Poly-time blackbox identity testing for sum of log-variate constant-width ROABPs

Pranav Bisht * Nitin Saxena †

Abstract

Blackbox polynomial identity testing (PIT) affords 'extreme variable-bootstrapping' (Agrawal et al, STOC'18; PNAS'19; Guo et al, FOCS'19). This motivates us to study *log*-variate read-once oblivious algebraic branching programs (ROABP). We restrict width of ROABP to a constant and study the more general sum-of-ROABPs model. We give the first poly(s)-time blackbox PIT for sum of *constant*-many, size-s, $O(\log s)$-variate constant-width ROABPs. The previous best for this model was *quasi*-polynomial time (Gurjar et al, CCC'15; CC'16) which is comparable to brute-force in the log-variate setting. Also, we handle unbounded-many such ROABPs if each ROABP computes a homogeneous polynomial.

Our new techniques comprise– (1) an ROABP computing a homogeneous polynomial can be made *syntactically* homogeneous in the same width; and (2) overcome the hurdle of *unknown* variable order in sum-of-ROABPs in the log-variate setting (over *any* field).

2012 ACM CCS concept: Theory of computation– Algebraic complexity theory, Fixed parameter tractability, Pseudorandomness and derandomization; Computing methodologies– Algebraic algorithms; Mathematics of computing– Combinatoric problems.
Keywords: identity test, hitting-set, ROABP, blackbox, log variate, width, diagonal, derandomization, homogeneous, sparsity.

1 Introduction

Polynomial Identity Testing (PIT) is the problem of testing whether a given multivariate polynomial is identically zero or not. The input polynomial to be tested is usually given in a compact representation– like an *algebraic circuit* or an *algebraic branching program* (ABP). The PIT algorithm is said to be efficient if its time complexity is polynomial in the input size of algebraic circuit resp. ABP. There are two main types of PIT algorithms– *blackbox* or *whitebox*. A blackbox PIT algorithm tests the zeroness of input polynomial using only evaluations of circuit, resp. ABP, over field points. However, a whitebox algorithm is allowed additional access to look 'inside' the circuit or ABP. The set of points $\mathcal{H}$ over which a blackbox PIT algorithm evaluates is also commonly known as a *hitting-set*. PIT admits a simple yet efficient randomized blackbox algorithm due to *Polynomial Identity Lemma* [Sch80, Zip79, DL78]. The primary focus of research in PIT is to derandomize it and get a poly-time deterministic blackbox algorithm. The problem of PIT also has interesting connections with circuit lower bounds [HS80, KI04, Agr05, AGS19] and many other well known problems like matching [MVV87, FGT17], primality testing [AKS04], polynomial factoring[KSS14] and polynomial equivalence [DdOS14]. Refer to [SY10, Sax09, Sax14, Sap16] for detailed surveys on PIT and lower bounds.

The model in focus for this paper is that of read-once oblivious ABPs (ROABPs) which is a popular special class of ABPs. An *ABP* is defined using a layered directed graph with a

*Department of Computer Science & Engineering, IIT Kanpur, India, pbisht@cse.iitk.ac.in
†CSE, IIT Kanpur, nitin@cse.iitk.ac.in

unique *source* and *sink* vertex. The graph has edges only among consecutive layers. Each edge is directed from one layer to the next and has some *linear* polynomial as its weight. The weight of a path is product of edge weights along the path. The polynomial computed by the ABP is then simply the sum of all weighted paths from source to sink. The length of an ABP is the length of the longest path from source to sink and *width* of an ABP is the maximum possible number of vertices in a layer. An ABP is called *read-once oblivious* (ROABP) if each variable is read in only one layer and each edge in a layer is labelled with a univariate (of arbitrary degree) in its corresponding variable. Equivalently, an ROABP of width w can be viewed as a product of n matrices $f(\mathbf{x}) = D_1(x_{\pi(1)}) \cdot D_2(x_{\pi(2)}) \dots D_n(x_{\pi(n)})$ where $D_1(x_{\pi(1)}) \in \mathbb{F}^{1\times w}[x_{\pi(1)}]$, $D_i(x_{\pi(i)}) \in \mathbb{F}^{w\times w}[x_{\pi(i)}]$ for $2 \leq i \leq n-1$ and $D_n(x_{\pi(n)}) \in \mathbb{F}^{w\times 1}[x_{\pi(n)}]$. Here, π is a permutation on the set $\{1, 2, \dots, n\}$ and it describes what we call the *variable order* of an ROABP.

In the whitebox regime, ROABP has a famous poly-time PIT algorithm [RS05]. However, in the blackbox regime, we only have quasi-poly-time PIT algorithms [FS13b, FSS14, AGKS15] and no known poly-time algorithms. The recent (variable-)*Bootstrapping* results [AGS19, KST19, GKSS19] tell us that it suffices to focus on PIT for extremely low-variate algebraic circuits. In the new light, one can ask for a poly-time blackbox PIT for *log*-variate ROABPs. Even this question is open. One can also ask for blackbox PIT of ROABPs with restriction on the width parameter. In [GKS17] they address this question and give a poly-time blackbox PIT for constant width ROABPs. However their algorithm works only for *known* variable order and for fields of characteristic either zero or sufficiently large. The combined restrictions on variables and width still gives interesting sub-models. Eg. [AGS19, Thm.22] shows that even solving PIT for log-variate width-2 *ABPs* will almost solve the complete PIT problem.

The sum of ROABPs model was also studied by researchers in PIT. For a constant number of ROABPs, [GKST16] give the first poly-time whitebox, and only a *quasi*-poly time blackbox PIT algorithm. One can then ask for *poly-time* blackbox PIT for sum of ROABPs under the restriction of constant width. This problem is also open. What if we also restrict the number of variables? It is a nontrivial model as the degree remains arbitrary. This brings us to the question of poly-time blackbox PIT for sum of constantly-many, constant-width, log-variate ROABPs. We answer this question in the affirmative.

Blackbox PIT for sum of constantly-many, log-variate constant-width ROABPs is in poly-time.

Furthermore, *if these ROABPs compute homogeneous polynomials then we can drop the 'constantly-many' restriction.*

An ABP is the algebraic analog of *boolean branching program* (BP). Barrington's Theorem [Bar89] proves that the complexity class NC^1 is captured by width-5 branching programs of polynomial size. An ROABP is the algebraic analog of a special branching program called oblivious binary decision diagram (*OBDD*); also called read-once ordered branching program (*ROBP*). The problem of PIT, or finding hitting-sets, for ROABPs is the algebraic analog of finding *pseudorandom generators* (PRG) for OBDD which is a well-studied open problem [Nis92, INW94, RR99, IMZ12, BCG18, FK18, HZ18]; being related to the RL=L question. Blackbox PIT for sum of ROABPs can be seen as the algebraic analog of PRG for XOR of OBDDs [GKST16, Sec.2.4]. See [SW97, GM96, BW98] for PRGs of XOR of OBDDs. In the *log*-variate boolean setting, PRG for XOR of OBDDs is trivial, since one can simply try all 2^n fixings of the boolean function $f : \{0,1\}^n \to \{0,1\}$. In the algebraic setting however, a polynomial can have arbitrary individual degree, thus making PIT a highly nontrivial problem.

1.1 Our results

The sum of 'few' constant-width ROABPs model is more expressive than that of a constant-width ROABP. Even a sum of two width-3 ROABPs cannot be computed by a single constant-width

ROABP as pointed out in Fact 1.

Theorem 1 (Sum of ROABPs)**.** *Let $\mathcal{P}$ be a set of n-variate polynomials, over a field $\mathbb{F}$, computed by a sum of c-many ROABPs, each of width-r and size-s. (The variable order of each ROABP is unknown.) Then, blackbox PIT for $\mathcal{P}$ can be solved in poly(s^c, r^{n3^c}) time.*

Remarks– 1) If $c, r = O(1)$ and $n = O(\log s)$, then the stated time-complexity is poly(s). Alternatively, for super-constant width $r = s^{o(1)}$, the complexity remains poly(s)-time up to $n = \log_r s = \omega(1)$.

2) For log-variate, the prior best complexity was $> s^n = s^{\Omega(\log s)}$, which is *super*-polynomial in the input-size. It is achieved by *brute-force* based on the Polynomial Identity Lemma.

3) [GKST16] gave a non-trivial blackbox PIT, in the general n-variate regime, for sum of constantly-many ROABPs. Their complexity is $> s^{c2^c \log s}$, which is *super*-polynomial time; even under the restrictions of constant-c, constant-width and log-variate.

4) [GKS17] gave a poly-time blackbox PIT for $c = 1$: a single constant-width ROABP without the restriction of log-variate. However, their algorithm assumes the knowledge of variable order. Moreover, it works only for fields of zero/large characteristic. However, our algorithm is efficient, in the log-variate setting, and neither requires the knowledge of variable order nor special characteristic.

In Theorem 1, c, the number of ROABPs, is assumed to be constant to get efficient blackbox PIT. We could allow an arbitrary c, if each ROABP computes a homogeneous polynomial.

Theorem 2 (Sum of Homog. ROABPs)**.** *Let $\mathcal{P}$ be a set of n-variate polynomials, over a field $\mathbb{F}$, computed by a sum of c-many ROABPs, each of width-r and size-s, each computing a* homogeneous *polynomial. (The variable order of each ROABP is unknown.) Then, blackbox PIT for $\mathcal{P}$ can be solved in poly(cr^n, s) time.*

Remarks– 1) If $r = O(1)$ and $n = O(\log s)$, then the stated time complexity is poly(cs)– polynomial in the input-size.

2) Without the restriction of homogeneity, the model subsumes the diagonal depth-3 model (note: poly-time blackbox PIT for $\sum\bigwedge\sum$ is open). This can be seen using the duality-trick of [Sax08], together-with [FSS14] variable-reduction map, which will convert a diagonal depth-3 model into sum of width-1 log-variate ROABPs (& unbounded c). See Appendix A for details.

[AFS+16] defined a model called k-pass ABP. An ABP is called an *oblivious* ABP if for every layer, edge labels are univariate polynomials in only a single variable. An oblivious ABP is said to be a k-pass ABP if there exists a permutation π on $\{1, 2, \ldots, n\}$ such that the ABP reads each variable k-times in the order: $x_{\pi(1)}, \ldots, x_{\pi(n)}, x_{\pi(1)}, \ldots, x_{\pi(n)}, \ldots, x_{\pi(1)}, \ldots, x_{\pi(n)}$.

Corollary 3 (k-pass ABP)**.** *Let $\mathcal{P}$ be a set of n-variate polynomials, over a field $\mathbb{F}$, computed by a sum of c, k-pass ABPs, each of width-r and size-s. Moreover, variable order of each k-pass ABP is unknown. Then, blackbox PIT for $\mathcal{P}$ can be solved in poly(s^c, r^{kn3^c}) time.*

Remark. If $c, r, k = O(1)$ and $n = O(\log s)$, then the stated time complexity is poly(s).

1.2 Previous works and motivation

ABPs: It is well known that ABPs subsume algebraic formulas. In turn, algebraic formulas subsume constant depth circuits (see [BOC92] and [Nis91, Lem.1]). Surprisingly, it was shown by a series of work that PIT for general circuits reduces to PIT for depth-3 circuits ($\sum\prod\sum$) [Bre74, VSBR83, AV08, Koi12, Tav15, GKKS16] (See Appendix B for definitions). In this paper we shall work with a special case of ABPs, called ROABPs. This is a natural first model to attack before solving the more general ABP model.

A single ROABP model is also highly non-trivial. In fact, PIT for log-variate ROABP subsumes PIT for the diagonal depth-3 model (See Appendix A). [RS05] gave the first poly-time whitebox PIT for non-commutative ABPs which also works for ROABPs. [FS13b] gave the first quasi-polynomial time blackbox PIT for ROABPs with known variable order. [AGKS15] gave a quasi-polynomial time blackbox PIT for general unknown order ROABPs. [GKST16] studied the sum of ROABPs model, which is much more general. In fact, it can be shown that it subsumes the multilinear depth-3 model, for which a poly-time blackbox PIT is still open [AGKS15, GKST16].

