POLYNOMIAL IDENTITY TESTING FOR ROABPS AND LOG-VARIATE CIRCUITS

PROJECT NO.: SERB/CS/2023507 *

Ankan Kar†
Project Designation: Senior Student Research Associate
Project Supervisor: **Prof. Nitin Saxena**‡

July 29, 2024

ABSTRACT

This project report focuses into Polynomial Identity Testing (PIT) for Read-Once Algebraic Branching Programs (ROABPs) and Log-Variate circuits. We systematically explore the development of PIT ideas in these models and their efficiency and unique properties. ROABPs, a specialized form of Algebraic Branching Programs (ABPs) where each variable appears only once per path, allow for potentially more efficient PIT algorithms due to their structured nature. This report also investigates a conjecture presented in [3], applying various methods and tried to examine its validity. Additionally, we provide an overview of Log-Variate circuits, highlighting their relevance in the context of PIT. All proofs not included in this paper can be found in the cited references.

1 Introduction

Arithmetic circuits, Algebraic Branching Programs (ABP), and Read-Once Algebraic Branching Programs (ROABP) are essential in algebraic complexity theory, focusing on computing polynomial functions efficiently. Arithmetic circuits, much like their Boolean counterparts, use addition and multiplication gates to perform their calculations. ABPs, utilize layered, directed graphs where each path represents a polynomial's computation, often providing a more compact representation. ROABPs are a more specialized version of ABPs, with the constraint that each variable appears only once along any path, simplifying analysis and potentially leading to more efficient algorithms. One interesting and challenging problem in this field is Polynomial Identity Testing (PIT), which checks whether a given model computes the zero polynomial. Efficient PIT algorithms, are vital for verifying computations in both arithmetic circuits and ABPs. For ROABPs, their unique structure sometimes allows for better PIT algorithms, making it a more understandable field of research compared to general ABPs. Understanding these models and their role in PIT is crucial. We will focus mainly on polynomial identity testing for ROABPs and gave a small overview of few methods that can be further developed to get better PIT algorithms. At last we also look into log-variate circuits along with PIT for it as well.

2 Circuits, ABPs & ROABPs

We will go through mainly different kind of ROABPs as in [3]. Before that we would define some other branching programs and get some general idea about them.

Definition 2.1. *([4]). **A Layered Algebraic branching program (ABP)** is a directed acyclic graph with one source and one sink. The vertices of the graph are partitioned into "levels" numbered from* 0 *to* d*, where edges may only go*

*This project was funded by J.C. Bose Fellowship and conducted at Indian Institute of Technology, Kanpur.
†Computer Science, Chennai Mathematical Institute, ankank.mcs2023@cmi.ac.in
‡CSE, Indian Institute of Technology Kanpur, nitin@cse.iitk.ac.in

from level i to level $i+1$. d is called the degree of the ABP. The source is the only vertex at level 0 and the sink is the only vertex at level d. Each edge is labeled with a homogeneous linear function of $x_1 ... x_n$ (i.e. a function of the form $\sum c_i x_i$). The width of an ABP is the maximum number of nodes in any layer, and the size of the ABP is the number of vertices.

In an ABP, each directed source-sink path computes a polynomial by multiplying the labels on the edges in the order they appear from source to sink. The ABP then computes the sum of all such polynomials, computed as:

$$f(x_1, \ldots, x_n) = (D_1 \cdot D_2 \cdot \ldots \cdot D_q)_{(1,1)}$$

where

$$D_i = \begin{pmatrix} \text{label}(u_1, v_1) & \text{label}(u_1, v_2) & \cdots \\ \text{label}(u_2, v_1) & \text{label}(u_2, v_2) & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}$$

which is equivalent to:

$$f(x_1, \ldots, x_n) = \sum_{\text{all paths from source to sink}} P_{\text{path}}$$

where

$$P_{\text{path}} = \prod_{\text{edges } (u,v) \in \text{path}} \text{label}(u, v)$$

We have the polynomial computed by the ABP same as $U^T(\prod_{i=0}^{q} D_i)V$, where $U, V \in \mathbb{F}^{w \times 1}$ and D_i is a $w \times w$ matrix for $1 \leq i \leq q$ such that; $U(\ell) = W(u, v_{0,\ell})$ for $1 \leq \ell \leq w$; $D_i(k, \ell) = W(v_{i-1,k}, v_{i,\ell})$ for $1 \leq \ell, k \leq w$ and $1 \leq i \leq q$; $V(k) = W(v_{q,k}, t)$ for $1 \leq k \leq w$ and w is the width of the ABP.

