Lower Bounds Against Non-Commutative Models of Algebraic Computation

Prerona Chatterjee (joint work with Pavel Hrubeš)
Tel Aviv University

January 24, 2023

The Question

Objects of study: Polynomials over some underlying field.

The Question

Objects of study: Polynomials over some underlying field.

$$f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$$

The Question

Objects of study: Polynomials over some underlying field.

$$f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$$

Question: Can it be computed efficiently using the given model of computation?

The Question

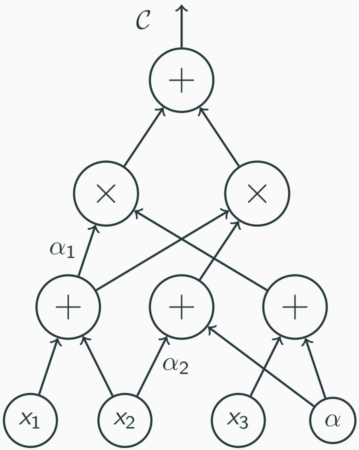
Objects of study: Polynomials over some underlying field.

$$f(\mathbf{x}) \in \mathbb{F}[\mathbf{x}]$$

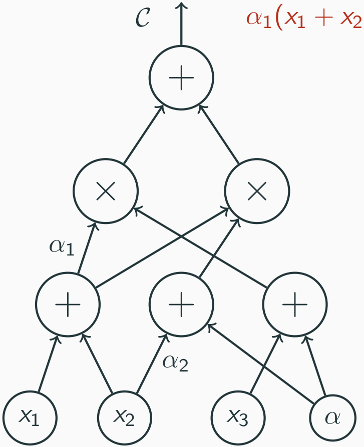
Question: Can it be computed efficiently using the given model of computation?

Model of interest today: Algebraic Circuits

Algebraic Circuits

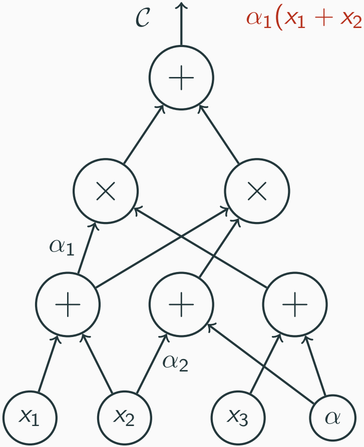


Algebraic Circuits



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

Algebraic Circuits



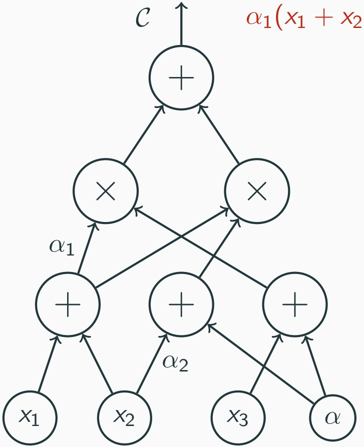
$\alpha_1(x_1 + x_2)(x_3 + \alpha) + (x_1 + x_2)(\alpha_2 x_2 + \alpha)$

Objects of Study

Polynomials over n variables of degree d.

Algebraic Circuits

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



Objects of Study

Polynomials over n variables of degree d.

Central Question: Find explicit polynomials that cannot be computed by circuits of size poly(n,d).

What is known?

A lot...

What is known?

A lot...

Super-polynomial Lower Bound Against Constant Depth Circuits

[Nisan-Wigderson], . . . , [Gupta-Kamath-Kayal-Saptharishi], . . . , [Kumar-Saraf], . . .

What is known?

A lot...

Super-polynomial Lower Bound Against Constant Depth Circuits

[Nisan-Wigderson], . . . , [Gupta-Kamath-Kayal-Saptharishi], . . . , [Kumar-Saraf], . . .

[Limaye-Srinivasan-Tavenas]: There is an explicit family of polynomials $\{f_{n,d}(\mathbf{x})\}_{n,d}$ such that any constant depth-Δ circuit computing $f_{n,d}(\mathbf{x})$ has must have size $n^{\Omega(d^{\frac{1}{4^{\Delta}}})}$.

What is known?

A lot...

Super-polynomial Lower Bound Against Constant Depth Circuits

[Nisan-Wigderson], . . . , [Gupta-Kamath-Kayal-Saptharishi], . . . , [Kumar-Saraf], . . .

[Limaye-Srinivasan-Tavenas]: There is an explicit family of polynomials $\{f_{n,d}(\mathbf{x})\}_{n,d}$ such that any constant depth-Δ circuit computing $f_{n,d}(\mathbf{x})$ has must have size $n^{\Omega(d^{\frac{1}{4^\Delta}})}$.

This is especially cool in the algebraic world.

Depth reduction results exist, which show that "good enough" super-polynomial lower bounds against constant depth circuits imply super-polynomial lower bounds against general circuits.

Ok! But what about general circuits?

Unfortunately, very little... :(

Ok! But what about general circuits?

Unfortunately, very little... :(

[Baur-Strassen]: Any algebraic circuit computing $\sum_{i=1}^{n} x_i^d$ has size at least $\Omega(n \log d)$.

Ok! But what about general circuits?

Unfortunately, very little... :(

[Baur-Strassen]: Any algebraic circuit computing $\sum_{i=1}^{n} x_i^d$ has size at least $\Omega(n \log d)$.

But do there exist "hard" polynomials?

Ok! But what about general circuits?

Unfortunately, very little... :(

[Baur-Strassen]: Any algebraic circuit computing $\sum_{i=1}^{n} x_i^d$ has size at least $\Omega(n \log d)$.

But do there exist "hard" polynomials? Yes! In fact a random polynomial is hard!

Ok! But what about general circuits?

Unfortunately, very little... :(

[Baur-Strassen]: Any algebraic circuit computing $\sum_{i=1}^{n} x_i^d$ has size at least $\Omega(n \log d)$.

