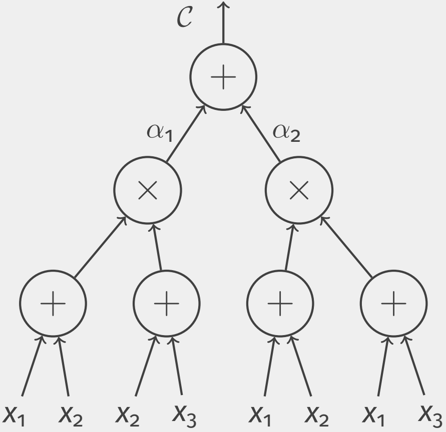
A Quadratic Lowerbound against Algebraic Branching Programs

Prerona Chatterjee
With *Mrinal Kumar* (IITB), *Adrian She* (UoT), *Ben Lee Volk* (Caltech)

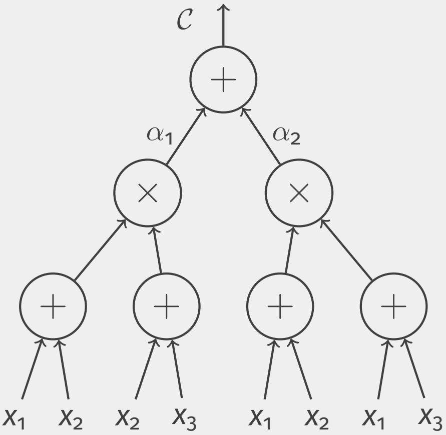
Tata Institute of Fundamental Research, Mumbai

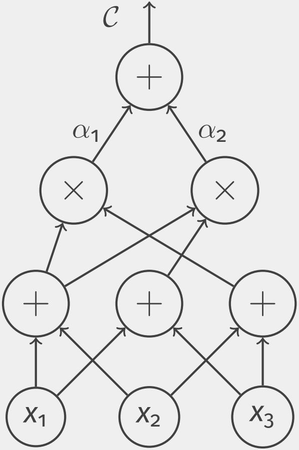
April 23, 2020

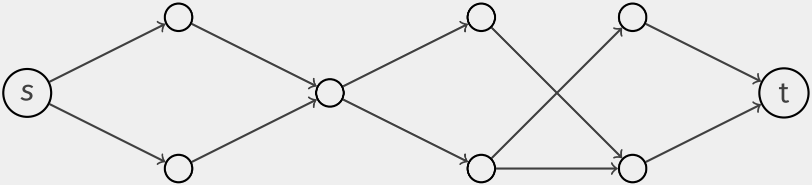
Algebraic Formulas and Algebraic Circuits



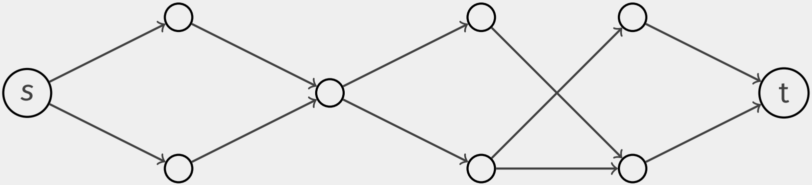
Algebraic Formulas and Algebraic Circuits