Note that we would define the formula complexity of a function f is denoted by $F(f)$, the circuit complexity by $C(f)$, the circuit depth complexity by $D(f)$, the ABP complexity by $B(f)$.

Now we will see the characterization of ABP as in [4]. Let f be a homogeneous function on n variables of degree d. For each $0 < k < d$, we define a real matrix $M_k(f)$ with dimensions as follows: there is a row for each sequence of k variables (called a k-term), and a column for each sequence of $d-k$ variables (called a $d-k$-term, allowing repetitions) out of the n possible variables. The entry at $\langle x_{i_1}, \ldots, x_{i_k} \rangle, \langle x_{j_1}, \ldots, x_{j_{d-k}} \rangle$ is defined to be the real coefficient of the monomial $x_{i_1} \cdots x_{i_k} x_{j_1} \cdots x_{j_{d-k}}$ in f.

$$M_f^{(k)}\left(\langle x_{i_1}, \ldots, x_{i_k} \rangle, \langle x_{j_1}, \ldots, x_{j_{d-k}} \rangle\right) = \textit{coefficient of the monomial } x_{i_1} \cdots x_{i_k} x_{j_1} \cdots x_{j_{d-k}} \textit{ in } f$$

Theorem 2.1. *([4], theorem 1). For any homogeneous function f of degree d,*

$$B(f) = \sum_{k=0}^{d} rank(M_k(f))$$

Our work focuses on a even restricted model, defined as follows.

Definition 2.2. ***A Read-Once Oblivious Algebraic Branching Program (ROABP)*** *in the variable set $x = \{x_1, \ldots, x_n\}$ is an ABP of depth n where each edge between layer $i-1$ and layer i is labeled with a univariate polynomial in $x_{\sigma(i)}$ of degree less than d and some permutation σ of $\{1, ..., n\}$.*

Here we have $q = n$, as variables doesn't repeat. The entries in D_i is from $\mathbb{F}[x_{\sigma(i)}]$. The order $(x_{\sigma(1)}, x_{\sigma(2)}, ..., x_{\sigma(n)})$ is said to be the variable order of the ROABP. In a similar way to ABP we can write

$$f(x_{\sigma(1)}, \ldots, x_{\sigma(n)}) = U^T (D_1 \cdot D_2 \cdot \ldots \cdot D_n) V$$

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

Definition 2.3. *([5]):* ***ROABP***$[\exists](n, d, w)$ *denotes n-variate polynomials of individual degree d that are computable by a width-w ROABP in some order $\sigma \in S_n$.*

Definition 2.4. *([5]): **ROABP**$[\forall](n, d, w)$ denotes n-variate polynomials of individual degree d that are computable by a width-w ROABP in every order.*

Definition 2.5. *(**Commutative ROABP**, [3]): A Read-Once Oblivious Algebraic Branching Program (ROABP) $U^T (\prod_{i=1}^{q} D_i) V$ is called a commutative ROABP if all D_i are polynomials over a commutative subalgebra of the matrix algebra. For instance, this is the case when the coefficients in the polynomials $D_i s$ are diagonal matrices. In a commutative ROABP, the order of the variables becomes irrelevant. Therefore, a polynomial computed by a commutative ROABP can be computed by an ROABP in any variable order.*

Definition 2.6. *(**Depth-3 Set-Multilinear Circuits**, [3]): A depth-3 set-multilinear circuit is a circuit of the form*

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

where $l_{i,j}$ are linear polynomials and $x_1, x_2, \ldots, x_q$ form a partition of x. These circuits are subsumed by ROABPs but are incomparable to commutative ROABPs. The corresponding polynomial over a k-dimensional algebra is

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

where $D_j = (l_{1,j}, l_{2,j}, \ldots, l_{k,j})$ and the algebra product is coordinate-wise. It follows that $C = (1, 1, \ldots, 1) \cdot D$. Polynomials D_j are over a commutative algebra, so techniques for commutative ROABPs apply to set-multilinear circuits.

