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

Pranav Bisht
Joint work with: Nitin Saxena

May 2, 2020

Department of Computer Science and Engg.
IIT Kanpur

Main Results (Informal)

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

Main Results (Informal)

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

Blackbox PIT for sum of unbounded-many, log-variate constant-width ROABPs is in poly-time, if each ROABP computes a homogeneous polynomial.

Content

Introduction

Polynomial Identity Testing (PIT)

- Simply test whether a given multivariate polynomial is identically zero or not.
- Identically zero means all coefficients in fully expanded form are 0.
- Input representation: Algebraic circuits, Algebraic Branching Programs (ABPs), Read-once Oblivious ABPs (ROABPs).

Polynomial Identity Testing

Two types of PIT algorithms:

1. Whitebox PIT: Have access to internal nodes of the circuit/ABP/ROABP.
2. Blackbox PIT: Can only evaluate circuit/ABP/ROABP on field points.

Blackbox PIT

Definition 1 (Blackbox PIT)

Let $\mathcal{P}$ be a set of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ of degree d. A blackbox PIT algorithm for $\mathcal{P}$ outputs a set of points $\mathcal{H} \subseteq \mathbb{F}^n$ such that if $f \in \mathcal{P}$ computes a non-zero polynomial, then $\exists \overline{\alpha} \in \mathcal{H}$ such that $f(\overline{\alpha}) \neq 0$.

Blackbox PIT

Definition 1 (Blackbox PIT)

Let $\mathcal{P}$ be a set of polynomials in $\mathbb{F}[x_1, \ldots, x_n]$ of degree d. A blackbox PIT algorithm for $\mathcal{P}$ outputs a set of points $\mathcal{H} \subseteq \mathbb{F}^n$ such that if $f \in \mathcal{P}$ computes a non-zero polynomial, then $\exists \overline{\alpha} \in \mathcal{H}$ such that $f(\overline{\alpha}) \neq 0$.

- Example: Size $d + 1$ hitting set for univariates.

Randomized Algorithm

Lemma 1 (PIT Lemma [Sch80] [Zip79] [DL77])

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ *be a non-zero polynomial of total degree* d*. Let* S *be any finite subset of* $\mathbb{F}$*, of size* $> d$ *and let* $\alpha_1, \alpha_2, \ldots, \alpha_n$ *be elements selected randomly from* S*. Then*

$$Pr_{\alpha_1,\ldots,\alpha_n \in_r S}[f(\alpha_1, \ldots, \alpha_n) = 0] \leq \frac{d}{|S|}$$

Randomized Algorithm

Lemma 1 (PIT Lemma [Sch80] [Zip79] [DL77])

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ be a non-zero polynomial of total degree d. Let S be any finite subset of $\mathbb{F}$, of size $> d$ and let $\alpha_1, \alpha_2, \ldots, \alpha_n$ be elements selected randomly from S. Then

$$Pr_{\alpha_1,\ldots,\alpha_n \in_r S}[f(\alpha_1, \ldots, \alpha_n) = 0] \leq \frac{d}{|S|}$$

- This gives poly-time randomized algorithm for PIT.
- Trivial derandomization: $(d+1)^n$ time deterministic algorithm for PIT.

Connections of PIT

- Lower Bounds.
- Primality Testing.
- Deciding existence of perfect matching in a graph.
- IP $=$ PSPACE.
- Polynomial Factoring.
- Polynomial Equivalence.

Standard example: Sparse PIT

- General PIT seems difficult as of now. Can we solve restricted cases?
- For eg, do we have poly-time blackbox PIT for the class of sparse polynomials?

Standard example: Sparse PIT

- General PIT seems difficult as of now. Can we solve restricted cases?
- For eg, do we have poly-time blackbox PIT for the class of sparse polynomials?
- Kronecker map: $x_i \to y^{d^{i-1}}$.
- [AB03] There exists $1 \leq r \leq \text{poly}(mn \log d)$ such that, $f \not\equiv 0 \Leftrightarrow f(y, y^{d \pmod r}, y^{d^2 \pmod r}, \ldots, y^{d^{n-1} \pmod r}) \not\equiv 0$.
- Sparse PIT map Φ preserves non-zeroness of a m-sparse, n variate, degree d polynomial.