But do there exist "hard" polynomials? Yes! In fact a random polynomial is hard!

[Hrubeš-Yehudayoff]: Over any field, most zero-one coefficient polynomials over n variables of degree d require circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$ to compute it.

Ok! But what about general circuits?

Unfortunately, very little... :(

[Baur-Strassen]: Any algebraic circuit computing $\sum_{i=1}^{n} x_i^d$ has size at least $\Omega(n \log d)$.

But do there exist "hard" polynomials? Yes! In fact a random polynomial is hard!

[Hrubeš-Yehudayoff]: Over any field, most zero-one coefficient polynomials over n variables of degree d require circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$ to compute it.

Find an explicit polynomial that is hard!

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

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

Non-Commutative Circuits: The multiplication gates, additionally, respect the order.

The Non-Commutative Setting

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

Non-Commutative Circuits: The multiplication gates, additionally, respect the order.

Can we do something better in this setting?

We should be able to...

[Nisan]: Exponential lower bound against non-commutative ABPs and formulas.

We should be able to...

[Nisan]: Exponential lower bound against non-commutative ABPs and formulas.

The best known lower bound against general ABPs, formulas is quadratic [C-Kumar-She-Volk].

We should be able to...

[Nisan]: Exponential lower bound against non-commutative ABPs and formulas.

The best known lower bound against general ABPs, formulas is quadratic [C-Kumar-She-Volk].

[Tavenas-Limaye-Srinivasan]: Super-polynomial separation between homogeneous non-commutative formulas and ABPs.

We should be able to...

[Nisan]: Exponential lower bound against non-commutative ABPs and formulas.

The best known lower bound against general ABPs, formulas is quadratic [C-Kumar-She-Volk].

[Tavenas-Limaye-Srinivasan]: Super-polynomial separation between homogeneous non-commutative formulas and ABPs.

No such result known in the general setting.

We should be able to...

[Nisan]: Exponential lower bound against non-commutative ABPs and formulas.

The best known lower bound against general ABPs, formulas is quadratic [C-Kumar-She-Volk].

[Tavenas-Limaye-Srinivasan]: Super-polynomial separation between homogeneous non-commutative formulas and ABPs.

No such result known in the general setting.

[Tavenas-Limaye-Srinivisan]: There is an explicit family of polynomials $\{f_{n,d}(\mathbf{x})\}_{n,d}$ such that any constant depth-Δ homogeneous circuit computing $f_{n,d}(\mathbf{x})$ must have size $n^{\Omega(d^{\frac{1}{\Delta}})}$.

Our Main Result

The best lower bound against NC circuits continues to be $\Omega(n \log d)$.

Our Main Result

The best lower bound against NC circuits continues to be $\Omega(n \log d)$.

Can we at least do better in the homogeneous case?

Our Main Result

The best lower bound against NC circuits continues to be $\Omega(n \log d)$.

Can we at least do better in the homogeneous case?

Theorem: Any homogeneous non-commutative circuit computing

$$\mathrm{OSym}_{n,d} = \sum_{1 \le i_1 < \cdots < i_d \le n} x_{i_1} \cdots x_{i_d}$$

has size $\Omega(nd')$ where $d' = \min(d, n-d)$.

A simple proof of an obvious fact

Obvious Fact: Any homogeneous circuit computing $x_1 \cdots x_d$ must have size $\Omega(d)$.

A simple proof of an obvious fact

Obvious Fact: Any homogeneous circuit computing $x_1 \cdots x_d$ must have size $\Omega(d)$.

f: Homogeneous non-commutative polynomial of degree d.

A simple proof of an obvious fact

Obvious Fact: Any homogeneous circuit computing $x_1 \cdots x_d$ must have size $\Omega(d)$.

f: Homogeneous non-commutative polynomial of degree d.

$f^{(i)}$: Polynomial got from f by setting variables in positions other than i, $i+1$ to 1.

A simple proof of an obvious fact

Obvious Fact: Any homogeneous circuit computing $x_1 \cdots x_d$ must have size $\Omega(d)$.

f: Homogeneous non-commutative polynomial of degree d.

$f^{(i)}$: Polynomial got from f by setting variables in positions other than i, $i+1$ to 1.

Example: $f = x_1 \cdots x_d$

A simple proof of an obvious fact

Obvious Fact: Any homogeneous circuit computing $x_1 \cdots x_d$ must have size $\Omega(d)$.

f: Homogeneous non-commutative polynomial of degree d.

$f^{(i)}$: Polynomial got from f by setting variables in positions other than i, $i+1$ to 1.

Example: $f = x_1 \cdots x_d \implies f^{(0)} = x_1, \; f^{(d)} = x_d, \; f^{(i)} = x_i x_{i+1}$ for every $1 \leq i \leq d-1$.

A simple proof of an obvious fact

f: Homogeneous non-commutative polynomial of degree d.

$f^{(i)}$: Polynomial got from f by setting variables in positions other than i, $i+1$ to 1.