Definition 2.7. *(**Diagonal ROABPs**, [5]): Diagonal ROABPs are ROABPs where all the $n(d+1)$ coefficient matrices are diagonal matrices. A diagROABP(n, d, w) represents n-variate polynomials of individual degree d that are computable by a width w diagonal ROABP.*

We can also get a hierarchy as given below:

$$\Sigma \bigwedge \Sigma \subsetneq \text{diagROABP} \subseteq \text{commROABP} \subseteq \text{ROABP}[\forall] \subsetneq \text{ROABP}[\exists]$$

3 Construction of ROABP

We can construct ROABP for many kinds of polynomials but here we will only show it for Elementary Symetric Polynomials.

Definition 3.1. *(**Elementary Symmetric Polynomials**): The n-variate elementary symmetric polynomial of degree d, denoted by ESym$_n^d$, is defined as follows:*

$$\mathit{ESym}_n^d(x) := \sum_{\substack{S \subset [n] \\ |S| = d}} \prod_{i \in S} x_i$$

Below in Figure 1 we have a construction of ROABP for ESym$_3^5$ taken from [5] which is provably tight owing to the characterisation result by [4].

Now some constructions of ROABPs as in [5] will be stated and described.

Construction 3.1. *For any $n, d \in \mathbb{N}$ such that $d \leq n$, the n-variate elementary symmetric polynomial of degree d, denoted by ESym$_n^d(x)$, can be expressed as:*

$$\mathit{ESym}_n^d(x) = (M(x_1) M(x_2) \cdots M(x_n))[1, d+1],$$

where $M(x_i)$ is a $(d+1) \times (d+1)$ matrix with:

$$M(x_i)[k, k] = 1 \quad \text{for } 1 \leq k \leq (d+1),$$
$$M(x_i)[k, k+1] = 1 \quad \text{for } 1 \leq k \leq d,$$

and all other entries are zero.

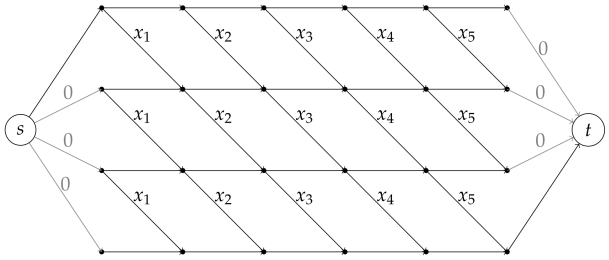

Figure 1: Commutative ROABP for ESym_3^5 (unlabelled edges have the label 1)

The matrix $M(x_i)$ can also be written as $(I + Ax_i)$, where A has 1s on its super-diagonal and zeros elsewhere, and I is the identity matrix. This gives:

$$\mathrm{ESym}_n^d(x) = \left(\prod_{i=1}^{n}(I + Ax_i)\right)[1, d+1] = u^T \left(\prod_{i\in[n]}(I + Ax_i)\right) v,$$

for suitable vectors $u, v \in \mathbb{C}^{d+1}$.

We can now see that:

- All coefficient matrices I and A commute, making it a commutative ROABP.
- $\left(\prod_{i=1}^{n}(I + Ax_i)\right) = \sum_{0\le j\le n} \mathrm{ESym}_n^j A^j = \sum_{0\le j\le d} \mathrm{ESym}_n^j A^j$, since $A^j = 0$ for $j \ge d+1$.
- The $(1, d+1)$-th entry of $(\prod_{i=1}^{n}(I + Ax_i))$ computes the coefficient of A^d, which is ESym_n^d.

Using univariate interpolation (Folklore), leads to a depth-3 multilinear circuit for ESym_n^d with top fan-in $n+1$, as in [6], and provides a nearly-optimal construction for a diagonal ROABP computing ESym_n^d.

