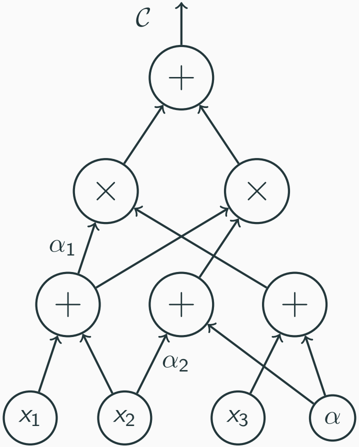
Lower Bounds Against Sums of Ordered Set-Multilinear ABPs

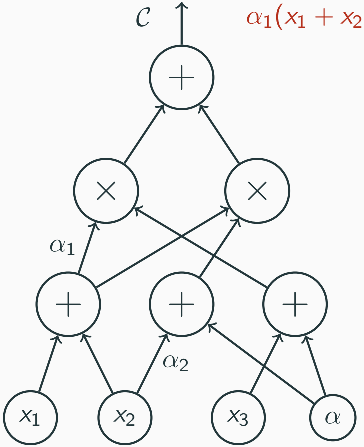
Prerona Chatterjee [with Deepanshu Kush (UoT), Shubhangi Saraf (UoT), Amir Shpilka (TAU)]
Tel Aviv University

January 30, 2024

Algebraic Models of Computation

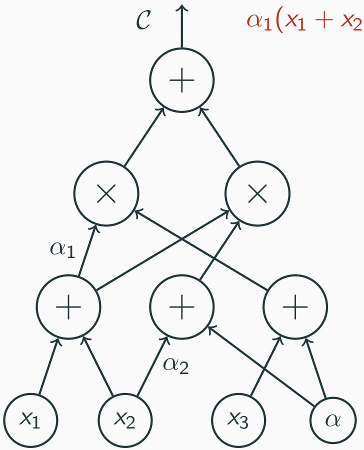


Algebraic Models of Computation

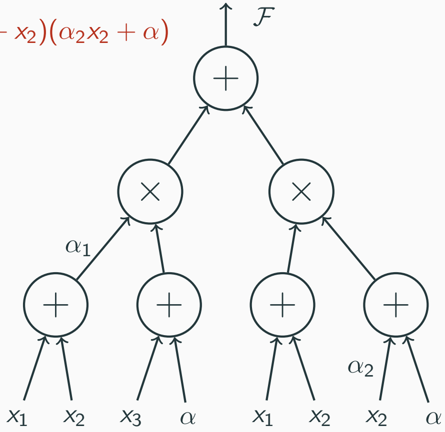


$\alpha_1(x_1 + x_2)(x_3 + \alpha) + (x_1 + x_2)(\alpha_2 x_2 + \alpha)$

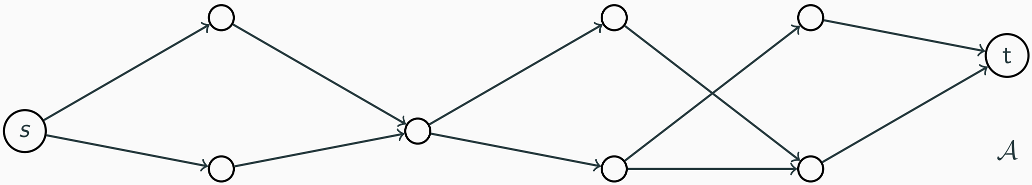
Algebraic Models of Computation



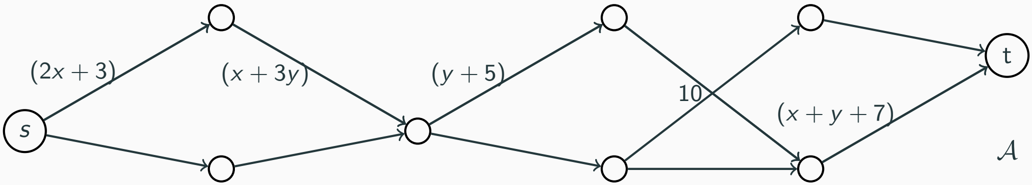
$$\alpha_1(x_1 + x_2)(x_3 + \alpha) + (x_1 + x_2)(\alpha_2 x_2 + \alpha)$$



Algebraic Branching Programs

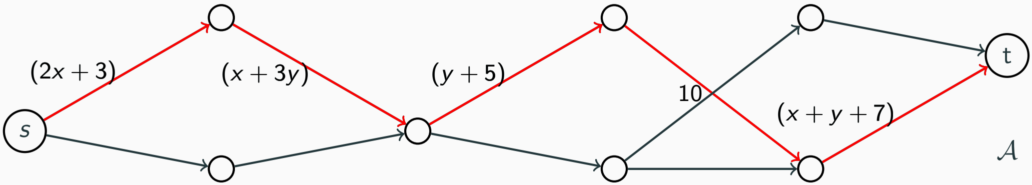


Algebraic Branching Programs



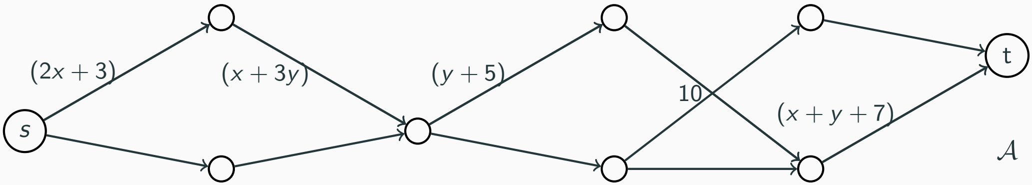
- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$

Algebraic Branching Programs



- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$
- Polynomial computed by the path $p = \mathrm{wt}(p)$: Product of the edge labels on p

Algebraic Branching Programs



- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$
- Polynomial computed by the path $p = \mathrm{wt}(p)$: Product of the edge labels on p
- Polynomial computed by the ABP: $f_{\mathcal{A}}(\mathbf{x}) = \sum_p \mathrm{wt}(p)$

Lower Bounds in Algebraic Circuit Complexity

Objects of Study: Polynomials over n variables of degree d.