Log-variate: There has been a recent line of work on 'Bootstrapping variables' in algebraic circuits. [AGS19] prove that solving blackbox PIT for circuits that depend only on the first $\log^{\circ c} s$ variables is sufficient to solve blackbox PIT for general circuits. Here c is a constant and $\log^{\circ c}$ is a composition of c logarithms. [KST19, GKSS19] further strengthened the results to ultimately show that even saving on *one* evaluation point from the brute-force hitting-set of constant-variate algebraic circuits would solve general PIT.

The well known diagonal depth-3 model is one of the lower hanging fruits in PIT. [FGS18] were able to separate the variable parameter from size, to give the first poly-time blackbox PIT for log-variate diagonal depth-3 circuits. The natural extension is to solve blackbox PIT for log-variate ROABPs. In fact, it can be shown that PIT for log-variate *commutative* ROABPs implies PIT for diagonal depth-3 model using the results of [FS13a, FSS14]. See Appendix A for details.

Constant width: The sum of constant-width ROABPs model is also quite non-trivial as evident from Fact 1. In part 2 of [KNS16, Thm.7], they construct g using an explicit family of 3-regular expander graphs such that g is computed by a multilinear depth-3 circuit with top fan-in *two* but requires a single ROABP of $2^{\Omega(n)}$ width to compute it. Careful inspection of the proof tells us: g can also be computed by a sum of just two width-3 ROABPs of size $\Theta(n)$. Even in the log-variate setting, g will require a single ROABP of super-constant width; which earlier lacked a poly-time blackbox PIT. We go around this fundamental obstruction in Theorem 1.

Fact 1. [KNS16, Thm.7] *There is an explicit family of $3n$-variate multilinear polynomials $\{g_n\}_{n\geq 1}$ which is computable by sum of two width-3 ROABPs of size $\Theta(n)$, but any single ROABP computing g must have width $2^{\Omega(n)}$.*

PIT for various models: Researchers have found poly-time or quasi-poly-time blackbox PIT algorithms for a variety of other restricted models. For example, Sparse PIT ($\sum\prod$ or depth-2) [BOT88, KS01, AB03], special depth-3 circuits [DS07, KS07, Sax08, KS09, KS11, SS11, SS12, dOSV16, FGS18], special depth-4 circuits [ASSS12, BMS13, SSS13, For15, KS16a, KS16b, PSS16], set-multilinear circuits [RS05, FS12, ASS13], ROABP related models [JQS10, FS12, FSS14, AFS$^+$16], and certain non-commutative models [GGOW16, LMP16].

Efficient PRGs have also been constructed for several special OBDDs– constant-width, regular, permutation [BRRY14, BDVY13, Ste12, De11, KNP11, BV10].

PIT also has implications in Geometric Complexity Theory (GCT). Refer to [Mul12b, Mul12a] for details. Mulmuley shows that solving blackbox PIT for diagonal depth-3 model is equivalent to derandomization of Noether's Normalization Lemma for certain explicit algebraic varieties. The concept of *Waring rank*, which has well-known connections in complexity theory, is also closely tied with homogeneous diagonal depth-3 model [IK99, Lan17, CHI$^+$18].

1.3 Proof techniques

Syntactic homogeneity for ROABP: Inspired by circuits we define *syntactic homogeneity* for ROABP (Definition 6). We prove that if a degree-d homogeneous polynomial has an ROABP

of width-r, then it also has a syntactic homogeneous ROABP of the *same* width, and in the same variable order (Theorem 7). Recall that if one applies the usual *homogenization* trick, for circuits/ABP [SY10, Thm.2.2], then the ROABP width blows up to $O(rd^2)$; making the width non-constant! Our new technique helps solve blackbox PIT for a constant-width log-variate ROABP; but also seems independently interesting.

Going beyond log-support concentration: Say, we consider sum of two constant-width log-variate ROABPs. The blackbox PIT algorithm of [GKST16] for sum of two ROABPs uses log-support concentration. They essentially prove that if a variable-shift log-support-concentrates a single ROABP, then it also log-support-concentrates sum of two ROABPs. However, in our log-variate setting, log-support concentration is a triviality and does not help in devising a nontrivial blackbox PIT. Instead, we develop a technique to directly use the hitting-set for a log-variate ROABP to get a hitting-set for sum of two such ROABPs (Section 4).

1.4 Proof ideas

Proof idea of Theorem 2: Let f be a d-degree, $n = O(\log s)$-variate polynomial computed by an ROABP of constant width r and size s. Let $f^{[d]}$ be the degree-d homogeneous part of f. We show that $f^{[d]}$ also has an ROABP of width $\leq r$ (Lemma 19). In our Structure Theorem 7, we show that $f^{[d]}$ can also be computed by a syntactically homogeneous ROABP of width $\leq r$. It provides us with a sparsity upper bound of r^n for $f^{[d]}$ (Lemma 8). Now, we can use sparse PIT map to get a poly-time blackbox PIT for f (Lemma 9). In Theorem 2, we are dealing with sum of ROABPs, where each ROABP computes a homogeneous polynomial. Thus, each ROABP computes a sparse polynomial and therefore the total sparsity of the sum is also polynomially bounded. This gives us the required poly-time blackbox PIT (proof in Section 3).

Proof idea of Theorem 1: Let us discuss the simpler case of blackbox PIT of $A + B$, where A resp. B is computed by an n-variate ROABP of width-r, in unknown (& different) variable order. This sketch can be extended to blackbox PIT for sum of c ROABPs (Section 4.2).

The PIT algorithm in [GKST16] for sum of two ROABPs uses the *dependency equations* from A (Definition 5). Suppose the variable order of A is $x_{\pi(1)} < \ldots < x_{\pi(n)}$. Their algorithm iteratively 'builds' the ROABP for B in this variable order of A. If $A \neq -B$, then (essentially) we will encounter the first variable, say $x_{\pi(k)}$, where B does not follow some dependency equation of A. This acts as a *certificate* of non-zeroness of $A + B$. This process can be done in poly-time in the whitebox setting.

In blackbox setting, the variable orders of both ROABPs are unknown and thus this idea cannot be implemented as it is. [GKST16] took the log-support-concentration route for their blackbox algorithm; which takes quasi-polynomial time. It is as bad as the brute-force complexity in our log-variate setting; so we need a new idea.

If $A \neq -B$, we know that the certificate of B not satisfying the dependency equations exists. Due to the ample 2^n-time available in log-variate setting, we essentially search for this certificate in poly-time by going over all prefixes $x_{\pi(1)} < \ldots < x_{\pi(k)}$ and checking the satisfiability of the dependency equations. Really going over all prefixes would take $n! > \omega(2^n)$ time; so instead we only go over *all* $(k-1)$*-size subsets* of $[n]$. This is because we are going to apply a sparse PIT map on the prefix variables (Claim 10) and hence the order within the prefix-subset does not matter.

We intend to map the prefix-variables to univariates in t_1 (Claim 10); and map the suffix-variables to a univariate in t_2 (Claim 11). Thus, reducing the variables to *three* (Lemma 12). During the variable-reduction, the violation of dependency equations gets preserved as our maps are based on the blackbox PIT for a single ROABP of width $O(r^3)$. The latter is indeed available to us by Lemma 9. The details and the proof are in Section 4.

The proof of Corollary 3 is an easy consequence of Theorem 1; since [AFS+16] showed that a constant-pass ABP of constant-width can be converted into a constant-width ROABP.

2 Notations and Preliminaries

2.1 Notations

We follow some of the notations from [GKST16]. Let $[n]$ denote the set $\{1, 2, \ldots, n\}$. Let $\mathbf{x} = (x_1, x_2, \ldots, x_n)$ be a tuple of n-variables. $\mathbf{x}_k$ or $\mathbf{x}_{\leq k}$ will denote the tuple of first k variables $(x_1, x_2, \ldots, x_k)$ and $\mathbf{x}_{>k}$ will denote the tuple of remaining variables $(x_{k+1}, x_{k+2}, \ldots, x_n)$. Let π denote the *variable order* of an ROABP, where $\pi : [n] \to [n]$ is some permutation. This means the variables are read in the order $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$. Let $\mathbb{F}[\mathbf{x}]$ denote the ring of polynomials in n-variables over some field $\mathbb{F}$. Let $\mathbb{F}^{w\times w}[\mathbf{x}]$ denote the ring of polynomials in n-variables over the matrix algebra of $w \times w$ matrices.

Let $A(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be a polynomial in n variables of degree d. Let $\mathbf{a}$ denote an exponent vector $(a_1, a_2, \ldots, a_n) \in \mathbb{N}^n$ such that $\mathbf{x}^{\mathbf{a}}$ denotes the monomial $\prod_{i=1}^{n} x_i^{a_i}$. Let $\text{coeff}(A)(\mathbf{x}^{\mathbf{a}}) \in \mathbb{F}$ denote the coefficient of the monomial $\mathbf{x}^{\mathbf{a}}$ in $A(\mathbf{x})$. The *sparsity* of a polynomial $A(\mathbf{x})$– sparsity(A) –is defined as the number of monomials with non-zero coefficients in A. We use $A^{[d]}$ to denote the degree-d *homogeneous part* of $A(\mathbf{x})$ and $A^{[<d]}$ to denote the remaining lower-degree terms.

Let $\mathbf{y}$ and $\mathbf{z}$ be a *partition* of $\mathbf{x}$ such that $|\mathbf{y}| = k$, then the *coefficient polynomial* $A_{(\mathbf{y},\mathbf{a})}$, also known as partial-derivative polynomial, denotes the coefficient of monomial $\mathbf{y}^{\mathbf{a}}$ in $A(\mathbf{x})$ which is a polynomial in $\mathbb{F}[\mathbf{z}]$. Similarly $A_{(\mathbf{z},\mathbf{b})} \in \mathbb{F}[\mathbf{y}]$ is the coefficient of monomial $\mathbf{z}^{\mathbf{b}}$ in $A(\mathbf{x})$.

Warning: $A_{(\mathbf{x},\mathbf{a})}$ and $\text{coeff}(A)(\mathbf{x}^{\mathbf{a}})$ are different. For example if $A(\mathbf{x}) = x_1x_2 + x_2^2 + 2x_1$, then $A_{(x_1,1)} = x_2 + 2$ while $\text{coeff}(A)(x_1) = 2$. Note that $A(\mathbf{x})$ can be expressed in multiple ways:

$$A(\mathbf{x}) = \sum_{\mathbf{a}\in\{0,1,\ldots,d\}^n} \text{coeff}(A)(\mathbf{x}^{\mathbf{a}}) \cdot \mathbf{x}^{\mathbf{a}}, \quad \text{and also,}$$

$$A(\mathbf{x}) = \sum_{\mathbf{a}\in\{0,1,\ldots,d\}^k} A_{(\mathbf{y},\mathbf{a})} \cdot \mathbf{y}^{\mathbf{a}}$$

A polynomial $A(\mathbf{x}) \in \mathbb{F}^{w\times w}[\mathbf{x}]$ is called a *matrix polynomial*, where the coefficients are $w \times w$ matrices of field constants. The *coeff.space* of $A(\mathbf{x})$ is defined as the span of all the coefficients of A: $\text{span}_{\mathbb{F}}\{\text{coeff}(A)(\mathbf{x}^{\mathbf{a}}) \mid \mathbf{a} \in \{0, 1, \ldots, d\}^n\}$. We can also define it for any prefix of variables.

For a set of polynomials $\mathcal{P}$, their $\mathbb{F}$-span is defined as: $\text{span}_{\mathbb{F}}\mathcal{P} := \left\{ \sum_{A\in\mathcal{P}} \alpha_A.A \mid \alpha_A \in \mathbb{F} \right\}$. The set $\mathcal{P}$ is called $\mathbb{F}$-linearly independent if $\sum_{A\in\mathcal{P}} \alpha_A.A = 0$ implies $\alpha_A = 0$ for all $A \in \mathcal{P}$. $\dim_{\mathbb{F}}\mathcal{P}$ is then defined as cardinality of the largest $\mathbb{F}$-linearly independent subset of $\mathcal{P}$.