A simple proof of an obvious fact

f: Homogeneous non-commutative polynomial of degree d.
$f^{(i)}$: Polynomial got from f by setting variables in positions other than i, $i+1$ to 1.

$$\mu(f) = \text{rank}\left(\text{span}_{\mathbb{F}}\left(\left\{f^{(0)}, f^{(1)}, \ldots, f^{(d)}\right\}\right)\right).$$

A simple proof of an obvious fact

f: Homogeneous non-commutative polynomial of degree d.
$f^{(i)}$: Polynomial got from f by setting variables in positions other than i, $i+1$ to 1.

$$\mu(f) = \text{rank}\left(\text{span}_{\mathbb{F}}\left(\left\{f^{(0)}, f^{(1)}, \ldots, f^{(d)}\right\}\right)\right).$$

$\mathcal{C}$: Homogeneous non-commutative circuit.

$$\mu(\mathcal{C}) = \text{rank}\left(\text{span}_{\mathbb{F}}\left(\bigcup_{g \in \mathcal{C}}\left\{g^{(0)}, g^{(1)}, \ldots, g^{(d)}\right\}\right)\right).$$

A simple proof of an obvious fact

$g^{(i)}$: Polynomial got from g by setting variables in positions other than i, $i+1$ to 1.

$$\mu(g) = \operatorname{rank}\left(\operatorname{span}_{\mathbb{F}}\left(\left\{g^{(0)}, g^{(1)}, \ldots, g^{(d)}\right\}\right)\right).$$

A simple proof of an obvious fact

$g^{(i)}$: Polynomial got from g by setting variables in positions other than i, $i+1$ to 1.

$$\mu(g) = \text{rank}\left(\text{span}_{\mathbb{F}}\left(\left\{g^{(0)}, g^{(1)}, \ldots, g^{(d)}\right\}\right)\right).$$

Claim: If $\mathcal{C}$ is a homogeneous non-commutative circuit of size s, then $\mu(\mathcal{C}) \leq s+1$.

A simple proof of an obvious fact

$g^{(i)}$: Polynomial got from g by setting variables in positions other than i, $i+1$ to 1.

$$\mu(g) = \text{rank}\left(\text{span}_{\mathbb{F}}\left(\left\{g^{(0)}, g^{(1)}, \ldots, g^{(d)}\right\}\right)\right).$$

Claim: If $\mathcal{C}$ is a homogeneous non-commutative circuit of size s, then $\mu(\mathcal{C}) \leq s+1$.

Proof Sketch: Use induction.

A simple proof of an obvious fact

$g^{(i)}$: Polynomial got from g by setting variables in positions other than i, $i+1$ to 1.

$$\mu(g) = \text{rank}\left(\text{span}_{\mathbb{F}}\left(\left\{g^{(0)}, g^{(1)}, \ldots, g^{(d)}\right\}\right)\right).$$

Claim: If $\mathcal{C}$ is a homogeneous non-commutative circuit of size s, then $\mu(\mathcal{C}) \leq s+1$.

Proof Sketch: Use induction. No change in rank at $+$ gates.

A simple proof of an obvious fact

$g^{(i)}$: Polynomial got from g by setting variables in positions other than i, $i+1$ to 1.

$$\mu(g) = \operatorname{rank}\left(\operatorname{span}_{\mathbb{F}}\left(\left\{g^{(0)}, g^{(1)}, \ldots, g^{(d)}\right\}\right)\right).$$

Claim: If $\mathcal{C}$ is a homogeneous non-commutative circuit of size s, then $\mu(\mathcal{C}) \leq s+1$.

Proof Sketch: Use induction. No change in rank at $+$ gates.
Rank can increase by at most 1 at $\times$ gates.

A simple proof of an obvious fact

$g^{(i)}$: Polynomial got from g by setting variables in positions other than i, $i+1$ to 1.

$$\mu(g) = \text{rank}\left(\text{span}_{\mathbb{F}}\left(\left\{g^{(0)}, g^{(1)}, \ldots, g^{(d)}\right\}\right)\right).$$

Claim: If $\mathcal{C}$ is a homogeneous non-commutative circuit of size s, then $\mu(\mathcal{C}) \leq s+1$.

Proof Sketch: Use induction. No change in rank at $+$ gates.
Rank can increase by at most 1 at $\times$ gates.

We already saw that for $f = x_1 \cdots x_d$, $\mu(f) = d+1$.

A simple proof of an obvious fact

$g^{(i)}$: Polynomial got from g by setting variables in positions other than i, $i+1$ to 1.

$$\mu(g) = \text{rank}\left(\text{span}_{\mathbb{F}}\left(\left\{g^{(0)}, g^{(1)}, \ldots, g^{(d)}\right\}\right)\right).$$

Claim: If $\mathcal{C}$ is a homogeneous non-commutative circuit of size s, then $\mu(\mathcal{C}) \leq s+1$.

Proof Sketch: Use induction. No change in rank at $+$ gates.
Rank can increase by at most 1 at $\times$ gates.

We already saw that for $f = x_1 \cdots x_d$, $\mu(f) = d+1$. Therefore $s \geq d$.

Using it to prove a "not so obvious" fact

Theorem: There exists an explicit monomial over $\{x_0, x_1\}$ of degree d such that any homogeneous non-commutative circuit computing it must have size $\Omega\left(\frac{d}{\log d}\right)$.

Using it to prove a “not so obvious” fact

Theorem: There exists an explicit monomial over $\{x_0, x_1\}$ of degree d such that any homogeneous non-commutative circuit computing it must have size $\Omega\left(\frac{d}{\log d}\right)$.

The tweak: For a homogeneous non-commutative polynomial f of degree d, define

$f^{(i)}$ by setting, in f, variables in positions other than $\{i, i+1, \ldots i + \log d\}$ to 1.

Using it to prove a “not so obvious” fact

Theorem: There exists an explicit monomial over $\{x_0, x_1\}$ of degree d such that any homogeneous non-commutative circuit computing it must have size $\Omega\left(\frac{d}{\log d}\right)$.

The tweak: For a homogeneous non-commutative polynomial f of degree d, define

$f^{(i)}$ by setting, in f, variables in positions other than $\{i, i+1, \ldots i+\log d\}$ to 1.

In this case, if $\mathcal{C}$ is a homogeneous non-commutative circuit of size s, then $\mu_\ell(\mathcal{C}) \leq O(s \log d)$.

Using it to prove a “not so obvious” fact

Theorem: There exists an explicit monomial over $\{x_0, x_1\}$ of degree d such that any homogeneous non-commutative circuit computing it must have size $\Omega\left(\frac{d}{\log d}\right)$.