Lower Bounds in Algebraic Circuit Complexity

Objects of Study: Polynomials over n variables of degree d.

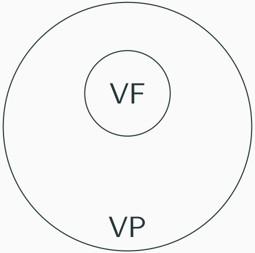
VP: Polynomials computable by circuits of size $\text{poly}(n, d)$.

Lower Bounds in Algebraic Circuit Complexity

Objects of Study: Polynomials over n variables of degree d.

VF: Polynomials computable by formulas of size $\text{poly}(n, d)$.

VP: Polynomials computable by circuits of size $\text{poly}(n, d)$.

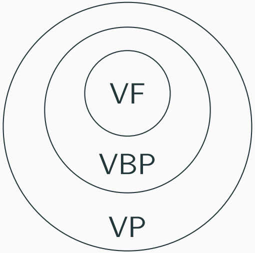


Objects of Study: Polynomials over n variables of degree d.

VF: Polynomials computable by formulas of size $\mathrm{poly}(n, d)$.

VBP: Polynomials computable by ABPs of size $\mathrm{poly}(n, d)$.

VP: Polynomials computable by circuits of size $\mathrm{poly}(n, d)$.

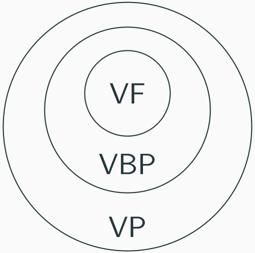


Objects of Study: Polynomials over n variables of degree d.

VF: Polynomials computable by formulas of size $\mathrm{poly}(n, d)$.

VBP: Polynomials computable by ABPs of size $\mathrm{poly}(n, d)$.

VP: Polynomials computable by circuits of size $\mathrm{poly}(n, d)$.



Are the inclusions tight?

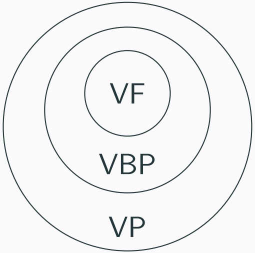
Lower Bounds in Algebraic Circuit Complexity

Objects of Study: Polynomials over n variables of degree d.

VF: Polynomials computable by formulas of size $\text{poly}(n, d)$.

VBP: Polynomials computable by ABPs of size $\text{poly}(n, d)$.

VP: Polynomials computable by circuits of size $\text{poly}(n, d)$.



Are the inclusions tight?

Central Question: Find explicit polynomials that cannot be computed by efficient circuits.

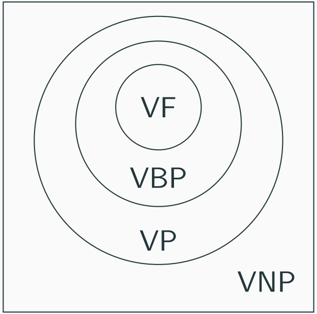
Objects of Study: Polynomials over n variables of degree d.

VF: Polynomials computable by formulas of size $\mathrm{poly}(n, d)$.

VBP: Polynomials computable by ABPs of size $\mathrm{poly}(n, d)$.

VP: Polynomials computable by circuits of size $\mathrm{poly}(n, d)$.

VNP: Explicit Polynomials



Are the inclusions tight?

Central Question: Find explicit polynomials that cannot be computed by efficient circuits.

Our Results

Set-Multilinearity

The variable set is divided into buckets.

$$\mathbf{x} = \mathbf{x}_1 \cup \cdots \cup \mathbf{x}_d \quad \text{where} \quad \mathbf{x}_i = \left\{x_{i,1}, \ldots x_{i,n_i}\right\}.$$

Set-Multilinearity

The variable set is divided into buckets.

$$\mathbf{x} = \mathbf{x}_1 \cup \cdots \cup \mathbf{x}_d \quad \text{where} \quad \mathbf{x}_i = \{x_{i,1}, \ldots x_{i,n_i}\}.$$

f is set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

every monomial in f has exactly one variable from $\mathbf{x}_i$ for each $i \in [d]$.

Set-Multilinearity

The variable set is divided into buckets.

$$\mathbf{x} = \mathbf{x}_1 \cup \cdots \cup \mathbf{x}_d \quad \text{where} \quad \mathbf{x}_i = \left\{x_{i,1}, \ldots x_{i,n_i}\right\}.$$

f is set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

every monomial in f has exactly one variable from $\mathbf{x}_i$ for each $i \in [d]$.

An algebraic formula is set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if every node in it

computes a set-multilinear polynomial with respect to S for some $S \subseteq \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

Set-Multilinearity

The variable set is divided into buckets.

$$\mathbf{x} = \mathbf{x}_1 \cup \cdots \cup \mathbf{x}_d \quad \text{where} \quad \mathbf{x}_i = \{x_{i,1}, \ldots x_{i,n_i}\}.$$

f is set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

every monomial in f has exactly one variable from $\mathbf{x}_i$ for each $i \in [d]$.

An algebraic formula is set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if every node in it

computes a set-multilinear polynomial with respect to S for some $S \subseteq \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

An ABP is set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if every path in it

computes a set-multilinear monomial with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

Ordered Set-Multilinear ABPs (osmABPs)

An ABP is ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP

Ordered Set-Multilinear ABPs (osmABPs)

An ABP is ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- there is a permutation $\sigma \in S_d$ such that

every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

Ordered Set-Multilinear ABPs (osmABPs)

An ABP is ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- there is a permutation $\sigma \in S_d$ such that

every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

$$\mathrm{IMM}_{n,d} = \sum_{1 \le i_1, \ldots, i_{d-1} \le n} x^{(1)}_{1,i_1} \cdot \left(\prod_{j=2}^{d-1} x^{(j)}_{i_{j-1},i_j} \right) \cdot x^{(d)}_{i_{d-1},i_d}.$$

Ordered Set-Multilinear ABPs (osmABPs)

An ABP is ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- there is a permutation $\sigma \in S_d$ such that

every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

$$\mathrm{IMM}_{n,d} = \sum_{1 \le i_1, \ldots, i_{d-1} \le n} x^{(1)}_{1,i_1} \cdot \left(\prod_{j=2}^{d-1} x^{(j)}_{i_{j-1},i_j} \right) \cdot x^{(d)}_{i_{d-1},i_d}.$$

Note: $\mathrm{IMM}_{n,d}$ is complete for the set of polynomials of degree d that are computable by osmABPs (also smABPs and ABPs) which had width at most n in every layer.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log \log n)}$.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log\log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log \log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.
- Separation is not tight: $\mathrm{IMM}_{n,n}$ is computable by a set-multilinear formula of size $n^{O(\log n)}$.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log \log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.
- Separation is not tight: $\mathrm{IMM}_{n,n}$ is computable by a set-multilinear formula of size $n^{O(\log n)}$.

