Multiple Facets of Algebraic Computation

C Ramya
IMSc, Chennai.

September 2023

Algebraic Computation

- Goal: Understand the amount of *computational resource* required to solve a given *computational problem* on a *computational model*.

Algebraic Computation

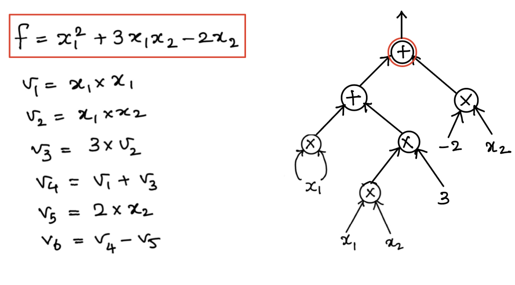
- Goal: Understand the amount of *computational resource* required to solve a given *computational problem* on a *computational model*.
- **Objects:** Polynomials. E.g., $f = x_1^2 + 3x_1x_2 - 2$.

Algebraic Computation

- Goal: Understand the amount of *computational resource* required to solve a given *computational problem* on a *computational model*.
- **Objects:** Polynomials. E.g., $f = x_1^2 + 3x_1x_2 - 2$.
  Resource: No. of arithmetic operations($+$, $\times$)

Algebraic Computation

- Goal: Understand the amount of *computational resource* required to solve a given *computational problem* on a *computational model*.
- **Objects:** Polynomials. E.g., $f = x_1^2 + 3x_1x_2 - 2$.
  Resource: No. of arithmetic operations($+$, $\times$)



- **Model:** Arithmetic circuits: DAGs with leaves labelled by variables or constants(from $\mathbb{F}$) and internal gates labelled by $\{+, \times\}$.

Complexity of polynomials

- *size*($\mathcal{C}$) - number of gates in circuit $\mathcal{C} \equiv$ no. of arithmetic operations to compute f.

Complexity of polynomials

- *size*$(\mathcal{C})$ - number of gates in circuit $\mathcal{C} \equiv$ no. of arithmetic operations to compute f.
- *depth*$(\mathcal{C})$ - length of longest path from input to output gate of $\mathcal{C}$.

Complexity of polynomials

- *size*($\mathcal{C}$) - number of gates in circuit $\mathcal{C} \equiv$ no. of arithmetic operations to compute f.
- *depth*($\mathcal{C}$) - length of longest path from input to output gate of $\mathcal{C}$.
- There can be several different circuits computing a given polynomial.

Complexity of polynomials

- *size*($\mathcal{C}$) - number of gates in circuit $\mathcal{C} \equiv$ no. of arithmetic operations to compute f.
- *depth*($\mathcal{C}$) - length of longest path from input to output gate of $\mathcal{C}$.
- There can be several different circuits computing a given polynomial.

Complexity of polynomials

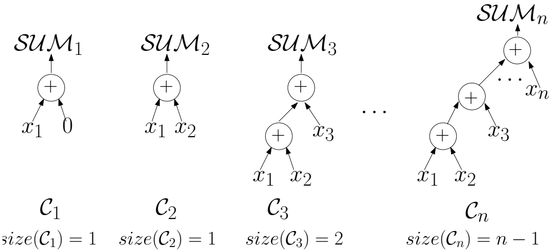
- *size*$(\mathcal{C})$ - number of gates in circuit $\mathcal{C} \equiv$ no. of arithmetic operations to compute f.
- *depth*$(\mathcal{C})$ - length of longest path from input to output gate of $\mathcal{C}$.
- There can be several different circuits computing a given polynomial.

Let $\mathcal{SUM}_n = x_1 + x_2 + \cdots + x_n$.

Complexity of polynomials

- *size*($\mathcal{C}$) - number of gates in circuit $\mathcal{C} \equiv$ no. of arithmetic operations to compute f.
- *depth*($\mathcal{C}$) - length of longest path from input to output gate of $\mathcal{C}$.
- There can be several different circuits computing a given polynomial.

Let $\mathcal{SUM}_n = x_1 + x_2 + \cdots + x_n$.



$SUM = (\mathcal{SUM}_n)_{n \geq 1}$ is a polynomial family.

Valiant's Conjecture

- Any n-variate degree-d polynomial can be computed by

Valiant's Conjecture

- Any n-variate degree-d polynomial can be computed by a depth two circuit of size $O\left(\binom{n+d}{d}\right)$.

Valiant's Conjecture

- Any n-variate degree-d polynomial can be computed by a depth two circuit of size $O\left(\binom{n+d}{d}\right)$.
- There *exists* n-variate degree-d polynomials that require arithmetic circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$.

Valiant's Conjecture

- Any n-variate degree-d polynomial can be computed by a depth two circuit of size $O\left(\binom{n+d}{d}\right)$.
- There *exists* n-variate degree-d polynomials that require arithmetic circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$.
- A polynomial family $(f_n)_{n\geq 1}$ is *efficiently computable* if for every n, $\deg(f_n) = \mathrm{poly}(n)$ and there is $\mathrm{poly}(n)$ size circuit for f_n.

Valiant's Conjecture

- Any n-variate degree-d polynomial can be computed by a depth two circuit of size $O\left(\binom{n+d}{d}\right)$.
- There *exists* n-variate degree-d polynomials that require arithmetic circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$.
- A polynomial family $(f_n)_{n\geq 1}$ is *efficiently computable* if for every n, $\deg(f_n) = \mathrm{poly}(n)$ and there is $\mathrm{poly}(n)$ size circuit for f_n.
  E.g.,: SUM, Symbolic Determinant $\det = (\det_n)_{n\geq 1}$.

Valiant's Conjecture

- Any n-variate degree-d polynomial can be computed by a depth two circuit of size $O\left(\binom{n+d}{d}\right)$.
- There *exists* n-variate degree-d polynomials that require arithmetic circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$.
- A polynomial family $(f_n)_{n\geq 1}$ is *efficiently computable* if for every n, $\deg(f_n) = \mathrm{poly}(n)$ and there is $\mathrm{poly}(n)$ size circuit for f_n.
  E.g.,: SUM, Symbolic Determinant $\det = (\det_n)_{n\geq 1}$.
- **Class** VP: class of efficiently computable polynomial families.

Valiant's Conjecture

- Any n-variate degree-d polynomial can be computed by a depth two circuit of size $O\left(\binom{n+d}{d}\right)$.
- There *exists* n-variate degree-d polynomials that require arithmetic circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$.
- A polynomial family $(f_n)_{n\geq 1}$ is *efficiently computable* if for every n, $\deg(f_n) = \mathrm{poly}(n)$ and there is $\mathrm{poly}(n)$ size circuit for f_n.
  E.g.,: SUM, Symbolic Determinant $\det = (\det_n)_{n\geq 1}$.
- **Class** VP: class of efficiently computable polynomial families.
- Are there polynomials that are *hard* to compute(outside VP)? YES.