Construction 3.2. *For any $n, d \in \mathbb{N}$ and distinct $a_0, a_1, \ldots, a_n \in \mathbb{C}$, there exist constants $\beta_0, \beta_1, \ldots, \beta_n \in \mathbb{C}$ such that*

$$\mathit{ESym}_n^d(x) = \sum_{0\le j\le n} \beta_j (1 + a_j x_1)(1 + a_j x_2)\cdots(1 + a_j x_n).$$

Construction 3.3. *For any $n, d \in \mathbb{N}$, we have:*

$$(x_1 + x_2 + \cdots + x_n)^d = (M(x_1)M(x_2)\cdots M(x_n))[1, d+1],$$

where $M(x_i)$ is a $(d+1)\times(d+1)$ matrix with:

$$M(x_i)[k, k+\ell] = \binom{d-k}{\ell} x_i^\ell \quad \text{for all } 0 \le k \le d \text{ and } 0 \le \ell \le (d-k).$$

We can write $M(x_i)$ as:

$$M(x_i) = I + A_1 x_i + \frac{A_2}{2!}x_i^2 + \cdots + \frac{A_d}{d!}x_i^d,$$

where A is a $(d+1)\times(d+1)$ matrix with $A[i, i+1] = (d-i)$ and other entries are zero.

We can now see that:

- All coefficient matrices I and powers of A commute, making this a commutative ROABP.
- The $(1, d+1)$-th entry of $M(x_1)M(x_2)\cdots M(x_n)$ computes the coefficient of A^d divided by $d!$.

Construction 3.4. *For any $n, d \in \mathbb{N}$ and distinct $a_0, a_1, \ldots, a_{nd} \in \mathbb{C}$, there exist $\beta_0, \beta_1, \ldots, \beta_{nd} \in \mathbb{C}$ such that:*

$$(x_1 + x_2 + \cdots + x_n)^d = \sum_{0\le j\le nd} \beta_j \prod_{i\in[n]} \left(1 + a_j x_i + \frac{a_j^2}{2!}x_i^2 + \frac{a_j^3}{3!}x_i^3 + \cdots + \frac{a_j^d}{d!}x_i^d\right).$$

4 PIT for ROABP

We will first explain the generators and hitting sets for ROABPs. Before that we need to get a small idea on hitting sets.

Definition 4.1. *A polynomial mapping $G = (G_1, \ldots, G_n) : \mathbb{F}^t \to \mathbb{F}^n$ is a **generator** for the circuit class $\mathcal{M}$ if for every nonzero n-variate polynomial $f \in \mathcal{M}$, the composition $f \circ G$ is not identically zero, i.e., $f(G) \not\equiv 0$. The image of the map G is denoted by Im$(G) = G(\mathbb{F}^t)$. Ideally, t should be very small compared to n.*

Definition 4.2. *A set of points H is called a **hitting set** for a class $\mathcal{C}$ of polynomials if for any nonzero polynomial P in $\mathcal{C}$, there exists a point in H where P evaluates to a nonzero value. An $f(n)$-time hitting set means that the hitting set can be generated in time $f(n)$ for input size n.*

First, we will describe a simple randomized algorithm known as the Schwartz-Zippel Algorithm. This algorithm is based on the observation that a nonzero low-degree polynomial does not have many zeros. The Schwartz-Zippel Algorithm is a probabilistic algorithm that leverages the property that a nonzero polynomial of low degree over a finite field has a limited number of zeros.

Lemma 4.1. *([7]): Let $f(x_1, \ldots, x_n)$ be a nonzero polynomial of degree at most r, and let $T \subseteq \mathbb{F}$. If we choose $a = (a_1, \ldots, a_n) \in T^n$ uniformly at random, then*

$$\Pr[f(a) = 0] \leq \frac{r}{|T|}.$$

This lemma suggests a randomized algorithm for Polynomial Identity Testing (PIT): given a polynomial $f(x_1, \ldots, x_n)$ of degree at most r, pick at random $a \in T^n$ and check whether $f(a) = 0$. If $f \not\equiv 0$, the probability of error is at most $\frac{r}{|T|}$, and if $f \equiv 0$, we are always correct. To achieve an error of at most ϵ, we should pick a set T of size $|T| \geq \frac{r}{\epsilon}$. This requires $n \cdot \lceil \log_2 \left(\frac{r}{\epsilon}\right) \rceil$ random bits. Another corollary of Lemma 4.1 is that there exists a small hitting set for all polynomial size arithmetic circuits. The proof follows from a straightforward application of the union bound.

Theorem 4.1. *([7]): For every integers n, r, s and a field $\mathbb{F}$ with $|\mathbb{F}| \geq \max(r^2, s)$, there exists a set $H \subseteq \mathbb{F}^n$ with $|H| =$ poly(r, s) that serves as a hitting set for all circuits of size at most s and degree at most r.*

First we will deal the case of known-order ROABPs and state the results as in [3]. Before that we need to know about basis isolation and some uses of it as described in [2].

