} \prec \mathbf{x}^{\mathbf{f}}$ holds true by the way a monomial ordering is defined. This equation contradicts the fact that $\mathbf{x}^{\mathbf{f}} \in B$, and completes the proof. ◀

4 Conclusion

Since it is known that one could focus solely on the PIT of VP circuits that depend only on the first $o(\log s)$ variables, we initiate a study of properties that are useful in that regime. These properties are– low-cone concentration and cone-closed basis. Their usefulness is proved in our monomial counting and coefficient extraction results. Using these concepts we solve an interesting special case of diagonal depth-3 circuits.

An open question is to make our approach work for field characteristic smaller than the degree. Another interesting problem is to employ the cone-closed basis properties of the $\Sigma \wedge \Sigma^n$ model to devise a poly-time blackbox PIT for general n.

In our second result, we proved that after shifting the variables by a basis isolating weight assignment, a polynomial has a cone-closed basis. Basis isolating weight assignment is much weaker than the one induced by lexicographic monomial ordering (or the Kronecker map). An interesting open question is to *efficiently* design a weight assignment (or, in general, polynomial map) that ensures a cone closed basis. Till now, no known blackbox PIT algorithm for ROABPs gives a polynomial time blackbox PIT algorithm for log (or sub-log) variate ROABPs. So, achieving cone-closed basis or low-cone concentration property (in polynomial time) for log (or sub-log) variate ROABPs is also interesting; then, the counting & extraction techniques developed in our first result will give a polynomial time blackbox PIT. This will solve some open problems posed in [2, Sec.6].

Acknowledgements M.F. & N.S. are grateful to the organizers of algebraic complexity workshops in 2014 (MPI Saarbrücken & TIFR Mumbai) that initiated the early discussions. N.S. thanks Manindra Agrawal for useful discussions. N.S. thanks the funding support from DST (DST/SJF/MSA-01/2013-14).

References

1 Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *FSTTCS 2005: Foundations of Software Technology and Theoretical Computer Science, 25th International Conference, Hyderabad, India, December 15-18, 2005, Proceedings*, pages 92–105, 2005.

2 Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. Technical report, https://www.cse.iitk.ac.in/users/nitin/research.html, 2017. (To appear in 50th ACM Symposium on Theory of Computing (STOC), 2018).

3 Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM Journal on Computing*, 44(3):669–697, 2015.

4 Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004.

5 Manindra Agrawal, Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. Jacobian hits circuits: hitting-sets, lower bounds for depth-d occur-k formulas & depth-3 transcendence degree-k circuits. In *STOC*, pages 599–614, 2012.

6 Manindra Agrawal, Chandan Saha, and Nitin Saxena. Quasi-polynomial hitting-set for set-depth-Δ formulas. In *Symposium on Theory of Computing Conference, STOC'13, Palo Alto, CA, USA, June 1-4, 2013*, pages 321–330, 2013.

7 M. Beecken, J. Mittmann, and N. Saxena. Algebraic Independence and Blackbox Identity Testing. *Inf. Comput.*, 222:2–19, 2013. (Conference version in ICALP 2011).

8 Michael Ben-Or and Prasoon Tiwari. A deterministic algorithm for sparse multivariate polynomial interpolation. In *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, STOC '88, pages 301–309, 1988.

9 Xi Chen, Neeraj Kayal, and Avi Wigderson. Partial derivatives in arithmetic complexity and beyond. *Foundations and Trends in Theoretical Computer Science*, 6(1-2):1–138, 2011.

10 Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193 – 195, 1978.

11 Zeev Dvir, Rafael Mendes de Oliveira, and Amir Shpilka. Testing Equivalence of Polynomials under Shifts. In *Proceedings of the 41st International Colloquium on Automata, Languages, and Programming, Part I*, volume 8572 of *Lecture Notes in Computer Science*, pages 417–428. Springer International Publishing, 2014. doi:10.1007/978-3-662-43948-7_35.