Valiant's Conjecture

- Any n-variate degree-d polynomial can be computed by a depth two circuit of size $O\left(\binom{n+d}{d}\right)$.
- There *exists* n-variate degree-d polynomials that require arithmetic circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$.
- A polynomial family $(f_n)_{n\geq 1}$ is *efficiently computable* if for every n, $\deg(f_n) = \mathrm{poly}(n)$ and there is $\mathrm{poly}(n)$ size circuit for f_n.
  E.g.,: SUM, Symbolic Determinant $\det = (\det_n)_{n\geq 1}$.
- **Class** VP: class of efficiently computable polynomial families.
- Are there polynomials that are *hard* to compute(outside VP)? YES.
- **Goal**: Find an *explicit polynomial* outside VP.
  Explicit: coefficient of any monomial is *reasonably* easy to compute.

Valiant's Conjecture

- Any n-variate degree-d polynomial can be computed by a depth two circuit of size $O\left(\binom{n+d}{d}\right)$.
- There *exists* n-variate degree-d polynomials that require arithmetic circuits of size $\Omega\left(\sqrt{\binom{n+d}{d}}\right)$.
- A polynomial family $(f_n)_{n\geq 1}$ is *efficiently computable* if for every n, $\deg(f_n) = \text{poly}(n)$ and there is $\text{poly}(n)$ size circuit for f_n.
  E.g.,: SUM, Symbolic Determinant $\det = (\det_n)_{n\geq 1}$.
- **Class** VP: class of efficiently computable polynomial families.
- Are there polynomials that are *hard* to compute(outside VP)? YES.
- **Goal**: Find an *explicit polynomial* outside VP.
  Explicit: coefficient of any monomial is *reasonably* easy to compute.

Valiant's Conjecture: Any circuit for perm_n requires size $n^{\omega(1)}$.

Towards Valiant's Hypothesis

(Baur,Strassen'83)Any circuit for $x_1^d + \cdots + x_n^d$ requires size $\Omega(n \log d)$.

Towards Valiant's Hypothesis

(Baur,Strassen'83)Any circuit for $x_1^d + \cdots + x_n^d$ requires size $\Omega(n \log d)$.
(Folklore) Any depth-2 circuit computing perm_n requires size $n!$.

Towards Valiant's Hypothesis

(Baur,Strassen'83)Any circuit for $x_1^d + \cdots + x_n^d$ requires size $\Omega(n \log d)$.
(Folklore) Any depth-2 circuit computing perm_n requires size $n!$.
Depth reduction $n^{\omega(\sqrt{d})}$ lower bound for depth-three circuits computing an explicit n-variate, degree d polynomial is sufficient to resolve Valiant's conjecture.

Towards Valiant's Hypothesis

(Baur,Strassen'83)Any circuit for $x_1^d + \cdots + x_n^d$ requires size $\Omega(n \log d)$.
(Folklore) Any depth-2 circuit computing perm_n requires size $n!$.
Depth reduction $n^{\omega(\sqrt{d})}$ lower bound for depth-three circuits computing an explicit n-variate, degree d polynomial is sufficient to resolve Valiant's conjecture.
(Limaye, Srinivasan, Tavenas '21) There is an explicit n-variate polynomial (in VP) of degree d such that any depth three circuit for it has size $n^{\Omega(\sqrt{d})}$.

Towards Valiant's Hypothesis

(Baur,Strassen'83)Any circuit for $x_1^d + \cdots + x_n^d$ requires size $\Omega(n \log d)$.
(Folklore) Any depth-2 circuit computing perm_n requires size $n!$.
Depth reduction $n^{\omega(\sqrt{d})}$ lower bound for depth-three circuits computing an explicit n-variate, degree d polynomial is sufficient to resolve Valiant's conjecture.
(Limaye, Srinivasan, Tavenas '21) There is an explicit n-variate polynomial (in VP) of degree d such that any depth three circuit for it has size $n^{\Omega(\sqrt{d})}$.

> Perhaps the principal embarrassment of complexity theory at the present time is its failure to provide techniques for proving non-trivial lower bounds on the complexity of some of the commonest combinatorial and arithmetic problems.

Towards Valiant's Hypothesis

(Baur,Strassen'83)Any circuit for $x_1^d + \cdots + x_n^d$ requires size $\Omega(n \log d)$.
(Folklore) Any depth-2 circuit computing perm_n requires size $n!$.
Depth reduction $n^{\omega(\sqrt{d})}$ lower bound for depth-three circuits computing an explicit n-variate, degree d polynomial is sufficient to resolve Valiant's conjecture.
(Limaye, Srinivasan, Tavenas '21) There is an explicit n-variate polynomial (in VP) of degree d such that any depth three circuit for it has size $n^{\Omega(\sqrt{d})}$.

> Perhaps the principal embarrassment of complexity theory at the present time is its failure to provide techniques for proving non-trivial lower bounds on the complexity of some of the commonest combinatorial and arithmetic problems.
>
> -Valiant(1975)

Proving Lower Bounds: A Toy Example

Let $\mathcal{C} = \{(\alpha x - \beta)^2 \mid \alpha, \beta \in \mathbb{C}\}$. Goal: Find an explicit $h(x) \notin \mathcal{C}$.

Proving Lower Bounds: A Toy Example

Let $\mathcal{C} = \{(\alpha x - \beta)^2 \mid \alpha, \beta \in \mathbb{C}\}$. Goal: Find an explicit $h(x) \notin \mathcal{C}$.

- Let $f(x) = ax^2 + bx + c$ be any quadratic polynomial. Then, $f(x)$ has a repeated root if and only if $b^2 - 4ac = 0$. Note, $\overline{\text{coeff}}(f) = (a, b, c)$.

Proving Lower Bounds: A Toy Example

Let $\mathcal{C} = \{(\alpha x - \beta)^2 \mid \alpha, \beta \in \mathbb{C}\}$. Goal: Find an explicit $h(x) \notin \mathcal{C}$.

- Let $f(x) = ax^2 + bx + c$ be any quadratic polynomial. Then, $f(x)$ has a repeated root if and only if $b^2 - 4ac = 0$. Note, $\overline{\text{coeff}}(f) = (a, b, c)$.
- Find a polynomial $h(x) = ax^2 + bx + c$ with non-zero discriminant.

Proving Lower Bounds: A Toy Example

Let $\mathcal{C} = \{(\alpha x - \beta)^2 \mid \alpha, \beta \in \mathbb{C}\}$. Goal: Find an explicit $h(x) \notin \mathcal{C}$.