2.2 Nisan's characterization

Algebraic circuit, ABP, ROABP and hitting-set map (or blackbox PIT) are formally defined in Appendix B. Also, refer to [For14, Gur15, Kor16] for excellent surveys on ROABPs.

Let $A(\mathbf{x})$ be a polynomial of degree d computed by an ROABP. Let the variable order be $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$. Let $\mathbf{y}$ and $\mathbf{z}$ be a partition of the variables such that $\mathbf{y}$ is a k length *prefix* of the variable order, that is, $y_i = x_{\pi(i)}$ for $i \in [k]$ and z is the remaining suffix. Nisan [Nis91] gave an exact width characterization for ROABPs. We will mostly follow the presentation of [GKST16] for this characterization; as it is a constructive proof.

Lemma 4. [GKST16, Lemmas 2.2, 2.5] *Let $A(\mathbf{x})$ be a polynomial of individual degree d, computed by an ROABP of width w with variable order $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$. For $k \in [n]$, $\mathbf{y} = (x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(k)})$ be the prefix of length k; and $\mathbf{z}$ be the suffix of length $n - k$. Then, $\dim_{\mathbb{F}}\{A_{(\mathbf{y},\mathbf{a})} \mid \mathbf{a} \in \{0, 1, \ldots, d\}^k\} \leq w$.*

Conversely, let $A(\mathbf{x})$ *be a polynomial of individual degree* d*, with* $\mathbf{x} = \{x_1, \ldots, x_n\}$ *and* $w \geq 1$*, such that for any* $k \in [n]$ *and* $\mathbf{y}_k = (x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(k)})$*, we have* $\dim_{\mathbb{F}}\{A_{(\mathbf{y}_k,\mathbf{a})} \mid \mathbf{a} \in \{0,1,\ldots,d\}^k\} \leq w$*. Then, there exists an ROABP of width* w *for* $A(\mathbf{x})$ *in the variable order* $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$*.*

We state the definition of *characterizing dependencies* almost as stated in [GKST16].

Definition 5. [GKST16, Defn.2.4] *Let* $A(\mathbf{x})$ *be a polynomial of individual degree* d *with variable-order* $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$*. Suppose, for each* $k \in [n]$ *and* $\mathbf{y} = (x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(k)})$*,* $\dim_{\mathbb{F}}\{A_{(\mathbf{y},\mathbf{a})} \mid \mathbf{a} \in \{0,1,\ldots,d\}^k\} \leq w$*. For* $k \in [n]$*, we define the* spanning set $\mathrm{span}_k(A)$ *and the* dependency set $\mathrm{depend}_k(A)$ *as subsets of* $\{0,1,\ldots,d\}^k$ *as follows. For* $k = 0$*, let* $\mathrm{depend}_0(A) := \varnothing$ *and* $\mathrm{span}_0(A) := \{\epsilon\}$*, where* $\epsilon = ()$ *denotes the empty tuple. For* $k \in [n]$*, let*

- $\mathrm{depend}_k(A) := \{(\mathbf{a}, j) \mid \mathbf{a} \in \mathrm{span}_{k-1}(A)$ *and* $0 \leq j \leq d\}$*, i.e.* $\mathrm{depend}_k(A)$ *contains all possible extensions of the tuples in* $\mathrm{span}_{k-1}(A)$*.*
- $\mathrm{span}_k(A) \subseteq \mathrm{depend}_k(A)$ *is a subset of size* $\leq w$*, such that for any* $\mathbf{b} \in \mathrm{depend}_k(A)$*, the polynomial* $A_{(\mathbf{y},\mathbf{b})}$ *is in the span of* $\{A_{(\mathbf{y},\mathbf{a})} \mid \mathbf{a} \in \mathrm{span}_k(A)\}$*.*

  Such dependencies of $\{A_{(\mathbf{y},\mathbf{a})} \mid \mathbf{a} \in \mathrm{depend}_k(A)\}$ *over* $\{A_{(\mathbf{y},\mathbf{a})} \mid \mathbf{a} \in \mathrm{span}_k(A)\}$ *comprise the* characterizing set of dependencies *(certifying the width of* A*).*

3 Sum of Homogeneous ROABPs: Proof of Theorem 2

In this section, we shall prove a structure theorem for the lead homogeneous (highest degree) part of a polynomial computed by an ROABP. It will ultimately give us the poly-time blackbox PIT promised in Theorem 2. We call an ROABP *syntactically homogeneous* if for any two nodes (u, v) in the ROABP, as source and sink respectively, the polynomial computed from $u \rightsquigarrow v$ is homogeneous. Clearly, a syntactically homogeneous ROABP computes a homogeneous polynomial of 'low' width, but the converse is not true because some edge label in its ROABP may be inhomogeneous or some intermediate path may be computing an inhomogeneous polynomial.

Throughout this paper, we work with unknown variable order of ROABP. If the ROABP computes a polynomial over $\mathbb{F}[\mathbf{x}]$, we assume an arbitrary variable order $(y_1, y_2, \ldots, y_n)$, where for all $i \in [n]$, $y_i = x_{\pi(i)}$ for some unknown permutation $\pi : [n] \to [n]$.

Definition 6 (Syntactic homogeneity)**.** *Let* $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ *be computed by an ROABP* $D(\mathbf{x})$ *of width* r *in variable order* $(y_1, \ldots, y_n)$*. Let* $D(\mathbf{x}) =: \prod_{i=1}^{n} D_i(y_i)$*, where* $D_1(y_1) \in \mathbb{F}^{1\times r}[y_1]$*,* $D_n(y_n) \in \mathbb{F}^{r\times 1}[y_n]$*, and* $D_i \in \mathbb{F}^{r\times r}[y_i]$ *for* $1 < i < n$*.*

We call ROABP $D(\mathbf{x})$*,* syntactically homogeneous*, if for all* $1 \leq i < n$*, each entry in the subproduct row-vector* $D_{\leq i} := \prod_{j=1}^{i} D_j \in \mathbb{F}^{1\times r}[\mathbf{y}_{\leq i}]$*, is a homogeneous polynomial; and so is each entry in the subproduct column-vector* $D_{>i} := \prod_{j=i+1}^{n} D_j \in \mathbb{F}^{r\times 1}[\mathbf{y}_{>i}]$*.*

In the following theorem, we prove that if a homogeneous polynomial f is computed by an ROABP of width w, then it can also be computed by a syntactically homogeneous ROABP of width $r \leq w$. More precisely, the optimal width ROABP for f constructed using Nisan's characterization (Lemma 4) has width $\leq r$ and we prove that it is syntactically homogeneous.

Theorem 7 (Structure Theorem)**.** *Let* $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ *be a degree* d *homogeneous polynomial computed by an ROABP* $C(\mathbf{x})$ *of width* w *in the variable order* $(y_1, \ldots, y_n)$*. Then,* f *also has a syntactically homogeneous ROABP* $D(\mathbf{x}) = \prod_{i=1}^{n} D_i(y_i)$ *of optimal width* $r \leq w$ *in the same variable order. Moreover,* $\forall i \in [n]$*, each entry in* $D_i(y_i)$ *is merely a monomial in* y_i*.*

Proof. If f is computed by a width w ROABP $C(\mathbf{x})$, it will also have an optimal ROABP $D(\mathbf{x})$ of width $r \leq w$ constructed using Nisan's characterization (Lemma 4). Here, we follow the construction as presented in [GKST16, Lem.2.5]. For all $i \in [n-1]$, we can write f as $f = D_{\leq i} \cdot D_{>i} = \sum_{j=1}^{r} g_j(\mathbf{y}_{\leq i}) h_j(\mathbf{y}_{>i})$. Fix i. Nisan's characterization picks the entries of $D_{\leq i}$ to be r $\mathbb{F}$-linearly independent polynomials $g_1, \ldots, g_r \in \mathbb{F}[\mathbf{y}_{\leq i}]$ and entries of $D_{>i}$ to another r $\mathbb{F}$-linearly independent polynomials $h_1, \ldots, h_r$. By construction, for each $j \in [r]$, $h_j =: f_{(\mathbf{y}_{\leq i}, \mathbf{e}_j)}$, where $\{\mathbf{e}_1, \ldots, \mathbf{e}_r\} := \text{span}_i(f)$. Observe that if f is a homogeneous polynomial, then so is each *coefficient polynomial* $h_j = f_{(\mathbf{y}_{\leq i}, \mathbf{e}_j)}$. Since f is a homogeneous polynomial and h_j's are $\mathbb{F}$-linearly independent homogeneous polynomials, this forces each g_j to be also homogeneous. We prove this non-trivial fact separately as Lemma 18 (Appendix C). Thus, for all $i \in [n-1]$, each entry in $D_{\leq i}$ and $D_{>i}$ is a homogeneous polynomial.

We now prove the second part of the theorem, that every entry in each intermediate matrix $D_i(y_i)$ is a monomial in y_i. For D_1, consider the partition $D = D_1 \cdot D_{>1}$. By syntactic homogeneity proved above, each entry of D_1 is homogeneous and thus is of the form $y_1^{b_j}$ (single monomial), for some $b_j \geq 0$. Similarly, each entry of D_n is also homogeneous, when considering the partition $D = D_{\leq n-1} \cdot D_n$. For $1 < i < n$, consider $D = D_{<i} \cdot D_i \cdot D_{>i}$. This looks like:

$$D = \begin{bmatrix} f_1 & f_2 & \cdots & f_r \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1r} \\ g_{21} & g_{22} & \cdots & g_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ g_{r1} & g_{r2} & \cdots & g_{rr} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_r \end{bmatrix} \tag{1}$$

where $f_1, \ldots, f_r \in \mathbb{F}[\mathbf{y}_{<i}]$, entries of current focus $g_{11}, \ldots, g_{rr} \in \mathbb{F}[y_i]$, and $h_1, \ldots, h_r \in \mathbb{F}[\mathbf{y}_{>i}]$. By syntactic homogeneity of $D_{<i}$, each f_k for $k \in [r]$, is homogeneous. Also, by Nisan's characterization $f_1, \ldots, f_r$ are $\mathbb{F}$-linearly independent. Note that each entry of $D_{\leq i}$ is inner product of $D_{<i} = [f_1, \ldots, f_r]$ with appropriate column of D_i. Wlog, let us consider the inner product with the first column. By syntactic homogeneity of $D_{\leq i}$, we know that $G := f_1 g_{11} + f_2 g_{21} + \ldots + f_r g_{r1}$ is homogeneous. Then, by Lemma 18 again, for each $k \in [r]$, g_{k1} is also a homogeneous polynomial in y_i. Similarly, for every other column in D_i. This shows that each matrix entry g_{ij} is homogeneous (& univariate); hence it is a monomial. □

As a simple corollary of our structure theorem, we get the following sparsity bound for a homogeneous polynomial computable by a width r ROABP.

Lemma 8 (Sparsity bound)**.** *If $f(\mathbf{x})$ is an n-variate homogeneous polynomial of width-r, then sparsity$(f) \leq r^n$.*

Proof. Let $D(\mathbf{x})$ be the width-r ROABP computing f over the field $\mathbb{F}$. By Theorem 7, wlog we can assume $D(\mathbf{x})$ to be syntactically homogeneous with width $\leq r$. Thus, each edge label in the ROABP D is a univariate monomial. In that case, each path from source to sink computes only a single monomial. The number of paths from source to sink is at most r^n. Hence the polynomial computes a sum of at most r^n monomials. □

Using just the Sparsity Lemma 8, we could now finish the proof of Theorem 2.

Proof of Theorem 2. Let $A(\mathbf{x})$, in $\mathcal{P}$, be $=: A_1(\mathbf{x}) + A_2(\mathbf{x}) + \ldots + A_c(\mathbf{x})$, where each $A_i(\mathbf{x})$ is a homogeneous polynomial. By Lemma 8, $\forall i \in [c]$, sparsity$(A_i) \leq r^n$. The total sparsity of $A(\mathbf{x})$ is then bounded by cr^n. One can thus simply apply the sparse PIT map (Lemma 17) for polynomials of degree d and sparsity $\leq cr^n$; which gives the required time-complexity of poly(n, d, cr^n) = poly(cr^n, s). □

We extend the above methods to prove another important property (Lemma 19, Appendix C): if a polynomial has width-r, then so does its lead homogeneous part. This immediately gives us blackbox PIT for a log-variate constant-width ROABP (possibly inhomogeneous).

