THEORY OF COMPUTING, Volume 13 (2), 2017, pp. 1–21
www.theoryofcomputing.org

SPECIAL ISSUE: CCC 2016

Identity Testing for Constant-Width, and Any-Order, Read-Once Oblivious Arithmetic Branching Programs

Rohit Gurjar* Arpita Korwar Nitin Saxena†

Received June 15, 2016; Revised April 9, 2017; Published May 15, 2017

Abstract: We give improved hitting sets for two special cases of Read-once Oblivious Arithmetic Branching Programs (ROABP). First is the case of an ROABP with known order of the variables. The best previously known hitting set for this case had size $(nw)^{O(\log n)}$ where n is the number of variables and w is the width of the ROABP. Even for a constant-width ROABP, nothing better than a quasi-polynomial bound was known. We improve the hitting-set size for the known-order case to $n^{O(\log w)}$. In particular, this gives the first polynomial-size hitting set for constant-width ROABP (known-order). However, our hitting set only works when the characteristic of the field is zero or large enough. To construct the hitting set, we use the concept of the rank of the partial derivative matrix. Unlike previous approaches which build up from mapping variables to monomials, we map variables to polynomials directly.

A conference version of this paper appeared in the Proceedings of the 31st Computational Complexity Conference, 2016 [16].

*Supported by DFG grant TH 472/4.

†Supported by DST-SERB.

ACM Classification: F.2.2

AMS Classification: 68Q25, 68W30

Key words and phrases: polynomial identity testing, hitting sets, arithmetic branching programs

 DOI: 10.4086/toc.2017.v013a002

The second case we consider is that of polynomials computable by width-w ROABPs in any order of the variables. The best previously known hitting set for this case had size $d^{O(\log w)}(nw)^{O(\log\log w)}$, where d is the individual degree. We improve the hitting-set size to $(ndw)^{O(\log\log w)}$.

1 Introduction

The polynomial identity testing (PIT) problem asks if a given multivariate polynomial is identically zero. The input to the problem is given via an arithmetic model computing a polynomial, for example, an arithmetic circuit, which is the arithmetic analogue of a Boolean circuit. The degree of the given polynomial is assumed to be polynomially bounded in the circuit size. Typically, any such circuit can compute a polynomial with exponentially many monomials (exponential in the circuit size). Thus, one cannot hope to write down the polynomial in a sum-of-monomials form. However, given such an input, it is possible to efficiently evaluate the polynomial at a point in the field. This property enables a randomized polynomial identity test with one-sided error. It is known that evaluating a small-degree nonzero polynomial over a random point gives a nonzero value with a good probability [13, 29, 33]. This gives us a randomized PIT—just evaluate the input polynomial, given as an arithmetic circuit, at random points.

Finding an efficient deterministic algorithm for PIT has been a major open question in complexity theory. The question is also related to arithmetic circuit lower bounds [1, 18, 21]. The PIT problem has been studied in two paradigms: (i) blackbox test, where one can only evaluate the polynomial at chosen points and (ii) whitebox test, where one has access to the description of the input circuit. A blackbox test for a family of polynomials is essentially the same as finding a hitting set—a set of points such that any nonzero polynomial in that family evaluates to a nonzero value on at least one of the points in the set. This work concerns finding hitting sets for a special model called read-once oblivious arithmetic branching programs (ROABP).

An *arithmetic branching program* (ABP) is a specialized arithmetic circuit. It is the arithmetic analogue of a Boolean branching program (also known as a binary decision diagram). It is a directed layered graph, with edges going from a layer of vertices to the next layer. The first and the last layers have one vertex each, called the source and the sink, respectively. Each edge of the graph has a label, which is a "simple" polynomial, for example, a univariate polynomial. For any path p, its weight is defined to be the product of labels on all the edges in p. The ABP computes a polynomial which is the sum of weights of all the paths from the source to the sink. Apart from its size, another important parameter for an ABP is its width. The width of an ABP is the maximum number of vertices in any of its layers. See Definition 2.1 for a formal definition of ABP.

ABPs are a strong model for computing polynomials. It is known that for any size-s arithmetic circuit of degree bounded by $\mathsf{poly}(s)$, one can find an ABP of size $\mathsf{quasi\text{-}poly}(s)$ computing the same polynomial [32, 31, 7] (see [23] for a complete proof). Even when the width is restricted to a constant, the ABP model is quite powerful. Ben-Or and Cleve [6] have shown that width-3 ABPs have the same expressive power as polynomial-sized arithmetic formulas.

An ABP is a *read-once oblivious ABP* or ROABP if each variable occurs in at most one layer of the edges and every layer has exactly one variable.[1] The read-once property severely restricts the power of the ABP. There is an explicit family of polynomials that can be computed by simple depth-3 ($\Sigma\Pi\Sigma$) circuits but requires exponential-size ROABPs [22] to compute it. The order of the variables in the consecutive layers is said to be the *variable order* of the ROABP. The variable order affects the size of the minimal ROABP computing a given polynomial. There are polynomials which have a small ROABP in one variable order but require exponential size in another variable order. Nisan [25] gave an exact characterization of the polynomials computed by width-w ROABPs in a certain variable order. In particular, he gave exponential lower bounds for this model.[2]

The question of whitebox identity testing of ROABPs has been settled by Raz and Shpilka [28], who gave a polynomial-time algorithm for this. However, though ROABPs are a relatively well-understood model, we still do not have a polynomial-time blackbox algorithm. The blackbox PIT question is studied with two variations: one where we know the variable order of the ROABP and the other where we do not know it. For known-order ROABPs, Forbes and Shpilka [15] gave the first efficient blackbox test with $(ndw)^{O(\log n)}$ time complexity, where n is the number of variables, w is the width of the ROABP and d is the individual-degree bound of each variable. For the unknown-order case, Forbes et al. [14] gave an $n^{O(d \log w \log n)}$-time blackbox test. Observe that the complexity of their algorithm is quasi-polynomial only when d is small. Subsequently, Agrawal et al. [2] removed the exponential dependence on the individual degree. They gave an $(ndw)^{O(\log n)}$-time blackbox test for the unknown-order case. Note that these results remain quasi-polynomial even in the case of constant width. Studying ROABPs has also led to PIT results for other computational models, for example, subexponential-size hitting sets for depth-3 multilinear circuits [12] and subexponential-time whitebox test for read-k oblivious ABPs [4].

Another motivation to study ROABPs comes from their Boolean analogues, called read-once ordered branching programs (ROBP).[3] ROBPs have been studied extensively, with regard to the RL versus L question (randomized log-space versus log-space). The problem of finding hitting sets for ROABP can be viewed as an analogue of finding pseudorandom generators (PRG) for ROBP. A pseudorandom generator for a Boolean function f is an algorithm which can generate a probability distribution (with a small sample space) with the property that f cannot distinguish it from the uniform random distribution (see [5] for details). Constructing an optimal PRG for ROBP, i. e., with $O(\log n)$ seed length or polynomial-size sample space, would imply $\mathsf{RL} = \mathsf{L}$. Although the known pseudorandom generators for ROBPs and hitting-set generators for ROABPs in similar settings have similar complexity, there is no known way to translate the construction of one to another. The best known PRG is of seed length $O(\log^2 n)$ ($n^{O(\log n)}$-size sample space), when variable order is known [26, 20, 27]. On the other hand, in the unknown-order case, the best known seed length is of size $n^{1/2+o(1)}$ [19]. Finding an $O(\log n)$-seed PRG even for constant-width known-order ROBPs has been a challenging open question. Though, some special cases of this question have been solved—width-2 ROBPs [8], or nearly solved—permutation and regular ROBPs [9, 10, 24, 11, 30].

Our first result addresses the analogous question in the arithmetic setting. We give the first polynomial-time blackbox test for constant-width known-order ROABPs. However, it works only for zero or large

[1] In a *read-once ABP*, each variable occurs only once on every source-sink path. An ROABP is a read-once ABP where every occurrence of a variable is in the same layer.

[2] The work of [25] is actually on non-commutative ABPs but the same results apply to ROABP.

[3] ROBPs are also known as Ordered Binary Decision Diagrams (OBDDs).

characteristic fields. Our idea is inspired by the pseudorandom generator for ROBPs by Impagliazzo, Nisan and Wigderson [20]. While their result does not give better PRGs for the constant-width case, we are able to achieve this in the arithmetic setting.

Theorem (Theorem 3.6)**.** Let $\mathcal{C}$ be the class of n-variate, individual-degree-d polynomials in $\mathbb{F}[x]$ computed by a width-w ROABP in the variable order $(x_1, x_2, \ldots, x_n)$. Then a hitting set of size $dn^{O(\log w)}$ can be constructed for $\mathcal{C}$, when $\mathrm{char}(\mathbb{F}) = 0$ or $\mathrm{char}(\mathbb{F}) > ndw^{\log n}$.

When $w < n$, the size of our hitting set is smaller than the previously known hitting sets. Furthermore, even in the regime when $w \geq n$, the size of our hitting set matches the previously best known hitting sets. We show that for a nonzero bivariate polynomial $f(x_1, x_2)$ computed by a width-w ROABP, the univariate polynomial $f(t^w, t^w + t^{w-1})$ is nonzero. For this, we use the notion of rank of the partial derivative matrix of a polynomial, defined by Nisan [25]. Our argument is that the rank of the partial derivative matrix of any bivariate polynomial which becomes zero on $(t^w, t^w + t^{w-1})$ is more than w, while for a polynomial computed by a width-w ROABP, this rank is at most w. We use the map $(x_1, x_2) \mapsto (t^w, t^w + t^{w-1})$ recursively in $\log n$ rounds to achieve the above mentioned hitting set. Our technique has a crucial difference from the previous works on ROABPs [14, 15, 2]. The starting point in all the previous techniques is a monomial map, i. e., each variable is mapped to a monomial. On the other hand, we argue with a polynomial map directly (where each variable is mapped to a univariate polynomial). We believe that our approach could lead to a polynomial-size hitting set for ROABPs and we now describe a concrete construction that we conjecture works. The goal would be to obtain a univariate n-tuple $(p_1(t), \ldots, p_n(t))$, such that any polynomial which becomes zero on $(p_1(t), \ldots, p_n(t))$ must have rank or evaluation dimension higher than w. We conjecture that $(t^r, (t+1)^r, \ldots, (t+n-1)^r)$ is one such tuple, where r is polynomially large (Conjecture 3.8).

