Reconstruction of full rank algebraic branching programs

Vineet Nair

Joint work with: Neeraj Kayal, Chandan Saha, Sebastien Tavenas
Arithmetic circuits

\[ f(x) \]

Input Variables: \( x_1, x_2, \ldots, x_n \)
Reconstruction problem

➢ \( f(\mathbf{x}) \in \mathbb{Q}[\mathbf{x}] \) is an \( m \)-variate degree \( d \) polynomial computable by a size \( s \) circuit in circuit class \( C \).

➢ Input:

\[ \alpha \in \mathbb{F}^m \quad \text{Blackbox access} \quad f(\alpha) \]
Reconstruction problem

➢ Input:

\[ \alpha \in \mathbb{F}_m \]

➢ Output: A small arithmetic circuit computing \( f \).

➢ The algorithm should run in time \( \text{poly}(m, s, d, b) \) where \( b \) is the bit length of the coefficients of \( f \).
Polynomial identity testing (PIT):

- **Input:**

  $\alpha \in F^m \rightarrow f(\alpha)$

  Is $f(x) = 0$ ?

- Randomized algorithm for PIT follows easily from Schwartz-Zippel lemma

- Unlike PIT no efficient randomized algorithm is known for reconstruction.
Previous works

- Over finite fields [Shp07],[KS09] gave quasi-poly time deterministic reconstruction algorithm for depth three circuits with constant number of product gates.
Previous works

- Over characteristic zero fields [Sinha16] gave a poly time randomized algorithm for depth three circuits with two product gates.

- [GKL12] gave poly time randomized algorithm for multilinear depth four circuits with two top-level product gates.
Previous works

- [SV09], [MV16] gave deterministic poly time reconstruction for read-once formulas

- [KS03], [FS13] gave deterministic quasi-poly time reconstruction for ROABPs, set-multilinear ABPs and non-commutative ABPs
Average-case reconstruction

- Progress in reconstruction is slow.
- Can we do reconstruction for most circuits in a circuit class $C$?
Average-case reconstruction

- Problem definition: The input $f$ is an $m$ variate degree $d$ polynomial picked according to a distribution $D$ on circuit class $C$.

- Output an efficient reconstruction algorithm for $f$.

- [GKL11], [GKQ13] gave randomized poly time algorithm for average-case reconstruction of multilinear formulas and formulas.
Algebraic branching programs (ABP)

- **Definition:** Consider the product of $d$ matrices as $X_1 \cdot X_2 \cdot \ldots \cdot X_d$, where $X_1$ is a row vector of length $w$, $X_d$ is a column vector of length $w$ and $X_2, \ldots, X_{d-1}$ are $w \times w$ matrices.

- Each entry of $X_i$, $i \in [d]$ is an affine form in $x$ variables. $|X| = m$, example $a_0 + a_1x_1 + \ldots + a_m$.

- Polynomial computed by the ABP is the entry in the $1 \times 1$ matrix computed as above. **Length** and **width** of the ABP is $w$ and $d$ respectively.
Distribution on ABPs

- **Random ABP:** Fix $w, d$ and $m$. Pick the constants of the linear forms independently and uniformly at random from a large set $S \subseteq \mathbb{Q}$.

- **Average-case reconstruction:** Design a reconstruction algorithm for random$(m,w,d,S)$ ABP.
Average-case reconstruction for ABPs

- **Input:** Blackbox access to \( f(x) \) computable by \( \text{random}(m,w,d,S) \) ABP.

- **Output:** A small ABP computing \( f \) with high probability.

- The algorithm should run in time \( \text{poly}(m,w,d,\rho) \) - (\( \rho \) bit length of an element in \( S \)).
Pseudo-random family

- A distribution $D$ on $m$ variate degree $d$ polynomial family with seed length $s=(md)^{O(1)}$ generates a pseudo-random family if
  
  - Every algorithm that distinguishes a polynomial coming from $D$ and uniformly random $m$-variate polynomial with a non-negligible bias runs in time exponential in $s$. 
Candidate family

- [Aar08] conjectures the family $\text{Det}_n(Ax)$ where every entry of $A \in F^{t \times m}$ is chosen uniformly at random from a finite field and $m << t=n^2$ is pseudo-random

