BLACKBOX IDENTITY TESTING FOR SUM OF SPECIAL ROABPS AND ITS BORDER CLASS

Pranav Bisht and Nitin Saxena

April 21, 2021

Abstract. We look at the problem of blackbox polynomial identity testing (PIT) for the model of read-once oblivious algebraic branching programs (ROABP), where the number of variables are logarithmic to the input size of ROABP. We restrict width of ROABP to a constant and study the more general sum-of-ROABPs model. This model is non-trivial due to the arbitrary individual-degree. 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; Computational Complexity'17) which is comparable to brute-force in the log-variate setting. We also show that we can work with unbounded-many such ROABPs if each ROABP computes a homogeneous polynomial (or more generally for *degree-preserving sums*). We also give poly-time PIT for the *border*.
We introduce two new techniques, both of which also work for the border version of the stated models. (1) The leading-degree-part of an ROABP can be made *syntactically homogeneous* in the same width. (2) There is a direct reduction from PIT of sum-of-ROABPs to PIT of single ROABP (over *any* field). Our methods improve the time complexity for PIT of sum-of-ROABPs in the log-variate regime.

Keywords. Identity test, hitting-set, ROABP, blackbox, log variate, width, diagonal, derandomization, homogeneous, sparsity, border complexity.

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

Contents

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* (Demillo & Lipton 1978; Ore 1922; Schwartz 1980; Zippel 1979). 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 (Agrawal 2005; Agrawal *et al.* 2019; Heintz & Schnorr 1980; Kabanets & Impagliazzo 2004), geometric complexity theory (Mulmuley 2012a,b) and many other well known problems like matching (Fenner *et al.* 2017; Mulmuley *et al.* 1987), primality testing (Agrawal *et al.* 2004) and polynomial factoring (Kopparty *et al.* 2014). Refer to Saptharishi (2016); Saxena (2009, 2014); Shpilka & Yehudayoff (2010) 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 special class of ABPs. An *ABP* is defined using a layered directed graph with a 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 labeled 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 $f(\mathbf{x}) = D_1(x_{\pi(1)}) \cdot D_2(x_{\pi(2)}) \cdots D_n(x_{\pi(n)})$ 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 $\{1, 2, \ldots, n\}$ and it describes what we call the *variable order* of an ROABP. Size of an ROABP is given by three parameters: w, n and degree d.

In the whitebox regime, ROABP has a well known poly-time PIT algorithm (Raz & Shpilka 2005). However, in the blackbox regime, we only have quasi-poly-time PIT algorithms (Agrawal *et al.* 2015; Forbes & Shpilka 2013b) and no known poly-time algorithms. Thus, one can also ask for blackbox PIT of ROABPs with restriction on the width parameter. In Gurjar *et al.* (2017a) they address this question and give a poly-time graybox (known variable order) PIT for constant width ROABPs. The constant width setting can be considered a necessary stepping stone before solving PIT for general width ROABPs.

The sum of ROABPs is another interesting model for PIT. For a constant number of ROABPs, Gurjar *et al.* (2017b) 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 give a positive answer for this question.

Blackbox Polynomial Identity Testing for sum of constantly-many, log-variate constant-width ROABPs is in polynomial time.

Under the single restriction of log-variate, we still get an improvement over Gurjar *et al.* (2017b). More generally, we give efficient PIT for the *border* version. Furthermore, *if the sum is degree preserving then we can drop the 'constantly-many' restriction.*

1.1. Our results.

PIT for sum of ROABPs. The main result of this work is showing a reduction from designing a blackbox PIT algorithm for sum of ROABPs to designing a blackbox PIT algorithm for a single ROABP.

THEOREM 1.1 (Reduction to one). *Let $T(r,n,d)$ be the time complexity of a blackbox PIT algorithm for a single ROABP of width r and degree d in n variables over any field $\mathbb{F}$. Then, blackbox PIT for sum of c-many ROABPs, each of width r and degree d in n variables, can be solved in time $T'(r,n,d,c) = \left(2^n \cdot T(2^c r^{3^c}, n, d)\right)^{O(c)}$ over $\mathbb{F}$.*

This reduction is poly-time when number of ROABPs, c is constant and number of variables is logarithmic in the input size, that is $n = O(\log(rd))$. Thus, in the log-variate setting, if we have a poly-time blackbox PIT for a single ROABP, then we show a poly-time blackbox PIT for sum of c ROABPs. Though, sum of even two ROABPs is provably harder than a single ROABP (see Fact 1.6), we still get an efficient PIT for sum using PIT for single ROABP.

Blackbox PIT for a single ROABP, over any field $\mathbb{F}$, has time complexity $(ndr)^{O(\log n)}$, which is only quasi-poly time. In the log-variate setting, a $d^n = d^{O(\log(rd))}$ (quasi-poly) time algorithm for sum of ROABPs is already trivial via brute force derandomization based on the Polynomial Identity Lemma. Thus, to extract a poly-time PIT for sum using above theorem, we need poly-time blackbox PIT for single ROABP. We indeed get one, when width r is constant and $n = O(\log d)$, in Lemma 3.8. This then gives us the following corollary.

COROLLARY 1.2 (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 degree d. Also, the variable order of each ROABP is unknown. Then, blackbox PIT for $\mathcal{P}$ can be solved in poly(d^c, r^{nc3^c}) time.*

Remark. *If $c, r = O(1)$ and $n = O(\log d)$, then input size is $O(d)$ and the stated time-complexity is poly(d).*

The trivial time complexity for blackbox PIT of $\mathcal{P}$ is $d^n = d^{O(\log d)}$. Gurjar *et al.* (2017b) gave a blackbox PIT algorithm for sum of c ROABPs in $(ndr)^{O(c2^c \log(ndr))}$ time. It is super-polynomial time, even under the restrictions of constant-c, constant-width and log-variate.

Gurjar *et al.* (2017a) gave a blackbox PIT algorithm for a single ROABP in $O(ndr^{\log n})$ time. It is poly-time for constant width without the log-variate restriction. However, their algorithm assumes the knowledge of variable order. Moreover, it works only for fields of characteristic either zero or larger than $ndr^{\log n}$. Our algorithm is efficient in the log-variate setting, does not require knowledge of variable order and works for all fields.

If we use Agrawal *et al.* (2015)'s blackbox PIT algorithm for single ROABP in Theorem 1.1, we get another efficient PIT for the sum of ROABPs as a corollary below.

Corollary 1.3 (Improved Sum PIT). *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 degree d. Also, the variable order of each ROABP is unknown. Then, blackbox PIT for $\mathcal{P}$ can be solved in poly$(2^{cn} \cdot n^{c \log n}, d^{c \log n}, r^{3^c \log n})$ time.*

Remark. *For $c = O(1)$ and $n = O(\log(rd))$, the stated time complexity is $(rd)^{O(\log\log(rd))}$. Thus, in the log-variate setting, without any width restriction, we give a more efficient PIT than the result of Gurjar* et al. *(2017b), which only yields a poly$(rd)^{O(\log(rd))}$ time algorithm, even with $n = O(\log(rd))$.*

In a subsequent work, Guo & Gurjar (2020) improved the blackbox PIT for a single ROABP. One can then also use that in conjunction with our Theorem 1.1 to get a further improved blackbox PIT for sum of ROABPs in the log-variate regime. This has been stated as Theorem 1.2 in Guo & Gurjar (2020) by citing an earlier version of Theorem 1.1 in this work.

PIT for degree-preserving sum of ROABPs. For some $k \in \mathbb{N}$, let $f_1(\mathbf{x}), f_2(\mathbf{x}) \ldots f_k(\mathbf{x})$ be any k polynomials in $\mathbb{F}[x_1, \ldots, x_n]$. We call $\sum_{i=1}^{k} f_i(\mathbf{x})$, a *degree-preserving sum*, if for $f(\mathbf{x}) = \sum_{i=1}^{k} f_i(\mathbf{x})$, we have $\deg(f) = \max_i \deg(f_i)$. In Corollary 1.2, the number of ROABPs c, is assumed to be constant to get efficient blackbox PIT. We could allow an arbitrary c, if the sum is degree-preserving. In other words, with the additional restriction of a degree-preserving sum, we bring down the double exponential dependence on c in Corollary 1.2 to polynomial dependence on c in the theorem below.

Theorem 1.4 (Degree-preserving sum). *Let $\mathcal{P}$ be a set of n-variate polynomials, over a field $\mathbb{F}$, computed by a degree-preserving sum of c ROABPs, each of width r and degree d. Also, the variable order of each ROABP is unknown. Then, blackbox PIT for $\mathcal{P}$ can be solved in poly(d, cr^n) time.*

Remark. *If $r = O(1)$ and $n = O(\log d)$, then the stated time complexity is poly(cd)– polynomial in the input-size. Consider the class of polynomials which can be computed by a sum of c ROABPs, where each ROABP computes a homogeneous polynomial. In Section 3, we show that such a sum can be expressed as a degree-preserving sum. Thus, we get a blackbox PIT for this class in the same time.*

If we could get a poly(d, cr^n) time PIT in the above theorem without the degree-preserving sum restriction, then we get poly-time PIT for the model of diagonal-depth 3 circuits (See Lemma 2.12).

PIT for border of sum of ROABPs. Let $\mathcal{C}$ be an algebraic class over field $\mathbb{F}$, like arithmetic circuit or ABP or ROABP. An *approximation closure* or *border* of class $\mathcal{C}$, denoted as $\overline{\mathcal{C}}$ is defined as follows: a family (f_n) is in $\overline{\mathcal{C}}$ if there are polynomials $f_{n,1}, \ldots, f_{n,t} \in \mathbb{F}[\mathbf{x}]$ such that the family (g_n) defined by

$$g_n(\mathbf{x}, \epsilon) := f_n(\mathbf{x}) + \epsilon f_{n,1}(\mathbf{x}) + \epsilon^2 f_{n,2}(\mathbf{x}) + \ldots + \epsilon^t f_{n,t}(\mathbf{x})$$

is in $\mathcal{C}$ over the field $\mathbb{F}(\epsilon)$, where t is called the error degree. Here ϵ is a new indeterminate and $\lim_{\epsilon \to 0} g_n(\mathbf{x}, \epsilon) = f_n(\mathbf{x})$. In other words, f_n is approximated by a polynomial g_n which has a circuit in class $\mathcal{C}$ over $\mathbb{F}(\epsilon)$. Although the circuit $C \in \mathcal{C}$ computing g_n might involve internal computations with ϵ in the denominator, the final output does not and is a polynomial over $\mathbb{F}[\epsilon][\mathbf{x}]$. Border classes can be more complicated because the degree of ϵ involved can be super-polynomial. So, poly-time PIT algorithms for border classes are rare. Below, we give blackbox PIT algorithm for the border class of sum of c ROABPs.

Theorem 1.5 (Reduction for border). *Let $T(r, n, d)$ be the time complexity of a blackbox PIT algorithm for a single ROABP of width r in n variables and degree d over any field $\mathbb{F}$. Then blackbox PIT for border of sum of c ROABPs, each of width r and degree d in n variables, can be solved in time $\left(2^n \cdot T(2^c r^{3^c}, n, d)\right)^{O(c)}$ over $\mathbb{F}$.*

Remark. *Since the above theorem achieves the same time complexity as in Theorem 1.1, all results for PIT of sum of ROABPs above extend to their border versions also.*

Thus, we also get $\text{poly}(d^c, r^{nc3^c})$ and $\text{poly}(2^{cn} \cdot n^{c \log n}, d^{c \log n}, r^{3^c \log n})$ time blackbox PIT algorithms for the border class of sum of c ROABPs analogous to Corollary 1.2 and Corollary 1.3, respectively. We will also get a new blackbox PIT for the border class with the time complexity achieved in Theorem 1.2 of Guo & Gurjar (2020).