Models of Interest

Algebraic Circuit

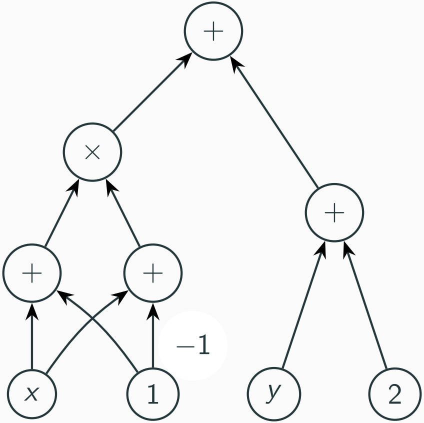


Figure 1: A circuit computing the polynomial $x^2 + y + 1$.

Algebraic Branching Program (ABP)

- Layered DAG with unique source s and sink t.
- Edges are labeled with linear polynomials.
- $C(\bar{x}) = \sum_{\text{path } p: s \rightsquigarrow t} w(p)$, where $w(p) = \prod_{e \in p} w(e)$.
- Size parameters: width, length (degree), number of variables.

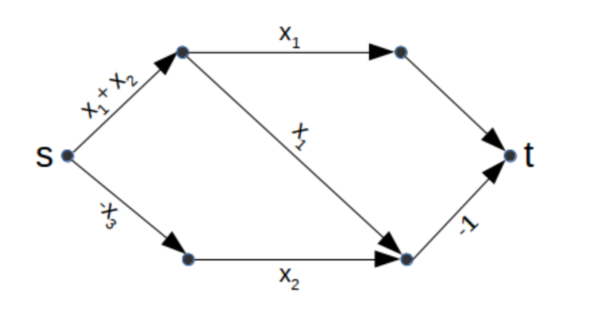
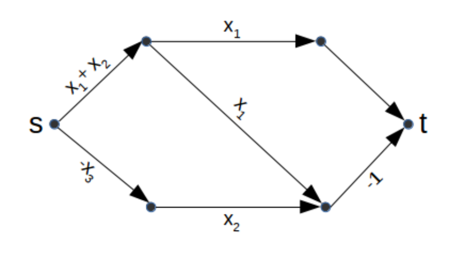


Figure 2: ABP of width 2, depth 3 computing $f = x_2x_3$.

Iterated Matrix Multiplication



$$\begin{bmatrix} x_1 + x_2 & -x_3 \end{bmatrix} \begin{bmatrix} x_1 & x_1 \\ 0 & x_2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 + x_2 & 0 \\ 0 & x_3 \end{bmatrix} \begin{bmatrix} x_1 & x_1 \\ 0 & x_2 \end{bmatrix} \begin{bmatrix} 1 \\ -1 \end{bmatrix}$$

$C(\bar{x}) = U^\top (\prod_{i=1}^{d} D_i) V$, where $U, V \in \mathbb{F}^{w \times 1}$ and $D_i \in \mathbb{F}[\bar{x}]^{w \times w}$.

Read-once oblivious ABP (ROABP)

- Each variable appears in a single layer only (read-once).
- Edge weights are univariate polynomials.
- $C(\bar{x}) = U^\top \cdot D_1(x_{\pi(1)}) D_2(x_{\pi(2)}) \cdots D_n(x_{\pi(n)}) \cdot V$.
- Size parameters: width, length (number of variables), degree.
- Variable order matters.

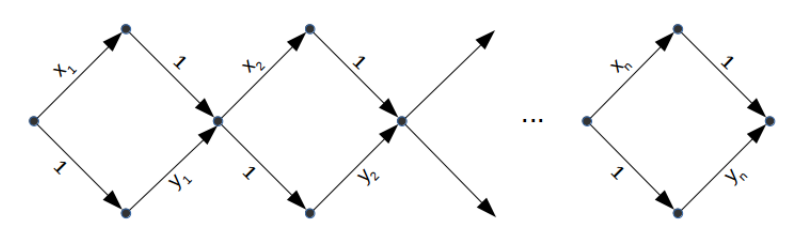


Figure 3: ROABP computing $f = (x_1 + y_1)(x_2 + y_2) \cdots (x_n + y_n)$.