We believe that these ideas from the arithmetic setting can help in constructing an optimal PRG for constant-width ROBP.

Our second result is for the class of polynomials which are computable by ROABPs in any variable order. To be precise, a polynomial $f(x)$ is in this class if for every permutation of the variables, there exists an ROABP of width-w that computes $f(x)$ in that variable order. This class of polynomials has a slightly better hitting set than the class of polynomials computed by ROABPs in a particular variable order, but still no polynomial-size hitting set is known. The previously best known hitting set for them has size $d^{O(\log w)}(nw)^{O(\log\log w)}$ [14]. We improve this to $(ndw)^{O(\log\log w)}$.

Theorem (Theorem 4.9)**.** For n-variate, individual-degree-d polynomials computed by width-w ROABPs in any order, a hitting set of size $(ndw)^{O(\log\log w)}$ can be constructed.

To obtain this result we follow the approach of Forbes et al. [14], which used the notion of rank concentration or low-support concentration, a technique introduced by Agrawal et al. [3]. We achieve rank concentration more efficiently using the basis isolation technique of Agrawal et al. [2]. The same technique also yields a more efficient concentration in depth-3 set-multilinear circuits (see Section 2 for the definition). However, it is not clear if it gives better hitting sets for them. The best known hitting set for them has size $n^{O(\log n)}$ [3].

2 Preliminaries

2.1 Definitions and notation

We write $[n]$ to denote the set $\{1,2,\ldots,n\}$ and $[\![d]\!]$ to denote the set $\{0,1,\ldots,d\}$. The character x will denote a list of variables, usually the list $(x_1,x_2,\ldots,x_n)$. For a field $\mathbb{F}$, we denote the ring of polynomials over $\mathbb{F}$ by $\mathbb{F}[x]$, and the field of rational functions over $\mathbb{F}$ by $\mathbb{F}(t)$. For a set x of n variables and for an exponent $a=(a_1,a_2,\ldots,a_n)\in\{0,1,2,\ldots\}^n$, we denote the monomial $\prod_{i=1}^n x_i^{a_i}$ by x^a. The *support* of a monomial x^a, denoted by $\mathrm{Supp}(a)$, is the set of variables appearing in that monomial, i. e., $\{x_i \mid i\in[n], a_i>0\}$. The *support size* of a monomial is the cardinality of its support, denoted by $\mathrm{supp}(a)$. A monomial is said to be ℓ-support if its support size is ℓ and $(<\ell)$-support if its support size is $<\ell$. For a polynomial $P(x)$, the coefficient of a monomial x^a in $P(x)$ is denoted by $\mathrm{coef}_P(x^a)$.

For a monomial x^a, the sum $\sum_i a_i$ is said to be its *degree* and a_i is said to be its *degree in variable x_i* for each i. Similarly, for a polynomial P, its degree (or degree in x_i) is the maximum degree (or maximum degree in x_i) of any monomial in P with a nonzero coefficient. We define the *individual degree* of P to be $\max_i\{\deg_{x_i}(P)\}$, where $\deg_{x_i}$ denotes degree in x_i.

To better understand polynomials computed by ROABPs, we often use polynomials over an algebra $\mathbb{A}$, i. e., polynomials whose coefficients come from $\mathbb{A}$. Matrix algebra is the vector space of matrices equipped with the matrix product. $\mathbb{F}^{m\times n}$ represents the set of all $m\times n$ matrices over the field $\mathbb{F}$. Note that the algebra of $w\times w$ matrices, has dimension w^2.

We often view a vector/matrix with polynomial entries, as a polynomial with vector/matrix coefficients. For example,

$$D(x,y)=\begin{pmatrix}1+x & y-xy\\ x+y & 1+xy\end{pmatrix}=\begin{pmatrix}1&0\\0&1\end{pmatrix}1+\begin{pmatrix}1&0\\1&0\end{pmatrix}x+\begin{pmatrix}0&1\\1&0\end{pmatrix}y+\begin{pmatrix}0&-1\\0&1\end{pmatrix}xy\,.$$

Here, the coef_D operator will return a matrix for any monomial, for example,

$$\mathrm{coef}_D(y)=\begin{pmatrix}0&1\\1&0\end{pmatrix}.$$

For a polynomial $D(x)\in\mathbb{A}[x]$ over an algebra, its *coefficient space* is the space spanned by its coefficients.

For a matrix R, we denote its entry in the i-th row and j-th column by $R(i,j)$.

As mentioned earlier, a deterministic blackbox PIT is equivalent to constructing a hitting set. A set of points $\mathcal{H}\in\mathbb{F}^n$ is called a *hitting set* for a class $\mathcal{C}$ of n-variate polynomials if for any nonzero polynomial P in $\mathcal{C}$, there exists a point in $\mathcal{H}$ where P evaluates to a nonzero value.

2.2 Arithmetic branching programs

Definition 2.1 (Arithmetic Branching Program (ABP))**.** An ABP is a layered directed acyclic graph with $q+1$ layers of vertices $\{V_0,V_1,\ldots,V_q\}$ and a source a and a sink b such that all the edges of the graph only go from a to V_0, V_{i-1} to V_i for any $i\in[q]$ and V_q to b. The edges have univariate polynomials as their weights and as a convention, the edges going out of u and the edges going into t have constant weights, i. e., weights from the field $\mathbb{F}$. The ABP is said to compute the polynomial

$$f(x)=\sum_{p\in\mathrm{paths}(a,b)}\prod_{e\in p}W(e)\,,$$

where $W(e)$ is the weight of the edge e.

The ABP has width-w if $|V_i| \leq w$ for all $i \in [\![q]\!]$. Without loss of generality we can assume $|V_i| = w$ for each $i \in [\![q]\!]$.

It is well-known that the sum over all paths in a layered graph can be represented by an iterated matrix multiplication. To see this, let the set of nodes in V_i be $\{v_{i,j} \mid j \in [w]\}$. It is easy to see that the polynomial computed by the ABP is the same as $A^{\mathsf{T}}(\prod_{i=1}^{q} D_i)B$, where $A, B \in \mathbb{F}^{w\times 1}$ and D_i is a $w \times w$ matrix for $1 \leq i \leq q$ such that

$$\begin{aligned} A(\ell) &= W(a, v_{0,\ell}) && \text{for } 1 \leq \ell \leq w\,, \\ D_i(k,\ell) &= W(v_{i-1,k}, v_{i,\ell}) && \text{for } 1 \leq \ell, k \leq w \text{ and } 1 \leq i \leq q\,, \\ B(k) &= W(v_{q,k}, b) && \text{for } 1 \leq k \leq w\,. \end{aligned}$$

2.2.1 Read-once oblivious ABP

An ABP is called a *read-once oblivious ABP* (ROABP) if the edge weights in different layers are univariate polynomials in distinct variables. Formally, there is a permutation π on the set $[q]$ such that the entries in the ith matrix D_i are univariate polynomials over the variable $x_{\pi(i)}$, i. e., they come from the polynomial ring $\mathbb{F}[x_{\pi(i)}]$. Here, q is the same as n, the number of variables. The order $(x_{\pi(1)}, x_{\pi(2)}, \ldots, x_{\pi(n)})$ is said to be the variable order of the ROABP.

Viewing $D_i(x_{\pi(i)}) \in \mathbb{F}^{w\times w}[x_{\pi(i)}]$ as a polynomial over the matrix algebra, we can write the polynomial computed by an ROABP as

$$f(x) = A^{\mathsf{T}} D_1(x_{\pi(1)}) D_2(x_{\pi(2)}) \cdots D_n(x_{\pi(n)}) B\,.$$

An equivalent representation of a width-w ROABP can be

$$f(x) = D_1(x_{\pi(1)}) D_2(x_{\pi(2)}) \cdots D_n(x_{\pi(n)})\,,$$

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 $D_n \in \mathbb{F}^{w\times 1}[x_{\pi(n)}]$.

2.2.2 Any-order ROABP

A polynomial $f(x)$ is said to be computed by width-w ROABPs in any order, if for every permutation σ of the variables, there exists a width-w ROABP in the variable order σ that computes the polynomial $f(x)$.

2.2.3 Set-multilinear circuits

A depth-3 set-multilinear circuit is a circuit of the form

$$f(x) = \sum_{i=1}^{k} l_{i,1}(x_1)\, l_{i,2}(x_2) \cdots\, l_{i,q}(x_q)\,,$$

where $l_{i,j}$s are linear polynomials and $x_1, x_2, \ldots, x_q$ form of a partition of the set x of variables. It is known that these circuits are subsumed by ROABPs [14]. However, the polynomials computed by these circuits may not be computable by ROABPs in any order. For example, the $2n$-variate polynomial $(x_1+y_1)(x_2+y_2)\cdots(x_n+y_n)$ has a linear-size set-multilinear circuit. But, every ROABP in the variable sequence $(x_1,x_2,\ldots,x_n,y_1,y_2,\ldots,y_n)$ that computes it has width $\geq 2^n$ (follows from Nisan's characterization [25]).