Definition 4.3. *A* weight assignment *w of the variables $x_1, x_2, \ldots, x_n$ is a map $w : \{x_1, x_2, \ldots, x_n\} \to \mathbb{N}$. This map extends to monomials $w : M(x) \to \mathbb{N}$ by $w(x^{\mathbf{a}}) := \sum_{i=1}^{n} w(x_i)a_i$ for $x^{\mathbf{a}} = \prod_{i=1}^{n} x_i^{a_i}$. An important tool is the construction of weight assignments that can separate polynomially many monomials, ensuring that distinct monomials receive different weights with high probability.*

Lemma 4.2. *For $n, s, \ell \in \mathbb{N}^+$ and $0 < \epsilon < 1$, there exist weight assignments $w_1, w_2, \ldots, w_N : \{x_1, x_2, \ldots, x_n\} \to [N \log N]$, where $N =$ poly$(n, s, \log \ell, \epsilon^{-1})$, such that for any s monomials $m_1, m_2, \ldots, m_s \in M(x)$ of individual degree less than ℓ, all but at most ϵ-fraction of w_i separate these monomials, i.e., $w_i(m_j) \neq w_i(m_{j'})$ for $j, j' \in [s]$ and $m_j \neq m_{j'}$. These weight assignments can be computed in polynomial time.*

Definition 4.4. *(Basis Isolating Weight Assignment): For a polynomial $f \in A[x]$, a weight assignment $w : \{x_1, x_2, \ldots, x_n\} \to \mathbb{N}$ is called basis isolating for f if there exists a set $S \subseteq M(x)$ of monomials such that their coefficients in f form a basis of span(f), and the following conditions hold:*

1. *$w(m) \neq w(m')$ for distinct $m, m' \in S$, and*
2. *For $m \in M(x) \setminus S$, coef$_f(m) \in$ span$\{$coef$_f(m') : m' \in S, w(m') < w(m)\}$.*

The following lemma states that if w is a basis isolating weight assignment, then the variable substitution map $x_i \mapsto y^{w(x_i)}$ preserves the nonzeroness of polynomials. This makes basis isolating weight assignments a very useful tool for polynomial identity testing (PIT).

Lemma 4.3. *Let $f(x) \in A[x]$, $\beta, \gamma \in \mathbb{F}^r$, and $g(x) = \beta^T f(x)\gamma \in F[x]$. Suppose $w : \{x_1, x_2, \ldots, x_n\} \to \mathbb{N}$ is a basis isolating weight assignment for f. Then $g(x) = 0$ if and only if $g(y^{w(x_1)}, y^{w(x_2)}, \ldots, y^{w(x_n)}) = 0$.*

Now we will get into describing how to get hitting set for known-order bivariate ROABP first and then n-variate ROABP and then to some other kinds of ROABP as well.

4.1 Bivariate ROABP

To construct a hitting set for read-once arithmetic branching programs (ROABPs), we first consider the bivariate case. A bivariate ROABP has the form $U^T D_1(x_1) D_2(x_2) T$, where $U, T \in \mathbb{F}^{w \times 1}$, $D_1 \in \mathbb{F}^{w \times w}[x_1]$, and $D_2 \in \mathbb{F}^{w \times w}[x_2]$. Any bivariate polynomial $f(x_1, x_2)$ computed by a width-w ROABP can be expressed as $f(x_1, x_2) = \sum_{r=1}^{w} g_r(x_1) h_r(x_2)$. To construct a hitting set for such polynomials, we utilize the partial derivative matrix M_f, defined as follows: for $f \in \mathbb{F}[x_1, x_2]$ with individual degree at most d, the partial derivative matrix M_f is a $(d+1) \times (d+1)$ matrix where

$$M_f(i, j) = \text{coeff}(x_1^i x_2^j) \in \mathbb{F}.$$

The rank of M_f gives the minimum width of an ROABP that computes f as in [4].

Lemma 4.4. *([3]): For any polynomial $f(x_1, x_2) = \sum_{r=1}^{w} g_r(x_1) h_r(x_2)$, the rank of the partial derivative matrix M_f satisfies*

$$\textit{rank}(M_f) \leq w.$$

Now using this lemma we get the hitting set as below.

