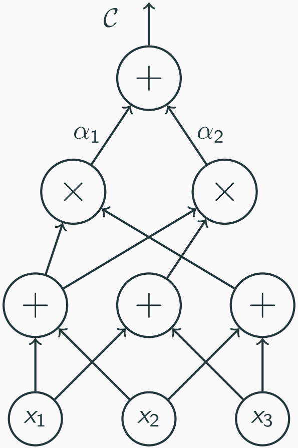
Lower Bounds in Algebraic Circuit Complexity

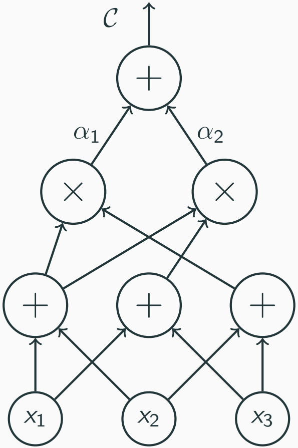
Prerona Chatterjee
Tata Institute of Fundamental Research, Mumbai

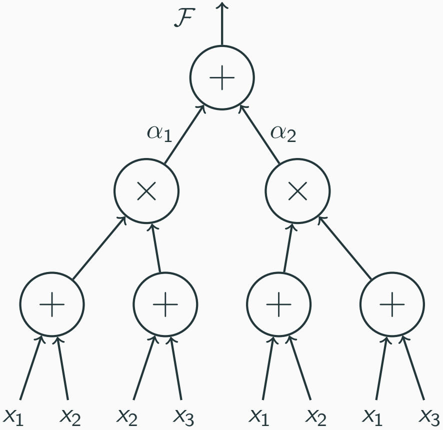
June 22, 2021

Algebraic Formulas and Algebraic Circuits



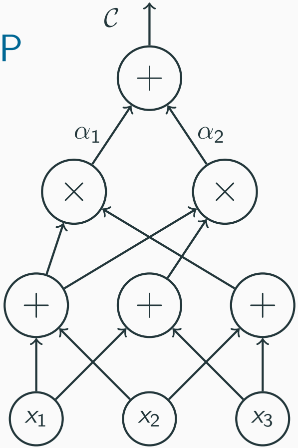
Algebraic Formulas and Algebraic Circuits

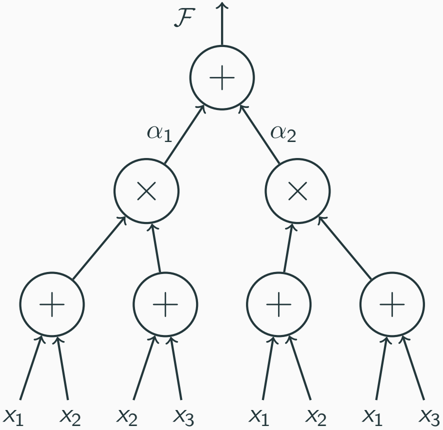




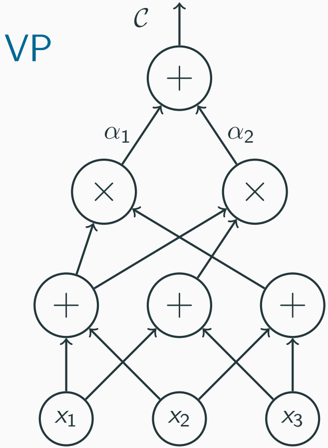
Algebraic Formulas and Algebraic Circuits