1.2. Previous works and motivation.

In this section, we will discuss the major motivations behind this work by showing connections of ROABP with other algebraic models of computation. We refer the reader to Section 2 for formal definitions of these models.

1.2.1. ABPs.

It is well known that ABPs subsume determinants and formulas. In turn, algebraic formulas subsume constant depth circuits (see Ben-Or & Cleve 1992 and Nisan 1991, Lem. 1). The combined restrictions on variables and width still gives interesting sub-models for ABPs. For example Agrawal *et al.* (2019, Thm.22)

Model	**Time**	**Reference**
$\sum\bigwedge\sum$	$(nd)^{O(\log n)}$	Agrawal *et al.* (2013)
—	$\mathrm{poly}(d, 2^n)$	Forbes *et al.* (2018)
ROABP	$(ndr)^{O(\log n)}$	Forbes & Shpilka (2013b)
—	$n^{O(d \log r \log n)}$	Forbes *et al.* (2014)
—	$(ndr)^{O(\log n)}$	Agrawal *et al.* (2015)
—	$O(ndr^{\log n})$	Gurjar *et al.* (2017a)
Sum of c —	$(ndr)^{O(c2^c \log(ndr))}$	Gurjar *et al.* (2017b)
Border version	$(2^n(nd)^{\log n} r^{3^c \log n})^{O(c)}$	**This** work
—	$\mathrm{poly}(d^c, r^{nc3^c})$	**This** work

Table 1.1: Time complexities of different PIT algorithms related to n-variate, degree d and width r ROABP model.

show that even solving PIT for log-variate width-2 *ABPs* will almost solve the complete PIT problem. The ROABP model is also quite nontrivial as a poly-time blackbox PIT is still open. Table 1.1 gives a comprehensive comparison between the time complexities of previous works on ROABP model and this work. In the table, algorithms of Forbes & Shpilka (2013b) and Gurjar *et al.* (2017a) work only for known variable orders. The work of Forbes *et al.* (2014) gives quasi-poly time PIT under the restriction of multilinearity or constant individual degree. Our border algorithms mentioned in Table 1.1 naturally also hold for the base class of sum of c ROABPs.

1.2.2. Log-variate. There has been a recent line of work on 'Bootstrapping variables' in algebraic circuits. Agrawal *et al.* (2019) 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. Kumar *et al.* (2019), Guo *et al.* (2019a) further showed that even saving on *one* evaluation point from the brute-force hitting-set of constant-variate algebraic circuits would solve general PIT. Although such bootstrapping results are not known for ROABPs, nonetheless log-variate ROABP is still an open interesting model for the reasons discussed below.

The well studied diagonal depth-3 model $\left(\sum\bigwedge\sum\right)$ is one of the lower hanging fruits in PIT. Forbes *et al.* (2018) were able to utilize low-variate setting 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 the general multivariate diagonal depth-3 model using the results of Forbes *et al.* (2014); Forbes & Shpilka (2013a). See Lemma 2.12 for details. Making progress in this direction has been the key motivation behind the tools and techniques developed in this work.

1.2.3. Constant width. The sum of few constant-width ROABPs model is more expressive than that of a constant-width ROABP. Kayal *et al.* (2016) observed that even a sum of two width-3 ROABPs cannot be computed by a single constant-width ROABP which is stated as Fact 1.6 here. Thus, this model is nontrivial and blackbox PIT for it is still open. Even in the log-variate setting, sum of two width-3 ROABPs will require a single ROABP of superconstant width. Thus, sum of constantly many, constant width, log-variate ROABPs lacked a poly-time blackbox PIT, which we solve in Corollary 1.2.

FACT 1.6 (Kayal *et al.* 2016, 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 $2^{\Omega(n)}$.*

1.2.4. PIT for Border. Border of a class can offer additional computational power. Forbes (2016) gives an interesting example. Consider the class of polynomials which are of the form $\alpha\ell_1^d + \beta\ell_2^d$, where $\ell_1, \ell_2 \in \mathbb{F}[\mathbf{x}]$ are homogeneous linear polynomials. For $d \geq 3$, it can be shown that $x^{d-1}y$ cannot be expressed as $\alpha\ell_1^d + \beta\ell_2^d$. However, $x^{d-1}y$ can be computed in the border of this class, as shown below

$$\begin{aligned} g &:= \frac{1}{d\epsilon}\left((x+\epsilon y)^d - x^d\right) \\ \lim_{\epsilon\to 0} g &= x^{d-1}y \end{aligned}$$

The fundamental problem in border complexity is to understand this difference in the computational power of a class $\mathcal{C}$ and its border class $\overline{\mathcal{C}}$. Generally, one would like to understand whether $\mathrm{VP} = \overline{\mathrm{VP}}$, where VP is the class of polynomials with polynomial sized algebraic circuits. However, this question is open even for more restricted classes like diagonal depth-3 circuits, depth-3 circuits, ABPs etc. Border classes play an important role in Mulmuley's 'GCT approach' to attack $\mathrm{P} \neq \mathrm{NP}$ conjecture (Mulmuley & Sohoni 2001, 2008), and in 'GCT Chasm' (Mulmuley 2012a,b). In particular, showing $\overline{\mathrm{VP}} \neq \mathrm{VNP}$ is an important structural question. We refer the reader to Bringmann *et al.* (2018) for an excellent discussion on border complexity.

An interesting question in border complexity is to come up with PIT algorithm for a border class. Of course to solve PIT for a border class $\overline{\mathcal{C}}$, we necessarily need to solve PIT for $\mathcal{C}$. However for those circuit classes where we have an efficient PIT algorithm for $\mathcal{C}$ and it is not known whether $\mathcal{C} = \overline{\mathcal{C}}$, it is an interesting problem to solve PIT for $\overline{\mathcal{C}}$.

For the algebraic class of ROABPs, one can show that the border does not offer any additional computation power, that is, $\overline{\mathrm{ROABP}} = \mathrm{ROABP}$ (Lemma 5.2). However, it is not clear if sum of constantly many ROABPs is equal to its border class. See Section 5.2 for a discussion on this. Therefore PIT for the border class of sum of ROABPs is an interesting problem. We solve it for sum of constantly-many, constant width, log-variate ROABPs.

1.3. Proof techniques.

1.3.1. Syntactic homogeneity for ROABP. Inspired by circuits we define *syntactic homogeneity* for ROABP (Definition 3.1). 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 3.3). Recall that if one applies the usual *homogenization* trick, for circuits/ABP (Shpilka & Yehudayoff 2010, 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. Moreover, Theorem 3.3 is independent of any restrictions on width or number of variables but we make use of these restrictions only in solving PIT.

1.3.2. Reduction from many to one. In Theorem 1.1, we give a reduction from designing a polynomial time blackbox PIT for sum of constantly many ROABPs to designing a polynomial time blackbox PIT for a single ROABP, in the log-variate setting. This reduction does not assume any restriction on width of ROABPs. This already gives us an improvement over Gurjar *et al.* (2017b) in Corollary 1.3. We need constant width in Corollary 1.2 only because poly-time blackbox PIT for an unbounded-width (log-variate) ROABP is yet to be found.

REMARK 1.7. *In a recent subsequent work, Guo & Gurjar (2020, Thm. 1.1) constructed explicit hitting sets of polynomial size for a single log-variate ROABP of width upto* $2^{O(\log d/ \log\log d)}$ *over field* $\mathbb{F}$ *with* $\mathrm{char}(\mathbb{F}) = 0$ *or* $\mathrm{char}(\mathbb{F}) > d$. *In their second theorem, they directly call our Theorem 1.1 to extend their result to sum of constantly many such ROABPs.*

Gurjar *et al.* (2017b) also give a reduction from PIT of sum of constantly many to PIT of single ROABP but it is an indirect one, taking quasi-polynomial time. Also, instead of a hitting set, they require something stronger, which a hitting set may not provide. They need an efficient shift that l-concentrates a single ROABP. A polynomial is said to be l-concentrated if all of its coefficients are in the linear span of its coefficients corresponding to monomials having variable support $< l$. They prove that this efficient shift also l-concentrates sum of constantly-many ROABPs. Although, they do not require log-variate restriction, their method only yields a quasi-poly time reduction, since they fix l to $O(\log s)$ (s is size of ROABP) and apply brute-force hitting set after the shift.

1.4. Organization. In Section 2, we recall preliminary tools and techniques that will be useful for rest of the paper. We first discuss the degree-preserving sum of ROABPs model in Section 3. Here, we develop a structure theorem (Theorem 3.3) and use it to prove Theorem 1.4. We also give a blackbox PIT algorithm for a single ROABP here. We discuss the more general sum of ROABPs model in Section 4. Here, we prove our reduction given in Theorem 1.1 and use it to prove Corollary 1.2 and Corollary 1.3. We give the border PIT results in Section 5.2. We prove Theorem 1.5 here. Finally, we conclude with few open questions in Section 6.

2. Notations and Preliminaries

2.1. Notations. We follow some of the notations from Gurjar *et al.* (2017b). 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 $\mathbf{x}_{\le k}$ 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 $\prod_{i=1}^{n} x_i^{a_i}$ and $|\mathbf{a}|_1$ denotes the degree of this monomial. Polynomial A is said to have individual degree b if $\deg_{x_i}(A) \le b$ for all $i \in [n]$. Let $\mathrm{coeff}(A)(\mathbf{x}^{\mathbf{a}}) \in \mathbb{F}$ 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* $A_{(\mathbf{y},\mathbf{a})}$, 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 $A(\mathbf{x})$. Observe that $A_{(\mathbf{x},\mathbf{a})}$ and $\mathrm{coeff}(A)(\mathbf{x}^{\mathbf{a}})$ 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 $\mathrm{coeff}(A)(x_1) = 2$.

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 *coefficient space* of $A(\mathbf{x})$ is defined as the span of all the coefficients of A: $\mathrm{span}_{\mathbb{F}}\{\mathrm{coeff}(A)(\mathbf{x}^{\mathbf{a}}) \mid \mathbf{a} \in \{0, 1, \ldots, d\}^n\}$. We can also define it for any prefix of variables.

For a set of polynomials $\mathcal{P}$, their $\mathbb{F}$-span is defined as: $\mathrm{span}_{\mathbb{F}}\mathcal{P} :=$

$\left\{ \sum_{A\in\mathcal{P}} \alpha_A \cdot A \mid \alpha_A \in \mathbb{F} \right\}$. The set $\mathcal{P}$ is called $\mathbb{F}$-linearly independent if $\sum_{A\in\mathcal{P}} \alpha_A \cdot A = 0$ implies $\alpha_A = 0$ for all $A \in \mathcal{P}$. $\mathrm{Dim}_{\mathbb{F}}\mathcal{P}$ is then defined as cardinality of the largest $\mathbb{F}$-linearly independent subset of $\mathcal{P}$.

2.2. Algebraic models of computation.

Algebraic circuit. An *algebraic circuit* or an arithmetic circuit C over $\mathbb{F}[\mathbf{x}]$, is defined as a directed acyclic graph with a unique root-vertex computing the polynomial. Each leaf-vertex is labeled by a *literal* – a variable or a field constant. Edge $u \to v$ is labeled 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 labeled 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 $\Sigma\Pi\Sigma\Pi$. 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 $\Sigma\Pi\Sigma$ computes a polynomial of the form $f = \sum_{i=1}^{k} \prod_{j=1}^{d_i} \ell_{ij}$, where each ℓ_{ij} is a *linear* polynomial. A *diagonal depth-3* circuit $\Sigma\wedge\Sigma$, computes a polynomial of the form $f = \sum_{i=1}^{k} \ell_i^{d_i}$, where each ℓ_i is a linear polynomial.