Example

<table>
<thead>
<tr>
<th>$x_1 + x_2$</th>
<th>$6x_1 + x_2$</th>
<th>$x_1 + 3x_2$</th>
<th>$5x_1 + 4x_2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$8x_1 + x_2$</td>
<td>$10x_1 + x_2$</td>
<td>$8x_1 + 3x_2$</td>
<td>$3x_1 + 2x_2$</td>
</tr>
<tr>
<td>$8x_1 + 2x_2$</td>
<td>$5x_1 + 4x_2$</td>
<td>$7x_1 + 9x_2$</td>
<td>$11x_1 + x_2$</td>
</tr>
<tr>
<td>$4x_1 + 3x_2$</td>
<td>$9x_1 + 3x_2$</td>
<td>$5x_1 + 6x_2$</td>
<td>$9x_1 + 7x_2$</td>
</tr>
</tbody>
</table>

$m = 2$, $n = 4$
Iterated matrix multiplication

- **Definition:** Consider the product of $d$ matrices as $X_1 \cdot X_2 \cdot \ldots \cdot X_d$, where $X_1$ is a **row** vector of length $w$, $X_d$ is a **column** vector of length $w$ and $X_2, \ldots, X_{d-1}$ are $wxw$ matrices.

- Each entry of $X_i$, $i \in [d]$ is a **distinct** variable. The variables are **disjoint** across matrices.

- $\text{IMM}_{w,d}$ is the entry in $1 \times 1$ matrix computed as above.
Consequence

- Det_n and IMM_{w,d} are affine projections of each other [Mahajan, Vinay 97].

- Hence, it makes sense to ask whether IMM_{w,d}(AX) where A ∈ F^{t \times m} is chosen uniformly at random from a finite S ⊆ Q and m << t = w^2(d-2) + 2w is pseudorandom.
Our Contribution
Main result

- **Main theorem:** An efficient average-case reconstruction algorithm for \( f(x) \in \mathbb{Q}[x] \) computed by \( \text{random}(m,w,d,S) \) ABP where \( m \geq w^2d \).

- The algorithm returns an ABP computing \( f(x) \) with probability \( 1 - \frac{1}{\text{poly}(w,d)} \).

- Running time is \( \text{poly}(m,w,d,b) \) where (\( b \) is the bit length of the coefficients of \( f \)).
Remarks

- Does not resolve Aaronson’s conjecture

- For $\text{IMM}_{w,d}$ the conjecture holds when $m \ll w^2d$

- Our result holds when $m \geq w^2d$

- Our result works even if the matrices are not of uniform width.
Full rank ABPs

- If $m \geq w^2d$ then the affine forms in the ABP are $\mathbb{Q}$-linearly independent with high probability.

- Full rank ABPs: the set of linear forms in $X_1, X_2, \ldots, X_d$ are $\mathbb{Q}$-linearly independent.

- Example:

  $\begin{bmatrix}
  x_1 + x_2 & x_2 + x_3 & x_3 + x_4 \\
  x_4 + x_5 & x_5 + x_6 & x_6 + x_7 \\
  x_7 + x_8 & x_8 + x_9 & x_9 + x_{10} \\
  x_{10} + x_{11} & x_{11} + x_{12} & x_{12} + x_{13} \\
  & & x_{13} + x_{14} \\
  & & x_{14} + x_{15} \\
  & & x_{15} + x_{16}
  \end{bmatrix}$
Full rank ABPs

- If $m \geq w^2d$ then the affine forms in the ABP are $Q$-linearly independent with high probability.

- Full rank ABPs: the set of linear forms in $X_1, X_2, \ldots, X_d$ are $Q$-linearly independent.

- Main result: We design an efficient randomized algorithm to reconstruct full rank ABPs.
Equivalent polynomials

- An $n$-variate polynomial $f$ is equivalent to an $n$-variate polynomial $g$ if there exists an invertible $A \in F^{nxn}$ such that $f(x) = g(Ax)$

Equivalence test:

Is there an invertible $A$ in $F^{nxn}$ such that $f(x) = g(Ax)$
Equivalent polynomials

- Equivalence test:

\[
\text{IMM}(\mathbf{x}) \quad \text{f}(\mathbf{x})
\]

Is there an invertible \( A \) in \( F^{nxn} \) such that
\[
f(\mathbf{x}) = \text{IMM}(A\mathbf{x})
\]

**Remark:** Computing a full rank ABP for \( f \) is the same as designing an efficient randomized equivalence test for IMM.
Group of symmetries of IMM

- **Group of symmetries:** For an $n$ variate polynomial $g(x)$ it is the set of all invertible $A \in \mathbb{F}^{n \times n}$ such that $g(Ax) = g(x)$.