Note that ROABPs can compute polynomials with individual-degree ≥ 1, but set-multilinear circuits cannot. It is not known whether all multilinear polynomials computed by ROABPs in any order can also be computed by polynomial-size set-multilinear circuits.

A set-multilinear circuit has a corresponding polynomial over a commutative algebra. For the polynomial $f(x)$ above, consider the polynomial over a k-dimensional algebra

$$D(x) = D_1(x_1)D_2(x_2)\cdots D_q(x_q)\,,$$

where $D_j = (l_{1,j}, l_{2,j}, \ldots, l_{k,j})$ and the algebra product is coordinate-wise product. It is easy to see that $f = (1,1,\ldots,1)\cdot D$. Note that the polynomials D_is are over a commutative algebra, that is, the order of the D_is in the product does not matter. Hence, some of our techniques for any-order ROABPs also work for set-multilinear circuits.

3 Hitting set for known-order ROABP

3.1 Bivariate ROABP

To construct a hitting set for ROABPs, we start with the bivariate case. Recall that a bivariate ROABP is of the form $A^{\mathsf{T}}D_1(x_1)D_2(x_2)B$, where $A,B\in\mathbb{F}^{w\times 1}$, $D_1\in\mathbb{F}^{w\times w}[x_1]$ and $D_2\in\mathbb{F}^{w\times w}[x_2]$. It is easy to see that a bivariate polynomial $f(x_1,x_2)$ computed by a width-w ROABP can be written as $f(x_1,x_2)=\sum_{r=1}^{w} g_r(x_1)h_r(x_2)$. To construct a hitting set for this polynomial, we will use the notion of a partial derivative matrix, defined by Nisan [25] in the context of lower bounds. Let the individual degree of the polynomial $f\in\mathbb{F}[x_1,x_2]$ be bounded by d. The *partial derivative matrix* M_f for f is a $(d+1)\times(d+1)$ matrix with

$$M_f(i,j) = \mathrm{coef}_f(x_1^i x_2^j) \in \mathbb{F}\,,$$

for all $i,j\in[\![d]\!]$. It is known that the rank of the matrix M_f equals the smallest possible width of any ROABP computing f [25].

Lemma 3.1 (rank $\leq$ width)**.** *For any polynomial* $f(x_1,x_2)=\sum_{r=1}^{w} g_r(x_1)h_r(x_2)$*, we have* $\mathrm{rank}(M_f)\leq w$.

Proof. Let us define $f_r = g_r h_r$, for all $r\in[w]$. Clearly, $M_f=\sum_{r=1}^{w} M_{f_r}$, as $f=\sum_{r=1}^{w} f_r$. We will show that $\mathrm{rank}(M_{f_r})\leq 1$, for all $r\in[w]$. As $f_r = g_r(x_1)h_r(x_2)$, its coefficients can be written as a product of coefficients from g_r and h_r, i. e.,

$$\mathrm{coef}_{f_r}(x_1^i x_2^j) = \mathrm{coef}_{g_r}(x_1^i)\,\mathrm{coef}_{h_r}(x_2^j)\,.$$

Now, it is easy to see that

$$M_{f_r} = u_r v_r^{\mathsf{T}}\,,$$

where $u_r, v_r\in\mathbb{F}^{d+1}$ with $u_r = (\mathrm{coef}_{g_r}(x_1^i))_{i=0}^{d}$ and $v_r = (\mathrm{coef}_{h_r}(x_2^i))_{i=0}^{d}$.

Thus, $\mathrm{rank}(M_{f_r})\leq 1$ and $\mathrm{rank}(M_f)\leq w$. ☐

One can also show that if $\mathrm{rank}(M_f) = w$ then there exists a width-w ROABP computing f. We skip this proof as we will not need it. Now, using the above lemma we give a hitting set for bivariate ROABPs.

Lemma 3.2. *Suppose* $\mathrm{char}(\mathbb{F}) = 0$*, or* $\mathrm{char}(\mathbb{F}) > d$*. Let*

$$f(x_1,x_2) = \sum_{r=1}^{w} g_r(x_1)h_r(x_2)$$

be a nonzero bivariate polynomial over $\mathbb{F}$ *of individual degree* d*. Then* $f(t^w, t^w + t^{w-1}) \neq 0$*.*

Proof. Let $\tilde{f}(t)$ be the polynomial after the substitution, i. e., $\tilde{f}(t) = f(t^w, t^w + t^{w-1})$. Any monomial $x_1^i x_2^j$ will be mapped to the polynomial $t^{wi}(t^w + t^{w-1})^j$, under the mentioned substitution. The highest power of t coming from this polynomial is $t^{w(i+j)}$. We will cluster together all the monomials for which this highest power is the same, i. e., $i+j$ is the same. The set of coefficients corresponding to any such cluster of monomials will form a *diagonal* in the matrix M_f. The set $\{M_f(i,j) \mid i+j=k\}$ is defined to be the *k-th diagonal* of M_f, for all $0 \leq k \leq 2d$. Let ℓ be the largest number such that the ℓ-th diagonal has at least one nonzero element, i. e.,

$$\ell = \max\{i+j \mid M_f(i,j) \neq 0\}\,.$$

As $\mathrm{rank}(M_f) \leq w$ (from Lemma 3.1), we claim that the ℓ-th diagonal has at most w nonzero elements. To see this, let $\{(i_1, j_1), (i_2, j_2), \ldots, (i_{w'}, j_{w'})\}$ be the set of indices where the ℓ-th diagonal of M_f has nonzero elements, i. e., the set $\{(i,j) \mid M_f(i,j) \neq 0,\ i+j=\ell\}$. Observe that $w' \leq d+1$. As $M_f(i,j) = 0$ for any $i+j > \ell$, it is easy to see that the rows $\{M_f(i_1), M_f(i_2), \ldots, M_f(i_{w'})\}$ are linearly independent. Thus, $w' \leq \mathrm{rank}(M_f) \leq w$.

Now, we claim that there exists an r with $w(\ell-1) < r \leq w\ell$ such that $\mathrm{coef}_{\tilde{f}}(t^r) \neq 0$. To see this, first observe that the highest power of t to which any monomial $x_1^i x_2^j$ with $i+j < \ell$ can contribute is $t^{w(\ell-1)}$. Thus, for any $w(\ell-1) < r \leq w\ell$, the term t^r can come only from the monomials $x_1^i x_2^j$ with $i+j \geq \ell$. We can ignore the monomials $x_1^i x_2^j$ with $i+j > \ell$ as $\mathrm{coef}_f(x_1^i x_2^j) = M_f(i,j) = 0$, when $i+j > \ell$. Now, for any $i+j = \ell$, the monomial $x_1^i x_2^j$ maps to

$$t^{w(\ell-j)}(t^w + t^{w-1})^j = t^{w\ell}(1+t^{-1})^j = \sum_{p=0}^{j} \binom{j}{p} t^{w\ell-p}\,.$$

Hence, for any $0 \leq p < w$,

$$\mathrm{coef}_{\tilde{f}}(t^{w\ell-p}) = \sum_{b=1}^{w'} M_f(i_b, j_b)\binom{j_b}{p}\,.$$

Here we assume that if $p > j_b$, then $\binom{j_b}{p} = 0$. Writing the above equation in the matrix form, we get,

$$\begin{pmatrix} \mathrm{coef}_{\tilde{f}}(t^{w\ell}) \\ \vdots \\ \mathrm{coef}_{\tilde{f}}(t^{w\ell-w+1}) \end{pmatrix} = C \begin{pmatrix} M_f(i_1, j_1) \\ \vdots \\ M_f(i_{w'}, j_{w'}) \end{pmatrix},$$

where C is a $w \times w'$ matrix with $C(a,b) = \binom{j_b}{a-1}$, for all $a \in [w]$ and $b \in [w']$. If all the columns of C are linearly independent, then clearly, $\mathrm{coef}_{\tilde{f}}(t^r) \neq 0$ for some $w(\ell-1) < r \leq w\ell$. We show the linear independence of the columns in Claim 3.3. To show this linear independence we need to assume that the numbers $\{j_b\}_b$ are all distinct. Hence, we need the field characteristic to be zero or strictly greater than d, as j_b can be as high as d for some $b \in [w']$.

Claim 3.3. *Let C' be the $w' \times w'$ submatrix of C with $C'(a,b) = \binom{j_b}{a-1}$, for all $a \in [w']$ and $b \in [w']$. Then C' has full rank.*

Proof. We will show that for any nonzero vector $\alpha := (\alpha_1, \alpha_2, \ldots, \alpha_{w'}) \in \mathbb{F}^{1\times w'}$, $\alpha C' \neq 0$. Consider the polynomial

$$h(y) = \alpha_1 + \alpha_2 \frac{y}{1!} + \alpha_3 \frac{y(y-1)}{2!} + \cdots + \alpha_{w'} \frac{y(y-1)\cdots(y-w'+2)}{(w'-1)!}\,.$$

As $h(y)$ is a nonzero polynomial of degree bounded by $w'-1$, it can have at most $w'-1$ roots. Thus, there exists an $b \in [w']$ such that

$$h(j_b) = \sum_{a=1}^{w'} \alpha_a \binom{j_b}{a-1} \neq 0\,. \qquad \square$$

This concludes the proof of Lemma 3.2. $\square$

As mentioned above, the hitting-set proof works only when the field characteristic is zero or greater than d. We given an example over a small characteristic field, which demonstrates that the problem is not with the proof technique, but with the hitting set itself. Let the field characteristic be 2. Consider the polynomial $f(x_1,x_2) = x_2^2 + x_1^2 + x_1$. Clearly, f has a width-2 ROABP. For a width-2 ROABP, the map in Lemma 3.2 would be $(x_1,x_2) \mapsto (t^2, t^2+t)$. However, $f(t^2,t^2+t) = 0$ (over $\mathbb{F}_2$). Hence, the hitting set does not work.