Remarks

- Set-multilinear formula LBs imply formula LBs in low degree.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log\log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.
- Separation is not tight: $\mathrm{IMM}_{n,n}$ is computable by a set-multilinear formula of size $n^{O(\log n)}$.

Remarks

- Set-multilinear formula LBs imply formula LBs in low degree.

  [Raz]: For $d = O(\frac{\log n}{\log\log n})$, if f is computable by a formula of size $s = \mathrm{poly}(n)$, then it is also computable by a set-multilinear formula of size $\mathrm{poly}(n)$.

[Tavenas-Limaye-Srinivasan]

The Result: Any set-multilinear formula computing $\mathrm{IMM}_{n,n}$ has size $n^{\Omega(\log\log n)}$.

- This shows a super-polynomial separation between osmABPs and set-multilinear formulas.
- Separation is not tight: $\mathrm{IMM}_{n,n}$ is computable by a set-multilinear formula of size $n^{O(\log n)}$.

Remarks

- Set-multilinear formula LBs imply formula LBs in low degree.

  [Raz]: For $d = O(\frac{\log n}{\log\log n})$, if f is computable by a formula of size $s = \mathrm{poly}(n)$, then it is also computable by a set-multilinear formula of size $\mathrm{poly}(n)$.
- $n^{\Omega(\log n)}$ set-multilinear formula LB for $\mathrm{IMM}_{n,n}$ implies formula LB due to self-reducibility.

[Kush-Saraf]

The Result: There is an n-variate polynomial, $G_{n,n}$, of degree n computable by a set-multilinear ABP of size $s = \text{poly}(n)$ such that any set-multilinear formula computing it has size $n^{\Omega(\log n)}$.

[Kush-Saraf]

The Result: There is an n-variate polynomial, $G_{n,n}$, of degree n computable by a set-multilinear ABP of size $s = \mathrm{poly}(n)$ such that any set-multilinear formula computing it has size $n^{\Omega(\log n)}$.

Wait! Why doesn't that lead to formula lower bounds?

- f is a projection of $\mathrm{IMM}_{s,n}$
- $\mathrm{IMM}_{s,n}$ is self-reducible

[Kush-Saraf]

The Result: There is an n-variate polynomial, $G_{n,n}$, of degree n computable by a set-multilinear ABP of size $s = \text{poly}(n)$ such that any set-multilinear formula computing it has size $n^{\Omega(\log n)}$.

Wait! Why doesn't that lead to formula lower bounds?

- f is a projection of $\mathrm{IMM}_{s,n}$
- $\mathrm{IMM}_{s,n}$ is self-reducible

But f need not be self-reducible unless f is computable by an osmABP.

[Kush-Saraf]

The Result: There is an n-variate polynomial, $G_{n,n}$, of degree n computable by a set-multilinear ABP of size $s = \text{poly}(n)$ such that any set-multilinear formula computing it has size $n^{\Omega(\log n)}$.

Wait! Why doesn't that lead to formula lower bounds?

- f is a projection of $\mathrm{IMM}_{s,n}$
- $\mathrm{IMM}_{s,n}$ is self-reducible

But f need not be self-reducible unless f is computable by an osmABP.

How different are the powers of smABP**s and** osmABP**s?**

Our Results

Our First Result: There is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear with respect to $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for each $i \in [d]$, such that:

- it has a set-multilinear branching program of size poly(n, d),
- but any ordered set-multilinear branching program computing $G_{n,d}$ requires width $n^{\Omega(d)}$.

Our Results

Our First Result: There is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear with respect to $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for each $i \in [d]$, such that:

- it has a set-multilinear branching program of size poly(n, d),
- but any ordered set-multilinear branching program computing $G_{n,d}$ requires width $n^{\Omega(d)}$.

$\sum$osmABP: Sum of ordered set-multilinear ABPs, each with a possibly different ordering.

Our Results

Our First Result: There is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear with respect to $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for each $i \in [d]$, such that:

- it has a set-multilinear branching program of size poly(n, d),
- but any ordered set-multilinear branching program computing $G_{n,d}$ requires width $n^{\Omega(d)}$.

$\sum$osmABP: Sum of ordered set-multilinear ABPs, each with a possibly different ordering.

Our Second Result: There is polynomial $G_{n,n}(\mathbf{x})$ which is set-multilinear with respect to a set of $\Theta(n)$ buckets, each of size $\Theta(n)$, such that

- it has a set-multilinear branching program of size poly(n),
- but any $\sum$osmABP computing $G_{n,n}(\mathbf{x})$ requires total-width $\exp(\Omega(n^{1/1000}))$.

The [Bhargav-Dwivedi-Saxena] Perspective

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)} s$.

The [Bhargav-Dwivedi-Saxena] Perspective

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)}s$.

What this implies: Super polynomial lower bound against total-width of $\sum$osmABP for a polynomial of degree $O\left(\frac{\log n}{\log \log n}\right)$ implies super-polynomial lower bound against ABPs.

The [Bhargav-Dwivedi-Saxena] Perspective

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)} s$.

What this implies: Super polynomial lower bound against total-width of $\sum$osmABP for a polynomial of degree $O\left(\frac{\log n}{\log \log n}\right)$ implies super-polynomial lower bound against ABPs.

Our Third Result: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

- $G_{n,d}$ is computable by a set-multilinear ABP of size $\text{poly}(n)$,
- any $\sum$osmABP of max-width $\text{poly}(n)$ computing $G_{n,d}$ requires total-width $2^{\Omega(d)}$.