Lemma 9 (Single ROABP)**.** *Let $\mathcal{P}$ be a set of polynomials, over a field $\mathbb{F}$, computed by an ROABP of width-r and size-s (in n-variables & unknown variable order). Then, blackbox PIT for $\mathcal{P}$ can be solved in poly(s, r^n) time.*

Proof. Let $f \in \mathcal{P}$ be of degree d. By Lemma 19, the ROABP width for $f^{[d]}$, the *lead homogeneous* part of f, is also upper bounded by r. Now, by Lemma 8, sparsity$(f^{[d]}) \leq r^n$. Let Φ be the efficient blackbox PIT map for polynomials of sparsity $\leq r^n$ (Lemma 17). Let t be a new variable. Note that $f \neq 0 \Rightarrow \text{coeff}\big(f(tx_1, tx_2, \ldots, tx_n)\big)(t^d) = f^{[d]} \neq 0$.

In poly(s, r^n) time, we can 'extract' coeff$\big(f(tx_1, \ldots, tx_n)\big)(t^d) = f^{[d]}(\mathbf{x})$ using the blackbox of f; by interpolating $f(tx_1, \ldots, tx_n)$ wrt t. Then, we can apply the non-zeroness preserving map Φ on this extracted coefficient; which completes the blackbox PIT for f in poly(s, r^n) time. □

4 Sum of ROABPs: Proof of Theorem 1

We start with the sum of two ROABPs, $A + B$; the blackbox PIT hence developed would extend to sum of c ROABPs in the next Section 4.2 .

4.1 Sum of two ROABPs

Testing $A + B = 0$ is same as testing equivalence of A and B. Let $A, B \in \mathbb{F}[\mathbf{x}]$ be polynomials of individual degree d, computed by width-r ROABPs, each of size s in n variables. Suppose A is computed in some unknown variable order $(y_1, y_2, \ldots, y_n)$, where $y_i = x_{\pi(i)}$ for each $i \in [n]$, for some permutation $\pi : [n] \to [n]$. We can assume that variable order of B is different from A, since otherwise $A + B$ can be computed by a single ROABP of width $\leq 2r$, and we are done by Lemma 9. The main idea in [GKST16] is to construct an ROABP for B in the variable order of A, by using the characterizing dependencies of A (Defn. 5). Note that the width of an ROABP can blow up exponentially when expressed in a different variable order (see [For15]). We can assume that B does not have ROABP of width r in the variable order of A since otherwise, we will again get a single ROABP of width $2r$ computing $A + B$ in which case we are done.

Thus we are in the setting: $A \neq -B$, and B does not have a width r ROABP in the order $(y_1, y_2, \ldots, y_n)$. By Lemma 4, there will be a minimum index $k \in [n]$ such that $\dim_{\mathbb{F}}\{A_{(\mathbf{y}_{\leq k}, \mathbf{a})} \mid \mathbf{a} \in \{0, 1, \ldots, d\}^k\} \leq r$, but $\dim_{\mathbb{F}}\{B_{(\mathbf{y}_{\leq k}, \mathbf{a})} \mid \mathbf{a} \in \{0, 1, \ldots, d\}^k\} > r$. In the language of dependency equations, there exists an exponent $\mathbf{b} \in \text{depend}_k(A)$ such that:

$$A_{(\mathbf{u},\mathbf{b})} = \sum_{\mathbf{a} \in \text{span}_k(A)} \alpha_{\mathbf{b},\mathbf{a}} \cdot A_{(\mathbf{u},\mathbf{a})} \tag{2}$$

$$B_{(\mathbf{u},\mathbf{b})} \neq \sum_{\mathbf{a} \in \text{span}_k(A)} \alpha_{\mathbf{b},\mathbf{a}} \cdot B_{(\mathbf{u},\mathbf{a})} \tag{3}$$

where $\mathbf{u} := \mathbf{y}_{\leq k} = (y_1, y_2, \ldots, y_k)$, and for each $\mathbf{b} \in \text{depend}_k(A)$: $\alpha_{\mathbf{b},\mathbf{a}} \in \mathbb{F}$ are the dependency coefficients defined by Equation 2. By the dependencies we could assume that B has the same ROABP as A for the first $k-1$ layers. We will design a map Φ_1 for the first $k-1$ variables of both A and B; such that $\Phi_1(B)$ continues to violate the dependency equations of $\Phi_1(A)$ at the y_k-layer (Claim 10 below).

For a polynomial $f \in \mathbb{F}[\mathbf{y}]$, let us use a short-hand $f_{(y_1^{a_1}y_2^{a_2})} \in \mathbb{F}[y_3,\ldots,y_n]$ to denote the coefficient polynomial of monomial $y_1^{a_1}y_2^{a_2}$ in f, which is same as $f_{((y_1,y_2),(a_1,a_2))}$ in the earlier notation.

Claim 10 (Prefix map)**.** *Follow the notation in Eqns. 2 & 3. Let $\Phi_1 : \mathbb{F}[y_1,\ldots,y_n] \to \mathbb{F}[t_1, y_k, y_{k+1}, \ldots, y_n]$ be a hitting-set map for any ROABP of width $\leq r$ on first $k-1$ variables (Φ_1 fixes $y_k,\ldots,y_n$). Let $\deg_{t_1}(\Phi_1(A))$ and $\deg_{t_1}(\Phi_1(B))$ be at most d'; let E denote the set $\{0,1,\ldots,d'\}$. Let $\mathrm{span}_2(\Phi_1(A))$ be a basis of size $\leq r$ such that there exists a set of constants $\{\gamma_{\mathbf{b},\mathbf{a}}\}$, with $\mathbb{F}$-dependencies, for every two-tuple $\mathbf{b} \in E^2$:*

$$\Phi_1(A)_{(t_1^{b_1}y_k^{b_2})} =: \sum_{\mathbf{a}\in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_1(A)_{(t_1^{a_1}y_k^{a_2})} \,. \tag{4}$$

Then, there exists $\mathbf{b} \in E^2$ with a dependency violation in $\Phi_1(B)$, i.e.

$$\Phi_1(B)_{(t_1^{b_1}y_k^{b_2})} \neq \sum_{\mathbf{a}\in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_1(B)_{(t_1^{a_1}y_k^{a_2})} \,. \tag{5}$$

Proof. Let ROABP for A be $D =: \prod_{i=1}^{n} D_i(y_i)$. Since B follows dependency equations of A for first $(k-1)$ layers, we write $B =: D_1D_2\ldots D_{k-1}\cdot D'_{\geq k}$ (by proof of Lemma 4 using Definition 5). Then, $\Phi_1(B) =: \Phi_1(D_1D_2\ldots D_{k-1})(t_1)\cdot D'_{\geq k}(\mathbf{y}_{\geq k})$. For the sake of contradiction, suppose $\Phi_1(B)$ follows the dependency equations of $\Phi_1(A)$ in the y_k layer too. This means $\forall \mathbf{b}$, LHS equals RHS in Equation 5. Since $|\mathrm{span}_2(\Phi_1(A))| \leq r$, this means $\Phi_1(B)$ has a width r ROABP in the first two layers (t_1 and y_k). In other words, coeff.space dimension for first two layers of $\Phi_1(B)$ is $\leq r$. Observe that then by Lemma 13, Φ_1 preserves the coeff.space dimension of the first k layers of B too.

Thus, $\dim_{\mathbb{F}}\{B_{(\mathbf{y}_{\leq k},\mathbf{a})} \mid \mathbf{a}\in E^k\} = \dim_{\mathbb{F}}\{\Phi_1(B)_{((t_1,y_k),\mathbf{a})} \mid \mathbf{a}\in E^2\} \leq r$. This contradicts our choice of k being the first variable up to which B does not have a width r representation, which meant $\dim_{\mathbb{F}}\{B_{(\mathbf{y}_{\leq k},\mathbf{a})} \mid \mathbf{a}\in E^k\} > r$ (as in Lemma 4). □

In the following claim, we study a map Φ_2 for $\mathbf{y}_{>k}$ variables such that we preserve Equations 4 and 5 under the action of map. Thus, eventually the image polynomial becomes *tri*variate!

Claim 11 (Suffix map)**.** *Continue with the notation of Claim 10. Let $\Phi_2 : \mathbb{F}[t_1, y_k, y_{k+1}, \ldots, y_n] \to \mathbb{F}[t_1, y_k, t_2]$ be a hitting-set map for any ROABP of width $\leq r^2(r+1)$ on last $n-k$ variables (Φ_2 fixes t_1, y_k). Then, there exists $\mathbf{b} \in E^2$ such that:*

$$\Phi_2\circ\Phi_1(A)_{(t_1^{b_1}y_k^{b_2})} = \sum_{\mathbf{a}\in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_2\circ\Phi_1(A)_{(t_1^{a_1}y_k^{a_2})} \tag{6}$$

$$\Phi_2\circ\Phi_1(B)_{(t_1^{b_1}y_k^{b_2})} \neq \sum_{\mathbf{a}\in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_2\circ\Phi_1(B)_{(t_1^{a_1}y_k^{a_2})} \,. \tag{7}$$

Proof. Equation 6 in this claim directly follows by applying map Φ_2 on Equation 4; it is true $\forall \mathbf{b}\in E^2$. Now we shall prove that in Equation 5 the difference polynomial $g := \Phi_1(B)_{(t_1^{b_1}y_k^{b_2})} - \sum_{\mathbf{a}\in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_1(B)_{(t_1^{a_1}y_k^{a_2})} \neq 0$ can be computed using a single ROABP of width at most $r^2(r+1)$.

Let σ represent the original variable order of B: $x_{\sigma(1)} < \ldots < x_{\sigma(n)}$. By assumption, B had a width r representation for the first $(k-1)$ layers in the variable order of A ($y_1 < \ldots < y_n$); implying $B = [P_1, P_2, \ldots, P_r]\cdot[Q_1, Q_2, \ldots, Q_r]^{\intercal}$, where $\forall i \in [r]$, $P_i \in \mathbb{F}[\mathbf{y}_{<k}]$ and $Q_i \in \mathbb{F}[\mathbf{y}_{\geq k}]$. Recall that, by construction [GKST16, Lem.2.5], Q_i's are certain coefficient polynomials of B; $Q_i = B_{(\mathbf{y}_{<k},\mathbf{a}_i)}$ where $\{\mathbf{a}_1,\ldots,\mathbf{a}_r\} = \mathrm{span}_{k-1}(A)$. Now, by Lemma 16, each Q_i has a width

r ROABP in the variable order inherited from B, that is, $\sigma(\mathbf{y}_{\geq k})$. Clearly, for each $a \in E$, $\Phi_1(B)_{(t_1,a)} = \sum_{i=1}^{r} \text{coeff}(\Phi_1(P_i))(t_1^a) \cdot Q_i$; implying $\Phi_1(B)_{(t_1,a)} \in \text{span}_{\mathbb{F}}\{Q_1, \ldots, Q_r\}$. For each $a \in E$, let $\Phi_1(B)_{(t_1,a)} =: Q'_a$, where Q'_a is the suitable $\mathbb{F}$-linear combination of $Q_1, \ldots, Q_r$. Observe that any $\mathbb{F}$-linear combination $\sum_{i=1}^{r} c_i Q_i$, where each $c_i \in \mathbb{F}$, can be computed by a single ROABP of width r^2; by placing ROABP of each Q_i in parallel. Thus, for each $a \in E$, Q'_a also has an ROABP of width r^2.

Moving one variable forward, again by applying Lemma 16 on each Q'_a, we know that for each $b \in E$, $Q'_{a(y_k,b)} := (Q'_a)_{(y_k,b)}$ also has an ROABP of width r^2; in the variable order $\sigma(\mathbf{y}_{>k})$. We can rewrite our polynomial g as $g = Q'_{b_1(y_k,b_2)} - \sum_{\mathbf{a} \in \text{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot Q'_{a_1(y_k,a_2)}$.