Now, we move on to getting a hitting set for an n-variate ROABP.

3.2 n-variate ROABP

Observe that the map given in Lemma 3.2 works irrespective of the degree of the polynomial, as long as the field characteristic is large enough. We plan to obtain a hitting set for general n-variate ROABP by applying this map recursively. For this, we use the standard divide and conquer technique. First, we make pairs of consecutive variables in the ROABP. For each pair (x_{2i-1}, x_{2i}), we apply the map from Lemma 3.2, using a new variable t_i. Thus, we go to $n/2$ variables from n variables. In Lemma 3.4, we use a hybrid argument to show that after this substitution the polynomial remains nonzero. Moreover, the new polynomial can be computed by a width-w ROABP. Thus, we can again use the same map on pairs of new variables. By repeating the halving procedure $\log n$ times we get a univariate polynomial. In each round the degree of the polynomial gets multiplied by w. Hence, after $\log n$ rounds, the degree of the univariate polynomial is bounded by $w^{\log n}$ times the original degree. Without loss of generality, let us assume that n is a power of 2.

Lemma 3.4 (Halving the number of variables)**.** *Suppose* $\mathrm{char}(\mathbb{F}) = 0$*, or* $\mathrm{char}(\mathbb{F}) > d$*. Let $f(x) = D_1(x_1)D_2(x_2)\cdots D_n(x_n)$ be a nonzero polynomial of individual degree d and computed by a width-w*

ROABP, where $D_1 \in \mathbb{F}^{1\times w}[x_1]$, $D_n \in \mathbb{F}^{w\times 1}[x_n]$ and $D_i \in \mathbb{F}^{w\times w}[x_i]$ for all $2 \le i \le n-1$. Let the map $\phi\colon x \to \mathbb{F}[t]$ be such that for any index $1 \le i \le n/2$,

$$\phi(x_{2i-1}) = t_i^w,$$
$$\phi(x_{2i}) = t_i^w + t_i^{w-1}.$$

Then $f(\phi(x)) \neq 0$. Moreover, the polynomial $f(\phi(x)) \in \mathbb{F}[t_1,t_2,\dots,t_{n/2}]$ is computed by a width-w ROABP in the variable order $(t_1,t_2,\dots,t_{n/2})$.

Proof. Let us apply the map in $n/2$ rounds, i. e., define a sequence of polynomials $(f = f_0, f_1, \dots, f_{n/2} = f(\phi(x)))$ such that the polynomial f_i is obtained by replacing (x_{2i-1}, x_{2i}) with $(\phi(x_{2i-1}), \phi(x_{2i}))$ in f_{i-1} for each $1 \le i \le n/2$. We will show that for each $1 \le i \le n/2$, if $f_{i-1} \neq 0$ then $f_i \neq 0$. Clearly this proves the first part of the lemma.

Note that f_{i-1} is a polynomial over variables $\{t_1,\dots,t_{i-1},x_{2i-1},\dots,x_n\}$. As $f_{i-1} \neq 0$, there exists a constant tuple $\alpha \in \mathbb{F}^{n-i-1}$ such that after replacing the variables $(t_1,\dots,t_{i-1}, x_{2i+1},\dots,x_n)$ with α, the polynomial f_{i-1} remains nonzero. After this replacement we get a polynomial $\hat{f}_{i-1}$ in the variables (x_{2i-1},x_{2i}). As f is computed by the ROABP $D_1 D_2 \cdots D_n$, the polynomial $\hat{f}_{i-1}$ can be written as $A^{\mathsf{T}} D_{2i-1}(x_{2i-1}) D_{2i}(x_{2i}) B$ for some $A, B \in \mathbb{F}^{w\times 1}$. In other words, $\hat{f}_{i-1}$ has a bivariate ROABP of width-w. Thus, $\hat{f}_{i-1}(\phi(x_{2i-1}),\phi(x_{2i}))$ is nonzero from Lemma 3.2. But, $\hat{f}_{i-1}(\phi(x_{2i-1}),\phi(x_{2i}))$ is nothing but the polynomial obtained after replacing the variables $(t_1,\dots,t_{i-1},x_{2i+1},\dots,x_n)$ in f_i with α. Thus, f_i is nonzero. This finishes the proof.

Now, we argue that $f(\phi(x))$ has a width-w ROABP. Let $\tilde{D}_i := D_{2i-1}(t_i^w) D_{2i}(t_i^w + t_i^{w-1})$ for all $1 \le i \le n/2$. Clearly, $\tilde{D}_1 \tilde{D}_2 \cdots \tilde{D}_{n/2}$ is a width-w ROABP computing $f(\phi(x))$ in variable order $(t_1,t_2,\dots,t_{n/2})$, as $\tilde{D}_1 \in \mathbb{F}^{1\times w}[t_1]$, $\tilde{D}_{n/2} \in \mathbb{F}^{w\times 1}[t_{n/2}]$ and $\tilde{D}_i \in \mathbb{F}^{w\times w}[t_i]$ for all $2 \le i \le n/2-1$. □

By applying the map ϕ in Lemma 3.4, we reduced an n-variate ROABP to an $(n/2)$-variate ROABP, while preserving the non-zeroness. The resulting ROABP has same width-w, but the individual degree goes up to become $2dw$, where d is the original individual degree. As our map ϕ is degree insensitive, we can apply a similar map again on the variables $\{t_i\}_{i=1}^{n/2}$. That is, for $1 \le i \le n/4$, define $\phi(t_{2i-1}) = s_i^w$ and $\phi(t_{2i}) = s_i^w + s_i^{w-1}$ for variables $\{s_1,s_2,\dots,s_{n/4}\}$. Now, we get an $(n/4)$-variate ROABP of individual degree $4dw^2$. It is easy to see that when the map ϕ is repeatedly applied in this way $\log n$ times, we get a nonzero univariate polynomial of degree $ndw^{\log n}$. Next lemma puts it formally. For ease of notation, we use the variable numbering from 0 to $n-1$. Let $p_0(t) = t^w$ and $p_1(t) = t^w + t^{w-1}$.

Lemma 3.5. *Suppose* $\mathrm{char}(\mathbb{F}) = 0$, *or* $\mathrm{char}(\mathbb{F}) \ge ndw^{\log n}$. *Let $f \in \mathbb{F}[x]$ be a nonzero polynomial of individual degree d and computed by a width-w ROABP in variable order $(x_0,x_1,\dots,x_{n-1})$. Let the map $\phi\colon \{x_0,x_1,\dots,x_{n-1}\} \to \mathbb{F}[t]$ be such that for any index $0 \le i \le n-1$,*

$$\phi(x_i) = p_{i_1}(p_{i_2}\cdots(p_{i_{\log n}}(t))),$$

where $i_{\log n}\, i_{\log n-1} \cdots i_1$ is the binary representation of i.

Then $f(\phi(x))$ is a nonzero univariate polynomial of degree $ndw^{\log n}$.

Note that the map ϕ crucially uses the knowledge of the variable order. In the last round when we are going from two variables to one, the individual degree is $ndw^{\log n-1}$ and Lemma 3.2 requires $\mathrm{char}(\mathbb{F})$

to be higher than the individual degree. Thus, having $\text{char}(\mathbb{F}) \geq ndw^{\log n}$ suffices. Hence, we get the following theorem.

Theorem 3.6. *Let $\mathcal{C}$ be the class of n-variate, individual-degree-d polynomials computed by width-w ROABPs. Then a hitting set for $\mathcal{C}$ of size $O(ndw^{\log n})$ can be constructed, when the variable order is known and the field characteristic is zero or at least $ndw^{\log n}$.*

Proof. Let $f(x)$ be a polynomial in class $\mathcal{C}$. From Lemma 3.5, $f(\phi(x)) \in \mathbb{F}[t]$ is a nonzero univariate polynomial of degree $ndw^{\log n}$. Thus, if we substitute $1 + ndw^{\log n}$ field values for the variable t, one of them will keep $f(\phi(x))$ nonzero. □

From this, we immediately get the following result for constant-width ROABPs. Note that when w is constant, the lower bound on the characteristic also becomes $\mathsf{poly}(n)$.

Corollary 3.7. *For the class of n-variate, individual-degree-d polynomials computed by constant width ROABPs (known variable order), a $\mathsf{poly}(n,d)$-size hitting set can be constructed, when the field characteristic is zero (or larger than poly(n,d)).*

As mentioned earlier, our approach can potentially lead to a polynomial-size hitting set for ROABPs. We make the following conjecture for which we hope to get a proof on the lines of Lemma 3.2.

Conjecture 3.8. *Suppose* $\text{char}(\mathbb{F}) = 0$. *Let $f(x) \in \mathbb{F}[x]$ be an n-variate, degree-d polynomial computed by a width-w ROABP. Then $f(t^r, (t+1)^r, \ldots, (t+n-1)^r) \neq 0$ for some r bounded by* $\mathsf{poly}(n,w,d)$.

4 Any-order ROABP

In this section, we give better hitting sets for the class of polynomials computed by ROABPs in any order. Recall that a polynomial $f(x) \in \mathbb{F}[x]$ is said to be computed by width-w ROABPs in any order if for any permutation $\sigma\colon [n] \to [n]$, $f(x)$ can be written as

$$A^{\mathsf{T}} D_1(x_{\sigma(1)}) D_2(x_{\sigma(2)}) \cdots D_n(x_{\sigma(n)}) B$$

for some $D_i \in \mathbb{F}^{w \times w}[x_{\sigma(i)}]$ for $1 \leq i \leq n$ and $A, B \in \mathbb{F}^{w \times 1}$.