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 $0 \le i \le d-1$ and are labeled 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.

Read-once oblivious algebraic branching program. 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* $(x_{\pi(1)}, \ldots, x_{\pi(n)})$ 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, ROABP $D(\mathbf{x}) = \prod_{i=1}^{n} D_i$, 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 sub product $\prod_{j=1}^{i} D_j$, and $D_{>i}$ denotes $\prod_{j=i+1}^{n} D_j$.

We state the definition of *characterizing dependencies* which defines an ROABP layer by layer.

Definition 2.1 (Gurjar *et al.* 2017b, Defn. 2.7). *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).*

2.3. Nisan's characterization. Nisan (1991) gave an exact width characterization for ROABPs. We follow the presentation of Gurjar *et al.* (2017b) for this characterization.

LEMMA 2.2 (Gurjar *et al.* 2017b, Lem. 2.4, 2.8). *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* $k \in [n]$, *let* $\mathbf{y} = (x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(k)})$ *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 need the following lemma later in Section 4.1, which is not difficult to prove (simply inspect the required coefficient).

LEMMA 2.3 (Gurjar *et al.* 2017b, Lem. 2.6). *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.*

2.4. Hitting set generator (HSG). 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 (Demillo & Lipton 1978; Ore 1922; Schwartz 1980; Zippel 1979).

We also have the notion of generator (hitting set generator) which is equivalent to a hitting set but is easier to work with PIT algorithms, especially recursive PIT algorithms.

Definition 2.4 (Generator). *Let $\mathcal{P}$ be a class of n-variate polynomials. Consider $\Phi = (f_1, f_2, \ldots, f_n)$, a tuple of k-variate polynomials where for each $i \in [n]$, $f_i \in \mathbb{F}[t_1, t_2, \ldots, t_k]$. Let $A(x_1, \ldots, x_n)$ be an n-variate polynomial. Define $\Phi : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{F}[t_1, \ldots, t_k]$ as $\Phi(A) = A(f_1, \ldots, f_n)$. We call Φ a k-seeded generator for class $\mathcal{P}$ if for every non-zero $A(\mathbf{x}) \in P$, $\Phi(A(\mathbf{x})) \neq 0$. Degree of generator Φ is defined as* $\deg(\Phi) := \max\{\deg(f_i)\}_{i=1}^n$.

For poly-time PIT, k should be constant. A generator $\Phi \in \mathbb{F}[\mathbf{t}]^n$ acts as a variable reduction map which converts an input polynomial $f \in \mathbb{F}[x_1, \ldots, x_n]$ to $\Phi(f) \in \mathbb{F}[t_1, \ldots, t_k]$ such that $f = 0$ if and only if $\Phi(f) = 0$. Let D be the degree of Φ, which makes $\Phi(f)$ a polynomial of individual degree dD. Thus, Φ gives us a hitting-set of size $(dD+1)^k$ by brute-force derandomization for $\Phi(f)$. In other words, we get a poly-time blackbox PIT for f when k is constant, Φ can be designed in poly-time and its degree is also polynomially bounded. An HSG for class $\mathcal{P}$ also yields a hitting set for $\mathcal{P}$. See Forbes (2015); Shpilka & Volkovich (2009) for equivalence of hitting-sets and generators.

Below we state the folklore trick of polynomial interpolation which recovers coefficients of a univariate polynomial from sufficiently many evaluations of the polynomial.

Lemma 2.5 (Lagrange Interpolation). *Let $\alpha_1, \ldots, \alpha_k$ be any k distinct points in $\mathbb{F}$. Suppose we are given evaluations of a polynomial $f(x) \in \mathbb{F}[x]$ of degree $k-1$ at these k points, $\beta_i = f(\alpha_i)$ for each $i \in [k]$. Then we can recover f as follows:*

$$\ell_i(x) = \prod_{\substack{1 \le j \le k \\ j \ne i}} \frac{x - \alpha_j}{\alpha_i - \alpha_j}$$

$$f(x) = \sum_{i=1}^{k} \beta_i \ell_i(x)$$

Proof. Observe that $\ell_i(\alpha_j) = 0$ for $i \ne j$ and 1 when $i = j$. Thus, $f(\alpha_i) = \beta_i$. Also $f(x)$ is the unique degree $k-1$ polynomial with these evaluations, since if there is another polynomial $g \ne f$ with same k evaluations, then $f - g$ is a non-zero polynomial of degree $\le k-1$ having k roots, which is a contradiction. □

Often we have a set of candidate maps for a class of polynomials $\mathcal{P}$, such that for each polynomial $f \in \mathcal{P}$, one of the maps in the set acts as an HSG for that particular f. The following lemma shows that we can replace these set of candidate generators with a single HSG for class $\mathcal{P}$.

Lemma 2.6 (Generator Interpolation). *Let $G = \{\Phi_1, \ldots, \Phi_k\}$ where each $\Phi_i \in \mathbb{F}[t]^n$. Suppose G is a set of candidate generators for a class of n-variate polynomials $\mathcal{P}$ such that for any non-zero $f \in \mathcal{P}$, there exists $i \in [k]$, $\Phi_i(f) \ne 0$. Then, there exists a single generator $\Psi \in \mathbb{F}[t, y]^n$ such that for every non-zero $f \in \mathcal{P}$, $\Psi(f) \ne 0$. Moreover,* $\deg_t(\Psi) = \max\{\deg(\Phi)\}_{\Phi \in G}$ *and* $\deg_y(\Psi) = |G| - 1 = k - 1$.

Proof. Let $\{\alpha_1, \ldots, \alpha_k\}$ be an arbitrary set of distinct constants. Define $\Psi \in \mathbb{F}[t, y]^n$ to be the Lagrange interpolation polynomial as follows:

$$\Psi = \sum_{i=1}^{k} \left(\prod_{\substack{1 \le j \le k \\ j \ne i}} \frac{y - \alpha_j}{\alpha_i - \alpha_j} \right) \Phi_i .$$

Observe that $\Psi|_{y=\alpha_i} = \Phi_i$ for each $i \in [k]$. We know that for any non-zero $f \in \mathcal{P}$, there exists $i \in [k]$, $\Phi_i(f) \neq 0$. Therefore $\Psi(f)|_{y=\alpha_i} \neq 0$. Hence $\Psi(f) \neq 0$ as a polynomial in $\mathbb{F}[y,t]$ since its evaluation at $y = \alpha_i$ is non-zero. □

2.5. Folklore facts. We mention few well known relevant tools and lemmas in this section. We start with the famous Kronecker map.

Lemma 2.7 (Kronecker 1882). *Let $f(\mathbf{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero multivariate polynomial of individual degree $< d$. Let the Kronecker HSG, $\Phi \in \mathbb{F}[t]^n$ be defined as $(t^{d^0}, t^{d^1}, \ldots, t^{d^{n-1}})$. Then, $\Phi(f) \neq 0$.*

Below we prove that a k-seeded HSG can be replaced with a single seed HSG with appropriate parameters.

Lemma 2.8. *Let $\Phi \in \mathbb{F}[t_1, \ldots, t_k]^n$ be a k-seeded hitting set generator for some class $\mathcal{P}$ of n-variate, degree d polynomials in $\mathbb{F}[\mathbf{x}]$ such that $\deg\{\Phi\} = D$. Then, there is a univariate (single seed) hitting set generator $\Psi \in \mathbb{F}[z]^n$ for class $\mathcal{P}$ with $\deg(\Psi) \leq (dD+1)^k$.*

Proof. Let $f \in \mathcal{P}$ be a non-zero polynomial. Since Φ is a generator for f, $\Phi(f) \neq 0$. Now, consider the Kronecker HSG $\Gamma = (z^{B^0}, z^{B^1}, \ldots, z^{B^{k-1}})$ for the polynomial $\Phi(f) \in \mathbb{F}[t_1, \ldots, t_k]$, where we set $B := dD + 1$. Since $\deg(\Phi(f)) < B$, by Lemma 2.7, $\Gamma(\Phi(f)) \neq 0$. Thus, we get a univariate HSG $\Psi := \Gamma \circ \Phi \in \mathbb{F}[z]^n$ such that for a non-zero $f \in \mathcal{P}$, $\Psi(f) \neq 0$. Observe that $\deg(\Psi) = \deg(\Gamma).\deg(\Phi) = B^{k-1}.D \leq (dD+1)^k$ □

Blackbox PIT for the class of sparse polynomials is achieved by the following lemma.

Lemma 2.9 (Sparse HSG; Klivans & Spielman 2001). *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero polynomial of individual degrees $\leq d$ such that $\mathrm{sparsity}(f) \leq m$. Let p be a prime larger than $\max(d, mn+1)$. Then, there is some $k \in [mn+1]$ such that the univariate polynomial $f'(y) := f(y, y^{k^1 \bmod p}, \ldots, y^{k^{n-1} \bmod p})$ is non-zero. This yields a HSG $\Psi \in \mathbb{F}[t]^n$ for the class of m-sparse polynomials such that $\deg(\Psi) = poly(m,n,d)$.*

Proof. We refer reader to Shpilka & Yehudayoff (2010, Theorem 4.12) for the proof of $f'(y) \coloneqq f(y, y^{k^1 \bmod p}, \ldots, y^{k^{n-1} \bmod p})$ being non-zero. Observe that this gives us a set G of HSGs for the class $\mathcal{P}$ of m-sparse polynomials, $G = \{\Phi_k\}_{k \in [mn+1]}$, where $\Phi_k = (y, y^{k^1 \bmod p}, \ldots, y^{k^{n-1} \bmod p})$. Using Lemma 2.6, we get a single bivariate HSG $\Phi \in \mathbb{F}[y,z]^n$ such that $\Phi(f) \neq 0$ for a non-zero $f \in \mathcal{P}$. By using Lemma 2.8, we also get a univariate HSG Ψ for class $\mathcal{P}$. Degree of $\Psi(f)$ is at most poly(m,n,d). We get a deterministic poly-time blackbox PIT for class $\mathcal{P}$ by evaluating $\Psi(f)$ on its degree+1 distinct points. □

Lemma 2.10 (Top Sparse). *Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a polynomial of degree d such that* sparsity$(f^{[d]}) \leq m$, *where $f^{[d]}$ denotes the top degree-d homogeneous part of f. Then, there is a hitting set generator $\Psi \in \mathbb{F}[t]^n$ for f with* $\deg(\Psi) =$ *poly(m,n,d).*

Proof. Note that if f is a non-zero polynomial of degree d, then $f^{[d]} \neq 0$. Moreover sparsity of $f^{[d]}$ is at most m. Thus by Lemma 2.9, we have an HSG $\Phi \in \mathbb{F}[y]^n$ for $f^{[d]}$ with $\deg(\Phi) =$ poly(m,n,d) such that $\Phi(f^{[d]}) \neq 0$. Let z be a new variable. Observe that,

$$f^{[d]}(\mathbf{x}) = \mathrm{coeff}\big(f(zx_1, zx_2, \ldots, zx_n)\big)(z^d).$$

Let HSG $\Phi = (g_1(y), \ldots, g_n(y))$. This gives us a bivariate HSG $\Phi' = (zg_1(y), \ldots, zg_n(y))$ for f. This is because,

$$\begin{aligned} f(zx_1, \ldots, zx_n) &= f^{[d]}(\mathbf{x})z^d + f^{[d-1]}(\mathbf{x})z^{d-1} + \ldots + f^{[0]}z^0 \\ f(zg_1, \ldots, zg_n) &= f^{[d]}(g_1, \ldots, g_n)z^d + \ldots + f^{[0]}z^0 \\ f(zg_1, \ldots, zg_n) &= \Phi(f^{[d]})z^d + \ldots + f^{[0]}z^0 \end{aligned}$$