Results for polynomials in VP, VNP

Our Fourth Result: There is polynomial family $\{F_{n,n}(\mathbf{x})\}$, in VP, which is set-multilinear with respect to a set of $\Theta(n)$ buckets, each of size $\Theta(n)$, such that

$$\text{any } \sum\text{osmABP computing it requires total-width } \exp(\Omega(n^{1/3})).$$

Results for polynomials in VP, VNP

Our Fourth Result: There is polynomial family $\{F_{n,n}(\mathbf{x})\}$, in VP, which is set-multilinear with respect to a set of $\Theta(n)$ buckets, each of size $\Theta(n)$, such that

$$\text{any } \sum \text{osmABP computing it requires total-width } \exp(\Omega(n^{1/3})).$$

The same result also holds for the Nisan-Wigderson polynomial family, which is in VNP.

Results for polynomials in VP, VNP

Our Fourth Result: There is polynomial family $\{F_{n,n}(\mathbf{x})\}$, in VP, which is set-multilinear with respect to a set of $\Theta(n)$ buckets, each of size $\Theta(n)$, such that

$$\text{any } \sum \text{osmABP computing it requires total-width } \exp(\Omega(n^{1/3})).$$

The same result also holds for the Nisan-Wigderson polynomial family, which is in VNP.

Our Fifth Result: For $\omega(\log n) = d \leq n$, there is polynomial family $\{F_{n,d}(\mathbf{x})\}$, in VP, which is set-multilinear with respect to a set of $\Theta(d)$ buckets, each of size $\Theta(n)$, such that

$$F_{n,d} \text{ cannot be computed by a } \sum \text{osmABP of total-width poly(n).}$$

Results for polynomials in VP, VNP

Our Fourth Result: There is polynomial family $\{F_{n,n}(\mathbf{x})\}$, in VP, which is set-multilinear with respect to a set of $\Theta(n)$ buckets, each of size $\Theta(n)$, such that

$$\text{any } \sum\text{osmABP computing it requires total-width } \exp(\Omega(n^{1/3})).$$

The same result also holds for the Nisan-Wigderson polynomial family, which is in VNP.

Our Fifth Result: For $\omega(\log n) = d \leq n$, there is polynomial family $\{F_{n,d}(\mathbf{x})\}$, in VP, which is set-multilinear with respect to a set of $\Theta(d)$ buckets, each of size $\Theta(n)$, such that

$$F_{n,d} \text{ cannot be computed by a } \sum\text{osmABP of total-width poly(n)}.$$

The same result also holds for the Nisan-Wigderson polynomial family, which is in VNP.

Related Results

[Bhargav-Dwivedi-Saxena]

Any $\sum$osmABP computing $\mathrm{IMM}_{n,n}$ which has max-width $n^{o(1)}$ must have $2^{\Omega(n)}$ summands.

Related Results

[Bhargav-Dwivedi-Saxena]

Any $\sum$ osmABP computing $\mathrm{IMM}_{n,n}$ which has max-width $n^{o(1)}$ must have $2^{\Omega(n)}$ summands.

[Arvind-Raja]

Any $\sum_{i=1}^{t}$ osmABP computing the $n \times n$ permanent polynomial has max-width $2^{\Omega(n/t)}$.

Sum of ROABPs

ROABPs: An ABP computing $f \in \mathbb{F}[x_1, \ldots x_n]$ is an ROABP, in order $\sigma \in S_n$, if there are n layers in the ABP and, for each $i \in [n]$, every edge in layer i is labelled a polynomial in $x_{\sigma(i)}$.

Sum of ROABPs

ROABPs: An ABP computing $f \in \mathbb{F}[x_1, \ldots x_n]$ is an ROABP, in order $\sigma \in S_n$, if there are n layers in the ABP and, for each $i \in [n]$, every edge in layer i is labelled a polynomial in $x_{\sigma(i)}$.

Note: $\sum$ osmABP LBs imply $\sum$ ROABP LBs.

Sum of ROABPs

ROABPs: An ABP computing $f \in \mathbb{F}[x_1, \ldots x_n]$ is an ROABP, in order $\sigma \in S_n$, if there are n layers in the ABP and, for each $i \in [n]$, every edge in layer i is labelled a polynomial in $x_{\sigma(i)}$.

Note: $\sum$ osmABP LBs imply $\sum$ ROABP LBs.

Proof Sketch:

- $f \in \mathbb{F}[\mathbf{x}_1, \ldots, \mathbf{x}_d]$ is a set-multilinear polynomial for which $\sum$ osmABP LB is known.

Sum of ROABPs

ROABPs: An ABP computing $f \in \mathbb{F}[x_1, \ldots x_n]$ is an ROABP, in order $\sigma \in S_n$, if there are n layers in the ABP and, for each $i \in [n]$, every edge in layer i is labelled a polynomial in $x_{\sigma(i)}$.

Note: $\sum$osmABP LBs imply $\sum$ROABP LBs.

Proof Sketch:

- $f \in \mathbb{F}[\mathbf{x}_1, \ldots, \mathbf{x}_d]$ is a set-multilinear polynomial for which $\sum$osmABP LB is known.
- Define $g_f(x_1, \ldots, x_d) = \sum_{\mathbf{e} \in [n]^d} \prod_{i=1}^{e_i} x_i^{e_i} \cdot \mathrm{coeff}_{x_{i,e_i}}(f)$.

Sum of ROABPs

ROABPs: An ABP computing $f \in \mathbb{F}[x_1, \ldots x_n]$ is an ROABP, in order $\sigma \in S_n$, if there are n layers in the ABP and, for each $i \in [n]$, every edge in layer i is labelled a polynomial in $x_{\sigma(i)}$.

Note: $\sum$ osmABP LBs imply $\sum$ ROABP LBs.

Proof Sketch:

- $f \in \mathbb{F}[\mathbf{x}_1, \ldots, \mathbf{x}_d]$ is a set-multilinear polynomial for which $\sum$ osmABP LB is known.
- Define $g_f(x_1, \ldots, x_d) = \sum_{\mathbf{e} \in [n]^d} \prod_{i=1}^{e_i} x_i^{e_i} \cdot \mathrm{coeff}_{x_{i,e_i}}(f)$.
- Suppose g_f is computable by a $\sum$ ROABP, $\sum \mathcal{A}_j$, with $\mathcal{A}_j$ ordered σ_j.