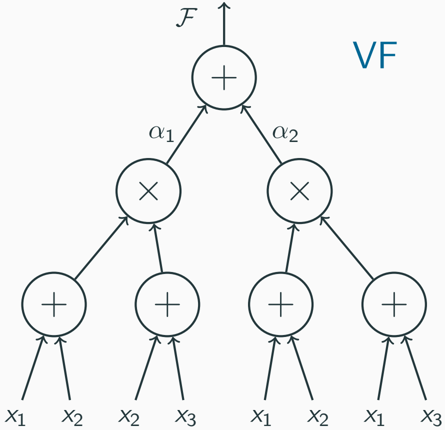
VP





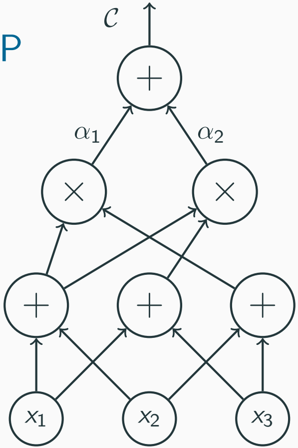
Algebraic Formulas and Algebraic Circuits



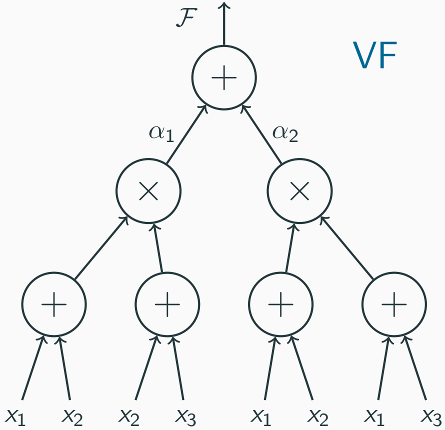


Algebraic Formulas and Algebraic Circuits

VP

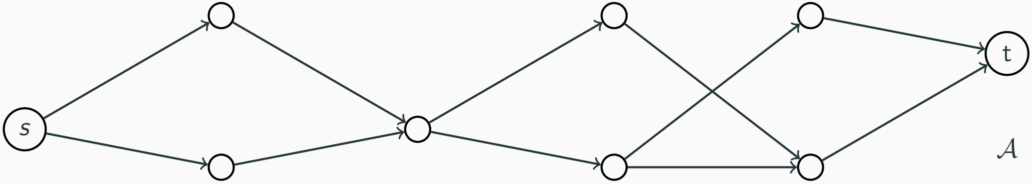


VF $\subseteq$ VP

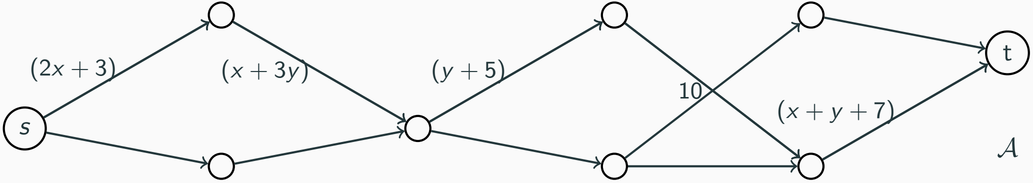


VF

Algebraic Branching Programs

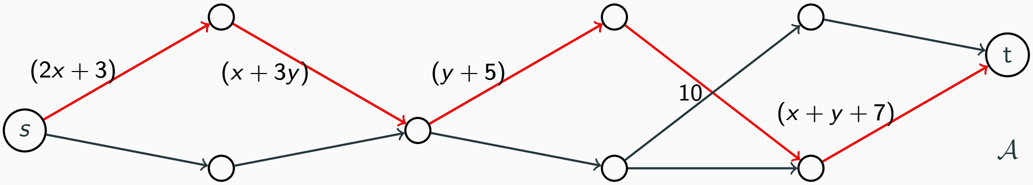


Algebraic Branching Programs



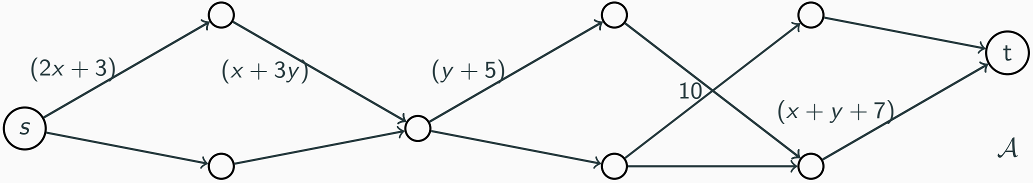
- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$

