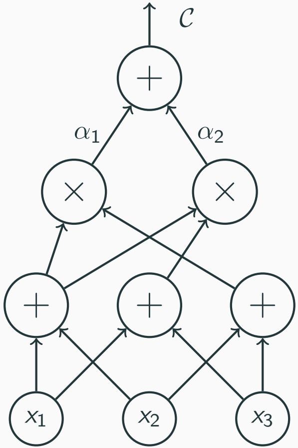
Separating ABPs and some Structured Formulas in the Non-Commutative Setting

Prerona Chatterjee

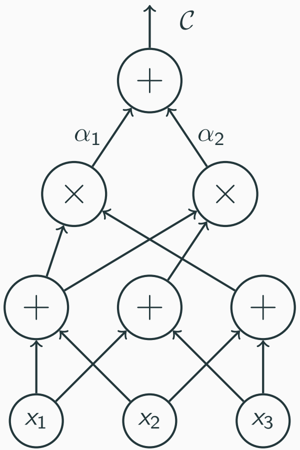
Tata Institute of Fundamental Research, Mumbai

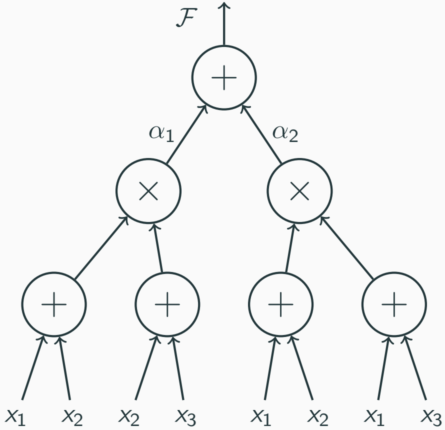
July 21, 2021

Algebraic Circuits and Formulas



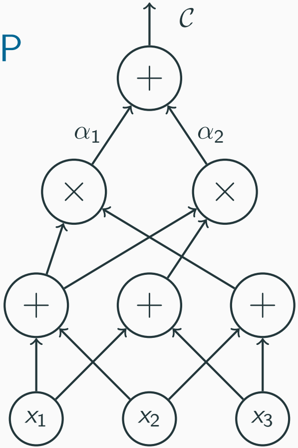
Algebraic Circuits and Formulas

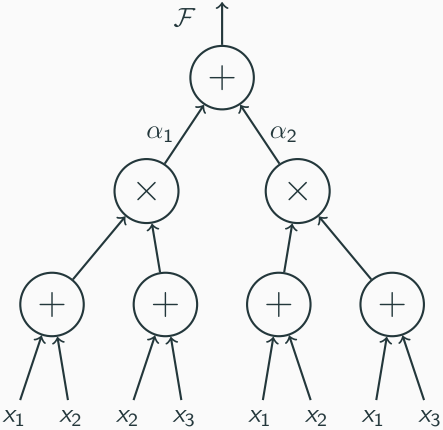




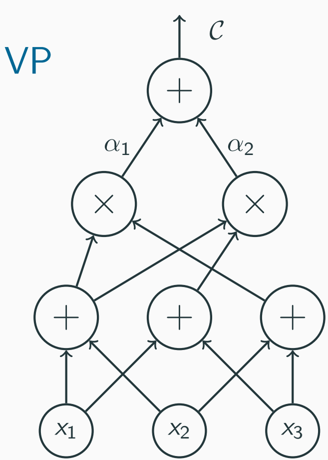
Algebraic Circuits and Formulas

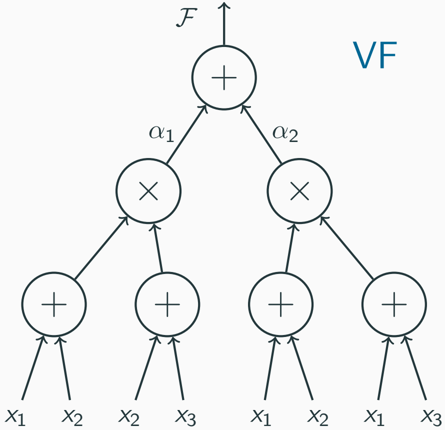
VP



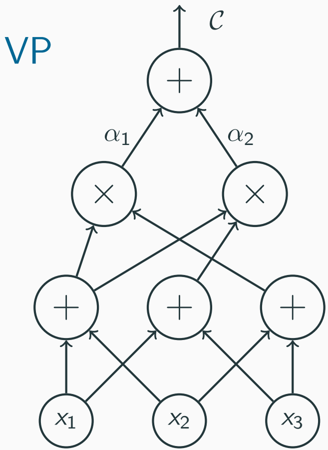


Algebraic Circuits and Formulas

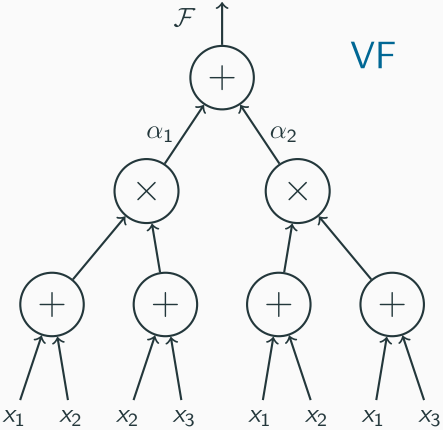




Algebraic Circuits and Formulas



VP ⊇ VF

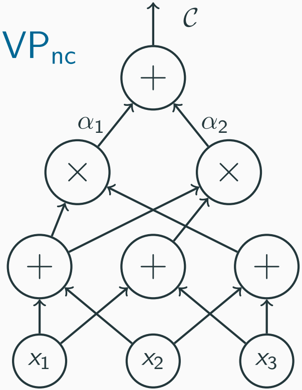


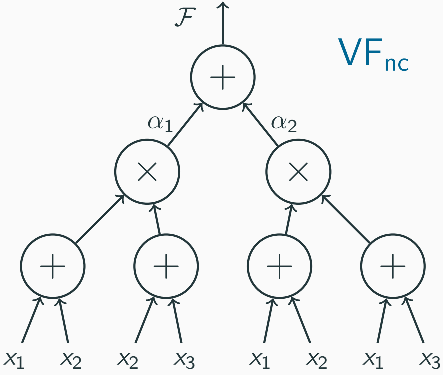
The Non-Commutative Setting

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

The Non-Commutative Setting

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

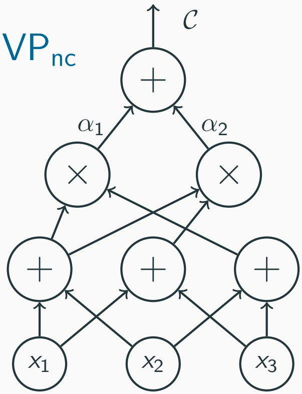




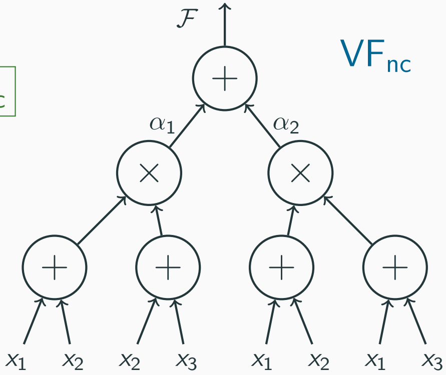
The Non-Commutative Setting

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

$\mathsf{VP}_{\mathsf{nc}}$



$\mathsf{VP}_{\mathsf{nc}} \supseteq \mathsf{VF}_{\mathsf{nc}}$



$\mathsf{VF}_{\mathsf{nc}}$