- Let $f(x) = ax^2 + bx + c$ be any quadratic polynomial. Then, $f(x)$ has a repeated root if and only if $b^2 - 4ac = 0$. Note, $\overline{\text{coeff}}(f) = (a, b, c)$.
- Find a polynomial $h(x) = ax^2 + bx + c$ with non-zero discriminant.

Consider $P(z_1, z_2, z_3) = z_2^2 - 4z_1z_3$. Then,

- $f(x) \in \mathcal{C} \Rightarrow P(\overline{\text{coeff}}(f)) = 0$.
- $P(z_1, z_2, z_3)$ is efficiently computable.
- There is polynomial $h(x) \in \mathbb{F}[x]^{\leq 2}$ such that $P(\overline{\text{coeff}}(h)) \neq 0$.

Proving Lower Bounds: A Toy Example

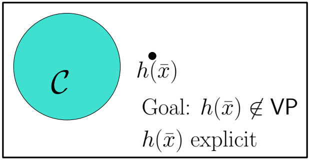
Let $\mathcal{C} = \{(\alpha x - \beta)^2 \mid \alpha, \beta \in \mathbb{C}\}$. Goal: Find an explicit $h(x) \notin \mathcal{C}$.

- Let $f(x) = ax^2 + bx + c$ be any quadratic polynomial. Then, $f(x)$ has a repeated root if and only if $b^2 - 4ac = 0$. Note, $\overline{\text{coeff}}(f) = (a, b, c)$.
- Find a polynomial $h(x) = ax^2 + bx + c$ with non-zero discriminant.

Consider $P(z_1, z_2, z_3) = z_2^2 - 4z_1z_3$. Then,

- $f(x) \in \mathcal{C} \Rightarrow P(\overline{\text{coeff}}(f)) = 0$.
- $P(z_1, z_2, z_3)$ is efficiently computable.
- There is polynomial $h(x) \in \mathbb{F}[x]^{\leq 2}$ such that $P(\overline{\text{coeff}}(h)) \neq 0$.

Proving Lower Bounds against $\mathcal{C}$: Find a property P that every polynomial in $\mathcal{C}$ satisfies and then find an explicit h that does not satisfy P.



Another Toy Example

Let $\mathcal{C}$ be the class of $\Sigma\Pi$-circuits. Any $\Sigma\Pi$ circuit computing the permanent requires size $n!$.

Another Toy Example

Let $\mathcal{C}$ be the class of $\Sigma\Pi$-circuits. Any $\Sigma\Pi$ circuit computing the permanent requires size $n!$.

- Let f be computable by a $\Sigma\Pi$ circuit of top fanin s.

Another Toy Example

Let $\mathcal{C}$ be the class of $\Sigma\Pi$-circuits. Any $\Sigma\Pi$ circuit computing the permanent requires size $n!$.

- Let f be computable by a $\Sigma\Pi$ circuit of top fanin s.
- Define $\mu : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{R}$ s.t. $\mu(f_1 + \cdots + f_s) \le \mu(f_1) + \cdots + \mu(f_s)$.

Another Toy Example

Let $\mathcal{C}$ be the class of $\Sigma\Pi$-circuits. Any $\Sigma\Pi$ circuit computing the permanent requires size $n!$.

- Let f be computable by a $\Sigma\Pi$ circuit of top fanin s.
- Define $\mu : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{R}$ s.t. $\mu(f_1 + \cdots + f_s) \leq \mu(f_1) + \cdots + \mu(f_s)$. E.g., $\mu(f) \triangleq$ number of monomials of f.

Another Toy Example

Let $\mathcal{C}$ be the class of $\Sigma\Pi$-circuits. Any $\Sigma\Pi$ circuit computing the permanent requires size $n!$.

- Let f be computable by a $\Sigma\Pi$ circuit of top fanin s.
- Define $\mu : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{R}$ s.t. $\mu(f_1 + \cdots + f_s) \leq \mu(f_1) + \cdots + \mu(f_s)$. E.g., $\mu(f) \triangleq$ number of monomials of f.

$$\begin{aligned}\mu(f) &= \mu(m_1 + \cdots + m_s)\\ &\leq \mu(m_1) + \cdots + \mu(m_s) \leq s\end{aligned}$$

Another Toy Example

Let $\mathcal{C}$ be the class of $\Sigma\Pi$-circuits. Any $\Sigma\Pi$ circuit computing the permanent requires size $n!$.

- Let f be computable by a $\Sigma\Pi$ circuit of top fanin s.
- Define $\mu : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{R}$ s.t. $\mu(f_1 + \cdots + f_s) \leq \mu(f_1) + \cdots + \mu(f_s)$. E.g., $\mu(f) \triangleq$ number of monomials of f.

$$\begin{aligned} \mu(f) &= \mu(m_1 + \cdots + m_s) \\ &\leq \mu(m_1) + \cdots + \mu(m_s) \leq s \end{aligned}$$

- Observe that $\mu(perm_n) = n!$

Another Toy Example

Let $\mathcal{C}$ be the class of ΣΠ-circuits. Any ΣΠ circuit computing the permanent requires size $n!$.

- Let f be computable by a ΣΠ circuit of top fanin s.
- Define $\mu : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{R}$ s.t. $\mu(f_1 + \cdots + f_s) \leq \mu(f_1) + \cdots + \mu(f_s)$. E.g., $\mu(f) \triangleq$ number of monomials of f.

$$\begin{aligned} \mu(f) &= \mu(m_1 + \cdots + m_s) \\ &\leq \mu(m_1) + \cdots + \mu(m_s) \leq s \end{aligned}$$

- Observe that $\mu(perm_n) = n!$ Therefore, $s \geq n!$.

Another Toy Example

Let $\mathcal{C}$ be the class of $\Sigma\Pi$-circuits. Any $\Sigma\Pi$ circuit computing the permanent requires size $n!$.

- Let f be computable by a $\Sigma\Pi$ circuit of top fanin s.
- Define $\mu : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{R}$ s.t. $\mu(f_1 + \cdots + f_s) \leq \mu(f_1) + \cdots + \mu(f_s)$. E.g., $\mu(f) \triangleq$ number of monomials of f.

$$\begin{aligned}\mu(f) &= \mu(m_1 + \cdots + m_s) \\ &\leq \mu(m_1) + \cdots + \mu(m_s) \leq s\end{aligned}$$

- Observe that $\mu(perm_n) = n!$ Therefore, $s \geq n!$.

For more(infact most) sophisticated circuit classes $\mathcal{C}$:

Construct a measure $\mu : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{R}$:

- $\mu(f)$ is *small* for $f \in \mathcal{C}$.
- $\mu(h)$ is *large* for an explicit polynomial h.