Algebraic Branching Programs



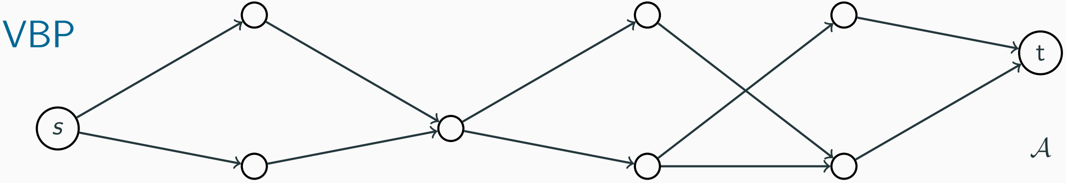
- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$
- Polynomial computed by the path $p = \text{wt}(p)$: Product of the edge labels on p

Algebraic Branching Programs



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

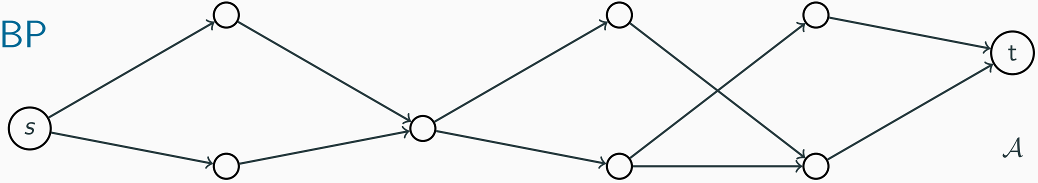
Algebraic Branching Programs



- 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: $f_{\mathcal{A}}(\mathbf{x}) = \sum_p \mathrm{wt}(p)$

Algebraic Branching Programs

VBP



- 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: $f_{\mathcal{A}}(\mathbf{x}) = \sum_p \mathrm{wt}(p)$

$$\mathrm{VF} \subseteq \mathrm{VBP} \subseteq \mathrm{VP}$$

A Recent Breakthrough

The Central Lower Bound Question:

Find explicit n-variate degree d polynomial that cannot be computed by $\text{poly}(n, d)$-size circuits.

A Recent Breakthrough

The Central Lower Bound Question:

Find explicit n-variate degree d polynomial that cannot be computed by $\text{poly}(n, d)$-size circuits.

A Superpolynomial Lower Bound against Constant Depth Circuits:

[Limaye-Srinivasan-Tavenas]: There exists an explicit n-variate degree d polynomial in VP such that no constant depth circuit of size $\text{poly}(n, d)$ can compute it.

A Recent Breakthrough

The Central Lower Bound Question:

Find explicit n-variate degree d polynomial that cannot be computed by $\text{poly}(n, d)$-size circuits.

A Superpolynomial Lower Bound against Constant Depth Circuits:

[Limaye-Srinivasan-Tavenas]: There exists an explicit n-variate degree d polynomial in VP such that no constant depth circuit of size $\text{poly}(n, d)$ can compute it.

Importance of Constant-Depth Circuits

[Agrawal-Vinay]: (Hom.) circuits of size s can be converted to (hom.) $\Sigma\Pi\Sigma\Pi$ of size $s^{O(\sqrt{d})}$.

A Recent Breakthrough

The Central Lower Bound Question:

Find explicit n-variate degree d polynomial that cannot be computed by $\text{poly}(n, d)$-size circuits.

A Superpolynomial Lower Bound against Constant Depth Circuits:

[Limaye-Srinivasan-Tavenas]: There exists an explicit n-variate degree d polynomial in VP such that no constant depth circuit of size $\text{poly}(n, d)$ can compute it.

Importance of Constant-Depth Circuits

[Agrawal-Vinay]: (Hom.) circuits of size s can be converted to (hom.) $\Sigma\Pi\Sigma\Pi$ of size $s^{O(\sqrt{d})}$.