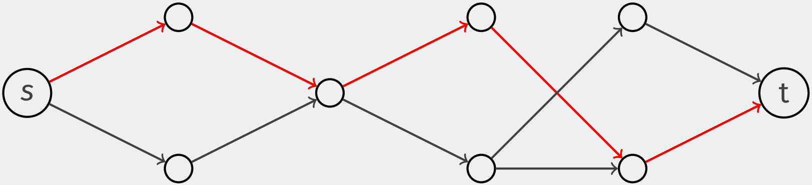

s
t

Algebraic Branching Programs

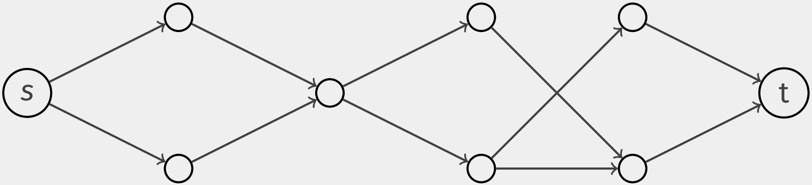


- Label on each edge: An affine linear form in $\{x_1, x_2, \ldots, x_n\}$

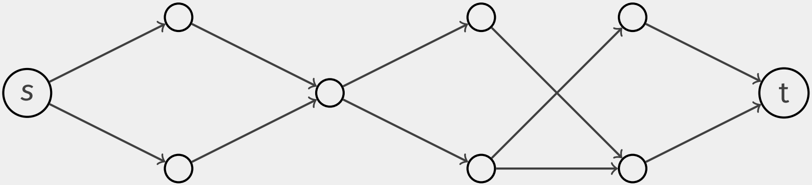
Algebraic Branching Programs



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



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



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

VF $\subseteq$ VBP $\subseteq$ VP

For Circuits [Baur-Strassen]:
Any fan-in 2 circuit computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ multiplication gates.

What was known earlier?

For Circuits [Baur-Strassen]:
Any fan-in 2 circuit computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ multiplication gates.

For Formulas [Kalorkoti]:
Any formula computing $\mathrm{Det}_{n \times n}$ requires $\Omega(n^3)$ wires.

What was known earlier?

For Circuits [Baur-Strassen]:
Any fan-in 2 circuit computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ multiplication gates.

For Formulas [Kalorkoti]:
Any formula computing $\mathrm{Det}_{n\times n}$ requires $\Omega(n^3)$ wires.
[SY]: Any formula computing $\sum_{i=1}^{n} \sum_{j=1}^{n} x_i^j y_j$ requires $\Omega(n^2)$ wires.

What was known earlier?

For Circuits [Baur-Strassen]:
Any fan-in 2 circuit computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ multiplication gates.

For Formulas [Kalorkoti]:
Any formula computing $\mathrm{Det}_{n\times n}$ requires $\Omega(n^3)$ wires.
[SY]: Any formula computing $\sum_{i=1}^{n} \sum_{j=1}^{n} x_i^j y_j$ requires $\Omega(n^2)$ wires.

Note: For an n-variate individual degree d polynomial, a lowerbound of $\Omega(nd)$ is trivial for formulas.

Old and New Results for ABPs

General ABPs [Baur-Strassen]:
Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ wires.

Old and New Results for ABPs

General ABPs [Baur-Strassen]:
Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ wires.

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

Old and New Results for ABPs

General ABPs [Baur-Strassen]:
Any ABP computing $\sum_{i=1}^{n} x_i^d$ requires $\Omega(n \log d)$ wires.

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

Our Main Result

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

Theorem (Kumar)

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

The Base Case: Proof Plan

Theorem (Kumar)

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

- Partition the vertices according to the layer it is in.

The Base Case: Proof Plan

Theorem (Kumar)

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

- Partition the vertices according to the layer it is in.
- Say $V_k = \{v_1, \ldots, v_{t_k}\}$ are the vertices in the k-th layer.

Theorem (Kumar)

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

- Partition the vertices according to the layer it is in.
- Say $V_k = \{v_1, \ldots, v_{t_k}\}$ are the vertices in the k-th layer.
- Show that $t_k \geq n/2$ for every $k \in [d-1]$.

Base Case: Proof Overview

$P_{n,d} = \sum_{i=1}^{n} x_i^d$

Base Case: Proof Overview

$P_{n,d} = \sum_{i=1}^{n} x_i^d$

Fix $k \in [d-1]$

Base Case: Proof Overview

$P_{n,d} = \sum_{i=1}^{n} x_i^d$

Fix $k \in [d-1]$

$V_k = \{v_1, \ldots, v_{t_k}\}$

Base Case: Proof Overview

$P_{n,d} = \sum_{i=1}^{n} x_i^d$ Fix $k \in [d-1]$ $V_k = \{v_1, \ldots, v_{t_k}\}$

$$P_{n,d} = \sum_{i=1}^{t_k} [s, v_i] \cdot [v_i, t]$$

Base Case: Proof Overview

$P_{n,d} = \sum_{i=1}^{n} x_i^d$

Fix $k \in [d-1]$

$V_k = \{v_1, \ldots, v_{t_k}\}$

$$P_{n,d} = \sum_{i=1}^{t_k} [s, v_i] \cdot [v_i, t] = \sum_{i=1}^{t_k} g_i \cdot h_i$$