12 Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Bipartite perfect matching is in quasi-NC. In *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2016, Cambridge, MA, USA, June 18-21, 2016*, pages 754–763, 2016.

13 Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Guest column: Parallel algorithms for perfect matching. *SIGACT News*, 48(1):102–109, 2017.

14 Michael A. Forbes. *Polynomial Identity Testing of Read-Once Oblivious Algebraic Branching Programs*. PhD thesis, Massachusetts Institute of Technology, 2014.

15 Michael A Forbes. Deterministic divisibility testing via shifted partial derivatives. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pages 451–465. IEEE, 2015.

16 Michael A. Forbes, Ankit Gupta, and Amir Shpilka. private communication, 2013.

17 Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Pseudorandomness for multilinear read-once algebraic branching programs, in any order. *Electronic Colloquium on Computational Complexity (ECCC)*, 20:132, 2013.

18 Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Symposium on Theory of Computing (STOC), New York, NY, USA, May 31 - June 03, 2014*, pages 867–875, 2014.

19 Michael A. Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In *STOC*, pages 163–172, 2012.

20 Michael A Forbes and Amir Shpilka. Explicit noether normalization for simultaneous conjugation via polynomial identity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 527–542. Springer, 2013.

21 Ignacio García-Marco, Pascal Koiran, Timothée Pecatte, and Stéphan Thomassé. On the complexity of partial derivatives. In *34th Symposium on Theoretical Aspects of Computer Science, STACS 2017, March 8-11, 2017, Hannover, Germany*, pages 37:1–37:13, 2017.

22 Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *IEEE 57th Annual Symposium on Foundations of Computer Science, FOCS 2016, 9-11 October 2016*, pages 109–117, 2016.

23 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth three. In *54th Annual IEEE Symposium on Foundations of Computer Science, FOCS 2013, 26-29 October, 2013, Berkeley, CA, USA*, pages 578–587, 2013.

24 Ankit Gupta, Pritish Kamath, Neeraj Kayal, and Ramprasad Saptharishi. Arithmetic circuits: A chasm at depth 3. *SIAM Journal on Computing*, 45(3):1064–1079, 2016.

25 Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Identity testing for constant-width, and any-order, read-once oblivious arithmetic branching programs. *Theory of Computing*, 13(2):1–21, 2017. (Preliminary version in CCC'16).

26 Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Computational Complexity*, pages 1–46, 2016. (Conference version in CCC 2015).

27 Rohit Gurjar and Thomas Thierauf. Linear matroid intersection is in quasi-nc. In *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, STOC 2017, Montreal, QC, Canada, June 19-23, 2017*, pages 821–830, 2017.

28 Rohit Gurjar, Thomas Thierauf, and Nisheeth K. Vishnoi. Isolating a vertex via lattices: Polytopes with totally unimodular faces. *CoRR*, abs/1708.02222, 2017.

29 Joos Heintz and Claus-Peter Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the 12th Annual ACM Symposium on Theory of Computing, April 28-30, 1980, Los Angeles, California, USA*, pages 262–272, 1980.

30 Stasys Jukna. *Extremal Combinatorics: With Applications in Computer Science*. Springer Publishing Company, Incorporated, 1st edition, 2010.

31 Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pages 355–364, 2003.

32 Neeraj Kayal. Algorithms for arithmetic circuits. *Electronic Colloquium on Computational Complexity (ECCC)*, 17:73, 2010.

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

34 Adam R. Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings on 33rd Annual ACM Symposium on Theory of Computing, July 6-8, 2001, Heraklion, Crete, Greece*, pages 216–223, 2001.

35 Swastik Kopparty, Shubhangi Saraf, and Amir Shpilka. Equivalence of polynomial identity testing and deterministic multivariate polynomial factorization. In *IEEE 29th Conference on Computational Complexity, CCC 2014, Vancouver, BC, Canada, June 11-13, 2014*, pages 169–180, 2014.