Comparative example

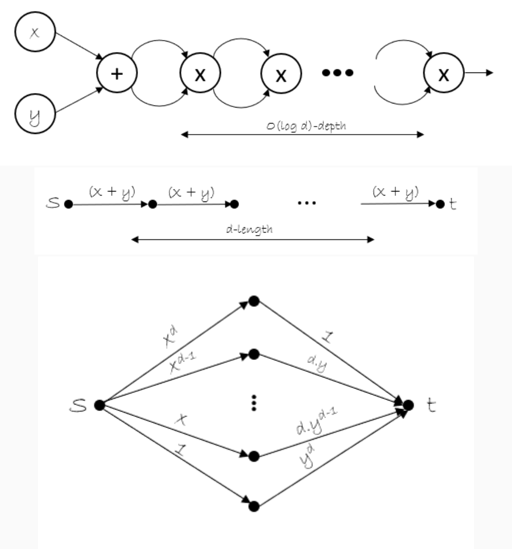


Figure 4: $(x + y)^d$ as computed by a circuit, ABP and ROABP resp.

Nisan's characterization

- ROABP is a complete model.
- [Nis91] gave a rank based measure $M(f)$ such that a polynomial f is computed by w-width ROABP, if and only if $M(f) \leq w$.

Nisan's characterization

- ROABP is a complete model.
- [Nis91] gave a rank based measure $M(f)$ such that a polynomial f is computed by w-width ROABP, if and only if $M(f) \leq w$.
- Both $det_{n\times n}$ and $per_{n\times n}$ have ROABPs of size $2^{\theta(n)}$ in any variable order [Nis91].
- While $det_{n\times n}$ has an ABP of size $O(n^3)$ [MV97], $per_{n\times n}$ is believed to be hard for ABPs.

Motivation

Motivation: ABPs

- PIT for ABPs is open (even lower bounds).
- [AGS19] show even PIT for log-variate width-2 ABPs will almost solve general PIT.

Motivation: ABPs

- PIT for ABPs is open (even lower bounds).
- [AGS19] show even PIT for log-variate width-2 ABPs will almost solve general PIT.
- Solving PIT for ROABPs is the natural first step since exponential lower bounds are already known but poly-time blackbox PIT is still open.

Motivation: Log-variate

- Bootstrapping results point that solving PIT for $\log^{\circ c} s$-variate circuits will solve general PIT.

Motivation: Log-variate

- Bootstrapping results point that solving PIT for $\log^{\circ c} s$-variate circuits will solve general PIT.
- [FGS18] give poly-time blackbox PIT for log-variate $\sum \bigwedge \sum$ (Diagonal depth-3) but general n-variate is still open.

Motivation: Log-variate

- Bootstrapping results point that solving PIT for $\log^{\circ c} s$-variate circuits will solve general PIT.
- [FGS18] give poly-time blackbox PIT for log-variate $\sum\bigwedge\sum$ (Diagonal depth-3) but general n-variate is still open.
- PIT for log-variate commutative ROABP $\Rightarrow$ PIT for general n-variate $\sum\bigwedge\sum$. [Sax08, FSS14]

Motivation: Log-variate

- Bootstrapping results point that solving PIT for $\log^{\circ c} s$-variate circuits will solve general PIT.
- [FGS18] give poly-time blackbox PIT for log-variate $\sum \bigwedge \sum$ (Diagonal depth-3) but general n-variate is still open.
- PIT for log-variate commutative ROABP $\Rightarrow$ PIT for general n-variate $\sum \bigwedge \sum$. [Sax08, FSS14]
- Note that we have quasi-poly time ($s^{O(\log s)}$) blackbox PIT by brute-force in log-variate regime. We need strictly poly-time.

Motivation: Constant-width

- [GKS17] solve PIT for constant-width ROABP but only for known variable order and for fields of zero or large characteristic.

Motivation: Constant-width

- [GKS17] solve PIT for constant-width ROABP but only for known variable order and for fields of zero or large characteristic.
- Sum of constant-width ROABPs >> Single constant-width ROABP.
- [KNS16] construct explicit polynomial computable by just sum of two width-3 ROABPs but requires $2^{\Omega(n)}$ width to compute using a single ROABP.