[Gupta-Kamath-Kayal-Saptharishi]: Size s circuits can be converted to $\Sigma\Pi\Sigma$ of size $s^{O(\sqrt{d})}$.

Still a LOOOOONG way to go: Lower Bounds for General Models

General Circuits and (previous best for) General ABPs [Baur-Strassen]:
Any algebraic circuit computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ wires.

Still a LOOOOONG way to go: Lower Bounds for General Models

General Circuits and (previous best for) General ABPs [Baur-Strassen]:
Any algebraic circuit computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ wires.

Restricted ABPs [Kumar]:
Any ABP with $(d+1)$ layers computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(nd)$ vertices.

Still a LOOOOONG way to go: Lower Bounds for General Models

General Circuits and (previous best for) General ABPs [Baur-Strassen]:
Any algebraic circuit computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ wires.

Restricted ABPs [Kumar]:
Any ABP with $(d+1)$ layers computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(nd)$ vertices.

General ABPs [C-Kumar-She-Volk]:

Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(nd)$ vertices.

Lower Bounds for General Formulas

General Formulas

[Kalorkoti]: Any formula computing the n^2-variate $\mathrm{Det}_n(\mathbf{x})$ requires $\Omega(n^3)$ wires.

Lower Bounds for General Formulas

General Formulas

[Kalorkoti]: Any formula computing the n^2-variate $\mathrm{Det}_n(\mathbf{x})$ requires $\Omega(n^3)$ wires.

[Shpilka-Yehudayoff] (using Kalorkoti's method): There is an n-variate multilinear polynomial such that any formula computing it requires $\Omega(n^2/\log n)$ wires.

General Formulas

[Kalorkoti]: Any formula computing the n^2-variate $\mathrm{Det}_n(\mathbf{x})$ requires $\Omega(n^3)$ wires.

[Shpilka-Yehudayoff] (using Kalorkoti's method): There is an n-variate multilinear polynomial such that any formula computing it requires $\Omega(n^2/\log n)$ wires.

[C-Kumar-She-Volk]:

$$\text{Any formula computing } \mathrm{ESym}_{n,0.1n}(\mathbf{x}) \text{ requires } \Omega(n^2) \text{ vertices.}$$

Natural Proofs

Are the proof techniques used against structured models useful against general models?

Natural Proofs

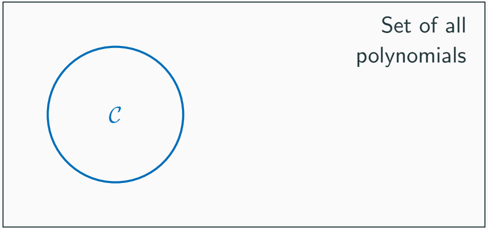
Are the proof techniques used against structured models useful against general models?

[Forbes-Shpilka-Volk, Grochow-Kumar-Saks-Saraf]: Defined Algebraically Natural Proofs.

Natural Proofs

Are the proof techniques used against structured models useful against general models?

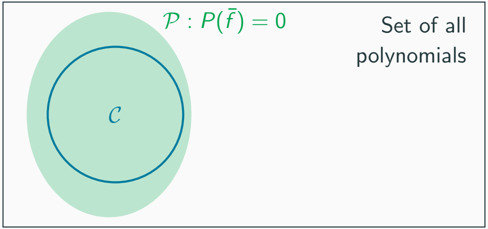
[Forbes-Shpilka-Volk, Grochow-Kumar-Saks-Saraf]: Defined Algebraically Natural Proofs.



Natural Proofs

Are the proof techniques used against structured models useful against general models?

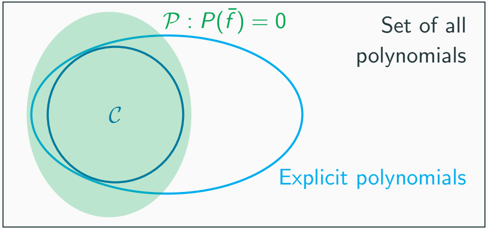
[Forbes-Shpilka-Volk, Grochow-Kumar-Saks-Saraf]: Defined Algebraically Natural Proofs.