Since $\Phi(f^{[d]}) \neq 0$, this implies $\Phi'(f) = f(zg_1, \ldots, zg_n) \neq 0$. Note that $\deg(\Phi') =$ poly(m,n,d). Now, by Lemma 2.8 we get a univariate HSG $\Psi \in \mathbb{F}[t]^n$ for f such that $\Psi(f) \neq 0$ and $\deg(\Psi) =$ poly(m,n,d). □

Lemma 2.11 (Parallel Sum). *Let A and B be polynomials computable by ROABPs of width w_1 and w_2 respectively, in the same variable order. Then $A + B$ can be computed by an ROABP of width $w_1 + w_2$ in this order.*

Proof. We compute the sum by joining the two ROABPs in parallel. Let s_1, t_1 be the source and sink vertices of ROABP A respectively and s_2, t_2 be that of B. Create a new source vertex s and a sink vertex t for $A+B$. Draw an edge from $s \to s_1$ with unit label and an edge $s \to s_2$ with unit label. Similarly, join $t_1 \to t$ and $t_2 \to t$ with unit labels. Clearly, the new ROABP is of width $w_1 + w_2$. □

Lemma 2.12 ($\sum\wedge\sum$ to ROABP; Forbes *et al.* 2014, Saxena 2008). *If we have poly-time blackbox PIT for log-variate (commutative) ROABPs, then we have poly-time blackbox PIT for (standard multivariate) diagonal depth-3 circuits. Moreover, if we have poly-time blackbox PIT for sum of log-variate, constant width (commutative) ROABPs, then also we get the same conclusion.*

Proof sketch. Forbes & Shpilka (2013a,b) exploited the fact that diagonal depth-3 circuits have low dimension partial-derivative space to show that non-zero polynomials computed by them have a nonzero *log*-support monomial. That is, a degree d polynomial $f = \sum_{i=1}^{k} \ell_i^{d_i}$ has $\dim_{\mathbb{F}}\{\partial^{<\infty}(f)\} = \mathrm{poly}(k, n, d) = \mathrm{poly}(s)$, where s is the size of circuit, ℓ_i's are linear polynomials and $\{\partial^{<\infty}(f)\}$ denotes the space of all partial derivatives of f. Note that a simple monomial like $x_1 x_2 \ldots x_n$ with support n has $\dim_{\mathbb{F}}\{\partial^{<\infty}(x_1 \ldots x_n)\} = 2^n$. With few additional observations, they prove that a non-zero f must compute a monomial with non-zero coefficient, that is supported on at most $O(\log s)$ variables.

Under the promise of such a log-support monomial, we can apply variable-reduction map Φ of Shpilka & Volkovich (2009), used in Forbes *et al.* (2014) or the map of Vaid (2015), to get from n to $O(\log n)$ variables. Both these maps preserve non-zeroness of f, i.e $\Phi(f) \neq 0$.

The maps are such defined that after applying either of them, 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 Saxena (2008), i.e $\Phi(f) = \sum_{i=1}^{t} \prod_{j=1}^{n'} f_{ij}(x_j)$ where $t =$ poly(s, d) and number of variables of $\Phi(f) =: n' = O(\log s)$. Observe that each product-of-univariates $\prod_{j=1}^{n'} f_{ij}(x_j)$, has a width-1 ROABP in any variable order (commutative). Thus, by Lemma 2.11, $\Phi(f)$ can be computed by an $O(\log s)$-variate, width $t =$ poly(s, d) ROABP. Thus, solving PIT for log-variate ROABP will solve PIT for $\Phi(f)$ and hence for f. Second part follows by observing that $\Phi(f)$ is also a sum of t-many width-1, log-variate ROABPs. □

3. PIT for Degree-preserving sum of ROABPs

We prove Theorem 1.4 in this section before proving Theorem 1.1 in the next section. We will be develop few tools here that will be needed later for the proof of Theorem 1.1.

3.1. Syntactically Homogeneous ROABP. 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, every syntactically homogeneous ROABP is an ROABP computing a homogeneous polynomial, but every ROABP computing a homogeneous polynomial is not syntactically homogeneous. This is because some edge label in the ROABP may be inhomogeneous or some intermediate path may be computing an inhomogeneous polynomial, which cancels out in the end.

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 $(y_1, \ldots, y_n)$, where for all $i \in [n]$, $y_i = x_{\pi(i)}$ for some unknown permutation $\pi : [n] \to [n]$.

DEFINITION 3.1 (Syntactic homogeneity). *Let $f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$ be computed by an ROABP $D(\mathbf{x})$ of width r in the 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* $D_{>i} := \prod_{j=i+1}^{n} D_j \in \mathbb{F}^{r\times 1}[\mathbf{y}_{>i}]$.

Although in this work, we only need Definition 3.1, it can be shown that it is equivalent to the informal definition of syntactic homogeneity stated at the start of this section. To see this observe that, Definition 3.1 clearly follows from homogeneity of the polynomials $[u \rightsquigarrow v]$ computed from u to v, for *any* two nodes (u, v) in ROABP. For the other side, let V be the vertex set of the layer which contains u. Then, note that $[s \rightsquigarrow v] = \sum_{w:w\in V}[s \rightsquigarrow w] \cdot [w \rightsquigarrow v]$. By Definition 3.1, $[s \rightsquigarrow v]$ is homogeneous and so is each $[s \rightsquigarrow w]$. Also, the set of polynomials $\{[s \rightsquigarrow w] \mid w \in V\}$ is $\mathbb{F}$-linearly independent by Nisan's characterization. Now, apply Lemma 3.2 below to get each $[w \rightsquigarrow v]$ (in particular $[u \rightsquigarrow v]$) to be homogeneous.

We first prove the following lemma, which we will need in the proof of Theorem 3.3.

Lemma 3.2. *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 the latter part has to be zero.

Let $\mathbf{z}^{\mathbf{a}}$ be any monomial in $g_k^{[\neq(d-d_k)]}$ with coefficient, say $c_k \neq 0$, 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 canceled since it is of degree $d_k + |\mathbf{a}|_1 \neq d$. Observe that this term can get canceled only by product of $\mathbf{z}^{\mathbf{a}}$ with those f_i that have degree d_k (simply by variable disjointedness & 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 $\mathrm{coeff}(g_i)(\mathbf{z}^{\mathbf{a}}) =: c_i$, 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 $d - \deg(f_k)$. □

If a homogeneous polynomial f is computed by an ROABP of width w, then the optimal width ROABP for f constructed using Nisan's characterization (Lemma 2.2) has width $r \leq w$. In the following theorem, we prove that it is also syntactically homogeneous.

Theorem 3.3 (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. Here, we follow the construction as presented in Gurjar *et al.* (2017b, Lem.2.8). 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 $\{\mathbf{e}_1, \ldots, \mathbf{e}_r\} := \mathrm{span}_i(f)$. 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, as proved in Lemma 3.2. 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 $D = D_{<i} \cdot D_i \cdot D_{>i}$.

$$D = \begin{bmatrix} f_1 & f_2 & \cdots & f_r \end{bmatrix} \begin{bmatrix} g_{11} & g_{12} & \cdots & g_{1r} \\ g_{21} & g_{22} & \cdots & g_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ g_{r1} & g_{r2} & \cdots & g_{rr} \end{bmatrix} \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_r \end{bmatrix} \tag{3.4}$$

Let entries of $D_{<i}$ be $f_1, \ldots, f_r \in \mathbb{F}[\mathbf{y}_{<i}]$. 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 $D_{<i}$ with appropriate column of $r \times r$ matrix D_i. Without loss of generality, let us consider the inner product with the first column whose entries are $g_{11}, g_{21}, \ldots, g_{r1} \in \mathbb{F}[y_i]$. 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 3.2 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) and hence it is a monomial. □

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

LEMMA 3.5 (Sparsity bound). *If $f(\mathbf{x})$ is an n-variate homogeneous polynomial, which can be computed by an ROABP 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 3.3, without loss of generality, 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. □

We extend the above methods to prove another important property below: if a polynomial has ROABP width-r, then so does its lead homogeneous part. For our paper, we only require proof for the highest degree homogeneous component, which we state below but the same proof works for the lowest degree homogeneous component as well. Let width(f) denote the minimum width in which f can be computed by an ROABP.

LEMMA 3.6 (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 of width($f^{[d]}$) $=: k$ in unknown variable order $(y_1, \ldots, y_n)$. For any fixed $\ell \in [n]$, consider the partition $\{y_1, \ldots, y_\ell\} \sqcup \{y_{\ell+1}, \ldots, y_n\}$. Without loss of generality, 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. For each $i \in [k]$, let h_i be the polynomial $(f^{[<d]})_{(\mathbf{y}_{\leq \ell}, \mathbf{e}_i)}$. Then,

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

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

$$c_1(g_1 + h_1) + \ldots + c_k(g_k + h_k) \;=\; 0$$
$$c_1 g_1 + \ldots + c_k g_k \;=\; -(c_1 h_1 + \ldots + c_k h_k)\;. \tag{3.7}$$

Let $d' := \max_i\{\deg(g_i) \mid c_i \neq 0\}$. We show that the LHS in (3.7) 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 in (3.7) is a polynomial of degree $< d'$, since $\deg(h_i) < \deg(g_i) \leq d'$, for each $i \in [k]$. This contradicts (3.7), thus proving $\{g_1 + h_1, \ldots, g_k + h_k\}$ to be $\mathbb{F}$-linearly independent. We conclude that $\text{width}(f) \geq k$. □

Together with sparsity bound, this immediately gives us blackbox PIT for a log-variate constant-width ROABP (possibly inhomogeneous).

LEMMA 3.8 (Single ROABP). *Let $\mathcal{P}$ be a set of polynomials, over a field $\mathbb{F}$, computed by an ROABP of width-r and degree-d in n variables and unknown variable order. Then, we can design a hitting set generator $\Psi \in \mathbb{F}[t]^n$ for $\mathcal{P}$ with $\deg(\Psi) = poly(d, r^n)$. Thus, we get a blackbox PIT algorithm for $\mathcal{P}$ in poly(d, r^n) time.*

PROOF. Let $f \in \mathcal{P}$ be of a non-zero polynomial of degree exactly d. Thus, $f^{[d]}$ – the *lead homogeneous* part of f, is also non-zero. By Lemma 3.6, the ROABP width for $f^{[d]}$ is also upper bounded by r. Now by Lemma 3.5, we get $\text{sparsity}(f^{[d]}) \leq r^n$. Thus by Lemma 2.10, we get a univariate HSG $\Psi \in \mathbb{F}[t]^n$ for f such that $\Psi(f) \neq 0$ and $\deg(\Psi) = \text{poly}(d, r^n)$. Thus, $\Psi(f)$ is a univariate polynomial in t, also with degree at most $\text{poly}(d, r^n)$. Evaluating $\Psi(f)$ on degree$(\Psi(f))$ +1 many points yields a $\text{poly}(d, r^n)$ time blackbox PIT. □

3.3. PIT for Degree-Preserving Sum. In a similar fashion as above, we also get a blackbox PIT for a degree-preserving sum of c ROABPs, giving us the proof of Theorem 1.4 below.