Another Toy Example

Let $\mathcal{C}$ be the class of $\Sigma\Pi$-circuits. Any $\Sigma\Pi$ circuit computing the permanent requires size $n!$.

- Let f be computable by a $\Sigma\Pi$ circuit of top fanin s.
- Define $\mu : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{R}$ s.t. $\mu(f_1 + \cdots + f_s) \leq \mu(f_1) + \cdots + \mu(f_s)$. E.g., $\mu(f) \triangleq$ number of monomials of f.

$$\begin{aligned} \mu(f) &= \mu(m_1 + \cdots + m_s) \\ &\leq \mu(m_1) + \cdots + \mu(m_s) \leq s \end{aligned}$$

- Observe that $\mu(perm_n) = n!$ Therefore, $s \geq n!$.

For more(infact most) sophisticated circuit classes $\mathcal{C}$:

Construct a measure $\mu : \mathbb{F}[x_1, \ldots, x_n] \to \mathbb{R}$:

- $\mu(f)$ is *small* for $f \in \mathcal{C}$.
- $\mu(h)$ is *large* for an explicit polynomial h.
- $\mu(f)$ is $\text{rank}(M_f)$ for a matrix M_f associated with polynomial f.

Proving Lower Bounds against $\mathcal{C}$

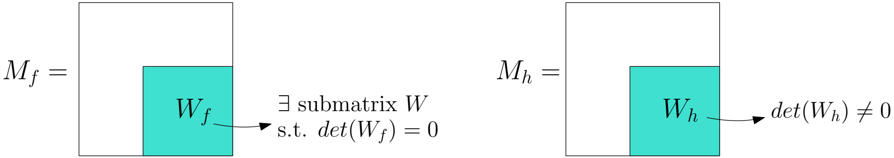
Most lower bound proofs against $\mathcal{C}$ construct a measure $\mu : \mathbb{F}[\bar{x}] \to \mathbb{R}$:

- $\mu(f)$ is *small* for $f \in \mathcal{C}$ (i.e., rank(M_f) is *small*)
- $\mu(h)$ is *large* for an explicit polynomial h. (i.e., rank(M_h) is *large*)

Proving Lower Bounds against $\mathcal{C}$

Most lower bound proofs against $\mathcal{C}$ construct a measure $\mu : \mathbb{F}[\bar{x}] \to \mathbb{R}$:

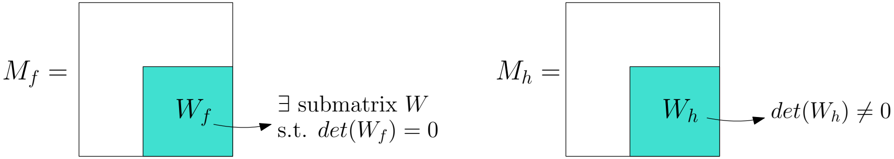
- $\mu(f)$ is *small* for $f \in \mathcal{C}$ (i.e., rank(M_f) is *small*)
- $\mu(h)$ is *large* for an explicit polynomial h. (i.e., rank(M_h) is *large*)



Proving Lower Bounds against $\mathcal{C}$

Most lower bound proofs against $\mathcal{C}$ construct a measure $\mu : \mathbb{F}[\bar{x}] \to \mathbb{R}$:

- $\mu(f)$ is *small* for $f \in \mathcal{C}$ (i.e., rank(M_f) is *small*)
- $\mu(h)$ is *large* for an explicit polynomial h. (i.e., rank(M_h) is *large*)



For $f \in \mathbb{F}[x_1, \ldots, x_n]$ of degree $d = \text{poly}(n)$: $M_f \in \mathbb{F}^{N \times N}, N = \binom{n+d}{n}$.

- $M_f[x^\alpha, x^\beta] =$ coefficient of x^α in $\frac{\partial f}{\partial x^\beta}$.
- Entries of M_f are linear in the coefficients of f.

Towards Algebraically Natural Proofs

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d. Then, $f = \sum c_m \cdot m$.
Coefficient-vector: $\overline{\text{coeff}}(f) = (c_1, c_2, \ldots, c_N) \in \mathbb{F}^N$ where $N = \binom{n+d}{n}$.
"Natural" lower bound proof for $\mathcal{C} \subseteq \mathbb{F}[x_1, \ldots, x_n]^{\leq d}$:

$\mathcal{C}$ has a *natural proof* if there is a non-zero polynomial $P(z_1, \ldots, z_N)$:

1. **Usefulness**: $\forall f \in \mathcal{C}$, $P(\overline{\text{coeff}}(f)) = 0$.
2. **Constructivity**: P has degree poly(N) and size poly(N).
3. **Largeness**: $P(\overline{\text{coeff}}(h)) \neq 0$ for candidate hard polynomial h (and for many more polynomials).

Towards Algebraically Natural Proofs

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d. Then, $f = \sum c_m \cdot m$.
Coefficient-vector: $\overline{\text{coeff}}(f) = (c_1, c_2, \ldots, c_N) \in \mathbb{F}^N$ where $N = \binom{n+d}{n}$.
"Natural" lower bound proof for $\mathcal{C} \subseteq \mathbb{F}[x_1, \ldots, x_n]^{\leq d}$:

$\mathcal{C}$ has a *natural proof* if there is a non-zero polynomial $P(z_1, \ldots, z_N)$:

1. **Usefulness**: $\forall f \in \mathcal{C}$, $P(\overline{\text{coeff}}(f)) = 0$.
2. **Constructivity**: P has degree poly(N) and size poly(N).
3. **Largeness**: $P(\overline{\text{coeff}}(h)) \neq 0$ for candidate hard polynomial h (and for many more polynomials).

Example 1: $\mathcal{C} = \{(\alpha x - \beta)^2 \mid \alpha, \beta \in \mathbb{C}\}$. Then, $P(z_1, z_2, z_3) = z_2^2 - 4z_1z_3$ such that $P(\overline{\text{coeff}}(f)) = 0$ for all $f \in \mathcal{C}$.

Towards Algebraically Natural Proofs

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d. Then, $f = \sum c_m \cdot m$.
Coefficient-vector: $\overline{\text{coeff}}(f) = (c_1, c_2, \ldots, c_N) \in \mathbb{F}^N$ where $N = \binom{n+d}{n}$.
"Natural" lower bound proof for $\mathcal{C} \subseteq \mathbb{F}[x_1, \ldots, x_n]^{\leq d}$:

$\mathcal{C}$ has a *natural proof* if there is a non-zero polynomial $P(z_1, \ldots, z_N)$:

1. **Usefulness**: $\forall f \in \mathcal{C}$, $P(\overline{\text{coeff}}(f)) = 0$.
2. **Constructivity**: P has degree $\text{poly}(N)$ and size $\text{poly}(N)$.
3. **Largeness**: $P(\overline{\text{coeff}}(h)) \neq 0$ for candidate hard polynomial h (and for many more polynomials).