We will also consider ROABPs which compute a polynomial over the matrix algebra, that is, polynomials whose coefficients are matrices. $D(x) \in \mathbb{F}^{w \times w}[x]$ is said to be computed by a width-w ROABP if $D(x) = D_1 D_2 \cdots D_n$ for some polynomials $D_i \in \mathbb{F}^{w \times w}[x_{\sigma(i)}]$ for $1 \leq i \leq n$.

Forbes et al. [14] gave a hitting set of size $d^{O(\log w)}(nw)^{O(\log \log w)}$ for n-variate, individual-degree-d polynomials computed by width-w ROABPs in any order. Note that when d is small, this hitting-set size is much better than that for ROABPs (with a particular variable order), i. e., $(ndw)^{O(\log n)}$ [2]. However when d is $\Omega(n)$, the size is comparable to the latter case. We improve the hitting-set size for polynomials computed by ROABPs in any order to $(ndw)^{O(\log \log w)}$. This is significantly better than the case of polynomials computed by ROABPs in a particular variable order.

4.1 Rank-concentration

Forbes et al. [14] constructed the hitting set using the notion of rank-concentration defined by Agrawal et al. [3]. Recall that $D(x)$ is a polynomial over an algebra if its coefficients come from the algebra.

Definition 4.1 ([3])**.** A polynomial $D(x)$ over an algebra is said to be ℓ-concentrated if its coefficients of $(<\ell)$-support monomials span all its coefficients. That is, for all $a \in \{0,1,2,\dots\}^n$

$$\mathrm{coef}_D(x^a) \in \mathrm{span}\{\mathrm{coef}_D(x^b) \mid b \in \{0,1,2,\dots\}^n,\ \mathrm{supp}(b) < \ell\}. \tag{4.1}$$

Note that for a nonzero polynomial over a field, ℓ-concentration simply means that one of its monomials of support $<\ell$ has a nonzero coefficient. As we will see later, it is easy to construct hitting sets for a polynomial which has low-support concentration. However, not every polynomial has a low-support concentration, for example, consider the following polynomial over a field: $f(x) = x_1x_2\dots x_n$. Agrawal et al. [3] observed that concentration can be achieved by a shift of variables, e. g., $f(x+1) = (x_1+1)(x_2+1)\cdots(x_n+1)$ has 1-concentration. For a polynomial $f(x)$, shift by a tuple $s = (s_1,s_2,\dots,s_n)$ would mean $f(x+s) = f(x_1+s_1,x_2+s_2,\dots,x_n+s_n)$.

To achieve concentration, it is often useful to consider shifts which are polynomials. In particular, we will be considering shifts by bivariate polynomials, i. e., $s(t_1,t_2) \in \mathbb{F}[t_1,t_2]^n$. As ultimately we are interested in hitting sets, the variables t_1 and t_2 can later be replaced by field values. The size of the hitting set, in this case, will be multiplied by δ^2, where δ is the maximum degree of any $s_i(t_1,t_2)$. Thus, for a bivariate shift $s(t_1,t_2)$, its degree will be viewed as the complexity measure. Note that for a polynomial $D(x) \in \mathbb{F}^{w\times w}[x]$, the coefficient of a monomial x^a in $D(x+s(t_1,t_2))$ will be from $\mathbb{F}[t_1,t_2]^{w\times w}$. So, when we talk of low-support concentration in $D(x+s(t_1,t_2))$, the span in (4.1) is taken over the field $\mathbb{F}(t_1,t_2)$.

Forbes et al. [14] construct the hitting set for polynomials computed by ROABPs in any order in two steps. Let $f(x)$ be an n-variate individual-degree-d polynomial computed by width-w ROABPs in any order. Their first step is to construct a tuple $s(t_1,t_2)$ of bivariate polynomials of degree $\mathrm{poly}(n)d^{O(\log w)}$ such that $f(x+s)$ has $O(\log w)$-concentration. We improve this step by constructing a new tuple $s(t_1,t_2)$ of degree $(ndw)^{O(\log\log w)}$, which has the same property.

We follow the second step of Forbes et al. [14] as it is. It is easy to see that $f(x+s)$ can also be computed by width-w ROABPs in any order (over the field $\mathbb{F}(t_1,t_2)$). They show that if a given polynomial, computed by ROABPs in any order, is ℓ-concentrated then there is a hitting set for it of size $(ndw)^{O(\log \ell)}$. This implies a hitting set $\mathcal{H}$ of size $(ndw)^{O(\log\log w)}$ for $f(x+s)$. Clearly, the set $\{h+s \mid h \in \mathcal{H}\}$ is a hitting set for $f(x)$. One can obtain a hitting set in $\mathbb{F}^n$ by replacing t_1 and t_2 with sufficiently many field values. By Schwartz-Zippel-DeMillo-Lipton Lemma, it will suffice to take more than $\deg_{t_1,t_2}(f(h+s)) = \deg(f)\cdot\deg(s)$ values. Thus, the final hitting-set size becomes $\deg(s)\cdot(ndw)^{O(\log\log w)}$. With our improved bound on $\deg(s)$, we get a hitting set of the desired size.

Now, we elaborate the first step of Forbes et al. [14], i. e., the construction of the shift $s(t_1,t_2)$. To achieve concentration they use the idea of Agrawal, Saha and Saxena [3], i. e., achieving concentration in small sub-ROABPs implies concentration in the given ROABP. For the sake of completeness, we rewrite the lemma using the terminology of this paper. We first clarify a notation which will be used often: for an n-tuple s and a polynomial $D(x)$ which only depends variables $(x_{i_1},x_{i_2},\dots,x_{i_\ell})$, the expression $D(x+s)$ will denote $D(x_{i_1}+s_{i_1},x_{i_2}+s_{i_2},\dots,x_{i_\ell}+s_{i_\ell})$.

Lemma 4.2 ([3, 14])**.** *Let $\ell < n$ be any number. Let s be the n-tuple such that for any distinct $i_1, i_2, \dots, i_\ell \in [n]$ and individual-degree-d polynomial*

$$D(x) = D_1(x_{i_1}) D_2(x_{i_2}) \cdots D_\ell(x_{i_\ell})$$

over the matrix algebra $\mathbb{F}^{w\times w}$, $D(x+s)$ is ℓ-concentrated. Then for any individual-degree-d polynomial $f(x) \in \mathbb{F}[x]$ computed by width-w ROABPs in any order, $f(x+s)$ is ℓ-concentrated.

Proof. Let $f'(x) = f(x+s)$. Consider any monomial x^a with support $\geq \ell$. We will show that its coefficient in $f'(x)$ is in the span of smaller support coefficients in $f'(x)$. Let $S = \{x_{i_1}, x_{i_2}, \dots, x_{i_\ell}\}$ be a set of ℓ variables contained in the support of monomial x^a. Let $\overline{S} = \{x_{i_{\ell+1}}, \dots, x_{i_n}\}$ be the rest of the variables. Let us write $x^a = x^b x^c$ with $\mathrm{Supp}(b) = S$ and $\mathrm{Supp}(c) \subseteq \overline{S}$. Since, $f(x)$ is computed by ROABPs in any order, it has an ROABP in the variable order $(x_{i_1}, \dots, x_{i_\ell}, x_{i_{\ell+1}}, \dots, x_{i_n})$. That is,

$$f(x) = A^{\mathsf{T}} D_1(x_{i_1}) \cdots D_\ell(x_{i_\ell}) D_{\ell+1}(x_{i_{\ell+1}}) \dots D_n(x_{i_n}) B$$

for some $D_j \in \mathbb{F}^{w\times w}[x_{i_j}]$ for $1 \leq j \leq n$ and $A, B \in \mathbb{F}^{w\times 1}$. Let

$$D(x) := D_1(x_{i_1}) \cdots D_\ell(x_{i_\ell}) \qquad \text{and} \qquad E(x) := D_{\ell+1}(x_{i_{\ell+1}}) \dots D_n(x_{i_n})\,.$$

Let $D'(x) = D(x+s)$ and $E'(x) = E(x+s)$. Clearly $f'(x) = A^{\mathsf{T}} D'(x) E'(x) B$. By the lemma hypothesis, $D'(x)$ is ℓ-concentrated. That is,

$$\mathrm{coef}_{D'}(x^b) \in \mathrm{span}\{\mathrm{coef}_{D'}(x^{b'}) \mid \mathrm{Supp}(b') \subseteq S,\ \mathrm{supp}(b') < \ell\}\,. \tag{4.2}$$

Note that we have $\mathrm{Supp}(b') \subseteq S$ because each monomial in $D'(x)$ comes from set S. It is easy to see that for any monomial $x^{b'}$ with $\mathrm{Supp}(b') \subseteq S$

$$\mathrm{coef}_{f'}(x^{b'} x^c) = A^{\mathsf{T}}\, \mathrm{coef}_{D'}(x^{b'})\, \mathrm{coef}_{E'}(x^c) B\,.$$

Thus, by left multiplying A^{T} and right multiplying $\mathrm{coef}_{D'}(x^c)B$ in (4.2), we get

$$\mathrm{coef}_{f'}(x^a) \in \mathrm{span}\{\mathrm{coef}_{f'}(x^{b'} x^c) \mid \mathrm{Supp}(b') \subseteq S,\ \mathrm{supp}(b') < \ell\}\,.$$

Note that $\mathrm{supp}(b') + \mathrm{supp}(c) < \mathrm{supp}(b) + \mathrm{supp}(c) = \mathrm{supp}(a)$. So, we can write

$$\mathrm{coef}_{f'}(x^a) \in \mathrm{span}\{\mathrm{coef}_{f'}(x^{a'}) \mid \mathrm{supp}(a') < \mathrm{supp}(a)\}\,.$$

In other words, for any monomial x^a with $\mathrm{supp}(a) \geq \ell$, $\mathrm{coef}_{f'}(x^a)$ is in the span of coefficients of support smaller than $\mathrm{supp}(a)$. This would mean that, in fact, all coefficients of $f'(x)$ are in the span of coefficients with support $< \ell$. $\square$