Proof (Theorem 1.4). Let $f(\mathbf{x})$ be a degree d polynomial computed by a degree-preserving sum, $f(\mathbf{x}) = \sum_{i=1}^{c} f_i(\mathbf{x})$, where for each $i \in [c]$, $f_i(\mathbf{x})$ is computed by a width r ROABP. For each $i \in [c]$, let $d_i := \deg(f_i)$. By Lemma 3.6, the top degree homogeneous part of each f_i, $f_i^{[d_i]}$ is also computed by a width $\leq r$ ROABP. Then, by Lemma 3.5, each $f_i^{[d_i]}$ has sparsity at most r^n. For a non-zero f of degree d, the (leading) degree-d part of f, $f^{[d]} \neq 0$. Since the sum is degree-preserving, there exists a subset of indices $S \subseteq [c]$ such that for all $j \in S$, $d_j = d$. This yields the homogeneous sum:

$$f^{[d]} = \sum_{j \in S} f_j^{[d]}.$$

Since $|S| \leq c$, $f^{[d]}$ is non-zero with sparsity at most cr^n. Thus, we get a univariate HSG $\Psi \in F[t]^n$ for f with $\deg(\Psi) = \text{poly}(d, cr^n)$, using Lemma 2.10. Thus, $\Psi(f)$ is a univariate polynomial of degree $\text{poly}(d, cr^n)$, which implies a blackbox PIT for f in $\text{poly}(d, cr^n)$ time. □

In the remark after Theorem 1.4, we stated that a sum of ROABPs, where each ROABP computes a homogeneous polynomial, can be expressed as a degree-preserving sum. We give a proof of this below.

Proof (Remark of Theorem 1.4). Let $f(\mathbf{x}) = \sum_{i=1}^{c} f_i(\mathbf{x})$, where for each $i \in [c]$, $f_i(\mathbf{x})$ is a homogeneous polynomial, say of degree d_i, computed by an ROABP of width r. Let degree of f be d. Let us consider the subset of indices $S \subseteq [c]$, defined as $S = \{j \mid j \in [c], d_j \leq d\}$. Since f is of degree d, observe that $f = \sum_{j \in S} f_j$, since homogeneous polynomials of degree $> d$ must cancel out. Thus, f is computed by a degree-preserving sum $\sum_{j \in S} f_j$. □

4. PIT for Sum of ROABPs

We start by showing below that any hitting-set map for a prefix of variables, also preserves the coefficient space dimension up to all *subsequent* variables. This will be used critically in proof of Claim 4.5 later.

Lemma 4.1 (Dim. preservation). *Let $D(\mathbf{y})$ be a matrix-product polynomial: $D(\mathbf{y}) = D_1(y_1)\cdots D_{k-1}(y_{k-1})\cdot D_k(y_k)\cdot D'(\mathbf{y}_{>k})$, where $D_1 \in \mathbb{F}^{1\times r}[y_1]$, $D_i \in \mathbb{F}^{r\times r}[y_i]$ for $2 \le i < k$, $D_k \in \mathbb{F}^{r\times r'}[y_k]$ with $r \le r'$ 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 generator for any width r ROABP in the first $k-1$ variables (Ψ keeps variables $y_k,\ldots,y_n$ as it is). Then, Ψ preserves the k-prefix coefficient space dimension of D, i.e.*

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

Proof. Consider the matrix product for $D(\mathbf{y})$ at $(k-1)^{th}$ layer: $D = D_{<k}\cdot D_k \cdot D'$, where $D_{<k} := \prod_{i=1}^{k-1} D_i \in \mathbb{F}^{1\times r}[\mathbf{y}_{<k}]$. Without loss of generality, let the entries of $D_{<k}\cdot D_k$ be the r' $\mathbb{F}$-linearly independent polynomials given by Nisan's characterization (Lemma 2.2). Similarly entries of D' are r' linearly independent polynomials given by coefficient-extraction of D:

$$\begin{aligned} D_{<k}\cdot D_k &=: [P_1, P_2, \ldots, P_{r'}],\\ D' &=: [Q_1, Q_2, \ldots, Q_{r'}]^{\intercal}. \end{aligned}$$

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:

$$c_1P_1 + c_2P_2 + \ldots + c_{r'}P_{r'} \;\neq\; 0\ . \tag{4.2}$$

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 an HSG (univariate in t) for any width-r $(k-1)$-variate ROABP, it is also an HSG (bivariate in t, y_k) for any width $\le r'$ k-variate ROABP (having a width r ROABP in first $k-1$ layers), since the k^{th} variable is left unchanged. Since P is indeed of width r' (& k-variate), therefore Ψ preserves the non-zeroness of (4.2), implying $\mathbb{F}$-linear independence of $\Psi(P_1),\ldots,\Psi(P_{r'}) \in \mathbb{F}[t, y_k]$. Moreover,

$$\Psi(D) = [\Psi(P_1),\ldots,\Psi(P_{r'})]\cdot[Q_1, Q_2,\ldots,Q_{r'}]^{\intercal}.$$

Hence, $\dim_{\mathbb{F}}\{\Psi(D)_{((t,y_k),\mathbf{a})} \mid \mathbf{a}\} = \dim_{\mathbb{F}}\{D_{(\mathbf{y}_{\le k},\mathbf{a})} \mid \mathbf{a}\} = r'$, since if the dimension reduces on applying Ψ, we get a new non-trivial dependency among $\{\Psi(P_i)\}_{i=1}^{r'}$, say $c_1\Psi(P_1) + \ldots + c_{r'}\Psi(P_{r'}) = 0$. But these coefficients will not form a dependency among $\{P_i\}_{i=1}^{r'}$, i.e. $c_1P_1 + \ldots + c_{r'}P_{r'} \neq 0$ and since Ψ preserves non-zeroness of (4.2), it leads to a contradiction. □

4.1. Sum of two ROABPs. We start with the sum of two ROABPs $A + B$. The blackbox PIT developed here would be extended to sum of c ROABPs in Section 4.2. 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 for all $i \in [n]$, $y_i = x_{\pi(i)}$ for some unknown permutation $\pi : [n] \to [n]$. We can assume that variable order of B is different from A, since otherwise by Lemma 2.11, $A+B$ can be computed by a single ROABP of width $\le 2r$. In that case, since we will be applying a hitting set map for a single ROABP of width $O(r^3)$ in Lemma 4.11, we are already done. The main idea in Gurjar *et al.* (2017b) is to construct an ROABP for B in the variable order of A, by using the characterizing dependencies of A (Definition 2.1). Note that the width of an ROABP can blow up exponentially when expressed in a different variable order (see Forbes 2015). 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 2.2, there will be a minimum index $k \in [n]$ such that

$$\begin{aligned}&\dim_{\mathbb{F}}\{A_{(\mathbf{y}_{\le k},\mathbf{a})} \mid \mathbf{a} \in \{0,1,\ldots,d\}^k\} \le r, \text{ and}\\ &\dim_{\mathbb{F}}\{B_{(\mathbf{y}_{\le k},\mathbf{a})} \mid \mathbf{a} \in \{0,1,\ldots,d\}^k\} > r\end{aligned}$$

In simple terms, k is the first layer in the variable order of A, where B does not have an ROABP of width $\le r$. Let us consider the dependency equations at this k^{th} layer for both A and B. Observe that there exists an exponent $\mathbf{b} \in \text{depend}_k(A)$ such that A

will satisfy its dependency equation while B will violate its dependency equation for this exponent. This is because, if B satisfies dependency equations for all $\mathbf{b} \in \text{depend}_k(A)$, then B also has a width $\leq r$ ROABP till layer k, by Lemma 2.2. This contradicts our choice of k. Thus, we have some $\mathbf{b} \in \text{depend}_k(A)$ such that

$$A_{(\mathbf{u},\mathbf{b})} =: \sum_{\mathbf{a} \in \text{span}_k(A)} \alpha_{\mathbf{b},\mathbf{a}} \cdot A_{(\mathbf{u},\mathbf{a})} \tag{4.3}$$

$$B_{(\mathbf{u},\mathbf{b})} \neq \sum_{\mathbf{a} \in \text{span}_k(A)} \alpha_{\mathbf{b},\mathbf{a}} \cdot B_{(\mathbf{u},\mathbf{a})} \tag{4.4}$$

where $\mathbf{u} := \mathbf{y}_{\leq k} = (y_1, y_2, \ldots, y_k)$, and $\alpha_{\mathbf{b},\mathbf{a}} \in \mathbb{F}$ are the dependency coefficients defined by (4.3). Note that B may violate the dependency equations of A before layer k, while having a width $\leq r$ representation. Unlike Gurjar *et al.* (2017b), in our proof, we are ignoring such a layer and care only about the layer, where we witness blow-up of width in ROABP of B. Such a layer will exist under our assumption that B does not have a width $\leq r$ ROABP in the variable order of A.

In the whitebox setting, one can essentially search for this violation/non-zeroness certificate and verify the satisfiability of dependency equations in poly-time. But in the blackbox setting, the unknown variable order creates a hurdle in searching for this certificate. Guessing the variable order by brute force takes $n! \approx n^n$ time. We will show later that for the purpose of PIT, we can get around this obstacle in 2^n time. We are okay with this overhead in the log-variate setting.

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. Let $\Phi_1 \in \mathbb{F}[t_1]^{k-1}$ be an HSG for $(k-1)$-variate ROABPs of width r. We extend Φ_1 to n variables by keeping the rest of the variables as it is, which makes $\Phi_1 \in \mathbb{F}[t_1, y_k, \ldots, y_n]^n$. We now show in Claim 4.5 below that B continues to violate the dependency equation (4.4) of A at the y_k-layer, under the image of map Φ_1. We get (4.6) and (4.7) below, analogous to (4.3) and (4.4), respectively.

Claim 4.5 (Prefix map). *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 generator for any ROABP of width $\leq r$*

on first $k-1$ variables (Φ_1 leaves variables $y_k, \ldots, y_n$ unchanged). 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 $\mathbf{b} \in E^2$, defined as*

$$\Phi_1(A)_{(t_1^{b_1} y_k^{b_2})} =: \sum_{\mathbf{a} \in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_1(A)_{(t_1^{a_1} y_k^{a_2})} \,. \tag{4.6}$$

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

$$\Phi_1(B)_{(t_1^{b_1} y_k^{b_2})} \neq \sum_{\mathbf{a} \in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_1(B)_{(t_1^{a_1} y_k^{a_2})} \,. \tag{4.7}$$

Proof. For the sake of contradiction, suppose $\Phi_1(B)$ follows the dependency equations of $\Phi_1(A)$ in the y_k layer. This means $\forall \mathbf{b}$, LHS equals RHS in (4.7). 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, coefficient space dimension for first two layers of $\Phi_1(B)$ is $\leq r$. Observe that then by Lemma 4.1, Φ_1 preserves the coefficient 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 2.2). □

In the following claim, we pick an HSG Φ_2 for $\mathbf{y}_{>k}$ variables such that the image of $\Phi_1(B)$ under Φ_2 continues to violate a dependency equation of the image of $\Phi_1(A)$ under Φ_2.