- **Characterization by symmetries:** $g(x)$ is characterized by its group of symmetries then
  - The group of symmetries of $f(x)$ and $g(x)$ are equal if and only if $f(x)$ is a constant multiple of $g(x)$

- **Main theorem 2:** $\text{IMM}_{w,d}$ is characterized by its group of symmetries.
Proof Ideas
Template of the reconstruction algorithm

Assume the input polynomial $f$ is computable by a full rank ABP

**Compute a full rank ABP**
1. Find the layer spaces
2. Glue them together

Do a polynomial identity test to check if the polynomial computed by the ABP is $f$

Output `f is not computable by a full rank ABP’

Output the full rank ABP computing $f$
Pre-processing

- Let an $m$ variate polynomial $f$ be computed by a width $w$ and length $d$ full rank ABP.
  - The number of edges is $n = w^2(d-2) + 2w$

- Two steps of pre-processing:
  - Variable reduction: At the end of this step we get an $n$ variate $f$ computable by a full rank ABP
  - Translation equivalence test: The entries in the matrices of the full rank ABP computing $f$ are linear forms (constant term is 0).
Multiple full rank ABPs for $f$

- Suppose $f$ is computable by a full rank ABP

\[ X_1 \cdot X_2 \cdot \ldots \cdot X_d \]

- Then this full rank ABP for $f$ is not unique

- The following transformations still compute $f$
  - Transposition
  - Left-right multiplication
  - Corner translations
Transposition

- Recall $X_1$ and $X_d$ are row and column vectors.

- Since the eventual product is a $1 \times 1$ matrix, the transpose of the product still computes $f$.

- Hence, $f$ is also computed by:

$$\mathsf{T}X_d \bullet \mathsf{T}X_2 \bullet \ldots \bullet \mathsf{T}X_1$$
Let $A$ be a $wxw$ invertible matrix with entries from $Q$.

- Replace $X_2$ with $X'_2 = X_2 \cdot A$ and $X'_3 = A^{-1} \cdot X_3$.

- $f$ is computed by the product

$$X_1 \cdot X'_2 \cdot X'_3 \cdot \ldots \cdot X_d$$
Corner translations

- Let $B$ be an anti-symmetric $w \times w$ matrix, then

\[ X_1 \cdot B \cdot X_1^T = 0 \]
Let $B_1, B_2, \ldots, B_w$ be anti-symmetric $wxw$ matrices.

Let $Y$ be the matrix such that the $i$-th column of $Y$ is $B_i \cdot ^TX_1$
Corner translations

- Replace $X_2$ with $X'_2 = X_2 + Y$

- Observe that $X_1 \cdot X'_2 = X_1 \cdot X_2$ as $X_1 Y = 0_{wxw}$

- $f$ is computed by the product

\[ X_1 \cdot X'_2 \cdot X_3 \cdot \ldots \cdot X_d = X_1 \cdot (X_2 + Y) \cdot X_3 \cdot \ldots \cdot X_d \]

- Similarly we can define corner translations for $X_{d-1}$
Uniqueness of the layer spaces

- Suppose $f$ is computable by a full rank ABP

$$X_1 \cdot X_2 \cdot \ldots \cdot X_d$$

- Let $X_i$ denote the $\mathbb{Q}$-linear space spanned by the linear forms in $X_i$

- $X_{1,2}$ and $X_{d-1,d}$ denote the $\mathbb{Q}$-linear space spanned by the linear forms in $X_1, X_2$ and $X_{d-1}, X_d$ respectively
Uniqueness of the layer spaces