Lemma 4.5. *([3]): Let $\textit{char}(\mathbb{F}) = 0$ or $\textit{char}(\mathbb{F}) > d$. Let $f(x_1, x_2) = \sum_{r=1}^{w} g_r(x_1) h_r(x_2)$ be a nonzero bivariate polynomial over F with individual degree d. Then*

$$f(t^w, t^w + t^{w-1}) \neq 0.$$

4.2 n-variate ROABP

We use the map from Lemma 4.5 to construct a hitting set for general n-variate ROABPs by applying it recursively. We pair consecutive variables (x_{2i-1}, x_{2i}) and apply the map with a new variable t_i, reducing the number of variables by half each time. Repeating this halving process $\log n$ times results in a univariate polynomial with degree at most $w^{\log n}$ times the original degree. WLOG we can assume n is a power of 2.

Lemma 4.6. *([3]): Let $\textit{char}(\mathbb{F}) = 0$ or $\textit{char}(\mathbb{F}) > d$. Let $f(x) = D_1(x_1) D_2(x_2) \cdots D_n(x_n)$ be a nonzero polynomial computed by a width-w and individual degree-d 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 $2 \leq i \leq n-1$. Let the map $\phi : x \to \mathbb{F}[t]$ such that for $1 \leq i \leq n/2$,*

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

Then $f(\phi(x)) \neq 0$. Moreover, the polynomial $f'(t_1, t_2, \ldots, t_{n/2}) := f(\phi(x))$ is computed by a width-w ROABP in the variable order $(t_1, t_2, \ldots, t_{n/2})$.

It is easy to see that when the map ϕ is repeatedly applied in the above lemma $\log n$ times, we get a nonzero univariate polynomial of degree $ndw^{\log n}$.

Lemma 4.7. *([3]): Let $\text{char}(\mathbb{F}) = 0$, or $\text{char}(\mathbb{F}) \geq ndw^{\log n}$. Let $f \in \mathbb{F}[x]$ be a nonzero polynomial, with individual degree d, computed by a width-w ROABP in variable order $(x_0, x_1, \ldots, x_{n-1})$. Let the map $\phi : \{x_0, x_1, \ldots, x_{n-1}\} \to F[t]$ be such that for any index $0 \leq i \leq n-1$,*

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

where $i_{\log n} i_{\log n - 1} \ldots i_1$ is the binary representation of i. Then $f(\phi(x))$ is a nonzero univariate polynomial with degree $ndw^{\log n}$.

Now we can state the following theorem for blackbox PIT.

Theorem 4.2. *([3]): For an n-variate, individual degree d, and width-w ROABP, there is a blackbox PIT with time complexity $O(ndw^{\log n})$, when the variable order is known and the field characteristic is zero or at least $ndw^{\log n}$.*

Corollary 4.1. *([3]): There is a polynomial time blackbox PIT for constant width ROABPs, with known variable order and field characteristic being zero (or polynomially large).*

Now in [3] they gave the below mentioned conjecture and thought of solving it somehow similar to Lemma 4.5.

Conjecture 4.1. *Let $\mathrm{char}(\mathbb{F}) = 0$. 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 $\mathrm{poly}(n, w, d)$.*

This is still open. I tried to gave some counter-examples to it for specific values of r but it does not prove the conjecture to be false at all. This is because we can anytime shift the values a little and get another values where the conjecture holds. So unless we get a counterexample for all $r \in poly(n, w, d)$ we can't say its false. But as of now we don't have much headway for its proof. But in the meantime we can try to prove it for a simpler model as given below, which is also not a easy task yet.

Subpart of Conjecture 4.1. *Let $\mathrm{char}(\mathbb{F}) = 0$. Let $f(x) \in \mathbb{F}[x]$ be an n-variate, degree-d polynomial computed by a width-w ROABP and has the form as $P(x) = \sum_{i=1}^{w} \prod_{j=1}^{n} (1 + a_{i,j}x_j)$ where $a_{ij} \in \mathbb{F}$ (specifically $\mathbb{F}$ can be $\mathbb{R}$ or $\mathbb{C}$). Then $f(t^r, (t+1)^r, \ldots, (t+n-1)^r) \neq 0$ for some r bounded by $\mathrm{poly}(n, w, d)$.*

Till now stated works only on known order bivariate and n-variate ROABPs. Now we will delve into a specific case, i.e., the log-variate circuits.