Claim 4.8 (Suffix map). *Assume the setup of Claim 4.5. Let Φ_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 generator for any ROABP of width $\leq r^2(r+1)$ on last $n-k$ variables (Φ_2 leaves variables t_1, y_k unchanged). Then, there exists $\mathbf{b} \in E^2$ such that:*

$$\Phi_2 \circ \Phi_1(A)_{(t_1^{b_1} y_k^{b_2})} = \sum_{\mathbf{a} \in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_2 \circ \Phi_1(A)_{(t_1^{a_1} y_k^{a_2})} \tag{4.9}$$

(4.10)

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

Proof. (4.9) in this claim directly follows by applying map Φ_2 on (4.6) and it is true $\forall \mathbf{b} \in E^2$. Now we shall prove that in (4.7) the difference polynomial g defined as,

$$g := \Phi_1(B)_{(t_1^{b_1} y_k^{b_2})} - \sum_{\mathbf{a} \in \operatorname{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_1(B)_{(t_1^{a_1} y_k^{a_2})} \neq 0$$

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 (Lemma 2.2), Q_i's are certain coefficient polynomials of B, $Q_i = B_{(\mathbf{y}_{<k}, \mathbf{a}_i)}$ where $\{\mathbf{a}_1, \ldots, \mathbf{a}_r\} = \operatorname{span}_{k-1}(A)$. Now, by Lemma 2.3, 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 $a \in E$,

$$\Phi_1(B)_{(t_1,a)} = \sum_{i=1}^{r} \operatorname{coeff}(\Phi_1(P_i))(t_1^a) \cdot Q_i,$$

implying $\Phi_1(B)_{(t_1,a)} \in \operatorname{span}_{\mathbb{F}}\{Q_1, \ldots, Q_r\}$. 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 by Lemma 2.11, 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 2.3 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

$$g = Q'_{b_1(y_k,b_2)} - \sum_{\mathbf{a} \in \operatorname{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot Q'_{a_1(y_k,a_2)}.$$

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, again by Lemma 2.11, 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 an HSG 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 (4.10). □

Suppose we are given a correct guess of the variable order of A, say $(y_1, y_2, \ldots, y_n)$. Then, using Claim 4.5 and Claim 4.8, we show that Φ is an HSG for $A + B$, because the dependency violation by B is preserved under the image of Φ. Moreover, $\Phi(A + B)$ is a trivariate polynomial which is easy to test for non-zeroness.

Now we handle the case where variable order of A is unknown. Instead of going through all $n!$ permutations, we only go over all k-sized subsets of $[n]$. This is because we are applying an HSG Φ_1 of a single ROABP (of unknown variable order) on the prefix variables and hence the order within the prefix-subset does not matter. For each choice of subset, we go over additional n choices by trying each variable as y_k. This process will lead to constructing a set G of HSGs such that for an input polynomial $A+B \neq 0$, there is an HSG $\Phi \in G$ such that $\Phi(A + B) \neq 0$. Moreover, our set G has size $\leq n.2^n$ which is $\text{poly}(r, d)$ in the log-variate setting. Then, using Lemma 2.6 we can combine all the generators into a single generator. We formalize this construction in the lemma below.

Lemma 4.11 (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 $\Gamma \in \mathbb{F}[t]^n$ be a hitting set generator for the class of width r, n-variate, d-degree ROABPs with degree of HSG Γ, $\deg(\Gamma) =: T(r, n, d)$. Then, one can design a hitting set generator $\Psi \in \mathbb{F}[z]^n$ for the sum $A + B$ in $\big(n2^n \cdot T(2r^3, n, d)\big)^{O(1)}$ time such that* $\deg(\Psi) = \big(n2^n \cdot T(2r^3, n, d)\big)^{O(1)}$.

Proof. We start with a non-zero input polynomial $A+B$. Suppose we know the correct variable order of A: $(y_1, \ldots, y_n)$ and the correct layer where B violates dependency equation of A, say the y_k layer. Invoke HSG Γ with appropriate parameters to get HSG

$\Phi_1 \in \mathbb{F}[t_1]^{k-1}$ for $(k-1)$-variate, width $\leq r$ ROABPs and HSG $\Phi_2 \in \mathbb{F}[t_2]^{n-k}$ for $(n-k)$-variate, width $\leq r^2(r+1)$ ROABPs as used in Claim 4.5 and Claim 4.8 respectively. Let $\Phi := (\Phi_1, y_k, \Phi_2) \in \mathbb{F}[t_1, y_k, t_2]^n$ be the concatenation of the two HSGs. We now show that if we guessed the correct order and layer, then Φ is an HSG for $A+B$, that is, $\Phi(A+B) \neq 0$. Claim 4.5 and Claim 4.8 together prove that $\Phi(B)$ violates a dependency equation of $\Phi(A)$ in the y_k layer. In other words $\Phi(A)$ is an ROABP of width r, in the variable order (t_1, y_k, t_2), while (4.10) 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 work with unknown variable order. For Claim 4.5 to hold, we only need to 'guess' the prefix set $\{y_1, \ldots, y_k\}$ 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 $\sum_{k=1}^{n} k\binom{n}{k} = \sum_{k=1}^{n} n\binom{n-1}{k-1} \leq n2^{n-1}$. We try Φ for each guess. For the Φ corresponding to correct guess of prefix and variable y_k, the above argument guarantees that Φ is an HSG for $A+B$. Thus, we get a collection G of candidate HSGs, one for each guess, with $|G| \leq n2^n$ such that one of them is guaranteed to work. Using Lemma 2.6, we get a single generator $\Psi' \in \mathbb{F}[t_1, y_k, t_2, s]^n$, which we can redefine to a univariate generator $\Psi \in \mathbb{F}[z]^n$ for $A+B$ using Lemma 2.8.

We now calculate the degree of HSG Ψ in terms of degree of the HSG Γ for a single ROABP. Observe that $\deg(\Phi_1) = T(r, k-1, d)$ and $\deg(\Phi_1) = T(r^2(r+1), n-k, d)$. Thus,

$$\deg(\Phi) = \max\{\deg(\Phi_1), \deg(\Phi_2)\} \leq T(2r^3, n, d).$$

Now, degree for the single generator Ψ' using Lemma 2.6 is

$$\deg(\Psi') = \max\{\deg(\Phi), \deg(s)\} = \max\{T(2r^3, n, d), n2^n\}.$$

Thus, by Lemma 2.8, degree of the univariate generator Ψ is

$$\deg(\Psi) \leq \left(n2^n \cdot T(2r^3, n, d)\right)^{O(1)}.$$

The time complexity for designing Ψ is also similarly bounded. □

Using the efficient HSG designed for a single constant-width, log-variate ROABP in Lemma 3.8, we get an efficient HSG for sum of two such ROABPs below.

Corollary 4.12. *Let $\mathcal{P}$ be a set of n-variate polynomials over a field $\mathbb{F}$, computed by a sum of two ROABPs, each of width r and degree d in unknown variable order. Then, blackbox PIT for $\mathcal{P}$ can be solved in poly(d, r^n) time.*

Proof. We have HSG Φ for single ROABP of width r with $\deg(\Phi) = \text{poly}(d, r^n)$ using Lemma 3.8. Now, by Lemma 4.11, we have HSG $\Psi \in \mathbb{F}[z]^n$ for $\mathcal{P}$ with $\deg(\Psi) = \text{poly}(d, 2^n, r^{3n}) = \text{poly}(d, r^n)$. Thus, $\deg(\Psi(f))$ for some $f \in \mathcal{P}$ is also poly(d, r^n). By evaluating $\Psi(f)$ on deg+1 points, we can check non-zeroness of $\Psi(f)$. This gives blackbox PIT for f. □

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 Gurjar *et al.* (2017b) 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 $(y_1, y_2, \ldots, y_n)$. Thus, we get dependency equations for A, as in (4.3). If the input sum is non-zero, then B will not follow some dependency of A.

Note that unlike the sum of two ROABPs 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 $Q := B_{(\mathbf{u},\mathbf{b})} - \sum_{\mathbf{a} \in \text{span}_k(A)} \gamma_{\mathbf{b},\mathbf{a}} \cdot B_{(\mathbf{u},\mathbf{a})}$, where $\mathbf{u} := (y_1, \ldots, y_k)$. Since $B = A_2 + \ldots + A_c$, we get

$$Q = \sum_{i=2}^{c} \left(A_{i(\mathbf{u},\mathbf{b})} - \sum_{\mathbf{a} \in \text{span}_k(A)} \gamma_{\mathbf{b},\mathbf{a}} \cdot A_{i(\mathbf{u},\mathbf{a})} \right). \tag{4.13}$$

Now each $A_{i(\mathbf{u},\mathbf{a})}$ and $A_{i(\mathbf{u},\mathbf{b})}$ have ROABPs of width r individually, by Lemma 2.3. Thus, for each of the $c-1$ summands in (4.13), we have an ROABP of width $r(r+1)$, by Lemma 2.11. 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 4.5.

$$\begin{aligned}\Phi_1(Q)_{(t_1^{b_1} y_k^{b_2})} = \sum_{i=2}^{c} \Big(\Phi_1(A_i)_{(t_1^{b_1} y_k^{b_2})} - \\ \sum_{\mathbf{a} \in \mathrm{span}_2(\Phi_1(A))} \gamma_{\mathbf{b},\mathbf{a}} \cdot \Phi_1(A_i)_{(t_1^{a_1} y_k^{a_2})} \Big) \, . \end{aligned} \tag{4.14}$$

In (4.14), each of the $c-1$ summands has an ROABP of width $r^2(r+1) \leq 2r^3$ (See proof of Claim 4.8). 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 4.15 (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. Let $\Gamma \in \mathbb{F}[t]^n$ be a hitting set generator for the class of width r, n-variate, d-degree ROABPs with degree of HSG Γ,* $\deg(\Gamma) =: T(r,n,d)$. *Then, one can design a hitting set generator $\Psi \in \mathbb{F}[z]^n$ for the sum $\sum_{i=1}^{c} A_i(\mathbf{x})$ in $\big(2^n \cdot T(2^c r^{3^c}, n, d)\big)^{O(c)}$ time, with* $\deg(\Psi) = \big(2^n \cdot T(2^c r^{3^c}, n, d)\big)^{O(c)}$.

PROOF. We prove by induction on c. Base case for $c = 2$ has been proved earlier in Lemma 4.11. 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. Suppose we have correctly guessed the variable order and the variable y_k. Then, we employ HSG $\Phi_1 \in \mathbb{F}[t_1]^{k-1}$ for first $k-1$ variables as used in Claim 4.5 and we get (4.6) for A while we get (4.14) for $B = \sum_{i=2}^{c} A_i$. Since B violates dependency equation of A, (4.14) is non-zero polynomial, which can be computed as a sum of $c-1$ ROABPs of width $\leq 2r^3$. By induction hypothesis, we can design a (univariate) HSG $\Phi_2 \in \mathbb{F}[t_2]^{n-k}$ 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 $\Phi := \Phi_2 \circ \Phi_1 \in \mathbb{F}[t_1, y_k, t_2]^n$ 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 $\Phi(A+B) \neq 0$ for $A + B \neq 0$.

Now, suppose the variable order is unknown. Then, observe that we only need to guess the prefix subset of variables and the correct variable y_k in it. We simply brute-force search for them. In other words, for each k-sized subset of $[n]$ and for each variable as y_k, we apply Φ. Thus, we get a set of candidate generators G of size $|G| \leq n2^n$ such that for any non-zero $A+B$, there is some generator $\Phi \in G$ for which $\Phi(A+B) \neq 0$. Using Lemma 2.6, we can combine the generators into a single generator $\Psi' \in \mathbb{F}[t_1, y_k, t_2, s]^n$. Using Lemma 2.8, we get a univariate generator $\Psi \in \mathbb{F}[z]^n$ for $\sum_{i=1}^{c} A_i$.

Now, we calculate $\deg(\Psi)$. Observe that

$$\begin{aligned}
\deg(\Phi_1) &= T(r, k-1, d) \leq T(r, n, d) \\
\deg(\Phi_2) &= \left(2^{n-k} \cdot T(2^{c-1} r^{3^c}, n-k, d)\right)^{O(c-1)} \\
&\leq \left(2^n \cdot T(2^c r^{3^c}, n, d)\right)^{O(c-1)} \\
\deg(\Phi) &\leq \left(2^n \cdot T(2^c r^{3^c}, n, d)\right)^{O(c-1)} \\
\deg(\Psi') &\leq \left(2^n \cdot T(2^c r^{3^c}, n, d)\right)^{O(c-1)} \\
\deg(\Psi) &\leq (d \cdot n2^n)^{O(1)} \cdot \left(2^n \cdot T(2^c r^{3^c}, n, d)\right)^{O(c-1)} \\
&\leq \left(2^n \cdot T(2^c r^{3^c}, n, d)\right)^{O(c)}
\end{aligned}$$

The first step follows since Φ_1 is HSG for single ROABP. Degree of Φ_2 is given by induction hypothesis. Degree of Φ is simply the maximum between $\deg(\Phi_1)$ and $\deg(\Phi_2)$. Degree of Ψ' is calculated using Lemma 2.6. Finally, by Lemma 2.8, we get degree of Ψ.