Sum of ROABPs

ROABPs: An ABP computing $f \in \mathbb{F}[x_1, \ldots x_n]$ is an ROABP, in order $\sigma \in S_n$, if there are n layers in the ABP and, for each $i \in [n]$, every edge in layer i is labelled a polynomial in $x_{\sigma(i)}$.

Note: $\sum$osmABP LBs imply $\sum$ROABP LBs.

Proof Sketch:

- $f \in \mathbb{F}[\mathbf{x}_1, \ldots, \mathbf{x}_d]$ is a set-multilinear polynomial for which $\sum$osmABP LB is known.
- Define $g_f(x_1, \ldots, x_d) = \sum_{\mathbf{e} \in [n]^d} \prod_{i=1}^{e_i} x_i^{e_i} \cdot \mathrm{coeff}_{x_{i,e_i}}(f)$.
- Suppose g_f is computable by a $\sum$ROABP, $\sum \mathcal{A}_j$, with $\mathcal{A}_j$ ordered σ_j.
- Construct a $\sum$osmABP, $\sum \mathcal{A}'_j$, with $\mathcal{A}'_j$ ordered σ_j, by replacing $x_i^{e_i}$ by x_{i,e_i} for $e_i \in [n_i]$ and erasing other components on each edge.

Sum of ROABPs

ROABPs: An ABP computing $f \in \mathbb{F}[x_1, \ldots x_n]$ is an ROABP, in order $\sigma \in S_n$, if there are n layers in the ABP and, for each $i \in [n]$, every edge in layer i is labelled a polynomial in $x_{\sigma(i)}$.

Note: $\sum$ osmABP LBs imply $\sum$ ROABP LBs.

Proof Sketch:

- $f \in \mathbb{F}[\mathbf{x}_1, \ldots, \mathbf{x}_d]$ is a set-multilinear polynomial for which $\sum$ osmABP LB is known.
- Define $g_f(x_1, \ldots, x_d) = \sum_{\mathbf{e} \in [n]^d} \prod_{i=1}^{e_i} x_i^{e_i} \cdot \mathrm{coeff}_{x_{i,e_i}}(f)$.
- Suppose g_f is computable by a $\sum$ ROABP, $\sum \mathcal{A}_j$, with $\mathcal{A}_j$ ordered σ_j.
- Construct a $\sum$ osmABP, $\sum \mathcal{A}'_j$, with $\mathcal{A}'_j$ ordered σ_j, by replacing $x_i^{e_i}$ by x_{i,e_i} for $e_i \in [n_i]$ and erasing other components on each edge.
- Check that this $\sum$ osmABP computes f and use known LB.

[Ramya-Rao]

There exists an explicit multilinear polynomial family $\{g_n(x_1, \ldots, x_n)\}$, in VP, such that any $\sum$ ROABP computing g_n has total width $2^{\Omega\left(\frac{n^{1/6}}{\log n}\right)}$.

Related Results

[Ramya-Rao]

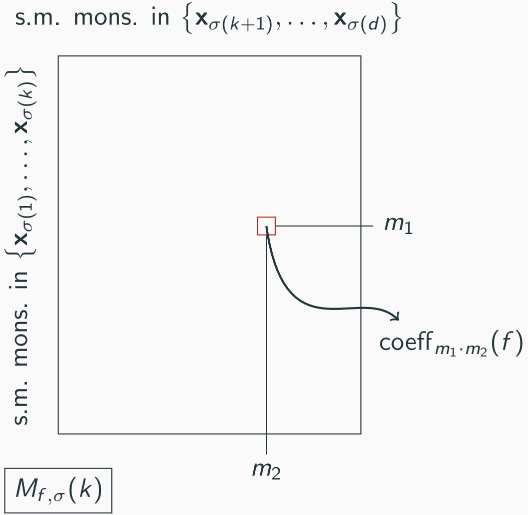
There exists an explicit multilinear polynomial family $\{g_n(x_1, \ldots, x_n)\}$, in VP, such that any $\sum$ ROABP computing g_n has total width $2^{\Omega\left(\frac{n^{1/6}}{\log n}\right)}$.

[Ghoshal-Rao]

There exists an explicit multilinear polynomial family $\{g_n(x_1, \ldots, x_n)\}$, in VBP, such that any $\sum$ ROABP computing g_n that has max-width poly(n) must have total width $2^{\Omega\left(n^{1/500}\right)}$.

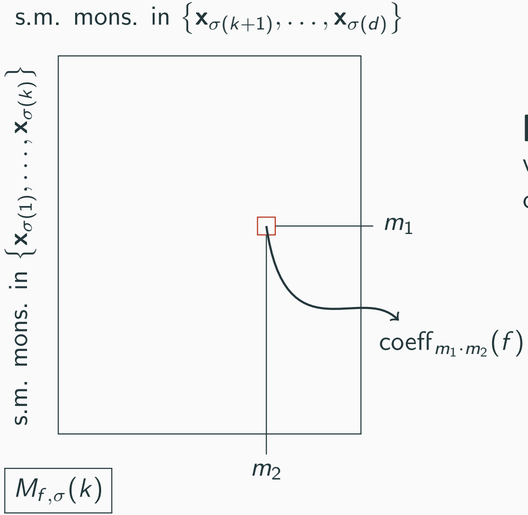
Proof Overviews

Lower Bound for osmABPs



f is a set-multilinear poly. w.r.t $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

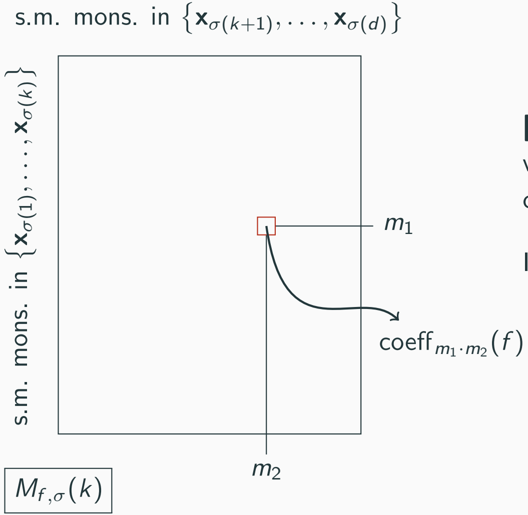
Lower Bound for osmABPs



f is a set-multilinear poly. w.r.t $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

[Nisan]: For every $1 \leq k \leq d$, the number of vertices in the k-th layer of the smallest osmABP(σ) computing f is equal to the rank of $M_{f,\sigma}(k)$.