Motivation: Constant-width

- [GKS17] solve PIT for constant-width ROABP but only for known variable order and for fields of zero or large characteristic.
- Sum of constant-width ROABPs $>>$ Single constant-width ROABP.
- [KNS16] construct explicit polynomial computable by just sum of two width-3 ROABPs but requires $2^{\Omega(n)}$ width to compute using a single ROABP.
- Hence, before this work, there was no poly-time PIT known even for sum of two log-variate constant-width ROABPs.

Motivation: Connections with other OPEN models

- PIT for log-variate commutative ROABPs $\Rightarrow$ PIT for $\sum\bigwedge\sum$ model. [Sax08, FSS14] (We solve for log-variate constant-width ROABPs)

Motivation: Connections with other OPEN models

- PIT for log-variate commutative ROABPs $\Rightarrow$ PIT for $\sum\bigwedge\sum$ model. [Sax08, FSS14] (We solve for log-variate constant-width ROABPs)
- PIT for sum of (unbounded-many) log-variate constant width ROABPs $\Rightarrow$ PIT for $\sum\bigwedge\sum$. (We solve it when each ROABP is restricted to compute a homogeneous polynomial)

Motivation: Connections with other OPEN models

- PIT for log-variate commutative ROABPs $\Rightarrow$ PIT for $\sum \bigwedge \sum$ model. [Sax08, FSS14] (We solve for log-variate constant-width ROABPs)
- PIT for sum of (unbounded-many) log-variate constant width ROABPs $\Rightarrow$ PIT for $\sum \bigwedge \sum$. (We solve it when each ROABP is restricted to compute a homogeneous polynomial)
- PIT for sum of (unbounded-many) ROABPs $\Rightarrow$ PIT for multilinear depth-3 ($\sum \prod \sum$).

Previous results

- [RS05]: Poly-time whitebox PIT for ROABPs.
- [FS13]: Quasi-poly time blackbox PIT for ROABPs of known var. order.
- [AGKS15]: Quasi-poly time blackbox PIT for ROABPs.
- [GKST16]: Quasi-poly time blackbox PIT for sum of constantly-many ROABPs.
- [GKS17]: Poly-time blackbox PIT for constant-width ROABPs of known var. order (over fields of zero or large characteristic).

Main Results

Our Results

Theorem 1 (Sum of ROABPs)

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

Our Results

Theorem 1 (Sum of ROABPs)

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

Poly(s) time for $r, c = O(1)$ and $n = O(\log s)$.

Both brute-force (d^n) and [GKST16] yield only $s^{O(\log s)}$ time blackbox PIT.

Our Results

Theorem 2 (Sum of Homog. ROABPs)

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

Our Results

Theorem 2 (Sum of Homog. ROABPs)

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

Poly(s) time for $r = O(1)$ and $n = O(\log s)$. (arbitrary c)

Final polynomial may be inhomogeneous.

New techniques:

- Syntactic homogeneity in same width.
- Bypassing log-support concentration for sum of ROABPs.

PIT: Sum of Homog. ROABPs

Definition 2 (Syntactically Homogeneous ROABP)

For any two nodes (u, v) in the ROABP, the polynomial computed from $u \rightsquigarrow v$ is homogeneous.

PIT: Sum of Homog. ROABPs

Definition 2 (Syntactically Homogeneous ROABP)

For any two nodes (u, v) in the ROABP, the polynomial computed from $u \rightsquigarrow v$ is homogeneous.

Theorem 3 (Structure Theorem)

If f is a homogeneous polynomial computed by an ROABP of width w, then it is also computed by a syntactically homogeneous ROABP of width $r \leq w$.

Proved using Nisan's characterization, variable disjointedness and degree argument.