The number of summands in g is $|\text{span}_2(\Phi_1(A))| + 1 \leq r + 1$, and each summand in g has a width r^2 ROABP. Hence, g can be computed by a single ROABP of width $\leq r^2(r+1)$; by placing each of the width r^2 ROABPs in parallel.

Finally, since Φ_2 is a hitting-set map for any $(n-k)$-variate ROABP of width $\leq r^2(r+1)$, it will preserve the non-zeroness of g; giving us the inequality in Eqn. 7. □

Using Claims 10 and 11, we can design a single map Φ for nonzero $A + B$ such that $\Phi(A)$ has a trivariate ROABP of width $\leq r$ in the order (t_1, y_k, t_2), whereas $\Phi(B)$ does not. This would prove $\Phi(A) \neq -\Phi(B)$; eventually certifying that $A + B \neq 0$.

Lemma 12 (Sum of two)**.** *Let $A(\mathbf{x})$ and $B(\mathbf{x})$ be two polynomials of individual degree d, each computed by an ROABP of width r. Let A have variable order $(y_1, y_2, \ldots, y_n)$, where $\forall i \in [n]$, $y_i := x_{\pi(i)}$, for some permutation $\pi : [n] \to [n]$. Then, one can design a map $\Phi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[t_1, y_k, t_2]$, for some $k \in [n]$, in poly(d, r^{3n}) time and $\deg(\Phi(A+B)) \leq$ poly(d, r^{3n}), such that $A + B = 0$ if and only if $\Phi(A + B) = 0$.*

This implies a poly(s, r^n)-time blackbox PIT for sum of two size-s ROABPs.

Proof. Observe that the maps Φ_1 and Φ_2, the way defined above, act on disjoint set of variables. Let $\Phi := \Phi_2 \circ \Phi_1$. Trivially, $A + B = 0$ implies $\Phi(A + B) = 0$. Conversely, $A + B \neq 0$ implies that $A \neq -B$. If B has a width $\leq r$ ROABP in the variable order of A then $A + B$ has width $\leq 2r$; in which case we are done by Lemma 9.

So, we assume that B requires width $> r$ ROABP in the variable order of A. Let us now work with the k and the variable order $y_1, \ldots, y_n$ as in the above proved claims. Claims 10 and 11 together prove that $\Phi(A)$ is an ROABP of width r, in the variable order (t_1, y_k, t_2); while Equation 7 points out that $\Phi(B)$ does not have width r ROABP in the same variable order. Thus, $\Phi(A + B) \neq 0$.

Let us now estimate the cost of constructing Φ. For Claim 10 to hold, we only need to 'guess' the prefix set $\{y_1, \ldots, y_{k-1}\}$ and the variable y_k. For each $k \in [n]$, we go over all k-sized subsets of $[n]$, and try k choices (of y_k) for each subset. The number of possibilities is at most $\sum_{k=1}^{n} k\binom{n}{k} = \sum_{k=1}^{n} n\binom{n-1}{k-1} \leq n2^{n-1}$.

A hitting-set map Φ_1, for any width-r $(k-1)$-variate ROABP, can be designed in time $T_1 \leq \text{poly}(d, r^k)$ (invoke Lemma 9 for $k-1$ variables). The remaining $n-k$ variables are fixed once we guess the correct choice of prefix. Since in Claim 11, Φ_2 needs to be a hitting-set map for any ROABP of width $r^3 + r^2$, it can be designed in time $T_2 \leq \text{poly}(d, r^{3(n-k)})$ (invoke Lemma 9 for $n-k$ variables).

With all the subset guesses, overall, our correct map $\Phi = \Phi_2 \circ \Phi_1$ can be designed in time $T \leq n2^{n-1} \cdot (T_1 + T_2) \leq \text{poly}(d, r^{3n})$ (for $r \geq 2$). The individual degree of polynomial in the image of Φ is bounded by $\text{poly}(d, r^{3n})$ using Lemmas 9 and 17. After designing Φ, one can use the trivial hitting-set for trivariate polynomials as the hitting-set for $A + B$. □

We show below that any hitting-set map for a prefix of variables, also preserves the coeff.space dimension up to all *subsequent* variables.

Lemma 13 (Coeff.space preservation). *Let $D(\mathbf{x}) = D_1(y_1)\cdots D_{k-1}(y_{k-1})\cdot D_k(y_k)\cdot D'(\mathbf{y}_{>k})$ be matrix-product for a polynomial; assume that the prefix $D_1(y_1)\cdots D_{k-1}(y_{k-1})$ is a matrix-product of width r, $D_k \in \mathbb{F}^{r\times r'}[y_k]$ (say $r \le r'$), and $D' \in \mathbb{F}^{r'\times 1}[\mathbf{y}_{>k}]$. Let $\Psi : \mathbb{F}[y_1,\ldots,y_n] \to \mathbb{F}[t, y_k, y_{k+1},\ldots,y_n]$ be a hitting-set map for any width r ROABP in the first $k-1$ variables. (Ψ fixes $y_k,\ldots,y_n$.) Then, Ψ preserves the k-prefix coeff.space of D, that is:*

$$\dim_{\mathbb{F}}\{D_{(\mathbf{y}_{\le k},\mathbf{a})} \mid \mathbf{a}\} \;=\; \dim_{\mathbb{F}}\{\Psi(D)_{((t,y_k),\mathbf{a})} \mid \mathbf{a}\}\ .$$

Proof. Consider the matrix product for $D(\mathbf{y})$ at $(k-1)^{th}$ layer: $D = D_{<k}\cdot D_k\cdot D'$, where $D_{<k} := \prod_{i=1}^{k-1} D_i \in \mathbb{F}^{1\times r}[\mathbf{y}_{<k}]$. Wlog let the entries of $D_{<k}\cdot D_k$ be the r' $\mathbb{F}$-linearly independent polynomials given by Nisan's characterization (Lemma 4): $D_{<k}\cdot D_k =: [P_1, P_2, \ldots, P_{r'}]$. Similarly entries of D' are r' linearly independent polynomials given by coefficient-extraction of D: $D' =: [Q_1, Q_2, \ldots, Q_{r'}]^\intercal$. View Ψ as mapping the first k variables to $\mathbb{F}[t, y_k]$ (keeping the rest $n-k$ variables unchanged). For $c_i \in \mathbb{F}$ ($i \in [r']$) not all zero, we have:

$$c_1P_1 + c_2P_2 + \ldots + c_{r'}P_{r'} \;\neq\; 0\ . \tag{8}$$

Note that the 'polynomial-vector' $D_{<k}\cdot D_k$ has a width r' ROABP (with r' output gates) which is derived from the 'partial' ROABP of D. In the same width, we could represent the polynomial $c_1P_1 + c_2P_2 + \ldots + c_{r'}P_{r'} =: P$.

Since Ψ is a hitting-set map (univariate in t) for any width-r $(k-1)$-variate ROABP, it preserves $\dim_{\mathbb{F}(y_k)}$ of the r coordinates of $D_{<k}(\mathbf{y}_{<k})$. Consequently, Ψ (now bivariate in t, y_k) preserves $\dim_{\mathbb{F}}$ of the r' coordinates of $D_{<k}(\mathbf{y}_{<k})\cdot D_k(y_k)$. In other words, Ψ is also a hitting-set map (bivariate) for any width $\le r'$ k-variate ROABP.

Since P is indeed of width r' (& k-variate), therefore, Ψ preserves the non-zeroness of Equation 8; implying $\mathbb{F}$-linear independence of $\Psi(P_1),\ldots,\Psi(P_{r'}) \in \mathbb{F}[t, y_k]$. Moreover, $\Psi(D) = [\Psi(P_1),\ldots,\Psi(P_{r'})]\cdot[Q_1, Q_2, \ldots, Q_{r'}]^\intercal$. Whence, $\dim_{\mathbb{F}}\{\Psi(D)_{((t,y_k),\mathbf{a})} \mid \mathbf{a}\} = \dim_{\mathbb{F}}\{D_{(\mathbf{y}_{\le k},\mathbf{a})} \mid \mathbf{a}\} = r'$. □

4.2 Sum of c ROABPs

Let the input be sum of c polynomials $A_1(\mathbf{x}), A_2(\mathbf{x}), \ldots, A_c(\mathbf{x})$, each of individual degree d, computed by ROABPs of width r. Again, we will assume the variable orders for each ROABP to be different, lest we reduce to a smaller sum instance. The simple recursive strategy used in [GKST16] is to reduce it to an instance of sum of two ROABPs. Let $A := A_1$ and $B := A_2 + \ldots + A_c$. Suppose A has r-width ROABP in some unknown variable order $(y_1, y_2, \ldots, y_n)$, where $y_i := x_{\pi(i)}$, for each $i \in [n]$. Thus, we get dependency equations for A, as in Equation 2. If the input sum is non-zero then B will not follow some dependency of A.

Note that unlike the 'sum of two' case, B is not computed by a single ROABP of width r. This is not a cause of worry, as we shall see now. Define $Q := B_{(\mathbf{u},\mathbf{b})} - \sum_{\mathbf{a}\in \mathrm{span}_k(A)} \gamma_{\mathbf{b},\mathbf{a}}\cdot B_{(\mathbf{u},\mathbf{a})}$, where $\mathbf{u} := (y_1,\ldots,y_k)$. Since $B = A_2 + \ldots + A_c$, we get

$$Q \;=\; \sum_{i=2}^{c}\left(A_{i(\mathbf{u},\mathbf{b})} \;-\; \sum_{\mathbf{a}\in \mathrm{span}_k(A)} \gamma_{\mathbf{b},\mathbf{a}}\cdot A_{i(\mathbf{u},\mathbf{a})}\right). \tag{9}$$

Now, as explained in Section 4.1, for each of the $c-1$ summands in Equation 9, we have an ROABP of width $r(r+1)$. We apply Φ_1 on first $k-1$ variables and get analogous dependency equations for $\Phi_1(A)$, $\Phi_1(B)$ (and $\Phi_1(Q) \neq 0$) as in Claim 10.

$$\Phi_1(Q)_{(t_1^{b_1}y_k^{b_2})} \;=\; \sum_{i=2}^{c}\left(\Phi_1(A_i)_{(t_1^{b_1}y_k^{b_2})} \;-\; \sum_{\mathbf{a}\in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}}\cdot \Phi_1(A_i)_{(t_1^{a_1}y_k^{a_2})}\right). \tag{10}$$

In Equation 10, each of the $c-1$ summands has an ROABP of width $r^2(r+1) \leq 2r^3$ (See proof of Claim 11). This effectively reduces the problem, of designing Φ_2, to an instance of blackbox PIT for sum of $c-1$ ROABPs of width $O(r^3)$, which can be solved recursively. We formalize this process in the following lemma.

Lemma 14 (Sum of c)**.** *Let $A_1(\mathbf{x}), A_2(\mathbf{x}), \ldots, A_c(\mathbf{x})$ be c polynomials of individual degree d, each computed by an ROABP of width r. Then, in poly(d^c, r^{n3^c}) time, one can design a hitting-set (univariate) map Ψ, with degree of $\Psi(\sum_{i=1}^{c} A_i)$ bounded by poly(d^c, r^{n3^c}), such that $\sum_{i=1}^{c} A_i = 0$ if and only if $\Psi(\sum_{i=1}^{c} A_i) = 0$.*

Proof. We prove by induction on c. Base case for $c = 2$ has been proved earlier in Lemma 12. Suppose $A = A_1$ has the unknown variable order $(y_1, \ldots, y_n)$ where $y_i = x_{\pi(i)}$ for each $i \in [n]$. Let y_k be the first layer where $B = A_2 + \ldots + A_c$ 'deviates' from A. Thus, by Claim 10, we get map Φ_1 and Equation 10, which is a sum of $c-1$ ROABPs of width $\leq 2r^3$. By induction hypothesis, we can design a (univariate) map Φ_2 for $\Phi_1(Q)_{(t_1^{b_1} y_k^{b_2})}$, which acts on the remaining $(n-k)$ variables, and preserves non-zeroness of the polynomial. Thus, altogether $\Phi := \Phi_2 \circ \Phi_1$ will preserve non-zeroness of $A+B$; since $\Phi(A)$ will satisfy all its dependency equations but $\Phi(B)$ would continue to violate one. This implies $A + B = 0$ if and only if $\Phi(A+B) = 0$. Note that Φ's image is trivariate; we can trivially redefine it so that the image is univariate (by Polynomial Identity Lemma or Kronecker map) and call this map Ψ.