The tweak: For a homogeneous non-commutative polynomial f of degree d, define

$f^{(i)}$ by setting, in f, variables in positions other than $\{i, i+1, \ldots i+\log d\}$ to 1.

In this case, if $\mathcal{C}$ is a homogeneous non-commutative circuit of size s, then $\mu_\ell(\mathcal{C}) \leq O(s \log d)$.

Therefore all we need is a monomial, f, over $\{x_0, x_1\}$ of degree d such that $\mu_\ell(f) \geq \Omega(d)$.

Using it to prove a "not so obvious" fact

de Bruijn Sequence (of order $\log d$**)**: It is a cyclic sequence in the alphabet $\{0, 1\}$ in which every string of length $\log d$, occurs exactly once as a substring.

Using it to prove a “not so obvious” fact

de Bruijn Sequence (of order $\log d$): It is a cyclic sequence in the alphabet $\{0, 1\}$ in which every string of length $\log d$, occurs exactly once as a substring.

Fact: There is a length-d de Bruijn sequence of order $\log d$.

Using it to prove a “not so obvious” fact

de Bruijn Sequence (of order $\log d$): It is a cyclic sequence in the alphabet $\{0, 1\}$ in which every string of length $\log d$, occurs exactly once as a substring.

Fact: There is a length-d de Bruijn sequence of order $\log d$.

Therefore, if B_d is the monomial corresponding to this de Bruijn sequence, then $\mu(B_d) \geq \Omega(d)$.

Using it to prove a "not so obvious" fact

de Bruijn Sequence (of order $\log d$**)**: It is a cyclic sequence in the alphabet $\{0, 1\}$ in which every string of length $\log d$, occurs exactly once as a substring.

Fact: There is a length-d de Bruijn sequence of order $\log d$.

Therefore, if B_d is the monomial corresponding to this de Bruijn sequence, then $\mu(B_d) \geq \Omega(d)$.

How can non-homogeneity possibly help in computing a monomial?

Using it to prove a “not so obvious” fact

de Bruijn Sequence (of order $\log d$**)**: It is a cyclic sequence in the alphabet $\{0, 1\}$ in which every string of length $\log d$, occurs exactly once as a substring.

Fact: There is a length-d de Bruijn sequence of order $\log d$.

Therefore, if B_d is the monomial corresponding to this de Bruijn sequence, then $\mu(B_d) \geq \Omega(d)$.

How can non-homogeneity possibly help in computing a monomial?

Question: Can we prove the same lower bound against general non-commutative circuits?

Getting back to the main result

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

Getting back to the main result

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

- Suppose a similar result was true in the homogeneous non-commutative setting.

Getting back to the main result

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

- Suppose a similar result was true in the homogeneous non-commutative setting.
- Suppose there is an n-variate, degree-d polynomial f such that

$$\mu(\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}) \geq \Omega(nd).$$

Getting back to the main result

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

- Suppose a similar result was true in the homogeneous non-commutative setting.
- Suppose there is an n-variate, degree-d polynomial f such that

$$\mu(\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}) \geq \Omega(nd).$$

Then we would have an $\Omega(nd)$ lower bound against homogeneous non-commutative circuits.

Getting back to the main result

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

- Suppose a similar result was true in the homogeneous non-commutative setting.
- Suppose there is an n-variate, degree-d polynomial f such that

$$\mu(\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}) \geq \Omega(nd).$$

Then we would have an $\Omega(nd)$ lower bound against homogeneous non-commutative circuits.

Note: $f = x_1 B_d(x_0^{(1)}, x_1^{(1)}) + \cdots + x_n B_d(x_0^{(n)}, x_1^{(n)})$ already (almost) has the required property.

Getting back to the main result

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

- A similar result is true in the homogeneous non-commutative setting.
- Suppose there is an n-variate, degree-d polynomial f such that

$$\mu(\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}) \geq \Omega(nd).$$

Then we would have an $\Omega(nd)$ lower bound against homogeneous non-commutative circuits.

Getting back to the main result

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

- A similar result is true in the homogeneous non-commutative setting.
- There is an n-variate, degree-d polynomial f such that

$$\mu(\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}) \geq \Omega(nd).$$

Then we would have an $\Omega(nd)$ lower bound against homogeneous non-commutative circuits.

Getting back to the main result

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

- A similar result is true in the homogeneous non-commutative setting.
- There is an n-variate, degree-d polynomial f such that

$$\mu(\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}) \geq \Omega(nd).$$

Therefore we have an $\Omega(nd)$ lower bound against homogeneous non-commutative circuits.

Getting back to the main result

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

- A similar result is true in the homogeneous non-commutative setting.
- There is an n-variate, degree-d polynomial f such that

$$\mu(\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}) \geq \Omega(nd).$$

Therefore we have an $\Omega(nd)$ lower bound against homogeneous non-commutative circuits.

Note: f has a non-homogeneous non-commutative circuit of size $O(n \log^2 d)$.

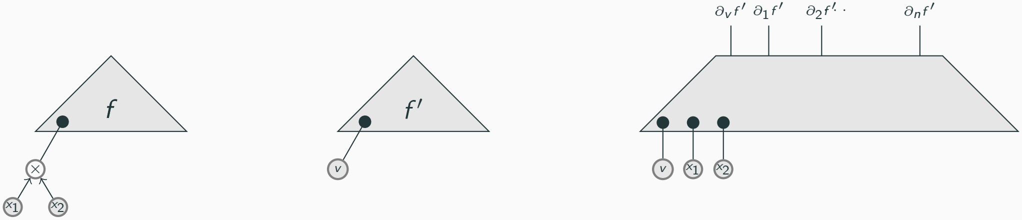
Proof of [Baur-Strassen]

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

Proof of [Baur-Strassen]