Example 1: $\mathcal{C} = \{(\alpha x - \beta)^2 \mid \alpha, \beta \in \mathbb{C}\}$. Then, $P(z_1, z_2, z_3) = z_2^2 - 4z_1z_3$ such that $P(\overline{\text{coeff}}(f)) = 0$ for all $f \in \mathcal{C}$.

Example 2: $\mathcal{C} = \{\Sigma\Pi\Sigma,\ \Sigma\Pi\Sigma\Pi,\ \Sigma \wedge \Sigma\}$.
Then, $P(z_1, \ldots, z_N) = \det(W)$ such that $P(\overline{\text{coeff}}(f)) = 0$ for all $f \in \mathcal{C}$.

Towards Algebraically Natural Proofs

Let $f \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d. Then, $f = \sum c_m \cdot m$.
Coefficient-vector: $\overline{\text{coeff}}(f) = (c_1, c_2, \ldots, c_N) \in \mathbb{F}^N$ where $N = \binom{n+d}{n}$.
"Natural" lower bound proof for $\mathcal{C} \subseteq \mathbb{F}[x_1, \ldots, x_n]^{\leq d}$:

$\mathcal{C}$ has a *natural proof* if there is a non-zero polynomial $P(z_1, \ldots, z_N)$:

1. **Usefulness**: $\forall f \in \mathcal{C}$, $P(\overline{\text{coeff}}(f)) = 0$.
2. **Constructivity**: P has degree poly(N) and size poly(N).
3. **Largeness**: $P(\overline{\text{coeff}}(h)) \neq 0$ for candidate hard polynomial h (and for many more polynomials).

Example 1: $\mathcal{C} = \{(\alpha x - \beta)^2 \mid \alpha, \beta \in \mathbb{C}\}$. Then, $P(z_1, z_2, z_3) = z_2^2 - 4z_1z_3$ such that $P(\overline{\text{coeff}}(f)) = 0$ for all $f \in \mathcal{C}$.

Example 2: $\mathcal{C} = \{\Sigma\Pi\Sigma,\ \Sigma\Pi\Sigma\Pi,\ \Sigma \wedge \Sigma\}$.
Then, $P(z_1, \ldots, z_N) = \det(W)$ such that $P(\overline{\text{coeff}}(f)) = 0$ for all $f \in \mathcal{C}$.

How far can natural proofs succeed?

Can we prove Valiant's Conjecture via natural proofs? $\mathrm{VP}(n)$ is class of n-variate degree-$\mathrm{poly}(n)$ polynomials computable by size $\mathrm{poly}(n)$ circuits.

How far can natural proofs succeed?

Can we prove Valiant's Conjecture via natural proofs? VP(n) is class of n-variate degree-poly(n) polynomials computable by size poly(n) circuits.

Question: Does there exist a non-zero polynomial $P(z_1, \ldots, z_N)$:

- $\forall f \in \mathrm{VP}(n)$, $P(\overline{\mathrm{coeff}}(f)) = 0$;
- P has degree poly(N) and size poly(N)?

How far can natural proofs succeed?

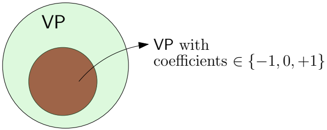
Can we prove Valiant's Conjecture via natural proofs? VP(n) is class of n-variate degree-poly(n) polynomials computable by size poly(n) circuits.

Question: Does there exist a non-zero polynomial $P(z_1, \ldots, z_N)$:

- $\forall f \in \mathrm{VP}(n)$, $P(\overline{\mathrm{coeff}}(f)) = 0$;
- P has degree poly(N) and size poly(N)?

Theorem (Chatterjee, Kumar, R., Saptharishi, Tengse)

Answer: *Yes, for polynomials with small integer coefficients.*



On the Existence of Natural Proofs

Theorem (Chatterjee, Kumar, R., Saptharishi, Tengse)

For n, d and $N = \binom{n+d}{n}$, there exists a non-zero $P(z_1, \ldots, z_N)$ such that

1. $P(\overline{\text{coeff}}(f)) = 0$ *for all* $f \in \text{VP}(n, d)$ *with* small *integer coefficients;*
2. $P(z_1, \ldots, z_N)$ *has size and degree* $\text{poly}(N)$*; and*
3. *there exists h having* small *integer coefficients with* $P(\overline{\text{coeff}}(h)) \neq 0$.

On the Existence of Natural Proofs

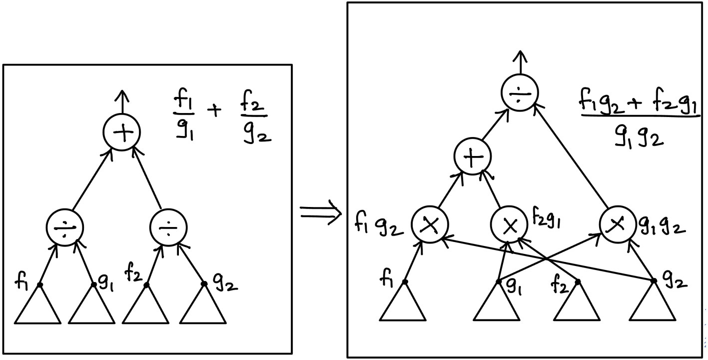
Theorem (Chatterjee, Kumar, R., Saptharishi, Tengse)

For n, d and $N = \binom{n+d}{n}$, there exists a non-zero $P(z_1, \ldots, z_N)$ such that

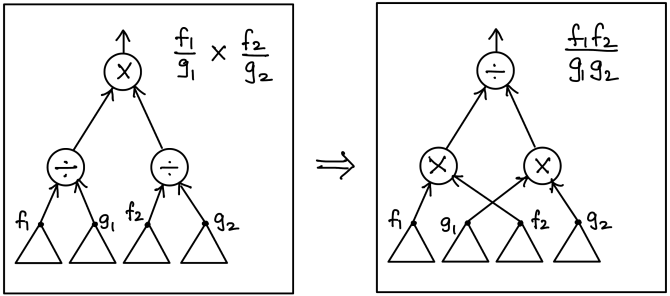
1. $P(\overline{\mathrm{coeff}}(f)) = 0$ *for all* $f \in \mathrm{VP}(n, d)$ *with* small *integer coefficients;*
2. $P(z_1, \ldots, z_N)$ *has size and degree* $\mathrm{poly}(N)$*; and*
3. *there exists* h *having* small *integer coefficients with* $P(\overline{\mathrm{coeff}}(h)) \neq 0$.