Lower Bound for osmABPs



f is a set-multilinear poly. w.r.t $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$.

[Nisan]: For every $1 \leq k \leq d$, the number of vertices in the k-th layer of the smallest osmABP(σ) computing f is equal to the rank of $M_{f,\sigma}(k)$.

If $\mathcal{A}$ is the smallest osmABP computing f, then

$$\mathrm{size}(\mathcal{A}) = \sum_{i=1}^{d} \mathrm{rank}(M_{f,\sigma}(k)).$$

Arc Partition

Fix an even $d \in \mathbb{N}$ arbitrarily. We will define a distribution $\mathcal{D}$ over the set $\{-k, +k\}^d$.

Arc Partition

Fix an even $d \in \mathbb{N}$ arbitrarily. We will define a distribution $\mathcal{D}$ over the set $\{-k, +k\}^d$.

$$P_1 = (1, 2),\ \mathcal{P}_1 = \{P_1\},\ [L_1, R_1] = [1, 2].$$

Arc Partition

Fix an even $d \in \mathbb{N}$ arbitrarily. We will define a distribution $\mathcal{D}$ over the set $\{-k, +k\}^d$.

$$P_1 = (1, 2),\ \mathcal{P}_1 = \{P_1\},\ [L_1, R_1] = [1, 2].$$

For $t = 2, \ldots, d/2$, each of the following happens with probability $1/3$:

Arc Partition

Fix an even $d \in \mathbb{N}$ arbitrarily. We will define a distribution $\mathcal{D}$ over the set $\{-k, +k\}^d$.

$$P_1 = (1, 2),\ \mathcal{P}_1 = \{P_1\},\ [L_1, R_1] = [1, 2].$$

For $t = 2, \ldots, d/2$, each of the following happens with probability $1/3$:

- $P_t = (L_{t-1} - 1, R_{t-1} + 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 1, R_{t-1} + 1]$

Arc Partition

Fix an even $d \in \mathbb{N}$ arbitrarily. We will define a distribution $\mathcal{D}$ over the set $\{-k, +k\}^d$.

$$P_1 = (1, 2),\ \mathcal{P}_1 = \{P_1\},\ [L_1, R_1] = [1, 2].$$

For $t = 2, \ldots, d/2$, each of the following happens with probability $1/3$:

- $P_t = (L_{t-1} - 1, R_{t-1} + 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 1, R_{t-1} + 1]$
- $P_t = (R_{t-1} + 1, R_{t-1} + 2)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1}, R_{t-1} + 2]$

Arc Partition

Fix an even $d \in \mathbb{N}$ arbitrarily. We will define a distribution $\mathcal{D}$ over the set $\{-k, +k\}^d$.

$$P_1 = (1, 2),\ \mathcal{P}_1 = \{P_1\},\ [L_1, R_1] = [1, 2].$$

For $t = 2, \ldots, d/2$, each of the following happens with probability $1/3$:

- $P_t = (L_{t-1} - 1, R_{t-1} + 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 1, R_{t-1} + 1]$
- $P_t = (R_{t-1} + 1, R_{t-1} + 2)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1}, R_{t-1} + 2]$
- $P_t = (L_{t-1} - 2, L_{t-1} - 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 2, R_{t-1} + 1]$

Arc Partition

Fix an even $d \in \mathbb{N}$ arbitrarily. We will define a distribution $\mathcal{D}$ over the set $\{-k, +k\}^d$.

$$P_1 = (1, 2),\ \mathcal{P}_1 = \{P_1\},\ [L_1, R_1] = [1, 2].$$

For $t = 2, \ldots, d/2$, each of the following happens with probability $1/3$:

- $P_t = (L_{t-1} - 1, R_{t-1} + 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 1, R_{t-1} + 1]$
- $P_t = (R_{t-1} + 1, R_{t-1} + 2)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1}, R_{t-1} + 2]$
- $P_t = (L_{t-1} - 2, L_{t-1} - 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 2, R_{t-1} + 1]$

For $t = 1, \ldots, d/2$, if $P_t = (i_t, j_t)$, then each of the following happens with probability $1/2$:

Arc Partition

Fix an even $d \in \mathbb{N}$ arbitrarily. We will define a distribution $\mathcal{D}$ over the set $\{-k, +k\}^d$.

$$P_1 = (1, 2),\ \mathcal{P}_1 = \{P_1\},\ [L_1, R_1] = [1, 2].$$

For $t = 2, \ldots, d/2$, each of the following happens with probability $1/3$:

- $P_t = (L_{t-1} - 1, R_{t-1} + 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 1, R_{t-1} + 1]$
- $P_t = (R_{t-1} + 1, R_{t-1} + 2)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1}, R_{t-1} + 2]$
- $P_t = (L_{t-1} - 2, L_{t-1} - 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 2, R_{t-1} + 1]$

For $t = 1, \ldots, d/2$, if $P_t = (i_t, j_t)$, then each of the following happens with probability $1/2$:

- i_t-th position is assigned $-k$ and j_t-th position is assigned $+k$

Arc Partition

Fix an even $d \in \mathbb{N}$ arbitrarily. We will define a distribution $\mathcal{D}$ over the set $\{-k, +k\}^d$.

$$P_1 = (1, 2),\ \mathcal{P}_1 = \{P_1\},\ [L_1, R_1] = [1, 2].$$

For $t = 2, \ldots, d/2$, each of the following happens with probability $1/3$:

- $P_t = (L_{t-1} - 1, R_{t-1} + 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 1, R_{t-1} + 1]$
- $P_t = (R_{t-1} + 1, R_{t-1} + 2)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1}, R_{t-1} + 2]$
- $P_t = (L_{t-1} - 2, L_{t-1} - 1)$, $\mathcal{P}_t = \mathcal{P}_{t-1} \cup \{P_t\}$, $[L_t, R_t] = [L_{t-1} - 2, R_{t-1} + 1]$

For $t = 1, \ldots, d/2$, if $P_t = (i_t, j_t)$, then each of the following happens with probability $1/2$:

- i_t-th position is assigned $-k$ and j_t-th position is assigned $+k$
- i_t-th position is assigned $+k$ and j_t-th position is assigned $-k$

The Hard Polynomial

$$G_{n,d} = \sum_{\mathcal{P}_{d/2}} \prod_{(i,j)\in\mathcal{P}} y_{i,j} y_{j,i} \cdot \left(\sum_{k=1}^{n} x_{i,k} x_{j,k} \right).$$

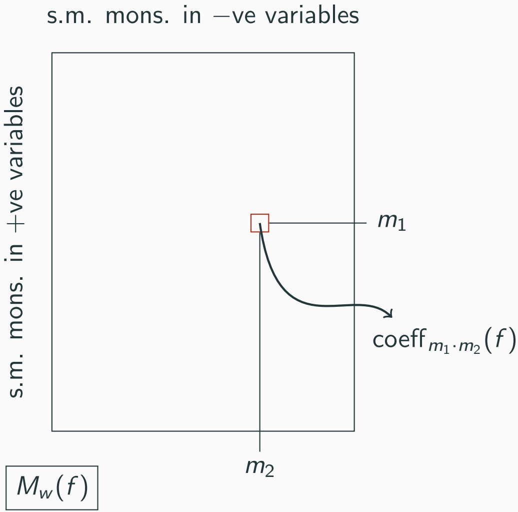
The Hard Polynomial

$$G_{n,d} = \sum_{\mathcal{P}_{d/2}} \prod_{(i,j)\in\mathcal{P}} y_{i,j} y_{j,i} \cdot \left(\sum_{k=1}^{n} x_{i,k} x_{j,k} \right).$$

Properties:

- $G_{n,d}$ is computable by a set-multilinear ABP of size poly(n, d).

The Hard Polynomial



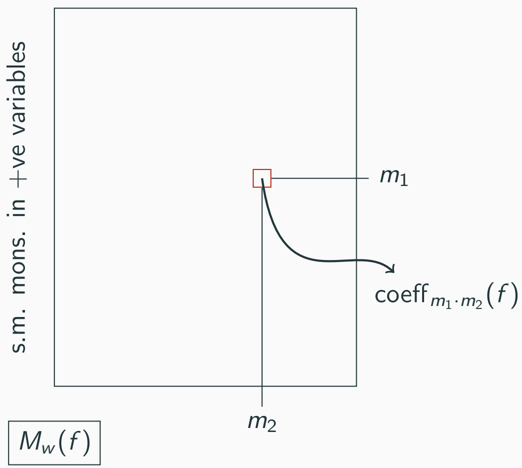
$$G_{n,d} = \sum_{\mathcal{P}_{d/2}} \prod_{(i,j)\in\mathcal{P}} y_{i,j} y_{j,i} \cdot \left(\sum_{k=1}^{n} x_{i,k} x_{j,k} \right).$$

Properties:

- $G_{n,d}$ is computable by a set-multilinear ABP of size $\mathrm{poly}(n, d)$.
- For every $w \in \{+k, -k\}^d$, that has non-zero mass in $\mathcal{D}$,

$$\mathrm{rank}(M_w(G_{n,d})) = \Omega(n^d).$$

The Hard Polynomial



$$G_{n,d} = \sum_{\mathcal{P}_{d/2}} \prod_{(i,j)\in\mathcal{P}} y_{i,j}y_{j,i} \cdot \left(\sum_{k=1}^{n} x_{i,k}x_{j,k}\right).$$

Properties:

- $G_{n,d}$ is computable by a set-multilinear ABP of size $\text{poly}(n, d)$.
- For every $w \in \{+k, -k\}^d$, that has non-zero mass in $\mathcal{D}$,
$$\text{rank}(M_w(G_{n,d})) = \Omega(n^d).$$
- For every $\sigma \in S_d$, there is some $\mathcal{P}$ such that for at least $d/8$ of the $P = (i, j) \in \mathcal{P}$, $i \in \{\sigma(1), \ldots \sigma(\frac{d}{2})\}$ & $j \in \{\sigma(1 + \frac{d}{2})), \ldots \sigma(d)\}$

[Bhargav-Dwivedi-Saxena]

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t. the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)}s$.

[Bhargav-Dwivedi-Saxena]

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t. the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)}s$.

Steps:

1. Homogenisation: Let $P_{n,d}$ can be computed by a homogeneous ABP of width s and length d.

[Bhargav-Dwivedi-Saxena]

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t. the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)} s$.

Steps:

1. Homogenisation: Let $P_{n,d}$ can be computed by a homogeneous ABP of width s and length d. Furthermore, all the edge labels are linear forms.

[Bhargav-Dwivedi-Saxena]

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t. the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)}s$.

Steps:

1. Homogenisation: Let $P_{n,d}$ can be computed by a homogeneous ABP of width s and length d. Furthermore, all the edge labels are linear forms.

2. Set-Multilinearisation:

[Bhargav-Dwivedi-Saxena]

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t. the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)}s$.

Steps:

1. Homogenisation: Let $P_{n,d}$ can be computed by a homogeneous ABP of width s and length d. Furthermore, all the edge labels are linear forms.

2. Set-Multilinearisation:

$$P_{n,d} = \prod_{i=1}^{d} M_i$$

[Bhargav-Dwivedi-Saxena]

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t. the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)} s$.

Steps:

1. Homogenisation: Let $P_{n,d}$ can be computed by a homogeneous ABP of width s and length d. Furthermore, all the edge labels are linear forms.