PIT: Sum of Homog. ROABPs

Example: ROABP and syntactically homogeneous ROABP resp. computing a homogeneous polynomial.

$$(x+y)^2 = \begin{bmatrix} \frac{35}{12} \\ \frac{-26}{3}(1+x+\frac{1}{2}x^2) \\ \frac{19}{2}(1+2x+2x^2) \\ \frac{-14}{3}(1+3x+\frac{9}{2}x^2) \\ \frac{11}{12}(1+4x+8x^2) \end{bmatrix}^\mathsf{T} \cdot \begin{bmatrix} 1 \\ 1+y+\frac{1}{2}y^2 \\ 1+2y+2y^2 \\ 1+3y+\frac{9}{2}y^2 \\ 1+4y+8y^2 \end{bmatrix}$$

$$(x+y)^2 = \begin{bmatrix} x^2 & 2x & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ y \\ y^2 \end{bmatrix}$$

PIT: Sum of Homog. ROABPs

- Syntactic Homogeneity $\Rightarrow$ Monomial edge weights.

PIT: Sum of Homog. ROABPs

- Syntactic Homogeneity $\Rightarrow$ Monomial edge weights.
- Monomial edge weights $\Rightarrow$ Sparsity(f) $\leq r^n$.

PIT: Sum of Homog. ROABPs

- Syntactic Homogeneity $\Rightarrow$ Monomial edge weights.
- Monomial edge weights $\Rightarrow$ Sparsity(f) $\leq r^n$.
- Thus, sparsity($f_1 + f_2 + \ldots + f_c$) $\leq cr^n$.

PIT: Sum of Homog. ROABPs

- Syntactic Homogeneity $\Rightarrow$ Monomial edge weights.
- Monomial edge weights $\Rightarrow$ Sparsity(f) $\leq r^n$.
- Thus, sparsity($f_1 + f_2 + \ldots + f_c$) $\leq cr^n$.
- Apply sparse PIT map.

PIT: Sum of Homog. ROABPs

- Syntactic Homogeneity $\Rightarrow$ Monomial edge weights.
- Monomial edge weights $\Rightarrow$ Sparsity(f) $\leq r^n$.
- Thus, sparsity($f_1 + f_2 + \ldots + f_c$) $\leq cr^n$.
- Apply sparse PIT map.
- Gives poly-time blackbox PIT for arbitrary sum of homog. ROABPs (constant-width, log-variate).

PIT: Single ROABP

Lemma 2

Let f be a degree-d polynomial computed by an ROABP of width w. Then, $f^{[d]}$ is also computed by an ROABP of width $r \leq w$.

$f^{[d]}$ denotes the degree-d homogeneous component of f.

PIT: Single ROABP

Lemma 2

Let f be a degree-d polynomial computed by an ROABP of width w. Then, $f^{[d]}$ is also computed by an ROABP of width $r \leq w$.

$f^{[d]}$ denotes the degree-d homogeneous component of f.

- By structure theorem, sparsity($f^{[d]}$) $\leq r^n$.
- Apply sparse PIT map on $f^{[d]}$.
- Gives poly-time blackbox PIT for constant-width, log-variate ROABP (possibly inhomogeneous).

PIT: Sum of ROABPs

Consider sum of two: $A + B$.

PIT: Sum of ROABPs

Consider sum of two: $A + B$.

- Case 1 (Easy): Both A and B have width r in same var. order.

PIT: Sum of ROABPs

Consider sum of two: $A + B$.

- Case 1 (Easy): Both A and B have width r in same var. order.
- Case 2: A and B have width r in different var. orders.

PIT: Sum of ROABPs

Consider sum of two: $A + B$.

- Case 1 (Easy): Both A and B have width r in same var. order.
- Case 2: A and B have width r in different var. orders.
  - [GKST16] whitebox idea: Iteratively build ROABP of B in var. order of A.

PIT: Sum of ROABPs

Consider sum of two: $A + B$.

- Case 1 (Easy): Both A and B have width r in same var. order.
- Case 2: A and B have width r in different var. orders.
  - [GKST16] whitebox idea: Iteratively build ROABP of B in var. order of A.
  - Case 2.a (Easy): B also has width r in var. order of A.

PIT: Sum of ROABPs

Consider sum of two: $A + B$.

- Case 1 (Easy): Both A and B have width r in same var. order.
- Case 2: A and B have width r in different var. orders.
  - [GKST16] whitebox idea: Iteratively build ROABP of B in var. order of A.
  - Case 2.a (Easy): B also has width r in var. order of A.
  - Case 2.b: B does not have width r in var. order of A.