Let $T(c, r)$ denote the time complexity of designing Ψ, where $T(2, r) \leq \text{poly}(d, r^n)$ by Lemma 12. Like in proof of Lemma 12, for each choice of prefix $(y_1, y_2, \ldots, y_k)$ and Φ_1, we will design Φ_2 to verify non-zeroness of the corresponding Equation 10. By induction hypothesis Φ_2 can be designed in $\leq T(c-1, 2r^3)$ time. Thus, we get a recursive formula for designing Ψ (which is a univariate map),

$$T(c, r) \;\leq\; n2^n \cdot \text{poly}(d, r^n) \cdot T(c-1, 2r^3)\ .$$

As a solution, we get $T(c, r) \leq n^c \cdot 2^{cn} \cdot \text{poly}(d^c, r^{n3^c}) \leq \text{poly}(d^c, r^{n3^c})$ (for $r \geq 2$).

Similarly, we get a recurrence for the degree of the image of Ψ: $S(c, r) \leq \text{poly}(d, r^n) \cdot S(c-1, 2r^3)$. We get the solution $S(c, r) \leq \text{poly}(d^c, r^{n3^c})$. □

Finally after having developed all the machinery, we complete the proof of Theorem 1. We also prove Corollary 3, which is an easy consequence of Theorem 1.

Proof of Theorem 1. Let the polynomial be $f \in \mathbb{F}[\mathbf{x}] = A_1 + \ldots + A_c$, computed by c ROABPs mentioned in the hypothesis of Theorem 1. By Lemma 14 we have a variable reduction map $\Psi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[t]$ that preserves the non-zeroness of f, i.e. $f = 0 \Leftrightarrow \Psi(f) = 0$. As argued in Lemma 14, this map Ψ is constructible in poly(d^c, r^{n3^c})-time and the individual degree of $\Psi(f)$ is similarly bounded.

Thus, one can evaluate $\Psi(f) \in \mathbb{F}[t]$, on sufficiently many field points, since it is univariate and the degree is polynomially bounded. This gives the blackbox PIT algorithm with the required time-complexity. □

Proof of Corollary 3. By [AFS$^+$16, Lem.2.5], a polynomial computed by a width-r, degree-d, k-pass ABP can also be computed by a width-r^{2k} ROABP of degree-dk, and the same underlying variable order. Thus for a sum of c such k-pass ABPs, we can apply Theorem 1 to achieve the stated time-complexity. □

5 Conclusion and Future Directions

We discover a structure theorem which helps us in bounding the sparsity of the lead homogeneous part of an ROABP. This upper bound is polynomially bounded when the ROABP is constant-width and log-variate. We apply it to get two important results: Lemma 9 and Theorem 2. In the log-variate setting, we then show how to solve blackbox PIT for sum of constantly-many such ROABPs by using blackbox PIT of a single such ROABP.

Note that we need constant width in Theorem 1 only because poly-time blackbox PIT for an unbounded-width (log-variate) ROABP is not known as of now. In fact, our algorithm for sum of c ROABPs in Section 4 can directly employ any given PIT algorithm of an unbounded-width ROABP:

Meta-Lemma. *If there exists a poly-time blackbox PIT algorithm for a log-variate ROABP, then there exists a poly-time blackbox PIT algorithm for sum of c-many log-variate ROABPs (for c constant).*

In the context of this work and previous related works, a variety of open problems arise:

- Design a poly(s)-time blackbox algorithm for $(\log s)$-variate, size-s ROABP. This will also solve diagonal depth-3 model [FSS14]. Wlog, ROABP can also be assumed to be syntactically homogeneous (Theorem 7, Lemma 19).
- For a polynomial computed by an ROABP, we proved our Structure Theorem 7 only for the lead-degree homogeneous part. (Here leading can be either highest degree or lowest degree.) Can we get a similar structure theorem for *intermediate* degree homogeneous components? We do know that width upper bound in Lemma 19 works only for leading homogeneous components. For example, $f = (x_1 + 1)(x_2 + 1)$ has a width 1 ROABP, and so does its leading components $f^{[2]} = x_1x_2$ and $f^{[0]} = 1$, but $f^{[1]} = x_1 + x_2$ requires width 2. However, for general polynomials can we still get a 'weaker' upper bound on the ROABP width of intermediate homogeneous components?
- In Theorem 2, can we remove the restriction of each ROABP computing a homogeneous polynomial? If we can work with *inhomogeneous* polynomials also, then that would solve diagonal depth-3 model (Appendix A). Design a poly(r^n, c, d)-time blackbox PIT for sum of c ROABPs; each of width r computing an n-variate polynomial of degree d?
- Design a poly(n, d)-time blackbox algorithm for an n-variate, d-degree polynomial; computed by ROABP of constant-width in *unknown* variable order, which works for *all* fields. This problem is open, since [GKS17] algorithm only works for known variable order and for fields with zero/large characteristic.

Acknowledgements

We thank Subhayan Saha, Sumanta Ghosh and Zeyu Guo for various discussions related to the algebraic models studied here. N.S. thanks the funding support from DST (DST/SJF/MSA-01/2013-14) and N. Rama Rao Chair.

References

[AB03] Manindra Agrawal and Somenath Biswas. Primality and identity testing via Chinese remaindering. *Journal of the ACM*, 50(4):429–443, July 2003. 4, 23

[AFS+16] Matthew Anderson, Michael Forbes, Ramprasad Saptharishi, Amir Shpilka, and Ben Lee Volk. Identity testing and lower bounds for read-k oblivious algebraic branching programs. In *Computational Complexity Conference (CCC)*, 2016. 3, 4, 6, 13

[AGKS15] 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. 2, 4, 23

[Agr05] Manindra Agrawal. Proving lower bounds via pseudo-random generators. In *FSTTCS*, volume 3821 of *Lecture Notes in Computer Science*, pages 92–105, 2005. 1

[AGS19] Manindra Agrawal, Sumanta Ghosh, and Nitin Saxena. Bootstrapping variables in algebraic circuits. *Proceedings of the National Academy of Sciences*, 116(17):8107–8118, 2019. (A preliminary version appeared in STOC, 2018). 1, 2, 4

[AKS04] Manindra Agrawal, Neeraj Kayal, and Nitin Saxena. PRIMES is in P. *Annals of mathematics*, pages 781–793, 2004. 1

[ASS13] Manindra Agrawal, Chandan Saha, and Nitin Saxena. Quasi-polynomial hitting-set for set-depth-Δ formulas. In *Symposium on Theory of Computing Conference, STOC, Palo Alto, CA, USA, June 1-4*, pages 321–330, 2013. 4

[ASSS12] 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. 4

[AV08] Manindra Agrawal and V. Vinay. Arithmetic circuits: A chasm at depth four. In *49th Annual IEEE Symposium on Foundations of Computer Science, FOCS ,October 25-28 2008, Philadelphia, PA, USA*, pages 67–75, 2008. 3

[Bar89] David A Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989. 2

[BCG18] Mark Braverman, Gil Cohen, and Sumegha Garg. Hitting sets with near-optimal error for read-once branching programs. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, pages 353–362, 2018. 2

[BDVY13] Andrej Bogdanov, Zeev Dvir, Elad Verbin, and Amir Yehudayoff. Pseudorandomness for width-2 branching programs. *Theory of Computing*, 9:283–293, 2013. 4

[BMS13] M. Beecken, J. Mittmann, and N. Saxena. Algebraic Independence and Blackbox Identity Testing. *Inf. Comput.*, 222:2–19, 2013. (Conference version in ICALP 2011). 4

[BOC92] Michael Ben-Or and Richard Cleve. Computing algebraic formulas using a constant number of registers. *SIAM Journal on Computing*, 21(1):54–58, 1992. (Preliminary version in STOC'88). 3

[BOT88] 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. 4

[Bre74] Richard P. Brent. The parallel evaluation of general arithmetic expressions. *Journal of the ACM*, 21(2):201–206, April 1974. 3

[BRRY14] Mark Braverman, Anup Rao, Ran Raz, and Amir Yehudayoff. Pseudorandom generators for regular branching programs. *SIAM J. Comput.*, 43(3):973–986, 2014. 4

[BV10] Joshua Brody and Elad Verbin. The coin problem and pseudorandomness for branching programs. In *51th Annual IEEE Symposium on Foundations of Computer Science, FOCS, October 23-26, 2010, Las Vegas, Nevada, USA*, pages 30–39, 2010. 4

[BW98] Jan Behrens and Stephan Waack. Equivalence test and ordering transformation for parity-OBDDs of different variable ordering. In *15th Symposium on Theoretical Aspects of Computer Science*, Lecture Notes in Computer Science 1373, pages 227–237. Springer-Verlag, 1998. 2

[CHI+18] Luca Chiantini, Jonathan D Hauenstein, Christian Ikenmeyer, Joseph M Landsberg, and Giorgio Ottaviani. Polynomials and the exponent of matrix multiplication. *Bulletin of the London Mathematical Society*, 50(3):369–389, 2018. 4

[DdOS14] 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. 1

[De11] Anindya De. Pseudorandomness for permutation and regular branching programs. In *Proceedings of the 26th Annual IEEE Conference on Computational Complexity, CCC 2011, San Jose, California, June 8-10, 2011*, pages 221–231, 2011. 4

[DL78] Richard A. Demillo and Richard J. Lipton. A probabilistic remark on algebraic program testing. *Information Processing Letters*, 7(4):193 – 195, 1978. 1, 22

[dOSV16] Rafael Mendes de Oliveira, Amir Shpilka, and Ben Lee Volk. Subexponential size hitting sets for bounded depth multilinear formulas. *Computational Complexity*, 25(2):455–505, 2016. 4

[DS07] Zeev Dvir and Amir Shpilka. Locally decodable codes with two queries and polynomial identity testing for depth 3 circuits. *SIAM Journal on Computing*, 36(5):1404–1434, 2007. 4

[FGS18] Michael A Forbes, Sumanta Ghosh, and Nitin Saxena. Towards blackbox identity testing of log-variate circuits. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018. 4

[FGT17] Stephen A. Fenner, Rohit Gurjar, and Thomas Thierauf. Guest column: Parallel algorithms for perfect matching. *SIGACT News*, 48(1):102–109, 2017. 1

[FK18] Michael A Forbes and Zander Kelley. Pseudorandom generators for read-once branching programs, in any order. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 946–955. IEEE, 2018. 2

[For14] Michael A. Forbes. *Polynomial Identity Testing of Read-Once Oblivious Algebraic Branching Programs*. PhD thesis, MIT, 2014. 6

[For15] Michael A. Forbes. Deterministic divisibility testing via shifted partial derivatives. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 451–465, 2015. 4, 9, 23

[FS12] Michael A. Forbes and Amir Shpilka. On identity testing of tensors, low-rank recovery and compressed sensing. In *Proceedings of the 44th ACM Symposium on Theory of Computing (STOC)*, pages 163–172, 2012. 4

[FS13a] 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. 4, 21

[FS13b] Michael A. Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *FOCS*, pages 243–252, 2013. 2, 4, 21

[FSS14] 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. 2, 3, 4, 14, 21

[GGOW16] 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. 4

[GKKS16] 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. (Preliminary version in FOCS'13). 3

[GKS17] 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). 2, 3, 14

[GKSS19] Zeyu Guo, Mrinal Kumar, Ramprasad Saptharishi, and Noam Solomon. Derandomization from algebraic hardness: Treading the borders. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 147–157. IEEE, 2019. 2, 4

[GKST16] 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). 2, 3, 4, 5, 6, 7, 8, 9, 10, 12, 22

