Lower Bounds for some Algebraic Models of Computation

Prerona Chatterjee

May 2, 2024

Q: Given a computational problem and constraints on the computational power at hand,

Complexity Theory

Q: Given a computational problem and constraints on the computational power at hand,

- design a computational model that captures the constraints

Q: Given a computational problem and constraints on the computational power at hand,

- design a computational model that captures the constraints
- study the amount of resource required by the model to complete the task.

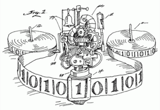
Complexity Theory

Q: Given a computational problem and constraints on the computational power at hand,

- design a computational model that captures the constraints
- study the amount of resource required by the model to complete the task.

Traditional Time Complexity

Given a boolean function f on n inputs, how many steps are required by a Turing machine to compute the f (in terms of n)?



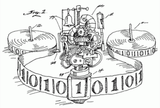
Complexity Theory

Q: Given a computational problem and constraints on the computational power at hand,

- design a computational model that captures the constraints
- study the amount of resource required by the model to complete the task.

Traditional Time Complexity

Given a boolean function f on n inputs, how many steps are required by a Turing machine to compute the f (in terms of n)?



Traditional Space Complexity

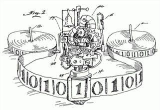
Given a boolean function f on n inputs, how much space is required by a Turing machine to compute the f (in terms of n)?

Complexity Theory

Q: Given a computational problem and constraints on the computational power at hand,

- design a computational model that captures the constraints
- study the amount of resource required by the model to complete the task.

Traditional Time Complexity
Given a boolean function f on n inputs, how many steps are required by a Turing machine to compute the f (in terms of n)?



Traditional Space Complexity
Given a boolean function f on n inputs, how much space is required by a Turing machine to compute the f (in terms of n)?

Circuit Complexity

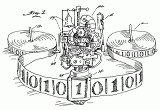
Complexity Theory

Q: Given a computational problem and constraints on the computational power at hand,

- design a computational model that captures the constraints
- study the amount of resource required by the model to complete the task.

Traditional Time Complexity

Given a boolean function f on n inputs, how many steps are required by a Turing machine to compute the f (in terms of n)?



Traditional Space Complexity

Given a boolean function f on n inputs, how much space is required by a Turing machine to compute the f (in terms of n)?

Circuit Complexity

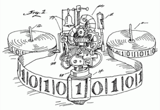
Communication Complexity

Complexity Theory

Q: Given a computational problem and constraints on the computational power at hand,

- design a computational model that captures the constraints
- study the amount of resource required by the model to complete the task.

Traditional Time Complexity
Given a boolean function f on n inputs, how many steps are required by a Turing machine to compute the f (in terms of n)?



Traditional Space Complexity
Given a boolean function f on n inputs, how much space is required by a Turing machine to compute the f (in terms of n)?

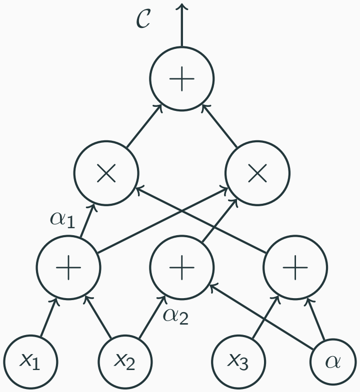
Circuit Complexity **Communication Complexity** **Quantum Complexity**

Algebraic Models of Computation

Q: Given $f(\mathbf{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d, how many $+, \times, -$ gates are needed to compute f?

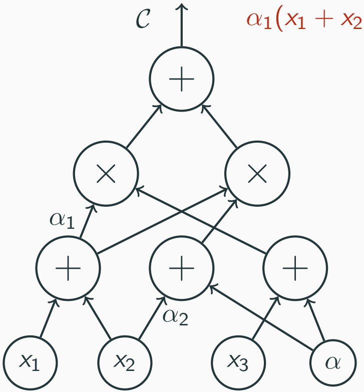
Algebraic Models of Computation

Q: Given $f(\mathbf{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d, how many $+, \times, -$ gates are needed to compute f?



Algebraic Models of Computation

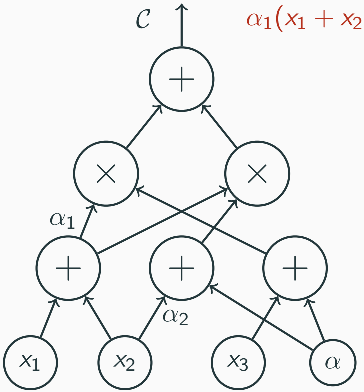
Q: Given $f(\mathbf{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d, how many $+, \times, -$ gates are needed to compute f?