PIT: Sum of ROABPs

Consider sum of two: $A + B$.

- Case 1 (Easy): Both A and B have width r in same var. order.
- Case 2: A and B have width r in different var. orders.
    - [GKST16] whitebox idea: Iteratively build ROABP of B in var. order of A.
    - Case 2.a (Easy): B also has width r in var. order of A.
    - Case 2.b: B does not have width r in var. order of A.

For Case 2.b, find the first layer where B deviates from A. Deviation is certified by B not satisfying the *dependency equations* of A.

PIT: Sum of ROABPs

Consider sum of two: $A + B$.

- Case 1 (Easy): Both A and B have width r in same var. order.
- Case 2: A and B have width r in different var. orders.
  - [GKST16] whitebox idea: Iteratively build ROABP of B in var. order of A.
  - Case 2.a (Easy): B also has width r in var. order of A.
  - Case 2.b: B does not have width r in var. order of A.

For Case 2.b, find the first layer where B deviates from A. Deviation is certified by B not satisfying the *dependency equations* of A.

This non-zeroness *certificate* can be found in poly-time in *whitebox* setting.

PIT: Sum of ROABPs

Var. orders are *unknown* in blackbox setting.

PIT: Sum of ROABPs

Var. orders are *unknown* in blackbox setting.

[GKST16] take shift route which takes quasi-poly time.

PIT: Sum of ROABPs

Var. orders are *unknown* in blackbox setting.

[GKST16] take shift route which takes quasi-poly time.

Our idea: Search for certificate in 2^n time.

PIT: Sum of ROABPs

Var. orders are *unknown* in blackbox setting.

[GKST16] take shift route which takes quasi-poly time.

Our idea: Search for certificate in 2^n time.

- Suppose B deviates at k^{th} layer.
- Go over all k-length prefixes. Can take $n!$ time which is super-poly time.

PIT: Sum of ROABPs

Var. orders are *unknown* in blackbox setting.

[GKST16] take shift route which takes quasi-poly time.

Our idea: Search for certificate in 2^n time.

- Suppose B deviates at k^{th} layer.
- Go over all k-length prefixes. Can take $n!$ time which is super-poly time.
- Correction: Go over all k-sized subsets. Takes $\leq 2^n$ time.
- It works since we apply PIT map of single ROABP on prefix, which is independent of var. order of prefix.

PIT: Sum of ROABPs

Suppose variable order of A is

$x_{\pi(1)} < x_{\pi(2)} < \ldots < x_{\pi(n)} = y_1 < y_2 < \ldots < y_n$.

PIT: Sum of ROABPs

Suppose variable order of A is

$x_{\pi(1)} < x_{\pi(2)} < \ldots < x_{\pi(n)} = y_1 < y_2 < \ldots < y_n$.

- Prefix PIT map $\Phi : \mathbb{F}[y_1, \ldots y_{k-1}] \to \mathbb{F}[t_1]$. (Guess the prefix variables)

PIT: Sum of ROABPs

Suppose variable order of A is
$x_{\pi(1)} < x_{\pi(2)} < \ldots < x_{\pi(n)} = y_1 < y_2 < \ldots < y_n$.

- Prefix PIT map $\Phi : \mathbb{F}[y_1, \ldots y_{k-1}] \to \mathbb{F}[t_1]$. (Guess the prefix variables)
- Deviation layer variable $y_k \to y_k$. (Guess correct y_k)

PIT: Sum of ROABPs

Suppose variable order of A is

$x_{\pi(1)} < x_{\pi(2)} < \ldots < x_{\pi(n)} = y_1 < y_2 < \ldots < y_n$.

- Prefix PIT map $\Phi : \mathbb{F}[y_1, \ldots y_{k-1}] \to \mathbb{F}[t_1]$. (Guess the prefix variables)
- Deviation layer variable $y_k \to y_k$. (Guess correct y_k)
- Suffix PIT map $\Psi : \mathbb{F}[y_{k+1}, \ldots y_n] \to \mathbb{F}[t_2]$.
- From n-variate to tri-variate.