36 Mrinal Kumar and Shubhangi Saraf. Arithmetic circuits with locally low algebraic rank. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 34:1–34:27, 2016.

37 Mrinal Kumar and Shubhangi Saraf. Sums of products of polynomials in few variables: Lower bounds and polynomial identity testing. In *31st Conference on Computational Complexity, CCC 2016, May 29 to June 1, 2016, Tokyo, Japan*, pages 35:1–35:29, 2016.

38 Guillaume Lagarde, Guillaume Malod, and Sylvain Perifel. Non-commutative computations: lower bounds and polynomial identity testing. *Electronic Colloquium on Computational Complexity (ECCC)*, 23:94, 2016.

39 Richard J. Lipton and Nisheeth K. Vishnoi. Deterministic identity testing for multivariate polynomials. In *Proceedings of the Fourteenth Annual ACM-SIAM Symposium on Discrete Algorithms, January 12-14, 2003, Baltimore, Maryland, USA.*, pages 756–760, 2003.

40 Ketan Mulmuley. Geometric complexity theory V: Efficient algorithms for Noether normalization. *Journal of the American Mathematical Society*, 30(1):225–309, 2017.

41 Ketan Mulmuley, Umesh V. Vazirani, and Vijay V. Vazirani. Matching is as easy as matrix inversion. In *Proceedings of the Nineteenth Annual ACM Symposium on Theory of Computing*, STOC '87, pages 345–354, 1987.

42 Ketan D Mulmuley. The GCT program toward the P vs. NP problem. *Communications of the ACM*, 55(6):98–107, 2012.

43 Ketan D. Mulmuley. Geometric complexity theory V: Equivalence between blackbox derandomization of polynomial identity testing and derandomization of Noether's normalization lemma. In *FOCS*, pages 629–638, 2012.

44 Noam Nisan and Avi Wigderson. Lower bounds for arithmetic circuits via partial derivatives (preliminary version). In *36th Annual Symposium on Foundations of Computer Science, Milwaukee, Wisconsin, 23-25 October 1995*, pages 16–25, 1995.

45 Anurag Pandey, Nitin Saxena, and Amit Sinhababu. Algebraic independence over positive characteristic: New criterion and applications to locally low algebraic rank circuits. In *41st International Symposium on Mathematical Foundations of Computer Science, MFCS 2016, August 22-26, 2016 - Kraków, Poland*, pages 74:1–74:15, 2016. (In print, Computational Complexity, 2018).

46 Chandan Saha, Ramprasad Saptharishi, and Nitin Saxena. A case of depth-3 identity testing, sparse factorization and duality. *Computational Complexity*, 22(1):39–69, 2013.

47 Ramprasad Saptharishi. personal communication, 2013.

48 Ramprasad Saptharishi. *Unified Approaches to Polynomial Identity Testing and Lower Bounds*. PhD thesis, Chennai Mathematical Institute, 2013.

49 Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Technical report, https://github.com/dasarpmar/lowerbounds-survey/, 2016.

50 Nitin Saxena. Diagonal circuit identity testing and lower bounds. In *ICALP*, volume 5125 of *Lecture Notes in Computer Science*, pages 60–71. Springer, 2008.

51 Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009.

52 Nitin Saxena. Progress on polynomial identity testing- II. In *Perspectives in Computational Complexity*, volume 26 of *Progress in Computer Science and Applied Logic*, pages 131–146. Springer International Publishing, 2014.

53 Nitin Saxena and C. Seshadhri. Blackbox identity testing for bounded top-fanin depth-3 circuits: The field doesn't matter. *SIAM Journal on Computing*, 41(5):1285–1298, 2012.

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

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

56 Avi Wigderson. Low-depth arithmetic circuits: technical perspective. *Communications of the ACM*, 60(6):91–92, 2017.

57 Jiang Zeng. A bijective proof of Muir's identity and the Cauchy-Binet formula. *Linear Algebra and its Applications*, 184:79–82, 1993.

58 Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposium on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226, 1979.