Now, for some $\ell \leq n$, the goal is to construct an n-tuple s such that for any distinct $i_1, i_2, \dots, i_\ell \in n$, shifting by s ensures ℓ-concentration in any ℓ-variate ROABP of the form $D(x) = D_1(x_{i_1}) D_2(x_{i_2}) \cdots D_\ell(x_{i_\ell})$. Note that Lemma 4.2 holds for any value of $\ell \leq n$. However, one cannot choose ℓ to be arbitrary small. The reason is that for an ℓ-variate polynomial over a k-dimensional algebra, one can hope to achieve

ℓ-concentration only when $\ell \geq \log(k+1)$. To see this, consider the polynomial $D(x) = \prod_{i=1}^{\ell}(1+v_i x_i)$ over the algebra of $k \times k$ diagonal matrices, with $k = 2^\ell$. Here, 1 stands for the matrix $\mathrm{diag}(1,1,\ldots,1)$. Define $v_1 = \mathrm{diag}(\alpha_1, \alpha_2, \ldots, \alpha_k)$ for some distinct α_is. And define $v_i = v_1^{2^{i-1}}$ for $2 \leq i \leq \ell$. It is not hard to see that the 2^ℓ coefficients of the polynomial D are $\{1, v_1, v_1^2, \ldots, v_1^{2^\ell - 1}\}$, which are linearly independent. Note that since shifting is an invertible operation, the 2^ℓ coefficients of $D(x+s)$ will also be linearly independent for any s. But, there are only $2^\ell - 1$ monomials with support $< \ell$. Hence, the coefficients of $(<\ell)$-support monomials cannot span all the coefficients in $D(x+s)$, for any shift s.

Note that the dimension of the algebra $\mathbb{F}^{w \times w}$ is bounded by w^2. To reiterate the goal, given n and w, we fix $\ell = \lceil \log(w^2+1) \rceil$ and we want to achieve ℓ-concentration in all polynomials computed by an ROABP of the form $D_1 D_2 \cdots D_\ell$ where $D_j \in \mathbb{F}^{w \times w}[x_{i_j}]$ for $1 \leq j \leq \ell$, for some distinct $i_1, i_2, \ldots, i_\ell \in [n]$. As now we are dealing with polynomials in a small number of variables, it should be easier to achieve the concentration.

Towards this goal, Forbes et al. [14] give a bit more general result. For any $\ell \geq \log(w^2+1)$, they construct a tuple $s \in \mathbb{F}[t_1, t_2]^n$ of degree $\mathsf{poly}(n) d^{O(\ell)}$ which has the following property: for any polynomial $D(x) \in \mathbb{F}^{w \times w}[x]$ which uses at most ℓ of the n variables and has individual-degree bound d, $D(x+s)$ has ℓ-concentration. Here, Forbes et al. [14] do not need that $D(x)$ is computed by an ROABP.

We, on the other hand, use the property that $D(x)$ is computed by a width-w, ℓ-variate ROABP and reduce the degree of $s(t_1,t_2)$ to $(ndw)^{O(\log \ell)}$. Our construction of $s(t_1,t_2)$ comes from the basis isolating weight assignment for ROABPs from Agrawal et al. [2]. We use the fact that for any polynomial over a k-dimensional algebra, shift by a basis isolating map achieves $\log(k+1)$-concentration [17].

4.2 Basis isolation

Let us first recall the definition of a basis isolating weight assignment. Let M denote the set of all monomials over the variable set x with individual-degree $\leq d$. Any function $\mathrm{w}\colon x \to \{0,1,2,\ldots\}$ can be naturally extended to the set of all monomials as follows: $\mathrm{w}(\prod_{i=1}^n x_i^{\gamma_i}) = \sum_{i=1}^n \gamma_i \mathrm{w}(x_i)$, for any $(\gamma_i)_{i=1}^n \in \{0,1,2,\ldots\}^n$. Note that if the variable x_i is replaced with $t^{\mathrm{w}(x_i)}$ for each i, then any monomial m just becomes $t^{\mathrm{w}(m)}$. Let $\mathbb{A}_k$ denote a k-dimensional algebra.

Definition 4.3 ([2]). A weight function $\mathrm{w}\colon x \to \{0,1,2,\ldots\}$ is called a basis isolating weight assignment for a polynomial $D(x) \in \mathbb{A}_k[x]$, if there exists a set of monomials $S \subseteq M$ ($|S| \leq k$) whose coefficients form a basis for the coefficient space of $D(x)$, such that

– for any $m, m' \in S$, $\mathrm{w}(m) \neq \mathrm{w}(m')$ and

– for any monomial $m \in M \setminus S$,

$$\mathrm{coef}_D(m) \in \mathrm{span}\{\mathrm{coef}_D(m') \mid m' \in S,\ \mathrm{w}(m') < \mathrm{w}(m)\}.$$

Gurjar et al. [17, Lemma 5.2] have shown that shifting by a basis isolating weight assignment achieves concentration. We write their lemma here without a proof. For a weight function $\mathrm{w}\colon x \to \{0,1,2,\ldots\}$, let t^{w} denote the tuple $(t^{\mathrm{w}(x_1)}, t^{\mathrm{w}(x_2)}, \ldots, t^{\mathrm{w}(x_n)})$.

Lemma 4.4 (Isolation to concentration)**.** *Let $D(x)$ be a polynomial over a k-dimensional algebra. Let* w *be a basis isolating weight assignment for $D(x)$. Then $D(x+t^{\mathrm{w}})$ is ℓ-concentrated (over $\mathbb{F}(t)$), where $\ell = \lceil \log(k+1) \rceil$.*

We now recall the construction complexity of a basis isolating weight assignment for ROABP from [2]. Here, we present a slightly modified version of their Lemma 8 (without proof), which easily follows from it.

Lemma 4.5. *For any numbers ℓ, n, k and d, we can construct a family $\mathcal{W}$ of $(knd)^{O(\log \ell)}$ integer weight assignments on variables $\{x_1, x_2, \ldots, x_n\}$ with weights bounded by $(knd)^{O(\log \ell)}$ which has the following property: Let $D(x)$ be an individual-degree-d polynomial over $\mathbb{A}_k$ of the form $D_1(x_{i_1})D_2(x_{i_2})\cdots D_\ell(x_{i_\ell})$ for some distinct $i_1, i_2, \ldots, i_\ell \in [n]$. Then one of the weight assignments in $\mathcal{W}$ is basis isolating for $D(x)$.*

Let $\mathcal{W}$ be the family constructed in Lemma 4.5 with $k = w^2$ and $\ell = \lceil \log(w^2+1) \rceil$. From Lemma 4.5 and Lemma 4.4, for any $D(x) = D_1(x_{i_1})D_2(x_{i_2})\cdots D_\ell(x_{i_\ell}) \in \mathbb{F}^{w\times w}[x]$ there exists a weight assignment $\mathrm{w} \in \mathcal{W}$ such that $D(x+t^{\mathrm{w}})$ is ℓ-concentrated (over $\mathbb{F}(t)$). However, we want a single tuple s which works for every $D(x)$. To get a single tuple, we combine the tuples in $\{t^{\mathrm{w}}\}_{\mathrm{w}\in\mathcal{W}}$ using the standard technique of Lagrange interpolation (also used in [14, 17]). Let $\{\alpha_{\mathrm{w}}\}_{\mathrm{w}\in\mathcal{W}}$ be distinct constants. Define

$$s(t_1,t_2) = \sum_{\mathrm{w}\in\mathcal{W}} t_1^{\mathrm{w}} \prod_{\substack{\mathrm{w}'\in\mathcal{W}\\ \mathrm{w}'\neq\mathrm{w}}} \frac{t_2-\alpha_{\mathrm{w}'}}{\alpha_{\mathrm{w}}-\alpha_{\mathrm{w}'}}\,.$$

Note that $s(t_1,\alpha_{\mathrm{w}}) = t_1^{\mathrm{w}}$. The following claim shows that if $D(x+t_1^{\mathrm{w}})$ is ℓ-concentrated for some $\mathrm{w}\in\mathcal{W}$, then $D(x+s(t_1,t_2))$ is also ℓ-concentrated.

Claim 4.6. *For a polynomial $D(x)$ over an algebra and a constant α_{w}, if $D'(x) = D(x+s(t_1,\alpha_{\mathrm{w}}))$ has ℓ-concentration (over $\mathbb{F}(t_1)$) then so does $D''(x) = D(x+s(t_1,t_2))$ (over $\mathbb{F}(t_1,t_2)$).*

Proof. It is easy to see that for any tuple s, coefficients of $D(x+s)$ are linear combinations of coefficients of D and vice versa (over an appropriate field). And since shifting is an invertible, it preserves the rank of all coefficients. That is,

$$\mathrm{rank}_{\mathbb{F}}\{\mathrm{coef}_D(x^a)\}_{x^a\in M} = \mathrm{rank}_{\mathbb{F}(t_1)}\{\mathrm{coef}_{D'}(x^a)\}_{x^a\in M} = \mathrm{rank}_{\mathbb{F}(t_1,t_2)}\{\mathrm{coef}_{D''}(x^a)\}_{x^a\in M}\,.$$

Let this rank be k. Let us represent each coefficient of D as a vector in $\mathbb{F}^k$. Then coefficients of D' and D'' come from $\mathbb{F}[t_1]^k$ and $\mathbb{F}[t_1,t_2]^k$, respectively. Let $M_\ell = \{x^a \in M \mid \mathrm{supp}(a) < \ell\}$. Since D' has ℓ-concentration,

$$\mathrm{rank}_{\mathbb{F}(t_1)}\{\mathrm{coef}_{D'}(x^a) \mid x^a \in M_\ell\} = k\,.$$