PIT: Sum of ROABPs

Suppose variable order of A is

$x_{\pi(1)} < x_{\pi(2)} < \ldots < x_{\pi(n)} = y_1 < y_2 < \ldots < y_n$.

- Prefix PIT map $\Phi : \mathbb{F}[y_1, \ldots y_{k-1}] \to \mathbb{F}[t_1]$. (Guess the prefix variables)
- Deviation layer variable $y_k \to y_k$. (Guess correct y_k)
- Suffix PIT map $\Psi : \mathbb{F}[y_{k+1}, \ldots y_n] \to \mathbb{F}[t_2]$.
- From n-variate to tri-variate.

Using PIT maps that work for a single ROABP of width $O(r^3)$ suffice to preserve the certificate under variable reduction.

PIT: Sum of ROABPs

This idea can be extended recursively to sum of c ROABPs. Set $A = A_1$ and $B = A_2 + \ldots + A_c$.

PIT: Sum of ROABPs

This idea can be extended recursively to sum of c ROABPs. Set $A = A_1$ and $B = A_2 + \ldots + A_c$.

Our algorithm is not limited to constant-width!

It can be seen as a reduction from PIT of sum of c ROABPs (any-width) to PIT of single ROABP (similar-width) in log-variate setting. (c-constant)

Future Directions

Following models are OPEN:

- Poly-time blackbox PIT for log-variate ROABPs (even commutative).

Future Directions

Following models are OPEN:

- Poly-time blackbox PIT for log-variate ROABPs (even commutative).
- Poly-time blackbox PIT for constant-width ROABPs (for unknown var. order and all fields).

Future Directions

Following models are OPEN:

- Poly-time blackbox PIT for log-variate ROABPs (even commutative).
- Poly-time blackbox PIT for constant-width ROABPs (for unknown var. order and all fields).
- Poly-time blackbox PIT for sum of unbounded-many log-variate, constant-width ROABPs.

References

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

[AGKS15] Manindra Agrawal, Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Hitting-sets for roabp and sum of set-multilinear circuits. *SIAM Journal on Computing*, 44(3):669–697, 2015.

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

References ii

[DL77] Richard A DeMillo and Richard J Lipton. A probabilistic remark on algebraic program testing. Technical report, GEORGIA INST OF TECH ATLANTA SCHOOL OF INFORMATION AND COMPUTER SCIENCE, 1977.

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

[FS13] Michael A Forbes and Amir Shpilka. Quasipolynomial-time identity testing of non-commutative and read-once oblivious algebraic branching programs. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 243–252. IEEE, 2013.

[FSS14] Michael A. Forbes, Ramprasad Saptharishi, and Amir Shpilka. Hitting sets for multilinear read-once algebraic branching programs, in any order. In *Symposium on Theory of Computing (STOC), New York, NY, USA, May 31 - June 03, 2014*, pages 867–875, 2014.

[GKS17] Rohit Gurjar, Arpita Korwar, and Nitin Saxena. Identity testing for constant-width, and any-order, read-once oblivious arithmetic branching programs. *Theory of Computing*, 13(2):1–21, 2017. (Preliminary version in CCC'16).

[GKST16] Rohit Gurjar, Arpita Korwar, Nitin Saxena, and Thomas Thierauf. Deterministic identity testing for sum of read-once oblivious arithmetic branching programs. *Computational Complexity*, pages 1–46, 2016. (Conference version in CCC 2015).

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

[MV97] Meena Mahajan and V Vinay. A combinatorial algorithm for the determinant. In *SODA*, pages 730–738, 1997.

[Nis91] Noam Nisan. Lower bounds for non-commutative computation. In *Proceedings of the 23rd ACM Symposium on Theory of Computing, ACM Press*. Citeseer, 1991.

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

[Sax08] Nitin Saxena. Diagonal circuit identity testing and lower bounds. In *International Colloquium on Automata, Languages, and Programming*, pages 60–71. Springer, 2008.

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

[Zip79] Richard Zippel. Probabilistic algorithms for sparse polynomials. In *International Symposium on Symbolic and Algebraic Manipulation*, pages 216–226. Springer, 1979.