Let $S(c, r, n, d)$ denote the time complexity of constructing Ψ. Similar to the argument above, we get the following recursive formula for designing Ψ, where $S(1, r, n, d) = T(r, n, d)^{O(1)}$

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

As a solution, we get $S(c, r, n, d) \leq \left(2^n \cdot T(2^c r^{3^c}, n, d)\right)^{O(c)}$. □

Now we complete the proof of our reduction from PIT of sum to PIT of single ROABP.

Proof (Theorem 1.1). The proof simply follows from Lemma 4.15 because of equivalence between hitting set generator and blackbox

PIT. That is, if we have an HSG of degree $T(r,n,d)$ which can also be constructed in the same time, then we have a poly$(T(r,n,d))$ time blackbox PIT and vice versa. See Section 2.4 and Shpilka & Yehudayoff (2010, Lemma 4.1) for exact equivalence between HSG and blackbox PIT. □

Finally after having developed all the machinery, we complete the proofs of Corollary 1.2 and Corollary 1.3.

PROOF (Corollary 1.2). Let $f \in \mathbb{F}[\mathbf{x}]$ be an n-variate polynomial computed by sum of c ROABPs, each of width r and degree d. Using Lemma 3.8, we have a blackbox PIT for a single ROABP of width r in poly(d, r^n) time. Now, in Theorem 1.1, set $T(r,n,d) :=$ poly(d, r^n) to get $T'(r,n,d,c) = \left(2^n \cdot d \cdot r^{n3^c}\right)^{O(c)} =$ poly(d^c, r^{nc3^c}). This gives us blackbox PIT algorithm for f with the required time complexity. □

PROOF (Corollary 1.3). Using result of Agrawal *et al.* (2015), we have a $(ndr)^{O(\log n)}$ time blackbox PIT for a single ROABP of width r and degree d in n variables. Therefore, we can set $T(r,n,d) :=$ $(ndr)^{O(\log n)}$ in Theorem 1.1 to get a poly$(2^{cn} \cdot n^{c \log n}, d^{c \log n}, r^{3^c \log n})$ time blackbox PIT for border of sum of c ROABPs. □

5. PIT for Border

In this section, we cover the proof of Theorem 1.5. Before that, we discuss few other things related to PIT for border classes.

For a class $\mathcal{C}$, where the border class $\overline{\mathcal{C}}$ is same as $\mathcal{C}$, the PIT algorithm will be same for both $\mathcal{C}$ and $\overline{\mathcal{C}}$. For example, the class of sparse polynomials and also the class of single ROABPs (Lemma 5.2). However, for classes where $\overline{\mathcal{C}} \neq \mathcal{C}$, blackbox PIT algorithms that work for $\mathcal{C}$ may not work for the border class $\overline{\mathcal{C}}$. The only thing we can say for such classes, in general, is that PIT for $\overline{\mathcal{C}}$ is in PSPACE, as PIT for $\overline{\mathrm{VP}}$ is in PSPACE (Forbes & Shpilka 2018; Guo *et al.* 2019b). PIT algorithms, which rely on a rank based measure, usually work for the border class also, since the rank based measure also works for the border class. An example of this are the PIT algorithms for the class $\mathcal{C}$ of diagonal depth-3 circuits, even though for this class it is unknown whether $\overline{\mathcal{C}}$ is same as $\mathcal{C}$.

LEMMA 5.1 (Border $\sum\bigwedge\sum$; Forbes 2016). *The blackbox PIT algorithms in Forbes & Shpilka (2013a), Forbes* et al. *(2014) and Forbes* et al. *(2018) for the class of diagonal depth-3 circuits also solve blackbox PIT for its border class in their same respective times.*

PROOF SKETCH. We discussed in the first part of the proof of Lemma 2.12 that for a polynomial f computed by size-s $\sum\bigwedge\sum$ circuit, $\dim\{\partial^{<\infty}(f)\} = \mathrm{poly}(s)$ which helps in proving that f has a non-zero monomial of $O(\log s)$ support. All of the works - Forbes & Shpilka (2013a), Forbes *et al.* (2014) and Forbes *et al.* (2018) build on this property to give efficient PIT algorithms for f.

Actually, one can also show that a polynomial g computed in the border of size-s $\sum\bigwedge\sum$ circuit also has $\dim\{\partial^{<\infty}(g)\} = \mathrm{poly}(s)$ as discussed in Forbes (2016). Its proof is very similar to the proof of Lemma 5.2 below. This then proves that g also has a non-zero monomial of $O(\log s)$ support. Thus, the above mentioned PIT algorithms also work for g. □

5.1. Border of ROABP. It turns out that for a single ROABP, border does not add any power, i.e. $\overline{\text{ROABP}} = \text{ROABP}$.

LEMMA 5.2 (Forbes 2016). *A polynomial $f \in \mathbb{F}[\mathbf{x}]$ in the border class of width w ROABPs can also be computed by an ROABP of width at most w.*

PROOF. Let $g = f + \epsilon h$, where g has an ROABP of width w over $\mathbb{F}(\epsilon)$. We need to show that the limit polynomial f also has ROABP-width $\leq w$ over $\mathbb{F}$. Let the unknown variable order of g be $(y_1, \ldots, y_n)$. By applying Nisan's characterization (Nisan 1991) on g, we know that for all $k \in [n]$, the matrix defined in Nisan (1991) for each layer, M_k has rank at most w over $\mathbb{F}(\epsilon)$. This means determinant of any $(w+1)\times(w+1)$ minor of M_k is identically zero. Observe that entries of M_k are coefficients of monomials of g which are in $\mathbb{F}[\epsilon][\mathbf{x}]$. Thus, determinant polynomial will remain zero even under the limit, $\epsilon \to 0$. Hence, for $f = \lim_{\epsilon\to 0} g$, each matrix M_k

also has rank at most w over $\mathbb{F}$. Thus by Nisan's characterization, f also has an ROABP of width at most w. This matrix is now commonly called as partial derivative matrix. The notion of rank of partial derivative matrix is equivalent to the notion of dimension of the space spanned by the coefficient polynomials as defined in Lemma 2.2 and used in this work. We refer the reader to chapter on The Partial Derivative Matrix in Saptharishi (2016) for details on this matrix and its connection with coefficient polynomials. □

Let f be a degree d polynomial computed by an ROABP of width w and let $f^{[d]}$ be its leading homogeneous degree-d part. Lemma 3.6 states that $f^{[d]}$ can also be computed an ROABP of width w. This fact also has a nice alternate proof via border complexity as follows. It is not difficult to show that for a polynomial f of degree-d in class $\mathcal{C}$, $f^{[d]}$ can be computed in the border class $\overline{\mathcal{C}}$. Since for ROABPs the border is the same, $f^{[d]}$ can also be computed by an ROABP of width $\leq w$.

5.2. Border of Sum of ROABPs. Although a single ROABP is closed under border, it is not clear if sum of constantly many ROABPs is equal to its border class. Let A and B be two ROABPs of width w in different variable orders. Let f be a polynomial computed in the border class of sum of two ROABPs. Then we can write $g = f + \epsilon h$, where g is computed by $A + B$ over $\mathbb{F}(\epsilon)$.

One might question whether f can be expressed as f_1+f_2, where f_1 is computed in the border of A and f_2 in the border of B. If this were true, then f could also be computed by sum of two ROABPs since both f_1 and f_2 individually would be computable by ROABPs of width w as stated in Lemma 5.2. But this line of thought is false for the following reason. Note that the polynomial g which approximates f is computed by sum of two ROABPs $A + B$ over $\mathbb{F}(\epsilon)$. In the edge weights of A, there maybe coefficients involving ϵ in denominator which get canceled only in the sum but stay individually, and therefore f is not directly expressible as $f_1 + f_2$, where both $f_1$and f_2 are individually computable in the border class of single ROABP.

Moreover, $g = A + B$ over $\mathbb{F}(\epsilon)$ may not have a single ROABP of same width over $\mathbb{F}(\epsilon)$, since width can blow up exponentially, as stated in Fact 1.6. Thus, border of sum of ROABPs cannot be directly expressed as border of a single ROABP of similar width. Hence, in all likelihood, the border class of sum of c ROABPs is more powerful than the class of sum of c ROABPs. This makes it an interesting candidate for the PIT question. In Theorem 1.5, we solve it along the same lines as Theorem 1.1 by showing an efficient reduction from PIT of border of sum to PIT of border of a single ROABP (in log-variate regime). We are able to achieve this because the proof technique of Theorem 1.1 is compatible with border, essentially because at the core they rely on rank based measure of Nisan's width characterization.

Proof (Theorem 1.5). Let $f(\mathbf{x})$ be a polynomial in the border of $A_1+A_2+\ldots+A_c$. That is, $g(\mathbf{x},\epsilon) = f(\mathbf{x})+\epsilon\cdot h(\mathbf{x},\epsilon)$, where $g(\mathbf{x},\epsilon)$ is computed by $A_1+\ldots+A_c$ over $\mathbb{F}(\epsilon)$ and $\lim_{\epsilon\to 0} g = f$. Our aim is to design an HSG Ψ for $f(\mathbf{x})$ such that $f \neq 0 \Rightarrow \Psi(f) \neq 0$.

Let us first work over the function field $\mathbb{F}(\epsilon)$. We follow a similar inductive strategy as Lemma 4.15 for g. Let $A = A_1$ and $B := A_2+\ldots+A_c$. Assume A has width r ROABP while f may not. Write f in the variable order of A. If f has an ROABP of width r in the variable order of A, then the HSG from Lemma 3.8 will work in the promised time, since border is closed for a single ROABP.

By Lemma 2.2, if f cannot be written as ROABP of width r in variable order of A, then there is a layer k, where f has width greater than r and hence f does not follow the dependency equations of A in that layer[1]. Thus, there exists $k \in [n]$, prefix $\mathbf{u}$; $S \subset \{0,1,\ldots,d\}^k$; and constants $\{\alpha_{\mathbf{a}} \in \mathbb{F}[\epsilon] \mid \mathbf{a} \in S\}$ such that

$$0 = \sum_{\mathbf{a}\in S} \alpha_{\mathbf{a}} \cdot A_{(\mathbf{u},\mathbf{a})} \tag{5.3}$$

$$0 \neq \sum_{\mathbf{a}\in S} \alpha_{\mathbf{a}} \cdot f_{(\mathbf{u},\mathbf{a})} \;. \tag{5.4}$$

The equality (5.3) above is derived from (4.3) by collecting all the terms on one side and considering size of S to be at most $r+1$.

[1] If no such layer exists, then f is in the border of a single ROABP of width r and we are done by Lemma 5.2

Similarly, (5.4) is derived by considering dependency equations for f at layer k. Observe that here, unlike Lemma 4.15, we are not working with B directly as in (4.4). This is because the inequality for B in (4.4) may become an equality in the limit $\epsilon \to 0$. Therefore, we work indirectly via f in (5.4) because we are assuming input polynomial $f = \lim_{\epsilon \to 0} g$ to be non-zero and use that to derive a nontrivial relationship as shown below.