Natural Proofs

Are the proof techniques used against structured models useful against general models?

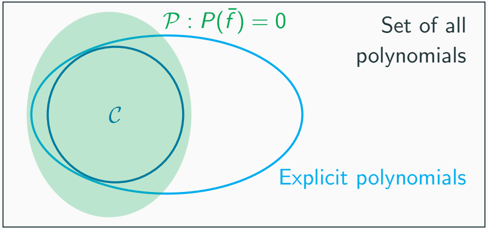
[Forbes-Shpilka-Volk, Grochow-Kumar-Saks-Saraf]: Defined Algebraically Natural Proofs.



Natural Proofs

Are the proof techniques used against structured models useful against general models?

[Forbes-Shpilka-Volk, Grochow-Kumar-Saks-Saraf]: Defined Algebraically Natural Proofs.

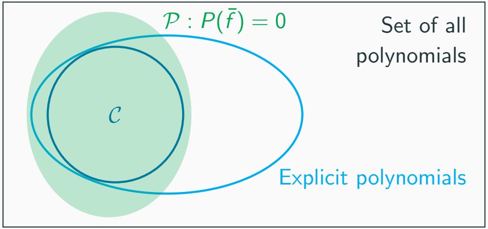


[C-Kumar-Ramya-Saptharishi-Tengse]:
Let VP′ be the polynomials in VP that additionally have $\{-1, 0, 1\}$ coefficients. Then, $\exists P(Z_1, \ldots, Z_N) \in \mathrm{VP}(N)$ such that $P(\bar{f}) = 0$ for all $f \in \mathrm{VP}'(n)$.

Natural Proofs

Are the proof techniques used against structured models useful against general models?

[Forbes-Shpilka-Volk, Grochow-Kumar-Saks-Saraf]: Defined Algebraically Natural Proofs.



[C-Kumar-Ramya-Saptharishi-Tengse]:
Let VP′ be the polynomials in VP that additionally have $\{-1, 0, 1\}$ coefficients. Then, $\exists P(Z_1, \ldots, Z_N) \in \mathrm{VP}(N)$ such that $P(\bar{f}) = 0$ for all $f \in \mathrm{VP}'(n)$.

[K-R-S-T]: Suppose the Permanent polynomial is 2^{n^ε}-hard for constant $\varepsilon > 0$. In this case, if VP has natural proofs, then there is also a natural proof P that has explicit non-roots.

The Non-Commutative Setting

$$f(x, y) = (x + y) \times (x + y)$$

The Non-Commutative Setting

$$f(x, y) = (x + y) \times (x + y) = x^2 + xy + yx + y^2$$

The Non-Commutative Setting

$$f(x, y) = (x + y) \times (x + y) = x^2 + xy + yx + y^2 \neq x^2 + 2xy + y^2$$

The Non-Commutative Setting

$$f(x, y) = (x + y) \times (x + y) = x^2 + xy + yx + y^2 \neq x^2 + 2xy + y^2$$

Lower Bounds for General Models

The Non-Commutative Setting

$$f(x, y) = (x + y) \times (x + y) = x^2 + xy + yx + y^2 \neq x^2 + 2xy + y^2$$

Lower Bounds for General Models

Circuits [Baur-Strassen]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

The Non-Commutative Setting

$$f(x, y) = (x + y) \times (x + y) = x^2 + xy + yx + y^2 \neq x^2 + 2xy + y^2$$

Lower Bounds for General Models

Circuits [Baur-Strassen]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nisan]: $2^{\Omega(n)}$ for a 2-variate, degree n polynomial in $\mathsf{VP}_{\mathsf{nc}}$.