[GM96] Jordan Gergov and Christoph Meinel. Mod-2-OBDDs - a data structure that generalizes EXOR-sum-of-products and ordered binary decision diagrams. *Formal Methods in System Design*, 8:273–282, 1996. 2

[Gur15] Rohit Gurjar. *Derandomizing PIT for ROABP and Isolation Lemma for Special Graphs*. PhD thesis, Indian Institute of Technology Kanpur, 2015. 6

[HS80] Joos Heintz and Claus P. Schnorr. Testing polynomials which are easy to compute (extended abstract). In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, STOC '80, pages 262–272, New York, NY, USA, 1980. ACM. 1

[HZ18] William Hoza and David Zuckerman. Simple optimal hitting sets for small-success RL. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 59–64. IEEE, 2018. 2

[IK99] Anthony Iarrobino and Vassil Kanev. *Power sums, Gorenstein algebras, and determinantal loci*. Springer Science & Business Media, 1999. 4

[IMZ12] Russell Impagliazzo, Raghu Meka, and David Zuckerman. Pseudorandomness from shrinkage. In *53rd Annual IEEE Symposium on Foundations of Computer Science, FOCS, New Brunswick, NJ, USA, October 20-23, 2012*, pages 111–119, 2012. 2

[INW94] Russell Impagliazzo, Noam Nisan, and Avi Wigderson. Pseudorandomness for network algorithms. In *Proceedings of the Twenty-sixth Annual ACM Symposium on Theory of Computing*, STOC, pages 356–364, New York, NY, USA, 1994. ACM. 2

[JQS10] Maurice J. Jansen, Youming Qiao, and Jayalal Sarma. Deterministic black-box identity testing π-ordered algebraic branching programs. In *FSTTCS*, pages 296–307, 2010. 4

[KI04] Valentine Kabanets and Russell Impagliazzo. Derandomizing polynomial identity tests means proving circuit lower bounds. *Computational Complexity*, 13(1-2):1–46, 2004. (Preliminary version in STOC' 03). 1

[KNP11] Michal Koucký, Prajakta Nimbhorkar, and Pavel Pudlák. Pseudorandom generators for group products: extended abstract. In *Proceedings of the 43rd ACM Symposium on Theory of Computing, STOC, San Jose, CA, USA, 6-8 June 2011*, pages 263–272, 2011. 4

[KNS16] Neeraj Kayal, Vineet Nair, and Chandan Saha. Separation between read-once oblivious algebraic branching programs (roabps) and multilinear depth three circuits. In *33rd Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 46:1–46:15, 2016. 4

[Koi12] Pascal Koiran. Arithmetic circuits: The chasm at depth four gets wider. *Theoretical Computer Science*, 448:56–65, 2012. 3

[Kor16] Arpita Korwar. *Polynomial Identity Testing and Lower Bounds for Sum of Special Arithmetic Branching Programs*. PhD thesis, Indian Institute of Technology Kanpur, 2016. 6

[Kro82] Leopold Kronecker. *Grundzuge einer arithmetischen Theorie der algebraischen Grossen*. Berlin, G. Reimer, 1882. 23

[KS01] Adam Klivans and Daniel A. Spielman. Randomness efficient identity testing of multivariate polynomials. In *Proceedings of the 33rd ACM Symposium on Theory of Computing (STOC)*, pages 216–223, 2001. 4

[KS07] Neeraj Kayal and Nitin Saxena. Polynomial identity testing for depth 3 circuits. *Computational Complexity*, 16(2):115–138, 2007. 4

[KS09] Neeraj Kayal and Shubhangi Saraf. Blackbox polynomial identity testing for depth 3 circuits. In *Proceedings of the 50th IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 198–207, 2009. 4

[KS11] Zohar Shay Karnin and Amir Shpilka. Black box polynomial identity testing of generalized depth-3 arithmetic circuits with bounded top fan-in. *Combinatorica*, 31(3):333–364, 2011. 4

[KS16a] 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. 4

[KS16b] 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. 4

[KSS14] 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. 1

[KST19] Mrinal Kumar, Ramprasad Saptharishi, and Anamay Tengse. Near-optimal bootstrapping of hitting sets for algebraic circuits. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 639–646. Society for Industrial and Applied Mathematics, 2019. 2, 4

[Lan17] Joseph M Landsberg. *Geometry and complexity theory*, volume 169. Cambridge University Press, 2017. 4

[LMP16] 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. 4

[Mul12a] Ketan D. Mulmuley. The GCT Program toward the P vs. NP problem. *Commun. ACM*, 55(6):98–107, June 2012. 4

[Mul12b] 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. 4

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

[Nis91] Noam Nisan. Lower bounds for non-commutative computation (extended abstract). In *Proceedings of the 23rd ACM Symposium on Theory of Computing, ACM Press*, pages 410–418, 1991. 3, 6

[Nis92] Noam Nisan. Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992. 2

[PSS16] 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). 4

[RR99] Ran Raz and Omer Reingold. On recycling the randomness of states in space bounded computation. In *Proceedings of the Thirty-First Annual ACM Symposium on Theory of Computing, May 1-4, 1999, Atlanta, Georgia, USA*, pages 159–168, 1999. 2

[RS05] Ran Raz and Amir Shpilka. Deterministic polynomial identity testing in non-commutative models. *Computational Complexity*, 14(1):1–19, 2005. 2, 4

[Sap16] Ramprasad Saptharishi. A survey of lower bounds in arithmetic circuit complexity. Technical report, https://github.com/dasarpmar/lowerbounds-survey/, 2016. 1

[Sax08] Nitin Saxena. Diagonal circuit identity testing and lower bounds. In *ICALP*, volume 5125 of *Lecture Notes in Computer Science*, pages 60–71. Springer, 2008. 3, 4, 21

[Sax09] Nitin Saxena. Progress on polynomial identity testing. *Bulletin of the EATCS*, 99:49–79, 2009. 1

[Sax14] 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. 1

[Sch80] Jacob T. Schwartz. Fast probabilistic algorithms for verification of polynomial identities. *Journal of the ACM*, 27(4):701–717, October 1980. 1, 22

[SS11] Nitin Saxena and C. Seshadhri. An almost optimal rank bound for depth-3 identities. *SIAM Journal on Computing*, 40(1):200–224, 2011. 4

[SS12] 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. 4

[SSS13] 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. 4

[Ste12] Thomas Steinke. Pseudorandomness for permutation branching programs without the group theory. *Electronic Colloquium on Computational Complexity (ECCC)*, 19:83, 2012. 4

[SV09] Amir Shpilka and Ilya Volkovich. Improved polynomial identity testing for read-once formulas. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, pages 700–713. Springer, 2009. 21, 23

[SW97] Petr Savický and Ingo Wegener. Efficient algorithms for the transformation between different types of binary decision diagrams. *Acta Informatica*, 34(4):245–256, 1997. 2

[SY10] Amir Shpilka and Amir Yehudayoff. Arithmetic circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science*, 5(3-4):207–388, 2010. 1, 5

[Tav15] Sébastien Tavenas. Improved bounds for reduction to depth 4 and depth 3. *Information and Computation*, 240:2–11, 2015. (Preliminary version in MFCS'13). 3

[Vai15] Rishabh Vaid. Blackbox identity testing for simple depth 3 circuits. Master's thesis, Indian Institute of Technology Kanpur, 2015. 21

[VSBR83] Leslie G. Valiant, Sven Skyum, S. Berkowitz, and Charles Rackoff. Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983. 3

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, EUROSAM '79, pages 216–226, London, UK, UK, 1979. Springer-Verlag. 1, 22

A Proofs from Section 1– Introduction

Lemma 15. *If we have poly-time blackbox PIT for sum of width-1, log-variate (commutative) ROABPs, then we have poly-time blackbox PIT for diagonal depth-3 circuits.*

Proof Sketch. [FS13a, FS13b] showed that diagonal depth-3 circuits have 'low' dimension partial-derivative space, and that such polynomials have a nonzero *log*-support monomial. Under the promise of such a log-support monomial, we can apply variable-reduction map of [SV09], used in [FSS14] or the map of [Vai15], to get from n to $O(\log n)$ variables. Both these maps preserve non-zeroness.

After applying the log-variate map, we will get to power-of-sums-of univariates form which we can convert to sum-of-products-of-univariates form using the duality-trick of [Sax08]. Moreover, each product-of-univariates has a width-1 ROABP; thus we have represented as sum of width-1 log-variate ROABPs (which are trivially commutative!). □

B Definitions from Section 2– Preliminaries

B.1 Algebraic circuit

An *algebraic circuit* $\mathcal{C}$, in $\mathbb{F}[\mathbf{x}]$, is defined as a directed acyclic graph with a unique root-vertex computing the polynomial. Each leaf-vertex is labelled by a *literal*– a variable or a field constant. Edge $u \to v$ is labelled with a field constant, which gets multiplied to the polynomial computed by vertex u and fed as input to vertex v. Each internal node-vertex is either labelled by $+$ or $\times$. A $+$ node computes the sum of all the incoming polynomials, while $\times$ node computes the product. The in-degree of a vertex is called its *fan-in*; and out-degree its *fan-out*. *Size* of an algebraic circuit is the size of the graph. *Depth* of the circuit is the length of the longest path from root to a leaf node.

An algebraic circuit with fan-out 1 is called an *algebraic formula*.

Algebraic circuits can be assumed to be layered with alternating layer of $+$ and $\times$ nodes, with the root node to be addition gate. A *depth-4* circuit is of the form $\sum\prod\sum\prod$. Thus, it computes a polynomial of the form $f = \sum_{i=1}^{k} \prod_{j=1}^{d_i} f_{ij}$, where each f_{ij} is a *sparse* polynomial.

A *depth-3* circuit $\sum\prod\sum$ computes a polynomial of the form $f = \sum_{i=1}^{k} \prod_{j=1}^{d_i} l_{ij}$, where each l_{ij} is a *linear* polynomial. A *diagonal depth-3* circuit $\sum\wedge\sum$, computes a polynomial of the form $f = \sum_{i=1}^{k} l_i^{d_i}$, where each l_i is a linear polynomial.

B.2 Algebraic branching program

An *algebraic branching program* (ABP) is a layered directed graph with a unique source vertex s and sink vertex t. The ABP of *depth-d* has $d+1$ layers– $V_0, V_1, \ldots, V_d$, where first layer $V_0 =: \{s\}$, and last layer $V_d =: \{t\}$. The directed edges go from V_i to V_{i+1}, for $0 \leq i \leq d-1$; and are labelled with *linear* polynomials from $\mathbb{F}[\mathbf{x}]$. The *weight of a path* p is $W(p) := \prod_{e \in p} W(e)$, where $W(e)$ denotes the weight (or label) of an edge. The final polynomial $f(\mathbf{x})$ *computed by the ABP* is then simply the sum of weight of all paths from source to sink: $f(\mathbf{x}) := \sum_{\text{path } p \,:s \rightsquigarrow t} W(p)$. The *length* of the ABP is the number of layers from s to t. The ABP has *width* w, if for $0 \leq i \leq d$, $|V_i| \leq w$. *Size* of the ABP is its graph size.

ABP also has an alternate algebraic representation in terms of matrix product. Let the set of vertices in i^{th}-layer V_i be $V_i =: \{v_{i,j} \mid j \in [w]\}$. Then, $f(\mathbf{x}) = \prod_{i=1}^{d} D_i$, where $D_1 \in \mathbb{F}^{1 \times w}[\mathbf{x}]$, $D_i \in \mathbb{F}^{w \times w}[\mathbf{x}]$ (for $2 \leq i \leq d-1$), and $D_d \in \mathbb{F}^{w \times 1}[\mathbf{x}]$ such that the entries are:

$$\begin{aligned} D_1(j) &:= W(s, v_{1,j}) \text{ , for } j \in [w] \\ D_i(j,k) &= W(v_{i-1,j}, v_{i,k}) \text{ , for } j,k \in [w] \text{ and } 2 \leq i \leq d-1 \\ D_d(k) &= W(v_{d-1,k}, t) \text{ , for } k \in [w] \text{ .} \end{aligned}$$

By default $W(u,v) := 0$, if there is no edge (u,v) in the ABP.