Recalling $A + B = g = f + \epsilon \cdot h$, we use (5.3) to deduce:

$$\sum_{\mathbf{a} \in S} \alpha_{\mathbf{a}} \cdot B_{(\mathbf{u},\mathbf{a})} \; = \; \sum_{\mathbf{a} \in S} \alpha_{\mathbf{a}} \cdot f_{(\mathbf{u},\mathbf{a})} + \epsilon \cdot h'$$

for the $h' \in \mathbb{F}[\epsilon, \mathbf{x}]$ that depends on h. Without loss of generality, we can assume that ϵ does not divide $\alpha_{\mathbf{a}}$ for some $\mathbf{a} \in S$, because if it does divide all $\alpha_{\mathbf{a}}$, then we can divide the above equation by ϵ to get a new equation of the same form. Moreover, $\{f_{(\mathbf{u},\mathbf{a})} \mid \mathbf{a} \in S\}$ are linearly independent polynomials over $\mathbb{F}$ (equivalently over $\mathbb{F}(\epsilon)$) and $|S| \leq r + 1$ in the above equation.

Using $\alpha \equiv \alpha(0) \bmod \epsilon$, we write down a nontrivial relationship

$$\sum_{\mathbf{a} \in S} \alpha_{\mathbf{a}} \cdot B_{(\mathbf{u},\mathbf{a})} \; = \; \sum_{\mathbf{a} \in S} \alpha_{\mathbf{a}}(0) \cdot f_{(\mathbf{u},\mathbf{a})} + \epsilon \cdot h'' \, . \tag{5.5}$$

It is nontrivial because its RHS does not vanish on setting $\epsilon = 0$ and it remains well-defined, while LHS is a sum of $(c-1)$ ROABPs over $\mathbb{F}(\epsilon)$, each of width $r(r+1)$. If our input $f = \lim_{\epsilon \to 0} g$ is non-zero, then LHS above must also be non-zero in the limit $\epsilon \to 0$, by considering (5.5) in conjunction with (5.4). Thus, we reduce to the problem of designing HSG for border of sum of $c - 1$ ROABPs.

Following the proof in Section 4, this reduces PIT of f to PIT of border of sum of $c - 1$ ROABPs each of width $O(r^3)$ (via the prefix and suffix maps). This can be solved recursively, till we reach border of a single ROABP of appropriate width. Since border of single ROABP is same as ROABP by Lemma 5.2, we can call HSG of Lemma 3.8 with appropriate parameters. The exact and formal details of unfolding the recursion are same as that in the proof of Lemma 4.15. □

6. Future Directions

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

- Design a poly(s)-time blackbox PIT algorithm for $(\log s)$-variate, size-s ROABP. This will also solve standard multivariate diagonal depth-3 model (Forbes *et al.* 2014). Without loss of generality, ROABP can also be assumed to be syntactically homogeneous (Theorem 3.3 and Lemma 3.6).

- In Theorem 1.4, can we remove the restriction of degree-preserving sum? If so, then that would solve diagonal depth-3 model (Lemma 2.12). 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 PIT 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 Gurjar *et al.* (2017a) algorithm only works for known variable order and for fields with zero or large characteristic.

- Bring down 2^n dependence in Theorem 1.1 to poly(n). Currently, the dependence on c is doubly exponential, both in this reduction and also in Gurjar *et al.* (2017b). One would also like to bring it down to poly(c) or even just to single exponential like $(ndr)^{O(c)}$.

Acknowledgements

We thank Subhayan Saha, Sumanta Ghosh and Zeyu Guo for various discussions related to the algebraic models studied here. We thank the anonymous reviewers for the suggestions to improve the writing. We especially thank the anonymous reviewer who gave an alternate proof of one of the lemmas here using border complexity, which inspired us to think about the border PIT algorithms. N.S. thanks the funding support from DST (DST/SJF/MSA-01/2013-14) and N. Rama Rao Chair.

References

Manindra Agrawal (2005). Proving Lower Bounds Via Pseudorandom Generators. In *FSTTCS*, volume 3821 of *Lecture Notes in Computer Science*, 92–105.

Manindra Agrawal, Sumanta Ghosh & Nitin Saxena (2019). Bootstrapping variables in algebraic circuits. *Proceedings of the National Academy of Sciences* **116**(17), 8107–8118. (A preliminary version appeared in STOC, 2018).

Manindra Agrawal, Rohit Gurjar, Arpita Korwar & Nitin Saxena (2015). Hitting-Sets for ROABP and Sum of Set-Multilinear Circuits. *SIAM Journal on Computing* **44**(3), 669–697.

Manindra Agrawal, Neeraj Kayal & Nitin Saxena (2004). PRIMES is in P. *Annals of mathematics* 781–793.

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

Michael Ben-Or & Richard Cleve (1992). Computing Algebraic Formulas Using a Constant Number of Registers. *SIAM Journal on Computing* **21**(1), 54–58. (Preliminary version in STOC'88).

Karl Bringmann, Christian Ikenmeyer & Jeroen Zuiddam (2018). On algebraic branching programs of small width. *Journal of the ACM (JACM)* **65**(5), 1–29.

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

Stephen A. Fenner, Rohit Gurjar & Thomas Thierauf (2017). Guest Column: Parallel Algorithms for Perfect Matching. *SIGACT News* **48**(1), 102–109.

Michael Forbes (2016). Some concrete questions on the border complexity of polynomials. Presentation given at the Workshop on Algebraic Complexity Theory WACT 2016 in Tel Aviv.

MICHAEL A. FORBES (2015). Deterministic Divisibility Testing via Shifted Partial Derivatives. In *Proceedings of the 56th IEEE Symposium on Foundations of Computer Science (FOCS)*, 451–465.

MICHAEL A FORBES, SUMANTA GHOSH & NITIN SAXENA (2018). 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.

MICHAEL A. FORBES, RAMPRASAD SAPTHARISHI & AMIR SHPILKA (2014). 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*, 867–875.

MICHAEL A FORBES & AMIR SHPILKA (2013a). Explicit Noether normalization for simultaneous conjugation via polynomial identity testing. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 527–542. Springer.

MICHAEL A. FORBES & AMIR SHPILKA (2013b). Quasipolynomial-Time Identity Testing of Non-commutative and Read-Once Oblivious Algebraic Branching Programs. In *FOCS*, 243–252.

MICHAEL A FORBES & AMIR SHPILKA (2018). A PSPACE construction of a hitting set for the closure of small algebraic circuits. In *Proceedings of the 50th Annual ACM SIGACT Symposium on Theory of Computing*, 1180–1192.

ZEYU GUO & ROHIT GURJAR (2020). Improved Explicit Hitting-Sets for ROABPs. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Schloss Dagstuhl-Leibniz-Zentrum für Informatik.

ZEYU GUO, MRINAL KUMAR, RAMPRASAD SAPTHARISHI & NOAM SOLOMON (2019a). Derandomization from Algebraic Hardness: Treading the Borders. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, 147–157. IEEE.

ZEYU GUO, NITIN SAXENA & AMIT SINHABABU (2019b). Algebraic Dependencies and PSPACE Algorithms in Approximative Complexity over Any Field. *Theory of Computing* **15**(1), 1–30.

Rohit Gurjar, Arpita Korwar & Nitin Saxena (2017a). Identity Testing for Constant-Width, and Any-Order, Read-Once Oblivious Arithmetic Branching Programs. *Theory of Computing* **13**(2), 1–21. (Preliminary version in CCC'16).

Rohit Gurjar, Arpita Korwar, Nitin Saxena & Thomas Thierauf (2017b). Deterministic Identity Testing for Sum of Read-Once Oblivious Arithmetic Branching Programs. *Computational Complexity* 1–46. (Conference version in CCC 2015).

Joos Heintz & Claus P. Schnorr (1980). Testing Polynomials Which Are Easy to Compute (Extended Abstract). In *Proceedings of the Twelfth Annual ACM Symposium on Theory of Computing*, STOC '80, 262–272. ACM, New York, NY, USA.

Valentine Kabanets & Russell Impagliazzo (2004). Derandomizing Polynomial Identity Tests Means Proving Circuit Lower Bounds. *Computational Complexity* **13**(1-2), 1–46. (Preliminary version in STOC' 03).

Neeraj Kayal, Vineet Nair & Chandan Saha (2016). Separation Between Read-once Oblivious Algebraic Branching Programs (ROABPs) and Multilinear Depth Three Circuits. In *33rd Symposium on Theoretical Aspects of Computer Science (STACS)*, 46:1–46:15.

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

Swastik Kopparty, Shubhangi Saraf & Amir Shpilka (2014). 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*, 169–180.

Leopold Kronecker (1882). *Grundzuge einer arithmetischen Theorie der algebraischen Grossen.* Berlin, G. Reimer.

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

Ketan Mulmuley, Umesh V. Vazirani & Vijay V. Vazirani (1987). Matching is as easy as matrix inversion. *Combinatorica* **7**, 105–113.

Ketan D. Mulmuley (2012a). The GCT Program Toward the P vs. NP Problem. *Commun. ACM* **55**(6), 98–107.

Ketan D. Mulmuley (2012b). Geometric Complexity Theory V: Equivalence between Blackbox Derandomization of Polynomial Identity Testing and Derandomization of Noether's Normalization Lemma. In *FOCS*, 629–638.

Ketan D Mulmuley & Milind Sohoni (2001). Geometric complexity theory I: An approach to the P vs. NP and related problems. *SIAM Journal on Computing* **31**(2), 496–526.

Ketan D Mulmuley & Milind Sohoni (2008). Geometric complexity theory II: Towards explicit obstructions for embeddings among class varieties. *SIAM Journal on Computing* **38**(3), 1175–1206.

Noam Nisan (1991). Lower Bounds for Non-Commutative Computation (Extended Abstract). In *Proceedings of the 23rd ACM Symposium on Theory of Computing, ACM Press*, 410–418.

Øystein Ore (1922). Über höhere kongruenzen. *Norsk Mat. Forenings Skrifter* **1**(7), 15.

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

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

Nitin Saxena (2008). Diagonal Circuit Identity Testing and Lower Bounds. In *ICALP*, volume 5125 of *Lecture Notes in Computer Science*, 60–71. Springer.

Nitin Saxena (2009). Progress on Polynomial Identity Testing. *Bulletin of the EATCS* **99**, 49–79.

NITIN SAXENA (2014). Progress on Polynomial Identity Testing- II. In *Perspectives in Computational Complexity*, volume 26 of *Progress in Computer Science and Applied Logic*, 131–146. Springer International Publishing.

JACOB T. SCHWARTZ (1980). Fast Probabilistic Algorithms for Verification of Polynomial Identities. *Journal of the ACM* **27**(4), 701–717.

AMIR SHPILKA & ILYA VOLKOVICH (2009). Improved polynomial identity testing for read-once formulas. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques*, 700–713. Springer.

AMIR SHPILKA & AMIR YEHUDAYOFF (2010). Arithmetic Circuits: A survey of recent results and open questions. *Foundations and Trends in Theoretical Computer Science* **5**(3-4), 207–388.

RISHABH VAID (2015). *Blackbox Identity Testing for Simple Depth 3 Circuits*. Master's thesis, Indian Institute of Technology Kanpur.

RICHARD ZIPPEL (1979). Probabilistic Algorithms for Sparse Polynomials. In *Proceedings of the International Symposiumon on Symbolic and Algebraic Computation*, EUROSAM '79, 216–226. Springer-Verlag, London, UK, UK.

Manuscript received 12 July 2020

PRANAV BISHT
Department of Computer Science & Engineering
IIT Kanpur
India, 208016
pbisht@cse.iitk.ac.in
https://pranavbisht.bitbucket.io/

NITIN SAXENA
Department of Computer Science & Engineering
IIT Kanpur
India, 208016
nitin@cse.iitk.ac.in
https://www.cse.iitk.ac.in/users/nitin/