2. Set-Multilinearisation:

$$P_{n,d} = \prod_{i=1}^{d} M_i = \prod_{i=1}^{d} \sum_{j=1}^{n} M_{ij}$$

[Bhargav-Dwivedi-Saxena]

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t. the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)} s$.

Steps:

1. Homogenisation: Let $P_{n,d}$ can be computed by a homogeneous ABP of width s and length d. Furthermore, all the edge labels are linear forms.

2. Set-Multilinearisation:

$$P_{n,d} = \prod_{i=1}^{d} M_i = \prod_{i=1}^{d} \sum_{j=1}^{n} M_{ij} = \sum_{\sigma \in S_d} \prod_{i=1}^{d} M_{i,\sigma(i)}.$$

[Bhargav-Dwivedi-Saxena]

The Result: Let $P_{n,d}(\mathbf{x})$ be a polynomial of degree d that is set-multilinear w.r.t. the partition $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, If $P_{n,d}$ can be computed by an ABP of size s, then it can also be computed by a $\sum$osmABP of max-width s and total-width $2^{O(d \log d)}s$.

Steps:

1. Homogenisation: Let $P_{n,d}$ can be computed by a homogeneous ABP of width s and length d. Furthermore, all the edge labels are linear forms.
2. Set-Multilinearisation: $P_{n,d}$ can be computed by a $\sum$osmABP of total width $d!s$.

$$P_{n,d} = \prod_{i=1}^{d} M_i = \prod_{i=1}^{d} \sum_{j=1}^{n} M_{ij} = \sum_{\sigma \in S_d} \prod_{i=1}^{d} M_{i,\sigma(i)}.$$

Our Proof Strategy

$$f = \sum_{i=1}^{t} g_i \quad \text{where} \quad g_i = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g^{(i)}_{u_{j-1},u_j}.$$

Our Proof Strategy

$$f = \sum_{i=1}^{t} g_i \quad \text{where} \quad g_i = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g^{(i)}_{u_{j-1},u_j}.$$

- Measure used: $\mu_w(f) = \mathrm{relrank}(M_w(f))$ for $w \in \{-k,+k\}^d$.

Our Proof Strategy

$$f = \sum_{i=1}^{t} g_i \quad \text{where} \quad g_i = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g^{(i)}_{u_{j-1},u_j}.$$

- Measure used: $\mu_w(f) = \mathrm{relrank}(M_w(f))$ for $w \in \{-k, +k\}^d$.
- Using subadditivity of μ_w,

$$\mu_w(f) \le t \cdot \mu_w(g_i).$$

Our Proof Strategy

$$f = \sum_{i=1}^{t} g_i \quad \text{where} \quad g_i = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g^{(i)}_{u_{j-1},u_j}.$$

- Measure used: $\mu_w(f) = \mathrm{relrank}(M_w(f))$ for $w \in \{-k,+k\}^d$.
- Using subadditivity of μ_w,

$$\mu_w(f) \leq t \cdot \mu_w(g_i).$$

- Using union bound we will get the lower bound if we can show that:

When $w \sim \mathcal{D}$, for $g = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g_{u_{j-1},u_j}$, with high probability $\mu_w(g)$ is far from full.

Our Proof Strategy (contd.)

To show: For $g = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g_{u_{j-1},u_j}$, $\mu_w(g)$ is far from full with high probability if $w \sim \mathcal{D}$.

Our Proof Strategy (contd.)

To show: For $g = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g_{u_{j-1},u_j}$, $\mu_w(g)$ is far from full with high probability if $w \sim \mathcal{D}$.

Let S_j be the bucket indices of the variables on which g_{u_{j-1},u_j} depends.

Our Proof Strategy (contd.)

To show: For $g = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g_{u_{j-1},u_j}$, $\mu_w(g)$ is far from full with high probability if $w \sim \mathcal{D}$.

Let S_j be the bucket indices of the variables on which g_{u_{j-1},u_j} depends.

$\mu_{w_{S_j}}(g_{u_{j-1},u_j})$ is small when the diff. between the no. of $+$ve and $-$ve indices in S_j is high.

Our Proof Strategy (contd.)

To show: For $g = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g_{u_{j-1},u_j}$, $\mu_w(g)$ is far from full with high probability if $w \sim \mathcal{D}$.

Let S_j be the bucket indices of the variables on which g_{u_{j-1},u_j} depends.

$\mu_{w_{S_j}}(g_{u_{j-1},u_j})$ is small when the diff. between the no. of $+$ve and $-$ve indices in S_j is high.

Want to show: For a lot of js, this happens with high probability when $w \sim \mathcal{D}$.

Our Proof Strategy (contd.)

To show: For $g = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g_{u_{j-1},u_j}$, $\mu_w(g)$ is far from full with high probability if $w \sim \mathcal{D}$.

Let S_j be the bucket indices of the variables on which g_{u_{j-1},u_j} depends.

$\mu_{w_{S_j}}(g_{u_{j-1},u_j})$ is small when the diff. between the no. of $+$ve and $-$ve indices in S_j is high.

Want to show: For a lot of js, this happens with high probability when $w \sim \mathcal{D}$.

- For hard polynomial in VP, choose q wisely and then use Chernoff bound.

Our Proof Strategy (contd.)

To show: For $g = \sum_{u_1,\ldots,u_{q-1}} \prod_{j=1}^{q} g_{u_{j-1},u_j}$, $\mu_w(g)$ is far from full with high probability if $w \sim \mathcal{D}$.

Let S_j be the bucket indices of the variables on which g_{u_{j-1},u_j} depends.

$\mu_{w_{S_j}}(g_{u_{j-1},u_j})$ is small when the diff. between the no. of $+$ve and $-$ve indices in S_j is high.

Want to show: For a lot of js, this happens with high probability when $w \sim \mathcal{D}$.

- For hard polynomial in VP, choose q wisely and then use Chernoff bound.
- For hard polynomial in VBP, choose q wisely and then use *many violations lemma*.

Open Questions

Open Threads

1. PIT for $\sum$ ROABP?

Open Threads

1. PIT for $\sum$ ROABP?
2. Super-quadratic lower bounds against smABPs?

Open Threads

1. PIT for $\sum$ ROABP?
2. Super-quadratic lower bounds against smABPs?

Thank you!