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

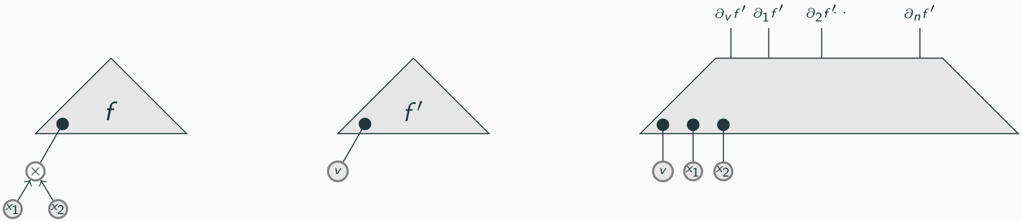
Step 1:



Proof of [Baur-Strassen]

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

Step 1:

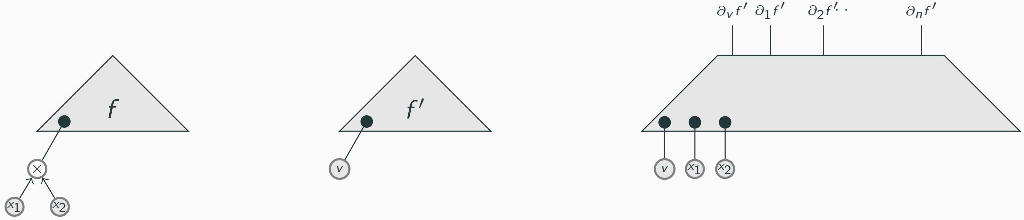


Step 2: Write each of $\{\partial_i f\}_i$ using $\partial_v f'$ and $\{\partial_i f'\}_i$.

Proof of [Baur-Strassen]

[Baur-Strassen]: If there is a circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

Step 1:



Step 2: Write each of $\{\partial_i f\}_i$ using $\partial_v f'$ and $\{\partial_i f'\}_i$. Add (the ≤ 10 extra) edges accordingly.

Making [Baur-Strassen] work in the homogeneous setting

Target: If there is a homogeneous circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a homogeneous circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

Making [Baur-Strassen] work in the homogeneous setting

Target: If there is a homogeneous circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a homogeneous circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

Weights: $w_i = \mathrm{wt}(x_i)$.

Making [Baur-Strassen] work in the homogeneous setting

Target: If there is a homogeneous circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a homogeneous circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

Weights: $w_i = \mathrm{wt}(x_i)$. Given $\mathbf{w} = (w_1, \ldots, w_n)$, define $\mathbf{w}$-homogeneous.

Making [Baur-Strassen] work in the homogeneous setting

Target: If there is a homogeneous circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a homogeneous circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \dots, \partial_{x_n} f\}$.

Weights: $w_i = \text{wt}(x_i)$. Given $\mathbf{w} = (w_1, \dots, w_n)$, define $\mathbf{w}$-homogeneous.

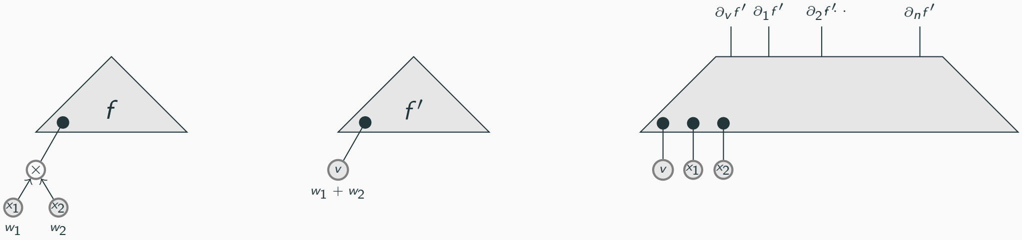
Lemma: If there is a $\mathbf{w}$-homogeneous circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a $\mathbf{w}$-homogeneous circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \dots, \partial_{x_n} f\}$.

Making [Baur-Strassen] work in the homogeneous setting

Target: If there is a homogeneous circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a homogeneous circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.

Weights: $w_i = \mathrm{wt}(x_i)$. Given $\mathbf{w} = (w_1, \ldots, w_n)$, define $\mathbf{w}$-homogeneous.

Lemma: If there is a $\mathbf{w}$-homogeneous circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a $\mathbf{w}$-homogeneous circuit of size at most $5s$ that simultaneously compute $\{\partial_{x_1} f, \partial_{x_2} f, \ldots, \partial_{x_n} f\}$.



Making [Baur-Strassen] work in the non-commutative setting

Formal derivatives (with respect to the first position)

Given a polynomial f and a variable x, f can be uniquely written as

$$f = x \cdot f_0 + f_1$$

where no monomial in f_1 contains x in the first position.

Making [Baur-Strassen] work in the non-commutative setting

Formal derivatives (with respect to the first position)

Given a polynomial f and a variable x, f can be uniquely written as

$$f = x \cdot f_0 + f_1$$

where no monomial in f_1 contains x in the first position.

We can then define the formal derivative to be $\partial_{1,x} f := f_0$.

Making [Baur-Strassen] work in the non-commutative setting

Formal derivatives (with respect to the first position)

Given a polynomial f and a variable x, f can be uniquely written as

$$f = x \cdot f_0 + f_1$$

where no monomial in f_1 contains x in the first position.

We can then define the formal derivative to be $\partial_{1,x} f := f_0$.

Chain rules can be defined formally as well.

Making [Baur-Strassen] work in the non-commutative setting

Formal derivatives (with respect to the first position)

Given a polynomial f and a variable x, f can be uniquely written as

$$f = x \cdot f_0 + f_1$$

where no monomial in f_1 contains x in the first position.

We can then define the formal derivative to be $\partial_{1,x} f := f_0$.

Chain rules can be defined formally as well.

Lemma: If there is a homogeneous NC circuit of size s computing $f \in \mathbb{F}[\mathbf{x}]$, then there is a homogeneous NC circuit of size at most $5s$ that simultaneously compute $\{\partial_{1,x_1} f, \ldots, \partial_{1,x_n} f\}$.