Lower Bounds for General Non-Commutative Models

Is there an explicit polynomial that is outside $\mathrm{VP}_{\mathrm{nc}}$?

Lower Bounds for General Non-Commutative Models

Is there an explicit polynomial that is outside $\mathsf{VP}_{\mathsf{nc}}$?

Circuits [Str73, BS83]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Lower Bounds for General Non-Commutative Models

Is there an explicit polynomial that is outside $\mathsf{VP}_{\mathsf{nc}}$? What about $\mathsf{VF}_{\mathsf{nc}}$?

Circuits [Str73, BS83]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Lower Bounds for General Non-Commutative Models

Is there an explicit polynomial that is outside $\mathsf{VP}_{\mathsf{nc}}$? What about $\mathsf{VF}_{\mathsf{nc}}$?

Circuits [Str73, BS83]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nis91]: $2^{\Omega(n)}$ for a 2-variate, degree n polynomial in $\mathsf{VP}_{\mathsf{nc}}$.

Lower Bounds for General Non-Commutative Models

Is there an explicit polynomial that is outside $\mathrm{VP_{nc}}$? What about $\mathrm{VF_{nc}}$?

Circuits [Str73, BS83]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nis91]: $2^{\Omega(n)}$ for a 2-variate, degree n polynomial in $\mathrm{VP_{nc}}$. So, $\mathrm{VF_{nc}} \neq \mathrm{VP_{nc}}$.