Hence, one can form an full rank matrix $L(t_1) \in \mathbb{F}[t_1]^{k\times k}$ which is given by

$$L(t_1) = \begin{pmatrix} \mathrm{coef}_{D'}(x^{a_1}) & \mathrm{coef}_{D'}(x^{a_2}) & \ldots & \mathrm{coef}_{D'}(x^{a_k}) \end{pmatrix}$$

for some $x^{a_1}, x^{a_2}, \ldots, x^{a_k} \in M_\ell$. Define $L'(t_1,t_2) \in \mathbb{F}[t_1,t_2]^{k\times k}$ to be the matrix

$$L'(t_1,t_2) = \begin{pmatrix} \mathrm{coef}_{D''}(x^{a_1}) & \mathrm{coef}_{D''}(x^{a_2}) & \ldots & \mathrm{coef}_{D''}(x^{a_k}) \end{pmatrix}.$$

From the definition of D' and D'', it is clear that $L'(t_1, \alpha_{\mathrm{w}}) = L(t_1)$. Since $\det(L) \neq 0$, we get that $\det(L') \neq 0$. Thus,

$$\mathrm{rank}_{\mathbb{F}(t_1,t_2)} \{\mathrm{coef}_{D''}(x^a) \mid x^a \in M_\ell\} \geq k\,.$$

However, k is the rank of all coefficients of D''. Hence, D'' has ℓ-concentration. □

Now, since $s(t_1,t_2)$ has the desired property from Lemma 4.2, $f(x+s(t_1,t_2))$ is ℓ-concentrated for any polynomial $f(x)$ computed by a width-w ROABP. Recall that $\deg_{t_1}(s)$ is bounded by $(ndw)^{O(\log\log w)}$ from the construction in Lemma 4.5. The same bound also holds on $\deg_{t_2}(s)$ because $|\mathcal{W}| = (ndw)^{O(\log\log w)}$.

Lemma 4.7. *Given n,d,w, one can compute a tuple $s(t_1,t_2) \in \mathbb{F}[t_1,t_2]^n$ of degree $(ndw)^{O(\log\log w)}$ such that for any n-variate, individual-degree-d polynomial $f(x) \in \mathbb{F}[x]$ computed by width-w ROABPs in any order, $f(x+s(t_1,t_2))$ is $O(\log w)$-concentrated.*

As mentioned before, $O(\log w)$-concentration in $f(x+s)$ means that it has an $O(\log w)$-support monomial with a nonzero coefficient. Lemma 4.7 gives a bivariate tuple $s(t_1,t_2)$ for the shift. We argue that one can substitute field values for t_1 and t_2 such that any chosen nonzero coefficient in $f(x+s)$ remains nonzero after the substitution. Note that any coefficient of $f(x+s)$ is a polynomial in t_1 and t_2 with its degree being at most $\deg(f)\cdot\deg(s)$, which is $(ndw)^{O(\log\log w)}$. Thus, by Schwartz-Zippel-DeMillo-Lipton Lemma, substituting $(ndw)^{O(\log\log w)}$ many field values for t_1 and t_2 suffices.

Now, we move on to the second step of Forbes, Shpilka and Saptharishi [14]. They give an $(ndw)^{O(\log\log w)}$-size hitting set for an already $O(\log w)$-concentrated polynomial which is computed by ROABPs in any order. They do this by reducing the PIT question to an $O(\log w)$-variate ROABP [14, Lemma 7.6].

Lemma 4.8 ([14])**.** *Let $f(x) \in \mathbb{F}[x]$ be an n-variate, individual-degree-d polynomial computed by width-w ROABPs in any order. Suppose $f(x)$ has an $(\leq \ell)$-support monomial with a nonzero coefficient. Then, there is a $\mathsf{poly}(n,w,d)$-time computable m-variate map $\phi : x \to \mathbb{F}[y_1,y_2,\dots,y_m]$ such that $f(\phi(x))$ is a nonzero polynomial of degree $< d^2n^4$, where $m = O(\ell^2)$. Moreover, $f(\phi(x))$ is computed by width-w, m-variate ROABPs in any order.*

From the results of [15, 2], we know that an m-variate, width-w ROABP has an $(mdw)^{O(\log m)}$-size hitting set. Combining Lemma 4.7 and Lemma 4.8 with this fact and putting $m = O(\log^2 w)$, we get the following.

Theorem 4.9. *For the class of n-variate, individual-degree-d polynomials computed by width-w ROABPs in any order, one can construct a hitting set of size $(ndw)^{O(\log\log w)}$.*

Concentration in Set-multilinear Circuits. Similar to Theorem 4.9, it would be interesting to achieve the same size hitting set for set-multilinear circuits. Recall from Section 2.2.3 that a polynomial computed by a depth-3 set-multilinear circuit can be written as $(1,1,\dots,1)\cdot D$, where $D = D_1(x_1)D_2(x_2)\cdots D_q(x_q)$ is a product of linear polynomials over a commutative algebra of dimension k. Here the partition $x = x_1 \cup x_2 \cup \cdots \cup x_q$ is unknown. Note that the polynomial D can also be expressed as

$$D = D_{\sigma(1)}(x_{\sigma(1)})D_{\sigma(2)}(x_{\sigma(2)})\cdots D_{\sigma(q)}(x_{\sigma(q)})$$

for any permutation σ on $[q]$. Hence, one can follow the same arguments as for polynomials computed by ROABPs in any order to get concentration in set-multilinear circuits. Hence, we get the following result analogous to Lemma 4.7.

Corollary 4.10. *Given n,k, one can compute an n-tuple $s(t_1,t_2)$ of degree $(nk)^{O(\log\log k)}$ such that for any n-variate polynomial $f(x)$ computed by a depth-3 set-multilinear circuit with top fan-in k, the polynomial $f(x+s(t_1,t_2))$ is $O(\log k)$-concentrated.*

However, it is not clear whether the second step of the hitting-set construction can be done for set-multilinear circuits, i. e., finding a better hitting set by assuming that the polynomial is already concentrated (Lemma 4.8).

5 Discussion

For our first result (Theorem 3.6), there are three directions for improvement. Ideally, one would like to have all three at once.

1. Find a similar hitting set for the unknown-order case. In fact, we conjecture that the same hitting set (Lemma 3.5) works for the unknown-order case as well.
2. Get a hitting set for all fields (including low-characteristic fields). It is easy to construct examples over small characteristic fields where our hitting set does not work.
3. Reduce the hitting-set size to polynomial. To achieve this, it seems one has to do away with the divide and conquer approach.

The map described in Conjecture 3.8 is a possible candidate for a polynomial-size hitting set for ROABPs and proving this conjecture would resolve two of the points above.

As mentioned earlier, we believe the ideas here may help in finding a better PRG for ROBPs. Studying such connections would in particular take us closer towards resolving a major open question of finding an $O(\log n)$-seed-length PRG for constant width ROBPs.

6 Acknowledgements

We thank the anonymous reviewer for suggesting that our techniques in Section 4 might work for a more general class of polynomials, namely polynomials computed by ROABPs in any order. We are thankful to Hervé Fournier, Sumanta Ghosh, Ramprasad Saptharishi for helpful discussions on the same.

References

[1] MANINDRA AGRAWAL: Proving lower bounds via pseudo-random generators. In *Proc. 25th Internat. Conf. on Foundation of Software Tech. and Theoret. Comput. Sci. (FSTTCS'05)*, volume 3821 of *LNCS*, pp. 92–105. Springer, 2005. [doi:10.1007/11590156_6] 2

[2] MANINDRA AGRAWAL, ROHIT GURJAR, ARPITA KORWAR, AND NITIN SAXENA: Hitting-sets for ROABP and sum of set-multilinear circuits. *SIAM J. Comput.*, 44(3):669–697, 2015. [doi:10.1137/140975103, arXiv:1406.7535] 3, 4, 11, 14, 15, 16

[3] MANINDRA AGRAWAL, CHANDAN SAHA, AND NITIN SAXENA: Quasi-polynomial hitting-set for set-depth-Δ formulas. In *Proc. 45th STOC*, pp. 321–330. ACM Press, 2013. [doi:10.1145/2488608.2488649, arXiv:1209.2333] 4, 12, 13

[4] MATTHEW ANDERSON, MICHAEL A. FORBES, RAMPRASAD SAPTHARISHI, AMIR SHPILKA, AND BEN LEE VOLK: Identity testing and lower bounds for read-k oblivious algebraic branching programs. In *Proc. 31st IEEE Conf. on Computational Complexity (CCC'16)*, pp. 30:1–30:25. IEEE Comp. Soc. Press, 2016. [doi:10.4230/LIPIcs.CCC.2016.30, arXiv:1511.07136] 3

[5] SANJEEV ARORA AND BOAZ BARAK: *Computational Complexity: A Modern Approach*. Cambridge Univ. Press, 1st edition, 2009. [doi:10.1017/CBO9780511804090] 3

[6] MICHAEL BEN-OR AND RICHARD CLEVE: Computing algebraic formulas using a constant number of registers. *SIAM J. Comput.*, 21(1):54–58, 1992. Preliminary version in STOC'88. [doi:10.1137/0221006] 2

[7] STUART J. BERKOWITZ: On computing the determinant in small parallel time using a small number of processors. *Inform. Process. Lett.*, 18(3):147–150, 1984. [doi:10.1016/0020-0190(84)90018-8] 2

[8] ANDREJ BOGDANOV, ZEEV DVIR, ELAD VERBIN, AND AMIR YEHUDAYOFF: Pseudorandomness for width-2 branching programs. *Theory of Computing*, 9(7):283–293, 2013. [doi:10.4086/toc.2013.v009a007] 3