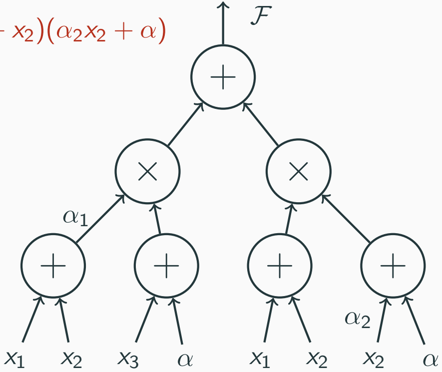
$\alpha_1(x_1 + x_2)(x_3 + \alpha) + (x_1 + x_2)(\alpha_2 x_2 + \alpha)$

Algebraic Models of Computation

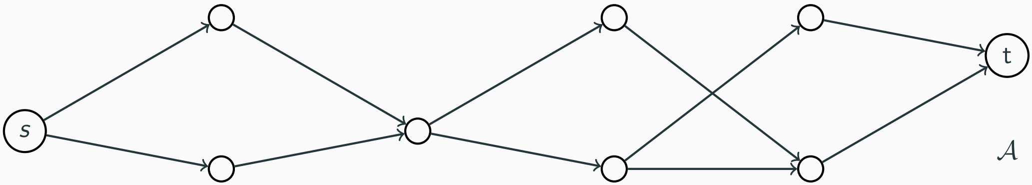
Q: Given $f(\mathbf{x}) \in \mathbb{F}[x_1, \ldots, x_n]$ of degree d, how many $+, \times, -$ gates are needed to compute f?



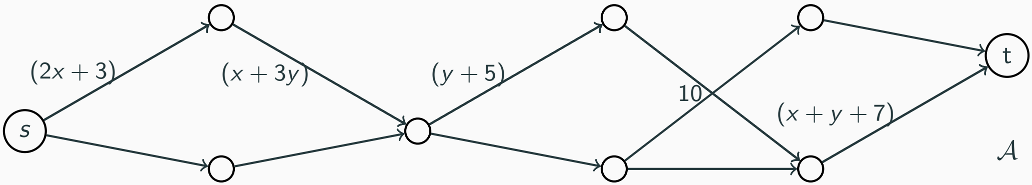
$$\alpha_1(x_1 + x_2)(x_3 + \alpha) + (x_1 + x_2)(\alpha_2 x_2 + \alpha)$$



Algebraic Branching Programs

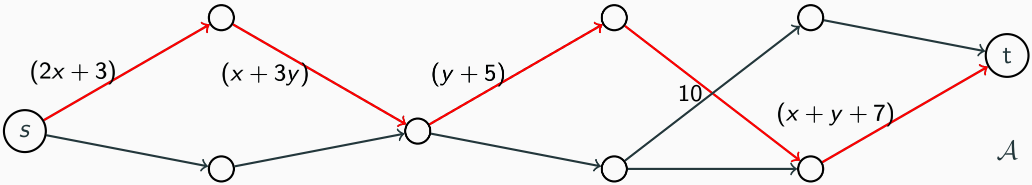


Algebraic Branching Programs



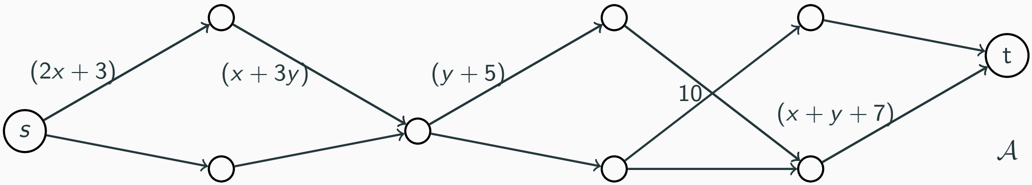
- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$

Algebraic Branching Programs



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

Algebraic Branching Programs



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

Lower Bounds in Algebraic Circuit Complexity

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

Lower Bounds in Algebraic Circuit Complexity

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

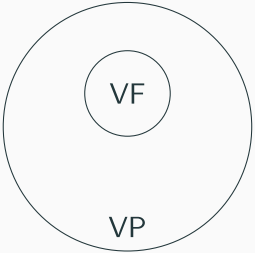
VP: Polynomials computable by circuits of size $\mathrm{poly}(n, d)$.

Lower Bounds in Algebraic Circuit Complexity

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

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

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

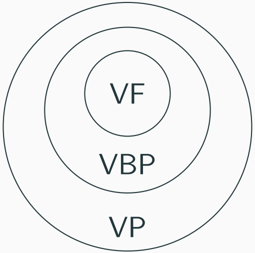


Objects of Study: Polynomials over n variables of degree d.