Lower Bounds for General Non-Commutative Models

Is there an explicit polynomial that is outside $\mathsf{VP}_{\mathsf{nc}}$? What about $\mathsf{VF}_{\mathsf{nc}}$?

Circuits [Str73, BS83]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nis91]: $2^{\Omega(n)}$ 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 lower bound against non-commutative Algebraic Branching Programs.

Lower Bounds for General Non-Commutative Models

Is there an explicit polynomial that is outside $\mathsf{VP}_{\mathsf{nc}}$? What about $\mathsf{VF}_{\mathsf{nc}}$?

Circuits [Str73, BS83]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nis91]: $2^{\Omega(n)}$ 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 lower bound against non-commutative Algebraic Branching Programs.

$\mathsf{VBP}_{\mathsf{nc}}$

Lower Bounds for General Non-Commutative Models

Is there an explicit polynomial that is outside $\mathsf{VP}_{\mathsf{nc}}$? What about $\mathsf{VF}_{\mathsf{nc}}$?

Circuits [Str73, BS83]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nis91]: $2^{\Omega(n)}$ 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 lower bound against non-commutative Algebraic Branching Programs.

$$\mathsf{VF}_{\mathsf{nc}} \subseteq \mathsf{VBP}_{\mathsf{nc}} \subseteq \mathsf{VP}_{\mathsf{nc}}$$

Lower Bounds for General Non-Commutative Models

Is there an explicit polynomial that is outside $\mathrm{VP_{nc}}$? What about $\mathrm{VF_{nc}}$?

Circuits [Str73, BS83]: $\Omega(n \log d)$ for an n-variate, degree d polynomial.

Formulas [Nis91]: $2^{\Omega(n)}$ for a 2-variate, degree n polynomial in $\mathrm{VP_{nc}}$. So, $\mathrm{VF_{nc}} \neq \mathrm{VP_{nc}}$.

But the proof is via a lower bound against non-commutative Algebraic Branching Programs.

$$\mathrm{VF_{nc}} \subseteq \mathrm{VBP_{nc}} \subseteq \mathrm{VP_{nc}}$$

So Nisan actually showed that $\mathrm{VBP_{nc}} \neq \mathrm{VP_{nc}}$.

The ABP vs Formula Question

The Question [Nis91]: Is $\mathsf{VF}_{\mathsf{nc}} = \mathsf{VBP}_{\mathsf{nc}}$?

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

X_1

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

X_1 X_2

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

$$X_1 = \{x_1\} \qquad X_2 = \{x_2\}$$

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

$X_1 = \{x_1\}$ $X_2 = \{x_2\}$ $x_1 x_1 x_2$

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

$X_1 = \{x_1\}$ $\quad X_2 = \{x_2\}$ $\quad x_1 x_1 x_2$ $\quad x_2$

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

$X_1 = \{x_1\}$ $X_2 = \{x_2\}$ $x_1 x_1 x_2$ x_2 $x_1 x_2 x_1$

The ABP vs Formula Question

The Question [Nis91]: Is $\mathrm{VF_{nc}} = \mathrm{VBP_{nc}}$?

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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

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

The ABP vs Formula Question

The Question [Nis91]: 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 $\{x_1, \ldots, x_n\}$.

Definitions Let $\{X_1, \ldots, X_m\}$ be a partition of the variables into buckets.

Abecedarian Polynomials: Polynomials in which every monomial has the form $X_1^* X_2^* \cdots X_m^*$.

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

Main Result:

There is a tight superpolynomial separation between *abecedarian* formulas and ABPs.

Proof Idea

The Statement: There is an explicit n^2-variate, degree-d abcd-polynomial $f_{n,d}(\mathbf{x})$ such that

Proof Idea

The Statement: There is an explicit n^2-variate, degree-d abcd-polynomial $f_{n,d}(\mathbf{x})$ such that

- abcd-ABP Upper Bound: the abcd-ABP complexity of $f_{n,d}(\mathbf{x})$ is $\Theta(nd)$;

Proof Idea

The Statement: There is an explicit n^2-variate, degree-d abcd-polynomial $f_{n,d}(\mathbf{x})$ such that

- abcd-ABP Upper Bound: the abcd-ABP complexity of $f_{n,d}(\mathbf{x})$ is $\Theta(nd)$;
- abcd-Formula Lower Bound: the abcd-formula complexity of $f_{n,\log n}(\mathbf{x})$ is $n^{\Theta(\log\log n)}$.