- What does this result suggest? An evidence for the power of *natural lower bound techniques* for proving super-polynomial lower bounds.

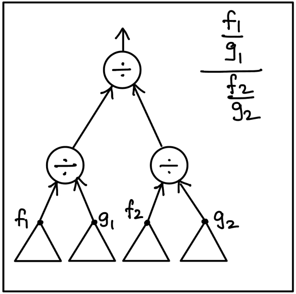
Strassen's Vermeidung von Divisionen



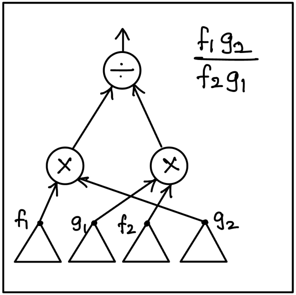
Strassen's Vermeidung von Divisionen



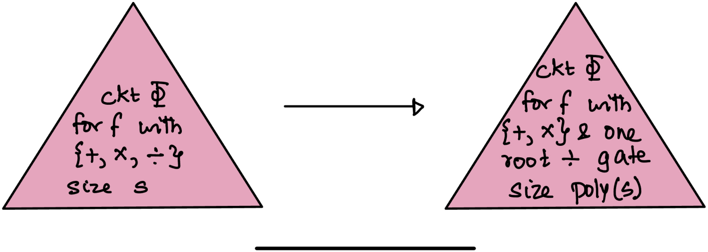
Strassen's Vermeidung von Divisionen



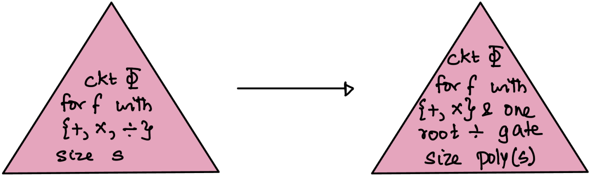
$\Rightarrow$



Strassen's Vermeidung von Divisionen



Strassen's Vermeidung von Divisionen



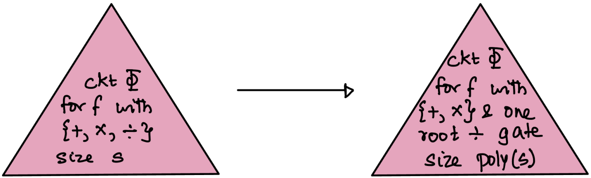
$$f = \frac{h}{g} = \frac{h}{1-(1-g)} = h \cdot \sum_{j=0}^{\infty} (1-g)^j$$

n-variate
deg d

Can truncate power series depending on the degree of f.

need $\bar{a} \in \mathbb{F}^n$ s.t. $g(\bar{a}) \neq 0$.

Strassen's Vermeidung von Divisionen



$$f = \frac{h}{g} = \frac{h}{1-(1-g)} = h \cdot \sum_{j=0}^{\infty} (1-g)^j$$

n-variate
deg d

Can truncate power series depending on the degree of f.

need $\bar{a} \in \mathbb{F}^n$ s.t. $g(\bar{a}) \neq 0$.

- Division gates can be eliminated with polynomial blow up in size.

Polynomial Identity Testing

- A polynomial($f \equiv 0$) is *identically zero* if all its coefficients are zero.
- E.g.: $(x+y)^2 - x^2 - y^2 - 2xy \equiv 0$ and $(x+y)^2 - x^2 - y^2 + 2xy \not\equiv 0$.

Polynomial Identity Testing

- A polynomial($f \equiv 0$) is *identically zero* if all its coefficients are zero.
- E.g.: $(x+y)^2 - x^2 - y^2 - 2xy \equiv 0$ and $(x+y)^2 - x^2 - y^2 + 2xy \not\equiv 0$.

Polynomial Identity Testing (PIT)

Given $f \in \mathbb{F}[x_1, \ldots, x_n]$ test if $f \equiv 0$.

Polynomial Identity Testing

- A polynomial($f \equiv 0$) is *identically zero* if all its coefficients are zero.
- E.g.: $(x+y)^2 - x^2 - y^2 - 2xy \equiv 0$ and $(x+y)^2 - x^2 - y^2 + 2xy \not\equiv 0$.

Polynomial Identity Testing (PIT)

Given $f \in \mathbb{F}[x_1, \ldots, x_n]$ test if $f \equiv 0$.

- Univariate case:

Polynomial Identity Testing

- A polynomial($f \equiv 0$) is *identically zero* if all its coefficients are zero.
- E.g.: $(x+y)^2 - x^2 - y^2 - 2xy \equiv 0$ and $(x+y)^2 - x^2 - y^2 + 2xy \not\equiv 0$.

Polynomial Identity Testing (PIT)

Given $f \in \mathbb{F}[x_1, \ldots, x_n]$ test if $f \equiv 0$.

- Univariate case: Any non-zero univariate polynomial of degree d has at most d roots. Easy to get a polynomial time algorithm.

Polynomial Identity Testing

- A polynomial($f \equiv 0$) is *identically zero* if all its coefficients are zero.
- E.g.: $(x+y)^2 - x^2 - y^2 - 2xy \equiv 0$ and $(x+y)^2 - x^2 - y^2 + 2xy \not\equiv 0$.

Polynomial Identity Testing (PIT)

Given $f \in \mathbb{F}[x_1, \ldots, x_n]$ test if $f \equiv 0$.

- Univariate case: Any non-zero univariate polynomial of degree d has at most d roots. Easy to get a polynomial time algorithm.
- Multivariate case: Can have infinitely many roots.

Polynomial Identity Testing

- A polynomial($f \equiv 0$) is *identically zero* if all its coefficients are zero.
- E.g.: $(x+y)^2 - x^2 - y^2 - 2xy \equiv 0$ and $(x+y)^2 - x^2 - y^2 + 2xy \not\equiv 0$.

Polynomial Identity Testing (PIT)

Given $f \in \mathbb{F}[x_1, \ldots, x_n]$ test if $f \equiv 0$.

- Univariate case: Any non-zero univariate polynomial of degree d has at most d roots. Easy to get a polynomial time algorithm.
- Multivariate case: Can have infinitely many roots.
- Randomized polynomial time algorithm for multivariate PIT is known.
- **Open Question:** Derandomizing PIT.

Non-commutative PIT

- Set of non-commuting variables $\{x_1, \ldots, x_n\}$ i.e., $x_i x_j \neq x_j x_i \ \forall i \neq j$. E.g., $(x_1 + x_2)(x_1 - x_2) \neq x_1^2 - x_2^2$.
- A non-commutative polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ is a combination of words.