Where are we at?

$\mathcal{C}$: Homogeneous circuit of size s computing f.

Where are we at?

$\mathcal{C}$: Homogeneous circuit of size s computing f.

$\mathcal{C}'$: Homogeneous circuit of size $5s$ that simultaneously compute $\{\partial_{1,x_1} f, \partial_{1,x_2} f, \ldots, \partial_{1,x_n} f\}$.

Where are we at?

$\mathcal{C}$: Homogeneous circuit of size s computing f.

$\mathcal{C}'$: Homogeneous circuit of size $5s$ that simultaneously compute $\{\partial_{1,x_1} f, \partial_{1,x_2} f, \ldots, \partial_{1,x_n} f\}$.

$$\mu(\mathcal{C}') \leq 5s + 1$$

Where are we at?

$\mathcal{C}$: Homogeneous circuit of size s computing f.

$\mathcal{C}'$: Homogeneous circuit of size $5s$ that simultaneously compute $\{\partial_{1,x_1} f, \partial_{1,x_2} f, \ldots, \partial_{1,x_n} f\}$.

$$\mu(\mathcal{C}') \leq 5s + 1$$

Task: Find n-variate, degree-d f such that if $\text{out}(\mathcal{C}') = \{\partial_{1,x_1} f, \partial_{1,x_2} f, \ldots, \partial_{1,x_n} f\}$, then

$$\mu(\text{out}(\mathcal{C}')) = \Omega(nd).$$

Where are we at?

$\mathcal{C}$: Homogeneous circuit of size s computing f.

$\mathcal{C}'$: Homogeneous circuit of size $5s$ that simultaneously compute $\{\partial_{1,x_1} f, \partial_{1,x_2} f, \ldots, \partial_{1,x_n} f\}$.

$$\mu(\mathcal{C}') \leq 5s + 1$$

Task: Find n-variate, degree-d f such that if $\text{out}(\mathcal{C}') = \{\partial_{1,x_1} f, \partial_{1,x_2} f, \ldots, \partial_{1,x_n} f\}$, then

$$\mu(\text{out}(\mathcal{C}')) = \Omega(nd).$$

Use the fact that $\mu(\text{out}(\mathcal{C}')) \leq \mu(\mathcal{C}')$ to complete the proof.

Recalling the measure and the polynomial

$f_1, \ldots, f_n$: Homogeneous non-commutative polynomials of degree d.

Recalling the measure and the polynomial

$f_1, \ldots, f_n$: Homogeneous non-commutative polynomials of degree d.

$f_i^{(j)}$: Polynomial got from f_i by setting variables in positions other than j, $j+1$ to 1.

Recalling the measure and the polynomial

$f_1, \ldots, f_n$: Homogeneous non-commutative polynomials of degree d.

$f_i^{(j)}$: Polynomial got from f_i by setting variables in positions other than j, $j+1$ to 1.

$$\mu(f_1, \ldots f_n) = \text{rank}\left(\text{span}_{\mathbb{F}}\left(\bigcup_{i=1}^{n}\left\{f_i^{(0)}, f_i^{(1)}, \ldots, f_i^{(d)}\right\}\right)\right).$$

Recalling the measure and the polynomial

$f_1, \ldots, f_n$: Homogeneous non-commutative polynomials of degree d.

$f_i^{(j)}$: Polynomial got from f_i by setting variables in positions other than j, $j+1$ to 1.

$$\mu(f_1, \ldots f_n) = \mathrm{rank}\left(\mathrm{span}_{\mathbb{F}}\left(\bigcup_{i=1}^{n}\left\{f_i^{(0)}, f_i^{(1)}, \ldots, f_i^{(d)}\right\}\right)\right).$$

The hard polynomial

$$\mathrm{OSym}_{n,\frac{n}{2}+1}(\mathbf{x}) = \sum_{1\le i_1<\cdots<i_{\frac{n}{2}+1}\le n} x_{i_1}x_{i_2}\cdots x_{i_{1+\frac{n}{2}}}$$

Polynomial with a large measure

$$f = \mathrm{OSym}_{n,\frac{n}{2}+1}(\mathbf{x}) = \sum_{1 \le i_1 < \cdots < i_{\frac{n}{2}+1} \le n} x_{i_1} x_{i_2} \cdots x_{i_{1+\frac{n}{2}}}$$

Polynomial with a large measure

$$f = \mathrm{OSym}_{n,\frac{n}{2}+1}(\mathbf{x}) = \sum_{1 \leq i_1 < \cdots < i_{\frac{n}{2}+1} \leq n} x_{i_1} x_{i_2} \cdots x_{i_{1+\frac{n}{2}}}$$

$$f_i = \partial_{1,x_i} f = \sum_{i < i_1 < \cdots < i_{\frac{n}{2}} \leq n} x_{i_1} x_{i_2} \cdots x_{i_{\frac{n}{2}}}$$

Polynomial with a large measure

$$f = \mathrm{OSym}_{n,\frac{n}{2}+1}(\mathbf{x}) = \sum_{1 \le i_1 < \cdots < i_{\frac{n}{2}+1} \le n} x_{i_1} x_{i_2} \cdots x_{i_{1+\frac{n}{2}}}$$

$$f_i = \partial_{1,x_i} f = \sum_{i < i_1 < \cdots < i_{\frac{n}{2}} \le n} x_{i_1} x_{i_2} \cdots x_{i_{\frac{n}{2}}}$$

Claim: The following set of size $\Omega(n^2)$ is linearly independent.

$$\left\{ f_i^{(j)} \;:\; 1 \le i \le \frac{n}{2}, \quad 0 < j < \frac{n}{2} \right\}.$$