Proof Idea

The Statement: There is an explicit n^2-variate, degree-d abcd-polynomial $f_{n,d}(\mathbf{x})$ such that

- abcd-ABP Upper Bound: the abcd-ABP complexity of $f_{n,d}(\mathbf{x})$ is $\Theta(nd)$;
- abcd-Formula Lower Bound: the abcd-formula complexity of $f_{n,\log n}(\mathbf{x})$ is $n^{\Theta(\log\log n)}$.

The Proof Idea:

Proof Idea

The Statement: There is an explicit n^2-variate, degree-d abcd-polynomial $f_{n,d}(\mathbf{x})$ such that

- abcd-ABP Upper Bound: the abcd-ABP complexity of $f_{n,d}(\mathbf{x})$ is $\Theta(nd)$;
- abcd-Formula Lower Bound: the abcd-formula complexity of $f_{n,\log n}(\mathbf{x})$ is $n^{\Theta(\log \log n)}$.

The Proof Idea:

4. Use known lower bound against homogeneous multilinear formulas [HY11].

Proof Idea

The Statement: There is an explicit n^2-variate, degree-d abcd-polynomial $f_{n,d}(\mathbf{x})$ such that

- abcd-ABP Upper Bound: the abcd-ABP complexity of $f_{n,d}(\mathbf{x})$ is $\Theta(nd)$;
- abcd-Formula Lower Bound: the abcd-formula complexity of $f_{n,\log n}(\mathbf{x})$ is $n^{\Theta(\log\log n)}$.

The Proof Idea:

1. Use low degree to make the abcd-formula structured.

4. Use known lower bound against homogeneous multilinear formulas [HY11].

Proof Idea

The Statement: There is an explicit n^2-variate, degree-d abcd-polynomial $f_{n,d}(\mathbf{x})$ such that

- abcd-ABP Upper Bound: the abcd-ABP complexity of $f_{n,d}(\mathbf{x})$ is $\Theta(nd)$;
- abcd-Formula Lower Bound: the abcd-formula complexity of $f_{n,\log n}(\mathbf{x})$ is $n^{\Theta(\log \log n)}$.

The Proof Idea:

1. Use low degree to make the abcd-formula structured.
2. Use the structured formula to amplify degree while keeping the structure intact.

4. Use known lower bound against homogeneous multilinear formulas [HY11].

Proof Idea

The Statement: There is an explicit n^2-variate, degree-d abcd-polynomial $f_{n,d}(\mathbf{x})$ such that

- abcd-ABP Upper Bound: the abcd-ABP complexity of $f_{n,d}(\mathbf{x})$ is $\Theta(nd)$;
- abcd-Formula Lower Bound: the abcd-formula complexity of $f_{n,\log n}(\mathbf{x})$ is $n^{\Theta(\log \log n)}$.

The Proof Idea:

1. Use low degree to make the abcd-formula structured.
2. Use the structured formula to amplify degree while keeping the structure intact.
3. Convert the structured abcd-formula into a homogeneous multilinear formula.
4. Use known lower bound against homogeneous multilinear formulas [HY11].

Concluding Remarks

Studying abcd-formulas might be enough: For any n-variate polynomial that is abecedarian with respect to a partition of size $O(\log n)$, if it can be computed efficiently by formulas, then they can also be computed efficiently by abcd-formulas.

Concluding Remarks

Studying abcd-formulas might be enough: For any n-variate polynomial that is abecedarian with respect to a partition of size $O(\log n)$, if it can be computed efficiently by formulas, then they can also be computed efficiently by abcd-formulas.

Super-polynomial separation between homogeneous formulas and ABPs [LST21]: There is an n^2-variate degree-n polynomial that can be computed by a homogeneous ABP of size $\Theta(n^2)$ but any homogeneous formula computing it must have size $n^{\Omega(\log \log n)}$.

Concluding Remarks

Studying abcd-formulas might be enough: For any n-variate polynomial that is abecedarian with respect to a partition of size $O(\log n)$, if it can be computed efficiently by formulas, then they can also be computed efficiently by abcd-formulas.

Super-polynomial separation between homogeneous formulas and ABPs [LST21]: There is an n^2-variate degree-n polynomial that can be computed by a homogeneous ABP of size $\Theta(n^2)$ but any homogeneous formula computing it must have size $n^{\Omega(\log \log n)}$.

Question: Can ideas from [LST21] and this paper be combined to answer Nisan's question?

Thank You !

Webpage: preronac.bitbucket.io

Email: prerona.ch@gmail.com