[9] MARK BRAVERMAN, ANUP RAO, RAN RAZ, AND AMIR YEHUDAYOFF: Pseudorandom generators for regular branching programs. *SIAM J. Comput.*, 43(3):973–986, 2014. Preliminary version in FOCS'10. [doi:10.1137/120875673] 3

[10] JOSHUA BRODY AND ELAD VERBIN: The coin problem and pseudorandomness for branching programs. In *Proc. 51st FOCS*, pp. 30–39. IEEE Comp. Soc. Press, 2010. [doi:10.1109/FOCS.2010.10] 3

[11] ANINDYA DE: Pseudorandomness for permutation and regular branching programs. In *Proc. 26th IEEE Conf. on Computational Complexity (CCC'11)*, pp. 221–231. IEEE Comp. Soc. Press, 2011. [doi:10.1109/CCC.2011.23] 3

[12] RAFAEL MENDES DE OLIVEIRA, AMIR SHPILKA, AND BEN LEE VOLK: Subexponential size hitting sets for bounded depth multilinear formulas. *Comput. Complexity*, 25(2):455–505, 2016. Preliminary version in CCC'15. [doi:10.1007/s00037-016-0131-1, arXiv:1411.7492] 3

[13] RICHARD A. DEMILLO AND RICHARD J. LIPTON: A probabilistic remark on algebraic program testing. *Inform. Process. Lett.*, 7(4):193 – 195, 1978. [doi:10.1016/0020-0190(78)90067-4] 2

[14] MICHAEL A. FORBES, RAMPRASAD SAPTHARISHI, AND AMIR SHPILKA: Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Proc. 46th STOC*, pp. 867–875. ACM Press, 2014. [doi:10.1145/2591796.2591816] 3, 4, 6, 11, 12, 13, 14, 15, 16

[15] MICHAEL A. FORBES AND AMIR SHPILKA: Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *Proc. 54th FOCS*, pp. 243–252. IEEE Comp. Soc. Press, 2013. [doi:10.1109/FOCS.2013.34, arXiv:1209.2408] 3, 4, 16

[16] ROHIT GURJAR, ARPITA KORWAR, AND NITIN SAXENA: Identity testing for constant-width, and commutative, read-once oblivious ABPs. In *Proc. 31st IEEE Conf. on Computational Complexity (CCC'16)*, pp. 29:1–29:16. IEEE Comp. Soc. Press, 2016. [doi:10.4230/LIPIcs.CCC.2016.29, arXiv:1601.08031] 1

[17] ROHIT GURJAR, ARPITA KORWAR, NITIN SAXENA, AND THOMAS THIERAUF: Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Comput. Complexity*, pp. 1–46, 2016. Preliminary version in CCC'15. [doi:10.1007/s00037-016-0141-z, arXiv:1411.7341] 14, 15

[18] JOOS HEINTZ AND CLAUS P. SCHNORR: Testing polynomials which are easy to compute (extended abstract). In *Proc. 12th STOC*, pp. 262–272. ACM Press, 1980. [doi:10.1145/800141.804674] 2

[19] RUSSELL IMPAGLIAZZO, RAGHU MEKA, AND DAVID ZUCKERMAN: Pseudorandomness from shrinkage. In *Proc. 53rd FOCS*, pp. 111–119. IEEE Comp. Soc. Press, 2012. Preliminary version in ECCC. [doi:10.1109/FOCS.2012.78] 3

[20] RUSSELL IMPAGLIAZZO, NOAM NISAN, AND AVI WIGDERSON: Pseudorandomness for network algorithms. In *Proc. 26th STOC*, pp. 356–364. ACM Press, 1994. [doi:10.1145/195058.195190] 3, 4

[21] VALENTINE KABANETS AND RUSSELL IMPAGLIAZZO: Derandomizing polynomial identity tests means proving circuit lower bounds. *Comput. Complexity*, 13(1):1–46, 2004. Preliminary version in STOC'03. [doi:10.1007/s00037-004-0182-6] 2

[22] NEERAJ KAYAL, VINEET NAIR, AND CHANDAN SAHA: Separation between read-once oblivious algebraic branching programs (ROABPs) and multilinear depth three circuits. In *Proc. 33rd Symp. Theoretical Aspects of Comp. Sci. (STACS'16)*, pp. 46:1–46:15. Schloss Dagstuhl, 2016. Preliminary version in ECCC. [doi:10.4230/LIPIcs.STACS.2016.46] 3

[23] PASCAL KOIRAN: Arithmetic circuits: The chasm at depth four gets wider. *Theoret. Comput. Sci.*, 448:56–65, 2012. [doi:10.1016/j.tcs.2012.03.041, arXiv:1006.4700] 2

[24] MICHAL KOUCKÝ, PRAJAKTA NIMBHORKAR, AND PAVEL PUDLÁK: Pseudorandom generators for group products: extended abstract. In *Proc. 43rd STOC*, pp. 263–272. ACM Press, 2011. [doi:10.1145/1993636.1993672] 3

[25] NOAM NISAN: Lower bounds for non-commutative computation (extended abstract). In *Proc. 23rd STOC*, pp. 410–418. ACM Press, 1991. [doi:10.1145/103418.103462] 3, 4, 7

[26] NOAM NISAN: Pseudorandom generators for space-bounded computation. *Combinatorica*, 12(4):449–461, 1992. Preliminary version in STOC'90. [doi:10.1007/BF01305237] 3

[27] RAN RAZ AND OMER REINGOLD: On recycling the randomness of states in space bounded computation. In *Proc. 31st STOC*, pp. 159–168. ACM Press, 1999. [doi:10.1145/301250.301294] 3

[28] RAN RAZ AND AMIR SHPILKA: Deterministic polynomial identity testing in non-commutative models. *Comput. Complexity*, 14(1):1–19, 2005. Preliminary version in CCC'04. [doi:10.1007/s00037-005-0188-8] 3

[29] JACOB T. SCHWARTZ: Fast probabilistic algorithms for verification of polynomial identities. *J. ACM*, 27(4):701–717, 1980. Preliminary version in EUROSAM'79. [doi:10.1145/322217.322225] 2

[30] THOMAS STEINKE: Pseudorandomness for permutation branching programs without the group theory. *Electron. Colloq. on Comput. Complexity (ECCC)*, 19(83), 2012. Available at ECCC. 3

[31] LESLIE G. VALIANT: Completeness classes in algebra. In *Proc. 11th STOC*, pp. 249–261. ACM Press, 1979. [doi:10.1145/800135.804419] 2

[32] LESLIE G. VALIANT, SVEN SKYUM, STUART J. BERKOWITZ, AND CHARLES RACKOFF: Fast parallel computation of polynomials using few processors. *SIAM J. Comput.*, 12(4):641–644, 1983. Preliminary version in MFCS'81. [doi:10.1137/0212043] 2

[33] RICHARD ZIPPEL: Probabilistic algorithms for sparse polynomials. In *Proc. Internat. Symp. Symbolic and Algebraic Comput. (EUROSAM'79)*, pp. 216–226. Springer, 1979. [doi:10.1007/3-540-09519-5_73] 2

AUTHORS

Rohit Gurjar
Postdoctoral fellow
Tel Aviv University
Tel Aviv, Israel
rohitgurjar0@gmail.com
`http://www.cse.iitk.ac.in/users/rgurjar`

Arpita Korwar
Ph. D. student
Indian Institute of Technology Kanpur
Kanpur, India
arpk@cse.iitk.ac.in
`http://www.cse.iitk.ac.in/users/arpk`

Nitin Saxena
Associate professor
Indian Institute of Technology Kanpur
Kanpur, India
nitin@cse.iitk.ac.in
http://www.cse.iitk.ac.in/users/nitin

ABOUT THE AUTHORS

ROHIT GURJAR is currently a postdoctoral fellow at Tel Aviv University, working with Amir Shpilka. His previous postdoc was at Aalen University (2015-16) with Thomas Thierauf. Before that he was at Indian Institute of Technology Kanpur for 10 long years for his B. Tech.-M. Tech. (2005-10) and Ph. D. (2010-15). He was very fortunate to have Manindra Agrawal and Nitin Saxena as his Ph. D. supervisors. His Ph. D. thesis was chosen for the ACM India Doctoral Dissertation Award, 2017. He is in general interested in theoretical computer science, and in particular in computational complexity and derandomization. Some problems on which he has worked are polynomial identity testing, perfect matching, and matrix completion. He likes hiking, cycling and listening to music.

ARPITA KORWAR is a Ph. D. student at IIT Kanpur advised by Manindra Agrawal and Nitin Saxena. Her work has also been influenced by Somenath Biswas. Her interests are in the theoretical aspects of computer science, especially in using algebraic techniques for computational complexity. Currently she is working on polynomial identity testing and arithmetic lower bounds with Hervé Fournier and Guillaume Malod at the University of Paris Diderot. She visited Thomas Thierauf at the University of Ulm a couple of times during her Ph. D. studies. She likes the outdoors and sports!

NITIN SAXENA received his Ph. D. from the Indian Institute of Technology Kanpur in 2006. He was truly fortunate to have Manindra Agrawal for his supervisor. He also spent stints at Princeton University (2003-04) and at the National University of Singapore (2004-05). He was a postdoc at the Centrum voor Wiskunde en Informatica Amsterdam (2006-08) and a faculty at the Hausdorff Center for Mathematics Bonn (2008-13). Nitin's long-term interests are in algebra-flavored computational complexity problems. He has contributed to primality testing, polynomial identity testing, polynomial independence, polynomial factoring and polynomial equivalence problems. Some of these works have been awarded the Gödel prize, Fulkerson prize, CCC best paper (2006), and ICALP best paper (2011) awards. He enjoys interacting with enthusiastic young researchers. In his spare (& non-spare) time he enjoys listening to music, watching movies, reading non-fiction, swimming and traveling.