The Non-Commutative Setting

$$f(x, y) = (x + y) \times (x + y) = x^2 + xy + yx + y^2 \neq x^2 + 2xy + y^2$$

Lower Bounds for General Models

Circuits [Baur-Strassen]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nisan]: $2^{\Omega(n)}$ for a 2-variate, degree n polynomial in $\mathsf{VP}_{\mathsf{nc}}$. So, $\mathsf{VF}_{\mathsf{nc}} \neq \mathsf{VP}_{\mathsf{nc}}$.

The Non-Commutative Setting

$$f(x, y) = (x + y) \times (x + y) = x^2 + xy + yx + y^2 \neq x^2 + 2xy + y^2$$

Lower Bounds for General Models

Circuits [Baur-Strassen]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nisan]: $2^{\Omega(n)}$ for a 2-variate, degree n polynomial in $\mathsf{VP}_{\mathsf{nc}}$. So, $\mathsf{VF}_{\mathsf{nc}} \neq \mathsf{VP}_{\mathsf{nc}}$.

But the proof is via a characterisation of the ABP complexity of non-commutative polynomials.

The Non-Commutative Setting

$$f(x, y) = (x + y) \times (x + y) = x^2 + xy + yx + y^2 \neq x^2 + 2xy + y^2$$

Lower Bounds for General Models

Circuits [Baur-Strassen]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nisan]: $2^{\Omega(n)}$ for a 2-variate, degree n polynomial in $\mathsf{VP}_{\mathsf{nc}}$. So, $\mathsf{VF}_{\mathsf{nc}} \neq \mathsf{VP}_{\mathsf{nc}}$.

But the proof is via a characterisation of the ABP complexity of non-commutative polynomials.

No lower bound known against general formulas that does not go via this characterisation.

The ABP vs Formulas Question

The Question [Nisan]: Is $\mathsf{VF}_{\mathsf{nc}} = \mathsf{VBP}_{\mathsf{nc}}$?

The ABP vs Formulas Question

The Question [Nisan]: Is $\mathrm{VF}_{\mathrm{nc}} = \mathrm{VBP}_{\mathrm{nc}}$?

Note: Every monomial in a non-commutative polynomial $f(x_1, \ldots, x_n)$ can be thought of as a word over the underlying variables $\{x_1, \ldots, x_n\}$.

The ABP vs Formulas Question

The Question [Nisan]: Is $\mathsf{VF}_{\mathsf{nc}} = \mathsf{VBP}_{\mathsf{nc}}$?

Note: Every monomial in a non-commutative polynomial $f(x_1, \ldots, x_n)$ can be thought of as a word over the underlying variables $\{x_1, \ldots, x_n\}$. We can therefore have the following definition:

Abecedarian Polynomials: Those in which every monomial is an alphabetically ordered word.

The ABP vs Formulas Question

The Question [Nisan]: Is $\mathsf{VF}_{\mathsf{nc}} = \mathsf{VBP}_{\mathsf{nc}}$?

Note: Every monomial in a non-commutative polynomial $f(x_1, \ldots, x_n)$ can be thought of as a word over the underlying variables $\{x_1, \ldots, x_n\}$. We can therefore have the following definition:

Abecedarian Polynomials: Those in which every monomial is an alphabetically ordered word.

Syntactically Abecedarian Formulas: Non-commutative formulas with a syntactic restriction that makes them naturally compute abecedarian polynomials.

The ABP vs Formulas Question

The Question [Nisan]: Is $\mathsf{VF}_{\mathsf{nc}} = \mathsf{VBP}_{\mathsf{nc}}$?

Note: Every monomial in a non-commutative polynomial $f(x_1, \ldots, x_n)$ can be thought of as a word over the underlying variables $\{x_1, \ldots, x_n\}$. We can therefore have the following definition:

Abecedarian Polynomials: Those in which every monomial is an alphabetically ordered word.

Syntactically Abecedarian Formulas: Non-commutative formulas with a syntactic restriction that makes them naturally compute abecedarian polynomials.

[Cha]: There is a tight superpolynomial separation between syntactically *abecedarian* formulas and ABPs.

Thank you !!