Polynomial with a large measure

$x_{\frac{n}{2}+1}x_{\frac{n}{2}+2}$ $\cdots$ $x_2x_{\frac{n}{2}+2}$ $\cdots$ $\cdots$ $x_{n-2}x_{n-1}$ $\cdots$ $x_{\frac{n}{2}-1}x_{n-1}$ $x_{n-1}x_n$ $\cdots$ $x_{\frac{n}{2}}x_n$

$(1, \frac{n}{2})$

$\vdots$

$(1, 1)$

$\vdots$

$\vdots$

$(\frac{n}{2} - 2, \frac{n}{2})$

$\vdots$

$(\frac{n}{2} - 2, 1)$

$(\frac{n}{2} - 1, \frac{n}{2})$

$\vdots$

$(\frac{n}{2} - 1, 1)$

Polynomial with a large measure

$x_{\frac{n}{2}+1}x_{\frac{n}{2}+2}$ $\cdots$ $x_2 x_{\frac{n}{2}+2}$ $\cdots$ $\cdots$ $x_{n-2}x_{n-1}$ $\cdots$ $x_{\frac{n}{2}-1}x_{n-1}$ $x_{n-1}x_n$ $\cdots$ $x_{\frac{n}{2}}x_n$

$(1, \frac{n}{2})$

$\vdots$

$(1, 1)$ $x_k x_l$

$\vdots$

$\vdots$

$(\frac{n}{2}-2, \frac{n}{2})$ (j, i) $\boxed{\mathrm{coeff}_{x_k x_l}(f_i^{(j)})}$

$\vdots$

$(\frac{n}{2}-2, 1)$

$(\frac{n}{2}-1, \frac{n}{2})$

$\vdots$

$(\frac{n}{2}-1, 1)$

Polynomial with a large measure

$x_{\frac{n}{2}+1}x_{\frac{n}{2}+2} \quad \cdots \quad x_2 x_{\frac{n}{2}+2} \quad \cdots \quad \cdots \quad x_{n-2}x_{n-1} \quad \cdots \quad x_{\frac{n}{2}-1}x_{n-1} \quad x_{n-1}x_n \quad \cdots \quad x_{\frac{n}{2}}x_n$

$(1, \frac{n}{2})$

$\vdots$

$(1, 1)$

$\vdots$

$\vdots$

$(\frac{n}{2} - 2, \frac{n}{2})$

$\vdots$

$(\frac{n}{2} - 2, 1)$

$(\frac{n}{2} - 1, \frac{n}{2})$

$\vdots$

$(\frac{n}{2} - 1, 1)$

$x_k x_l$

$(j, i) \qquad \boxed{\mathrm{coeff}_{x_k x_l}(f_i^{(j)})}$

The matrix is lower triangular with the diagonal entries being all 1.

Polynomial with a large measure

$x_{\frac{n}{2}+1}x_{\frac{n}{2}+2} \quad \cdots \quad x_2 x_{\frac{n}{2}+2} \quad \cdots \quad \cdots \quad x_{n-2}x_{n-1} \quad \cdots \quad x_{\frac{n}{2}-1}x_{n-1} \quad x_{n-1}x_n \quad \cdots \quad x_{\frac{n}{2}}x_n$

$(1, \frac{n}{2})$

$\vdots$

$(1, 1)$

$\vdots$

$\vdots$

$(\frac{n}{2} - 2, \frac{n}{2})$

$\vdots$

$(\frac{n}{2} - 2, 1)$

$(\frac{n}{2} - 1, \frac{n}{2})$

$\vdots$

$(\frac{n}{2} - 1, 1)$

$x_k x_l$

$(j, i) \quad \boxed{\text{coeff}_{x_k x_l}(f_i^{(j)})}$

The matrix is lower triangular with the diagonal entries being all 1.

This completes the proof of the main result.

The lower bound is tight

There is a homogeneous non-commutative circuit of size $O(n^2)$ that computes $\mathrm{OSym}_{n,\frac{n}{2}+1}(\mathbf{x})$.

The lower bound is tight

There is a homogeneous non-commutative circuit of size $O(n^2)$ that computes $\mathrm{OSym}_{n,\frac{n}{2}+1}(\mathbf{x})$.

How?

The lower bound is tight

There is a homogeneous non-commutative circuit of size $O(n^2)$ that computes $\mathrm{OSym}_{n,\frac{n}{2}+1}(\mathbf{x})$.

How?

Use the following fact recursively.

$$\mathrm{OSym}_{n,d}(x_1,\ldots,x_n) = \mathrm{OSym}_{n-1,d-1}(x_1,\ldots,x_{n-1}) \cdot x_n + \mathrm{OSym}_{n-1,d}(x_1,\ldots,x_{n-1}).$$

Better Upper bound in the non-homogeneous setting

There is a non-commutative circuit of size $O(n \log^2 n)$ that computes all the elementary symmetric polynomials simultaneously.

Better Upper bound in the non-homogeneous setting

There is a non-commutative circuit of size $O(n \log^2 n)$ that computes all the elementary symmetric polynomials simultaneously.

How?

Better Upper bound in the non-homogeneous setting

There is a non-commutative circuit of size $O(n \log^2 n)$ that computes all the elementary symmetric polynomials simultaneously.

How?

$\mathrm{OSym}_{n,d}(x_1, \ldots, x_n) = \mathsf{coeff}_{t^d} \left(\prod_{i=1}^{n}(1 + tx_i)\right)$

Better Upper bound in the non-homogeneous setting

There is a non-commutative circuit of size $O(n \log^2 n)$ that computes all the elementary symmetric polynomials simultaneously.

How?

$$\mathrm{OSym}_{n,d}(x_1, \ldots, x_n) = \mathsf{coeff}_{t^d}\left(\prod_{i=1}^{n}(1 + tx_i)\right) = \mathsf{coeff}_{t^d}\left(\prod_{i=1}^{\frac{n}{2}}(1 + tx_i) \cdot \prod_{i=\frac{n}{2}+1}^{n}(1 + tx_i)\right).$$

Better Upper bound in the non-homogeneous setting

There is a non-commutative circuit of size $O(n \log^2 n)$ that computes all the elementary symmetric polynomials simultaneously.