Non-commutative PIT

- Set of non-commuting variables $\{x_1, \ldots, x_n\}$ i.e., $x_i x_j \neq x_j x_i \ \forall i \neq j$. E.g., $(x_1 + x_2)(x_1 - x_2) \neq x_1^2 - x_2^2$.
- A non-commutative polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ is a combination of words.
- ncPIT: Given a non-commutative polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ test if $f \equiv 0$. E.g., $x_1 x_2 - x_2 x_1 \not\equiv 0$ in the non-commutative world.

Non-commutative PIT

- Set of non-commuting variables $\{x_1, \ldots, x_n\}$ i.e., $x_i x_j \neq x_j x_i \; \forall i \neq j$. E.g., $(x_1 + x_2)(x_1 - x_2) \neq x_1^2 - x_2^2$.
- A non-commutative polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ is a combination of words.
- ncPIT: Given a non-commutative polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ test if $f \equiv 0$. E.g., $x_1 x_2 - x_2 x_1 \not\equiv 0$ in the non-commutative world.
- Non-commutative circuit: arithmetic circuit whose $\times$ gate respects the ordering.

Non-commutative PIT

- Set of non-commuting variables $\{x_1, \ldots, x_n\}$ i.e., $x_i x_j \neq x_j x_i \; \forall i \neq j$. E.g., $(x_1 + x_2)(x_1 - x_2) \neq x_1^2 - x_2^2$.
- A non-commutative polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ is a combination of words.
- ncPIT: Given a non-commutative polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ test if $f \equiv 0$. E.g., $x_1 x_2 - x_2 x_1 \not\equiv 0$ in the non-commutative world.
- Non-commutative circuit: arithmetic circuit whose $\times$ gate respects the ordering.
- (Amitsur-Levitski '50) Let $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ be non-zero polynomial of degree $\leq 2d - 1$. Then, there exists $(A_1, \ldots, A_n)$ $\in Mat_d^n(\mathbb{F})$ such that $f(A_1, \ldots, A_n) \neq 0$ as a matrix.

Non-commutative PIT

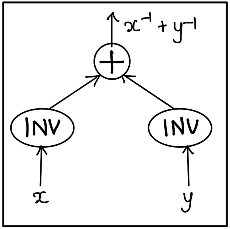
- Set of non-commuting variables $\{x_1, \ldots, x_n\}$ i.e., $x_i x_j \neq x_j x_i \ \forall i \neq j$. E.g., $(x_1 + x_2)(x_1 - x_2) \neq x_1^2 - x_2^2$.
- A non-commutative polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ is a combination of words.
- ncPIT: Given a non-commutative polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ test if $f \equiv 0$. E.g., $x_1 x_2 - x_2 x_1 \not\equiv 0$ in the non-commutative world.
- Non-commutative circuit: arithmetic circuit whose $\times$ gate respects the ordering.
- (Amitsur-Levitski '50) Let $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ be non-zero polynomial of degree $\leq 2d - 1$. Then, there exists $(A_1, \ldots, A_n)$ $\in Mat_d^n(\mathbb{F})$ such that $f(A_1, \ldots, A_n) \neq 0$ as a matrix.
- (Bogdanov, Wee 2005) Randomized polynomial time algorithm for ncPIT on circuits with polynomial degree.

Non-commutative PIT

- Set of non-commuting variables $\{x_1, \ldots, x_n\}$ i.e., $x_i x_j \neq x_j x_i \ \forall i \neq j$. E.g., $(x_1 + x_2)(x_1 - x_2) \neq x_1^2 - x_2^2$.
- A non-commutative polynomial $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ is a combination of words.
- ncPIT: Given a non-commutative polynomial $f \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ test if $f \equiv 0$. E.g., $x_1 x_2 - x_2 x_1 \not\equiv 0$ in the non-commutative world.
- Non-commutative circuit: arithmetic circuit whose $\times$ gate respects the ordering.
- (Amitsur-Levitski '50) Let $f(x_1, \ldots, x_n) \in \mathbb{F}\langle x_1, \ldots, x_n \rangle$ be non-zero polynomial of degree $\leq 2d - 1$. Then, there exists $(A_1, \ldots, A_n)$ $\in Mat_d^n(\mathbb{F})$ such that $f(A_1, \ldots, A_n) \neq 0$ as a matrix.
- (Bogdanov, Wee 2005) Randomized polynomial time algorithm for ncPIT on circuits with polynomial degree.
- **Open:** Randomized polynomial time algorithm for ncPIT on circuits of polynomial size.

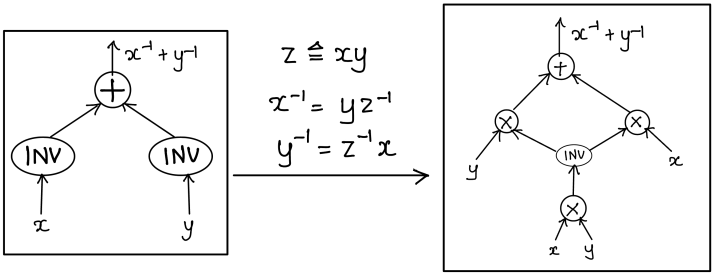
Non-Commutative circuits with division

- Circuits with $+, \times$(ordered multiplication gates) and INV gates. An INV gate has one input and computes g^{-1} on input g.



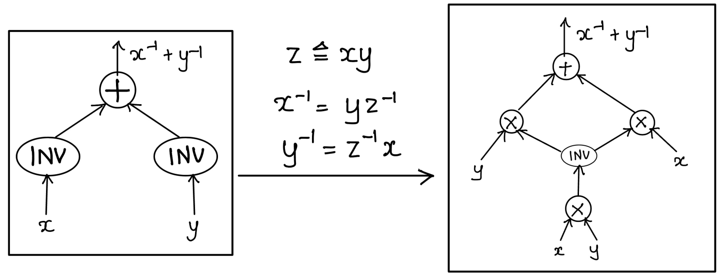
Non-Commutative circuits with division

- Circuits with $+, \times$(ordered multiplication gates) and INV gates. An INV gate has one input and computes g^{-1} on input g.



Non-Commutative circuits with division

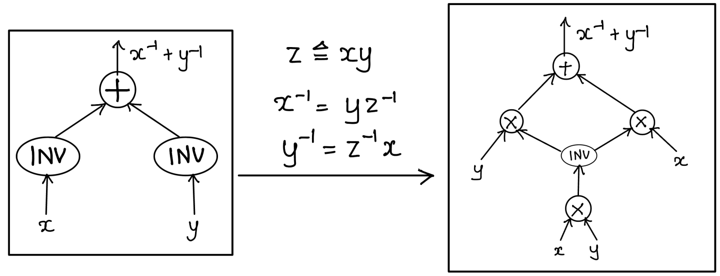
- Circuits with $+, \times$(ordered multiplication gates) and INV gates. An INV gate has one input and computes g^{-1} on input g.