5 PIT for Log-Variate Circuits

Simply speaking log-variate circuits means the number of variables involved is at most logarithmic with respect to the circuit size. We would state the results as in [1].

Studying PIT for log-variate models is crucial, as achieving $\mathrm{poly}(s)$-time blackbox PIT for size-s, degree-s, and $\log^{\circ c} s$-variate circuits ($\log^{\circ c}$ means composition of log c-times) can solve PIT entirely. Additionally, $\mathrm{poly}(s)$-time blackbox PIT for size-s and $\log^{*} s$-variate $\Sigma \wedge \Sigma\Pi$ circuits could partially solve PIT and prove either $E \not\subseteq \#P/\mathrm{poly}$ or VP $\neq$ VNP.

Here the measure for rank concentration will be the cone size. Using this cone-size we will look at a blackbox PIT algorithm for circuit models with 'low' dimensional partial derivative space.

For a circuit C, we denote its size by $|C|$. For a monomial m, $\mathrm{coef}_m(C)$ represents the coefficient of m in the polynomial computed by C. Additionally, we use C to refer to the polynomial itself computed by the circuit.

Definition 5.1. *(Cone of a monomial):A monomial x^e is considered a sub-monomial of x^f if $e \leq f$ (coordinate-wise). It is referred to as a proper sub-monomial of x^f if $e \leq f$ and $e \neq f$.*

For a monomial x^e, the cone of x^e is defined as the set of all its sub-monomials. The size of this set, known as the cone-size of x^e, is given by $\prod(e+1) := \prod_{i \in [n]} (e_i + 1)$, where $e = (e_1, \ldots, e_n)$. A set S of monomials is considered cone-closed if it contains every sub-monomial of each monomial in S.

Lemma 5.1. *(Coefficient Extraction, [1]): Let C be a blackbox circuit that computes an n-variate polynomial of degree d over a field of size greater than d. For any monomial $m = \prod_{i \in [n]} x_i^{e_i}$, there exists an algorithm with a runtime of $poly(|C|, d, cs(m))$ to compute the coefficient of m in C, where $cs(m)$ denotes the cone-size of m.*

Now we need to check how many low-cone monomials there can be. In the log-variate case there are quasi-polynomially many.

Lemma 5.2. *(Counting Low Cones, [1]): The number of n-variate monomials with a cone-size of at most k is $O(rk^2)$, where $r := \left(\frac{3n}{\log k}\right)^{\log k}$.*

Now using the last two lemmas we can state the following theorem.

Theorem 5.1. *([1]): Let $\mathbb{F}$ be a field with characteristic 0 or greater than d. Let $\mathbb{P}$ be a set of n-variate polynomials of degree d, over $\mathbb{F}$, computed by circuits of bitsize s. Suppose that for every $P \in \mathbb{P}$, the dimension of the partial derivative space of P is at most k. Then, blackbox PIT for $\mathbb{P}$ can be solved in time $(sdk)^{O(1)} \cdot \left(\frac{3n}{\log k}\right)^{O(\log k)}$.*

We can see that When $n = O(\log k) = O(\log sd)$, the bound becomes polynomial, yielding a polynomial-time blackbox PIT algorithm for log-variate circuits that have a low-dimensional partial derivative space. Then we get the immediate following corollary.

Corollary 5.1. *([1]): Let $\mathbb{F}$ be a field with characteristic 0 or greater than d. Consider a set $\mathbb{P}$ of n-variate, degree-d polynomials over F, computable by circuits with bit-size s, where $n = O(\log sd)$. If the partial derivative space of each $P \in \mathbb{P}$ has poly(sd) dimension, then blackbox PIT for $\mathbb{P}$ can be solved in poly(sd) time.*

Now we can look into depth-3 diagonal circuits $\Sigma \wedge \Sigma$.

Definition 5.2. ***(Depth-3 diagonal circuit and its rank, [1]).*** *A depth-3 diagonal circuit is of the form $\Sigma \wedge \Sigma$ (sum-power-sum). It computes a polynomial expressed as $C(x) = \sum_{i\in[k]} c_i \ell_i^{d_i}$, where ℓ_i are linear polynomials over F and $c_i \in F$.*

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

Lemma 5.3. *(Variable reduction, [1]): Let $P(x)$ be an n-variate d-degree polynomial computed by a size-s depth-3 diagonal circuit over some sufficiently large field $\mathbb{F}$. Then, there exists a poly(nds)-time computable nonzeroness preserving variable reduction map which converts P to another rk(P)-variate degree-d polynomial computed by a poly(s)-size depth-3 diagonal circuit.*

Then we get this following theorem.