If \( X'_1 \cdot X'_2 \cdot \ldots \cdot X'_d \) computes \( f \) then

either

\[
X'_i = X_i \quad \text{for} \quad i \in [d] \setminus \{2, d-1\}
\]

\[
X'_{1,2} = X_{1,2} \quad \text{and} \quad X'_{d-1,d} = X_{d-1,d}
\]

or

\[
X'_i = X_{d-i} \quad \text{for} \quad i \in [d] \setminus \{2, d-1\}
\]

\[
X'_{1,2} = X_{d-1,d} \quad \text{and} \quad X'_{d-1,d} = X_{1,2}
\]
Uniqueness of the layer spaces

\[ \chi_1, \chi_3, \chi_4, \ldots, \chi_{d-2}, \chi_d \]

\[ \chi_{1,2}, \chi_{d-1,d} \]
Uniqueness of the layer spaces

$\chi_d \quad \chi_{d-2} \quad \chi_{d-3} \quad \ldots \quad \chi_3 \quad \chi_1$

$\chi_{d-1,d} \quad \chi_{1,2}$
Group of symmetries of IMM

- The set of all invertible $A \in F^{n \times n}$ such that $\text{IMM}_{w,d}(Ax) = \text{IMM}_{w,d}$.

- We show that the group of symmetries are generated by the following subgroups:
  - $T$ denotes the group corresponding to transpositions
  - $M$ denotes the group corresponding to left-right multiplications
  - $C$ denotes the group corresponding to corner translations
Group of symmetries of IMM

Main Theorem:

\[ G_{\text{IMM}} = C \rtimes H, \text{ where } H = M \rtimes T \]

- \( C \) is a normal subgroup in \( G_{\text{IMM}} \) and \( M \) is a normal subgroup in \( H \)

- We also show that IMM is characterized by its group of symmetries

- That is any polynomial with the same symmetry group is a constant multiple of IMM
Computing the full rank ABP: Step 1

- Computing the layer spaces:
  - Study the Lie algebra of the group of symmetries of \( \text{IMM}_{w,d} \)

- [Kay12] Lie algebra of the group of symmetries of \( f \) is the set of matrices
  \[ A = (a_{i,j}) \quad i,j \in [n] \]

\[ \sum_{i,j \in [n]} x_j \frac{\partial f}{\partial x_i} = 0 \]

- We just use the vector space property of the algebra
Invariant spaces

- **Invariant space:** Let $M: \mathbb{Q}^n \to \mathbb{Q}^n$ be a linear operator. $U \subseteq \mathbb{Q}^n$ is an invariant space if $M(U) \subseteq U$

- The definition can be extended to a set of linear operators $\{M_1, M_2, \ldots, M_n\}$

- The layer spaces of an $f$ computed by a full rank ABP are intimately connected to the invariant spaces of Lie algebra of $f$
Computing the layer spaces

- Compute a basis of the Lie algebra of $f$
- Compute the irreducible invariant spaces of the Lie algebra of $f$
- Compute the layer spaces from the irreducible invariant spaces

- Easy: involves solving a set of linear dependencies
- Since $f$ and IMM are equivalent their Lie algebras are conjugates of each other
- We show that the layer spaces are in fact the irreducible invariant spaces in some sense
Computing the full rank ABP: Step 2

- Ordering the layer spaces: We use evaluation dimension to order the layer spaces.

- Definition:
  - Evaluation Dimension for a polynomial $H(\mathbf{x})$ is defined with respect to a set of variables $S \subseteq \mathbf{x}$
  - $\text{Evaldim}_S[H(\mathbf{x})]$ is equal to

    $$\dim(\text{span}\{H(\mathbf{x}) \mid x_j = a_j \text{ for } x_j \in S, \text{ where } a_j \in F\})$$
Ordering the layer spaces

- We make the variables in distinct layers are disjoint by mapping the basis vectors of the layer spaces to distinct variables.

- Then we find the ordering inductively.
Ordering the layer spaces

- **Base Case**

  Evaluation dimension = \( w \)

  ![Diagram showing evaluation dimension equals \( w \) with dots and a shaded area]

  Evaluation dimension = \( w^2 \)

  ![Diagram showing evaluation dimension equals \( w^2 \) with dots and a shaded area]
Ordering the layer spaces

- Inductive Step

  *Evaluation dimension = \( w \)*

  \[ \cdots \cdots \cdots \]

  *Evaluation dimension = \( w^2 \)*

  \[ \cdots \cdots \cdots \]
Thank You