- Hua's identity: $(x + xy^{-1}x)^{-1} \equiv x^{-1} + (x + y)^{-1}$.

Non-Commutative circuits with division

- Circuits with $+, \times$(ordered multiplication gates) and INV gates. An INV gate has one input and computes g^{-1} on input g.



- Hua's identity: $(x + xy^{-1}x)^{-1} \equiv x^{-1} + (x + y)^{-1}$.
- Nested inversions cannot always be eliminated. e.g., $(u + xy^{-1}z)^{-1}$.
- Inversion height: number if nested inversions.

Rational Identity Testing

- A rational expression $r(x_1, \ldots, x_n)$ computes the zero function[1] if
  - r has a nonempty domain of definition
  - for each $d \in \mathbb{N}$ and substitution $(A_, \ldots, A_n) \in (Mat_d(\mathbb{F}))^n$, $r(A_, \ldots, A_n) = 0$ as a matrix when defined.

[1](zero function in the free skew-field)

[2]formula is a circuit whose underlying graph is a tree.

Rational Identity Testing

- A rational expression $r(x_1, \ldots, x_n)$ computes the zero function[1] if
  - r has a nonempty domain of definition
  - for each $d \in \mathbb{N}$ and substitution $(A_, \ldots, A_n) \in (Mat_d(\mathbb{F}))^n$, $r(A_, \ldots, A_n) = 0$ as a matrix when defined.
- RIT: Given a non-commutative circuit(with inverses) computing r decide if $r \equiv 0$.

[1](zero function in the free skew-field)

[2]formula is a circuit whose underlying graph is a tree.

Rational Identity Testing

- A rational expression $r(x_1, \ldots, x_n)$ computes the zero function[1] if
  - r has a nonempty domain of definition
  - for each $d \in \mathbb{N}$ and substitution $(A_, \ldots, A_n) \in (Mat_d(\mathbb{F}))^n$, $r(A_, \ldots, A_n) = 0$ as a matrix when defined.
- RIT: Given a non-commutative circuit(with inverses) computing r decide if $r \equiv 0$.
- **Open:** Subexponential-time randomized white-box algorithm for noncommutative circuits.

[1](zero function in the free skew-field)

[2]formula is a circuit whose underlying graph is a tree.

Rational Identity Testing

- A rational expression $r(x_1, \ldots, x_n)$ computes the zero function[1] if
  - r has a nonempty domain of definition
  - for each $d \in \mathbb{N}$ and substitution $(A_, \ldots, A_n) \in (Mat_d(\mathbb{F}))^n$, $r(A_, \ldots, A_n) = 0$ as a matrix when defined.
- RIT: Given a non-commutative circuit(with inverses) computing r decide if $r \equiv 0$.
- **Open:** Subexponential-time randomized white-box algorithm for noncommutative circuits.

(Garg et al. '20, Ivanyos et al. '18) Deterministic polynomial time algorithm in the white-box model for non-commutative formula[2].

(Derksen, Makam '17) Randomized polynomial time in black-box model for non-commutative formula.

[1](zero function in the free skew-field)

[2]formula is a circuit whose underlying graph is a tree.

Lower Bounds $\implies$ RIT algorithm

- A polynomial identity for $d \times d$ matrix algebra is a noncommutative polynomial $p(x_1, \ldots, x_n)$ that vanishes on $d \times d$ matrix substitutions.

Lower Bounds $\implies$ RIT algorithm

- A polynomial identity for $d \times d$ matrix algebra is a noncommutative polynomial $p(x_1, \ldots, x_n)$ that vanishes on $d \times d$ matrix substitutions.
- $s(x_1, \ldots, x_{2d}) = \sum_{\sigma} \mathrm{sgn}(\sigma) x_{\sigma(1)} \cdots x_{\sigma(2d)}$ is a polynomial identity for $\mathbb{F}^{d \times d}$.

Lower Bounds $\implies$ RIT algorithm

- A polynomial identity for $d \times d$ matrix algebra is a noncommutative polynomial $p(x_1, \ldots, x_n)$ that vanishes on $d \times d$ matrix substitutions.
- $s(x_1, \ldots, x_{2d}) = \sum_{\sigma} \mathrm{sgn}(\sigma) x_{\sigma(1)} \cdots x_{\sigma(2d)}$ is a polynomial identity for $\mathbb{F}^{d \times d}$.

Conjecture(Bogdanov, Wee '05): The minimum size of a branching program of a *polynomial identity* for the $d \times d$ matrix algebra is $2^{\Omega(d)}$.

Lower Bounds $\implies$ RIT algorithm

- A polynomial identity for $d \times d$ matrix algebra is a noncommutative polynomial $p(x_1, \ldots, x_n)$ that vanishes on $d \times d$ matrix substitutions.
- $s(x_1, \ldots, x_{2d}) = \sum_{\sigma} \mathrm{sgn}(\sigma) x_{\sigma(1)} \cdots x_{\sigma(2d)}$ is a polynomial identity for $\mathbb{F}^{d \times d}$.

Conjecture(Bogdanov, Wee '05): The minimum size of a branching program of a *polynomial identity* for the $d \times d$ matrix algebra is $2^{\Omega(d)}$.

Theorem (Arvind, Chatterjee, Ghosal, Mukhopadhyay, R.,)

If BW conjecture is true then there is a deterministic subexponential time blackbox RIT algorithm for rational formulas of of size s over n variables and inversion height $\approx \frac{\log s}{\log \log s}$.

Lower Bounds $\implies$ RIT algorithm

- A polynomial identity for $d \times d$ matrix algebra is a noncommutative polynomial $p(x_1, \ldots, x_n)$ that vanishes on $d \times d$ matrix substitutions.
- $s(x_1, \ldots, x_{2d}) = \sum_{\sigma} \mathrm{sgn}(\sigma) x_{\sigma(1)} \cdots x_{\sigma(2d)}$ is a polynomial identity for $\mathbb{F}^{d \times d}$.

Conjecture(Bogdanov, Wee '05): The minimum size of a branching program of a *polynomial identity* for the $d \times d$ matrix algebra is $2^{\Omega(d)}$.

Theorem (Arvind, Chatterjee, Ghosal, Mukhopadhyay, R.,)

If BW conjecture is true then there is a deterministic subexponential time blackbox RIT algorithm for rational formulas of of size s over n variables and inversion height $\approx \frac{\log s}{\log \log s}$.

(Hrubes, Wigderson) A rational formula of size s has inversion height $O(\log s)$.

Thank you