Theorem 5.2. *(Log-rank $\Sigma \wedge \Sigma$, [1]): Let $\mathbb{F}$ be a field of characteristic 0 or $> d$. Let $\mathbb{P}$ be the set of n-variate d-degree polynomials P, computable by depth-3 diagonal circuits of bitsize s, with rk$(P) = O(\log sd)$. Then, blackbox PIT for $\mathbb{P}$ can be solved in poly(sd)-time.*

There are also various algorithms to get cone closed basis but we are not getting into it. But if needed one can look at [1]. Finally we get the following theorem.

Theorem 5.3. *([1]): Let $f(x) \in \mathbb{F}[x]^k$ be an n-variate d-degree polynomial over $\mathbb{F}^k$ and $\text{char}\,\mathbb{F} = 0$ or $> d$. Let $w = (w_1, \ldots, w_n) \in \mathbb{N}^n$ be a basis isolating weight assignment of $f(x)$. Then, $f(x+t^w) := f(x_1+t^{w_1}, \ldots, x_n+t^{w_n})$ has a cone-closed basis over $F(t)$.*

6 Acknowledgement

I would like to express my sincere gratitude to my project supervisor, Prof. Nitin Saxena, for his guidance and detailed explanations of the underlying theory. My heartfelt thanks also goes to Prateek Dwivedi for his guidance and for helping me understand the research flow in this field. Additionally, I am deeply thankful to my friends who supported me during this project.

References

[1] Michael A. Forbes, Sumanta Ghosh, and Nitin Saxena. "Towards Blackbox Identity Testing of Log-Variate Circuits". In: *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*. Ed. by Ioannis Chatzigiannakis et al. Vol. 107. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018, 54:1–54:16. ISBN: 978-3-95977-076-7. DOI: 10.4230/LIPIcs.ICALP.2018.54. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.ICALP.2018.54.

[2] Zeyu Guo and Rohit Gurjar. "Improved Explicit Hitting-Sets for ROABPs". In: *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX/RANDOM 2020)*. Ed. by Jarosław Byrka and Raghu Meka. Vol. 176. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2020, 4:1–4:16. ISBN: 978-3-95977-164-1. DOI: 10.4230/LIPIcs.APPROX/RANDOM.2020.4. URL: https://drops.dagstuhl.de/entities/document/10.4230/LIPIcs.APPROX/RANDOM.2020.4.

[3] Rohit Gurjar, Arpita Korwar, and Nitin Saxena. "Identity Testing for Constant-Width, and Any-Order, Read-Once Oblivious Arithmetic Branching Programs". In: *Theory of Computing* 13.2 (2017), pp. 1–21. DOI: 10.4086/toc.2017.v013a002. URL: https://theoryofcomputing.org/articles/v013a002.

[4] Noam Nisan. "Lower bounds for non-commutative computation". In: *Proceedings of the Twenty-Third Annual ACM Symposium on Theory of Computing*. STOC '91. New Orleans, Louisiana, USA: Association for Computing Machinery, 1991, pp. 410–418. ISBN: 0897913973. DOI: 10.1145/103418.103462. URL: https://doi.org/10.1145/103418.103462.

[5] C. Ramya and Anamay Tengse. "On Finer Separations between Subclasses of Read-once Oblivious ABPs". In: *Electron. Colloquium Comput. Complex.* TR22 (2022). URL: https://api.semanticscholar.org/CorpusID:246015432.

[6] Amir Shpilka and Avi Wigderson. "Depth-3 arithmetic circuits over fields of characteristic zero". In: *Computational Complexity* 10.1 (2001), pp. 1–27. ISSN: 1420-8954. DOI: 10.1007/PL00001609. URL: https://doi.org/10.1007/PL00001609.

[7] Amir Shpilka and Amir Yehudayoff. *Arithmetic Circuits: A Survey of Recent Results and Open Questions*. 2010. DOI: 10.1561/0400000039.