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

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

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



Lower Bounds in Algebraic Circuit Complexity

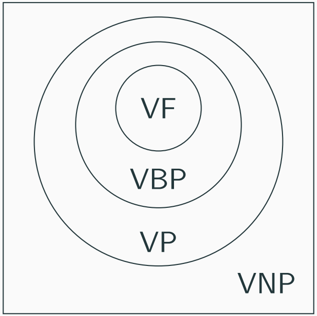
Objects of Study: Polynomials over n variables of degree d.

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

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

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

VNP: Explicit Polynomials



Lower Bounds in Algebraic Circuit Complexity

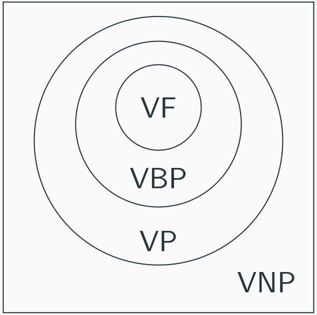
Objects of Study: Polynomials over n variables of degree d.

VF: Polynomials computable by formulas of size poly(n, d).

VBP: Polynomials computable by ABPs of size poly(n, d).

VP: Polynomials computable by circuits of size poly(n, d).

VNP: Explicit Polynomials



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

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

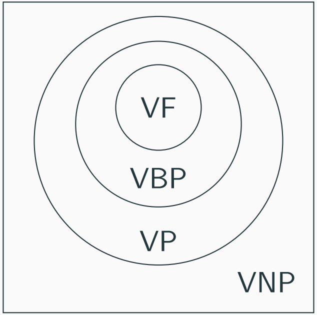
VF: Polynomials computable by formulas of size $\mathrm{poly}(n, d)$.

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

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

VNP: Explicit Polynomials

$$\boxed{\mathrm{VP} = \mathrm{VNP} \overset{\text{G.R.H.}}{\Longrightarrow} \mathrm{P} = \mathrm{NP}}$$



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

Towards Better ABP Lower Bounds

[C-Kumar-She-Volk 22]: Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(nd)$ vertices.

Towards Better ABP Lower Bounds

[C-Kumar-She-Volk 22]: Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(nd)$ vertices.

[Bhargav-Dwivedi-Saxena 24]: Super polynomial lower bound against total-width of $\sum$ osmABP for a polynomial of degree $d = O\left(\frac{\log n}{\log \log n}\right) \implies$ super-polynomial lower bound against ABPs.

Towards Better ABP Lower Bounds

[C-Kumar-She-Volk 22]: Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(nd)$ vertices.

[Bhargav-Dwivedi-Saxena 24]: Super polynomial lower bound against total-width of $\sum$osmABP for a polynomial of degree $d = O\left(\frac{\log n}{\log \log n}\right) \implies$ super-polynomial lower bound against ABPs.

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

- $G_{n,d}$ is computable by a set-multilinear ABP of size $\mathrm{poly}(n)$,
- any $\sum$osmABP computing $G_{n,d}$ must have super-polynomial total-width.

Set-Multilinearity

The variable set is divided into buckets.

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

Set-Multilinearity

The variable set is divided into buckets.

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

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

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

Set-Multilinearity

The variable set is divided into buckets.

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

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

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

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

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

Super-Polynomial Lower Bound against $\sum$osmABPs

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

Super-Polynomial Lower Bound against $\sum$osmABPs

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

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

Super-Polynomial Lower Bound against $\sum$osmABPs

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

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

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

Super-Polynomial Lower Bound against $\sum$osmABPs

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

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

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

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

Super-Polynomial Lower Bound against $\sum$osmABPs

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

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

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

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

Super-Polynomial Lower Bound against $\sum$osmABPs

For $\sigma \in S_d$, an ABP is σ-ordered set-multilinear with respect to $\{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$ if

- there are d layers in the ABP
- every edge in layer i is labelled by a homogeneous linear form in $\mathbf{x}_{\sigma(i)}$

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

[C-Kush-Saraf-Shpilka 24]: For $\omega(\log n) = d \leq n$, there is a polynomial $G_{n,d}(\mathbf{x})$ which is set-multilinear w.r.t $\mathbf{x} = \{\mathbf{x}_1, \ldots, \mathbf{x}_d\}$, where $|\mathbf{x}_i| \leq n$ for every $i \in [d]$, such that:

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

Open Threads

1. PIT for $\sum$ osmABP?

Open Threads

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

Open Threads

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

Question?