B.3 Read-once oblivious algebraic branching program (ROABP)

An ABP is called *read-once oblivious* ABP (ROABP) if each variable appears in only one layer and instead of linear polynomials, edge weights are univariate polynomials. So, ROABP has length equal to the number of variables n. The *variable order* $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$ of ROABP is the order of variables as they appear in edge weights between the layers $i-1$ to i, for $i \in [n]$ in the ROABP. *Size* of the ROABP is the sum of its graph size and the individual degrees (of the univariate edge-labels).

In the matrix product form, $D(\mathbf{x}) = \prod_{i=1}^{n} D_i$, where $D_1 \in \mathbb{F}^{1 \times w}[x_{\pi(1)}]$, $D_i \in \mathbb{F}^{w \times w}[x_{\pi(i)}]$ for $2 \leq i \leq n-1$, and $D_n \in \mathbb{F}^{w \times 1}[x_{\pi(n)}]$. One can also view D_i as a univariate polynomial with coefficients coming from w-dimensional vectors or $w \times w$ matrices. For ROABP $D(\mathbf{x})$, $D_{\leq i}$ denotes the subproduct $\prod_{j=1}^{i} D_j$, and $D_{>i}$ denotes $\prod_{j=i+1}^{n} D_j$.

We need the following lemma in Section 4.1, which is not difficult to prove (simply inspect the required coefficient).

Lemma 16. [GKST16, Lem.2.3] *Let $A(\mathbf{x})$ be a polynomial of individual degree d, computed by an ROABP of width w. Let $\mathbf{y} = (x_{i_1}, x_{i_2}, \ldots, x_{i_k})$ be any k variables of $\mathbf{x}$. Then, the coefficient polynomial $A_{(\mathbf{y},\mathbf{a})}$ can be computed by an ROABP of width w, for every $\mathbf{a} \in \{0,1,\ldots,d\}^k$. Moreover, all these ROABPs have the same variable order, inherited from the variable order of the ROABP for A.*

B.4 Hitting-set map

A *hitting-set* for a class $\mathcal{P}$ of n-variate, d-degree polynomials over $\mathbb{F}$, is defined as the set $\mathcal{H} \subseteq \mathbb{F}^n$ of field points such that, for all nonzero $f \in \mathcal{P}$, there exists at least one point $\boldsymbol{\alpha} \in \mathcal{H}$ which *hits* f, i.e. $f(\boldsymbol{\alpha}) \neq 0$. The notion of 'efficient blackbox PIT' is equivalent to 'a small-sized explicit hitting-set'. Any $\mathcal{P}$ has a hitting-set of size $(d+1)^n$; by brute-force derandomization of Polynomial Identity Lemma [Sch80, DL78, Zip79].

Most of the fast blackbox PIT algorithms give a variable-reduction map $\Phi : \mathbb{F}[\mathbf{x}] \to \mathbb{F}[\mathbf{y}]$ which converts the input polynomial f to a *constant*-variate polynomial $\Phi(f) \in \mathbb{F}[y_1, \ldots, y_k]$ such that $f = 0$ if and only if $\Phi(f) = 0$. Let D be the individual degree of $\Phi(f)$; thus Φ gives us

a hitting-set of size $(D+1)^k$ by brute-force derandomization for $\Phi(f)$. In other words, we get a poly-time blackbox PIT for f when Φ can be designed in poly-time and its individual degree D is also polynomially-bounded. We call such a map as *hitting-set map*, or blackbox PIT map, in our paper. It is also commonly known as a k-seed *hitting-set generator* (*hsg*) in literature. See [SV09, For15] for details on hitting-set and generators.

We now state the efficient *Kronecker map*, or the sparse PIT map, which gives poly-time blackbox PIT for the family of sparse polynomials. A polynomial f is called *s-sparse* if the number of monomials with nonzero coefficients in f are upper bounded by s.

Lemma 17 (Sparse PIT)**.** [Kro82, AB03, AGKS15] *Let $\mathcal{M}$ be the set of all monomials, in n variables $\mathbf{x}$, with individual degree $\leq d$. For any s, there is a (deterministic) polynomial-time constructible set of $N := ns\log(d+1)$ weight-functions $\mathbf{w}$: $\mathbf{x} \to [2NlogN]$, such that*

for any set $A \subseteq \mathcal{M}^2$ of s pairs of monomials, at least one of the weight functions $w \in \mathbf{w}$ separates all the pairs in A; i.e., for all $(m, m') \in A$, $w(m) \neq w(m')$.

C Proofs from Section 3 – Structure Theorem

We need the following lemma in the proof of our Structure Theorem 7.

Lemma 18. *Let $\mathbf{y}$ and $\mathbf{z}$ be a partition of variable set $\mathbf{x}$. Suppose $f \in \mathbb{F}[\mathbf{x}]$ is a homogeneous polynomial of degree d having a variable disjoint decomposition as $f = \sum_{i=1}^{r} f_i g_i$, where for all $i \in [r]$, $f_i \in \mathbb{F}[\mathbf{y}]$ and $g_i \in \mathbb{F}[\mathbf{z}]$. Suppose $f_1, \ldots, f_r$ are $\mathbb{F}$-linearly independent and each f_i is also a homogeneous polynomial. Then, for each $i \in [r]$, g_i is also a homogeneous polynomial.*

Proof. For the sake of contradiction, suppose there exists a g_k, for some $k \in [r]$, which is not homogeneous. Let f_k be its corresponding polynomial which is homogeneous and has degree, say d_k. Since f is homogeneous of degree d, let $g_k = g_k^{[d-d_k]} + g_k^{[\neq(d-d_k)]}$, where $g_k^{[d-d_k]}$ is the degree $(d-d_k)$ homogeneous part of g_k and $g_k^{[\neq(d-d_k)]}$ is the rest of the polynomial. We will prove that latter part has to be zero.

Let $\mathbf{z}^{\mathbf{a}}$ be any monomial in $g_k^{[\neq(d-d_k)]}$ with coefficient $c_k \neq 0$, where degree of monomial $|\mathbf{a}|_1 \neq d-d_k$. The nonzero term $f_k \cdot c_k \mathbf{z}^{\mathbf{a}}$ in f has to get cancelled since it is of degree $d_k + |\mathbf{a}|_1 \neq d$. Observe that this term can get cancelled only by product of $\mathbf{z}^{\mathbf{a}}$ with those f_i that have degree d_k (simply by variable disjointness & degree comparison). For $\ell \leq r$, let $f_{i_1}, f_{i_2}, \ldots, f_{i_\ell}$ be the polynomials in $\{f_1, \ldots, f_r\}$ of degree exactly d_k. Let $\text{coeff}(g_i)(\mathbf{z}^{\mathbf{a}}) =: c_i$, for $i \in [r]$; where c_i can be possibly zero except for c_k. Then,

$$\begin{aligned} f_{i_1} \cdot (c_{i_1}\mathbf{z}^{\mathbf{a}}) + f_{i_2} \cdot (c_{i_2}\mathbf{z}^{\mathbf{a}}) + \ldots + f_{i_\ell} \cdot (c_{i_\ell}\mathbf{z}^{\mathbf{a}}) &= 0 \\ \Rightarrow \; c_{i_1} f_{i_1} + c_{i_2} f_{i_2} + \ldots + c_{i_\ell} f_{i_\ell} &= 0\,. \end{aligned}$$

Since $c_k \neq 0$, this contradicts $\mathbb{F}$-linear independence of $f_1, \ldots, f_r$. Thus, $g_k^{[\neq(d-d_k)]}$ is zero. Hence, $\forall k \in [r]$, g_k is a homogeneous polynomial; of degree $d - \deg(f_k)$. □

In the following lemma we prove that the lead homogeneous part of a polynomial f can be computed by an ROABP in the *same* width as that of f. For our paper, we only prove for the highest degree homogeneous component, but similar proof holds for the lowest degree homogeneous component.

Lemma 19 (Homogeneous-part width)**.** *Let $f(\mathbf{x}) = f^{[d]} + f^{[<d]}$ be a polynomial of degree-d, in $\mathbb{F}[x_1, \ldots, x_n]$; where $f^{[d]}$ is the (lead) degree-d homogeneous component of f, and $f^{<[d]}$ is the rest of the polynomial f. Then, width$(f^{[d]}) \leq$ width(f), in the same variable order.*

Proof. Let $f^{[d]}$ have an ROABP in a variable order $y_1 < y_2 < \ldots < y_n$, where $y_i = x_{\pi(i)}$ for some permutation $\pi : [n] \to [n]$. Let $\text{width}(f^{[d]}) =: k$. For a fixed $\ell \in [n]$, consider the partition $\{y_1, \ldots, y_\ell\} \sqcup \{y_{\ell+1}, \ldots, y_n\}$. Wlog, there are k coefficient polynomials of $f^{[d]}$– $g_1, \ldots, g_k \in \mathbb{F}[\mathbf{y}_{>\ell}]$ –that are $\mathbb{F}$-linearly independent. For some $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \{0, 1, \ldots, d\}^\ell$, these are precisely $g_i =: (f^{[d]})_{(\mathbf{y}_{\leq \ell}, \mathbf{e}_i)}$, for each $i \in [k]$. We claim that the k coefficient-operators $\mathbf{e}_1, \ldots, \mathbf{e}_k \in \{0, 1, \ldots, d\}^\ell$, that worked for $f^{[d]}$, will also 'work' for f.

Formally, the set of polynomials $\{f_{(\mathbf{y}_{\leq \ell}, \mathbf{e}_1)}, \ldots, f_{(\mathbf{y}_{\leq \ell}, \mathbf{e}_k)}\}$ will also be $\mathbb{F}$-linearly independent. Let h_i be the polynomial (for each $i \in [k]$):

$$f_{(\mathbf{y}_{\leq \ell}, \mathbf{e}_i)} \;=\; (f^{[d]})_{(\mathbf{y}_{\leq \ell}, \mathbf{e}_i)} \;+\; (f^{[<d]})_{(\mathbf{y}_{\leq \ell}, \mathbf{e}_i)} \;=:\; g_i \;+\; h_i \,.$$

Here, $\forall i \in [k]$, h_i is of degree strictly less than that of g_i. Observe that the coefficient-operators $(f^{[d]})_{(\mathbf{y}_{\leq \ell}, \mathbf{e}_i)}$ respect homogeneity. Therefore, $\forall i \in [k]$, g_i is a nonzero *homogeneous* polynomial of degree $d_i := d - |\mathbf{e_i}|_1$. Since $g_1, \ldots, g_k$ are $\mathbb{F}$-linearly independent, any $\mathbb{F}$-linear combination $c_1 g_1 + c_2 g_2 + \ldots + c_k g_k$ is nonzero, whenever $c_i \in \mathbb{F}$ are not all zero. Now, we prove our claim that the polynomials $g_1 + h_1, \ldots, g_k + h_k$ are $\mathbb{F}$-linearly independent.

Suppose not, then there exist $c_1, \ldots, c_k \in \mathbb{F}$ not all zero such that

$$\begin{aligned} c_1(g_1 + h_1) + \ldots + c_k(g_k + h_k) \;&=\; 0 \\ \Rightarrow \quad c_1 g_1 + \ldots + c_k g_k \;&=\; -(c_1 h_1 + \ldots + c_k h_k) \,. \end{aligned} \tag{11}$$

Let $d' := \max_i\{\deg(g_i) \mid c_i \neq 0\}$. Note that LHS above is a nonzero polynomial of degree exactly d'. This is because g_i are homogeneous; so, if degree of LHS is $< d'$, then all the g_i of degree d' have to cancel among themselves. This cannot happen since they are linearly independent. Thus, LHS is of degree d'. But, RHS is a polynomial of degree $< d'$; since, $\deg(h_i) < \deg(g_i) \leq d'$, for each $i \in [k]$. This contradicts Eqn.11, thus proving $\{g_1 + h_1, \ldots, g_k + h_k\}$ to be $\mathbb{F}$-linearly independent. We conclude that $\text{width}(f) \geq k$. □