How?

$$\mathrm{OSym}_{n,d}(x_1, \ldots, x_n) = \mathsf{coeff}_{t^d}\left(\prod_{i=1}^{n}(1 + tx_i)\right) = \mathsf{coeff}_{t^d}\left(\prod_{i=1}^{\frac{n}{2}}(1 + tx_i) \cdot \prod_{i=\frac{n}{2}+1}^{n}(1 + tx_i)\right).$$

Think of $\quad f = \prod_{i=1}^{\frac{n}{2}}(1 + tx_i), g = \prod_{i=\frac{n}{2}+1}^{n}(1 + tx_i) \in \mathbb{F}\langle \mathbf{x} \rangle [t].$

Better Upper bound in the non-homogeneous setting

There is a non-commutative circuit of size $O(n \log^2 n)$ that computes all the elementary symmetric polynomials simultaneously.

How?

$$\mathrm{OSym}_{n,d}(x_1, \ldots, x_n) = \mathrm{coeff}_{t^d}\left(\prod_{i=1}^{n}(1 + tx_i)\right) = \mathrm{coeff}_{t^d}\left(\prod_{i=1}^{\frac{n}{2}}(1 + tx_i) \cdot \prod_{i=\frac{n}{2}+1}^{n}(1 + tx_i)\right).$$

Think of $f = \prod_{i=1}^{\frac{n}{2}}(1 + tx_i), g = \prod_{i=\frac{n}{2}+1}^{n}(1 + tx_i) \in \mathbb{F}\langle \mathbf{x} \rangle [t]$.

Do polynomial multiplication recursively $\log n$ times.

Better Upper bound in the non-homogeneous setting

There is a non-commutative circuit of size $O(n\log^2 n)$ that computes all the elementary symmetric polynomials simultaneously.

How?

$$\mathrm{OSym}_{n,d}(x_1,\dots,x_n) = \mathsf{coeff}_{t^d}\left(\prod_{i=1}^{n}(1+tx_i)\right) = \mathsf{coeff}_{t^d}\left(\prod_{i=1}^{\frac{n}{2}}(1+tx_i)\cdot\prod_{i=\frac{n}{2}+1}^{n}(1+tx_i)\right).$$

Think of $\quad f=\prod_{i=1}^{\frac{n}{2}}(1+tx_i),\ g=\prod_{i=\frac{n}{2}+1}^{n}(1+tx_i)\in\mathbb{F}\langle\mathbf{x}\rangle[t].$

Do polynomial multiplication recursively $\log n$ times. Note that polynomial multiplication can be done in time $O(n\log n)$ using FFT.

Open Questions

- Can we show a $\tilde{\Omega}(d)$ lower bound against general non-commutative circuits?

Open Questions

- Can we show a $\tilde{\Omega}(d)$ lower bound against general non-commutative circuits?
- Can we show a quadratic lower bound for a constant variate polynomial?

Open Questions

- Can we show a $\tilde{\Omega}(d)$ lower bound against general non-commutative circuits?
- Can we show a quadratic lower bound for a constant variate polynomial?

Conjecture: If

$$f = x_1 x_0^{d-1} f_1 + x_0 x_1 x_0^{d-2} f_2 + \cdots + x_0^{d-1} x_1 f_d$$

can be computed by a non-commutative circuit of size s, then $\{f_1, \ldots, f_d\}$ can be simultaneously computed by a non-commutative circuit of size $d + O(s)$.

Open Questions

- Can we show a $\tilde{\Omega}(d)$ lower bound against general non-commutative circuits?
- Can we show a quadratic lower bound for a constant variate polynomial?

Conjecture: If

$$f = x_1 x_0^{d-1} f_1 + x_0 x_1 x_0^{d-2} f_2 + \cdots + x_0^{d-1} x_1 f_d$$

can be computed by a non-commutative circuit of size s, then $\{f_1, \ldots, f_d\}$ can be simultaneously computed by a non-commutative circuit of size $d + O(s)$.

If true, then the answer to the second question is "yes".

Hardness Amplification

[Carmossino-Impagliazzo-Lovett-Mihajlin]: Super-linear lower bounds $\left(n^{\Omega\left(\frac{\omega}{2}+\varepsilon\right)}\right)$ against non-commutative circuits for constant degree polynomials imply exponential lower bounds.

Hardness Amplification

[Carmossino-Impagliazzo-Lovett-Mihajlin]: Super-linear lower bounds $\left(n^{\Omega\left(\frac{\omega}{2}+\varepsilon\right)}\right)$ against non-commutative circuits for constant degree polynomials imply exponential lower bounds.

- We seem to understand very little in the low degree (let alone constant degree) setting.

Hardness Amplification

[Carmossino-Impagliazzo-Lovett-Mihajlin]: Super-linear lower bounds $\left(n^{\Omega\left(\frac{\omega}{2}+\varepsilon\right)}\right)$ against non-commutative circuits for constant degree polynomials imply exponential lower bounds.

- We seem to understand very little in the low degree (let alone constant degree) setting.
- All the advantages of the non-commutative setting seems to be lost if degree is constant.

Hardness Amplification

[Carmossino-Impagliazzo-Lovett-Mihajlin]: Super-linear lower bounds $\left(n^{\Omega\left(\frac{\omega}{2}+\varepsilon\right)}\right)$ against non-commutative circuits for constant degree polynomials imply exponential lower bounds.

- We seem to understand very little in the low degree (let alone constant degree) setting.
- All the advantages of the non-commutative setting seems to be lost if degree is constant.

Question: Can we show a similar statement (or any non-trivial hardness amplification statement) in the non-constant degree setting?

Open Questions

Thank you!