Base Case: Proof Overview

$P_{n,d} = \sum_{i=1}^{n} x_i^d$ Fix $k \in [d-1]$ $V_k = \{v_1, \ldots, v_{t_k}\}$

$$P_{n,d} = \sum_{i=1}^{t_k} [s, v_i] \cdot [v_i, t] = \sum_{i=1}^{t_k} g_i \cdot h_i = \sum_{i=1}^{t_k} (g_i' + \alpha_i) \cdot (h_i' + \beta_i)$$

Base Case: Proof Overview

$P_{n,d} = \sum_{i=1}^{n} x_i^d$ Fix $k \in [d-1]$ $V_k = \{v_1, \ldots, v_{t_k}\}$

$$P_{n,d} = \sum_{i=1}^{t_k} [s, v_i] \cdot [v_i, t] = \sum_{i=1}^{t_k} g_i \cdot h_i = \sum_{i=1}^{t_k} (g'_i + \alpha_i) \cdot (h'_i + \beta_i)$$

Note: $\deg(g'_i), \deg(h'_i) \leq [d-1]$ for every $i \in [d-1]$.

$P_{n,d} = \sum_{i=1}^{n} x_i^d$ Fix $k \in [d-1]$ $V_k = \{v_1, \ldots, v_{t_k}\}$

$$P_{n,d} = \sum_{i=1}^{t_k} [s, v_i] \cdot [v_i, t] = \sum_{i=1}^{t_k} g_i \cdot h_i = \sum_{i=1}^{t_k} (g_i' + \alpha_i) \cdot (h_i' + \beta_i)$$

For $R = \sum_{i=1}^{t_k} (\alpha_i \cdot h_i' + \beta_i \cdot g_i' + \alpha_i \cdot \beta_i)$,

$$P_{n,d} = \left(\sum_{i=1}^{t_k} g_i' \cdot h_i' \right) + R$$

Base Case: Proof Overview

$P_{n,d} = \sum_{i=1}^{n} x_i^d$ Fix $k \in [d-1]$ $V_k = \{v_1, \ldots, v_{t_k}\}$

$$P_{n,d} = \sum_{i=1}^{t_k} [s, v_i] \cdot [v_i, t] = \sum_{i=1}^{t_k} g_i \cdot h_i = \sum_{i=1}^{t_k} (g_i' + \alpha_i) \cdot (h_i' + \beta_i)$$

For $R = \sum_{i=1}^{t_k} (\alpha_i \cdot h_i' + \beta_i \cdot g_i' + \alpha_i \cdot \beta_i)$,

$$P_{n,d} = \left(\sum_{i=1}^{t_k} g_i' \cdot h_i' \right) + R \implies P_{n,d} - R = \sum_{i=1}^{t_k} g_i' \cdot h_i'$$

Base Case: Proof Overview

$P_{n,d} = \sum_{i=1}^{n} x_i^d$ Fix $k \in [d-1]$ $V_k = \{v_1, \ldots, v_{t_k}\}$

$$P_{n,d} = \sum_{i=1}^{t_k} [s, v_i] \cdot [v_i, t] = \sum_{i=1}^{t_k} g_i \cdot h_i = \sum_{i=1}^{t_k} (g_i' + \alpha_i) \cdot (h_i' + \beta_i)$$

For $R = \sum_{i=1}^{t_k} (\alpha_i \cdot h_i' + \beta_i \cdot g_i' + \alpha_i \cdot \beta_i)$,

$$P_{n,d} = \left(\sum_{i=1}^{t_k} g_i' \cdot h_i' \right) + R \implies P_{n,d} - R = \sum_{i=1}^{t_k} g_i' \cdot h_i'$$

where $\deg(R) \leq d - 1$.

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g_i' \cdot h_i'$$

Base Case: Proof Overview (contd.)

$P_{n,d} - R = \sum_{i=1}^{t_k} g'_i \cdot h'_i$

$P_{n,d} = \sum_{i=1}^{n} x_i^d$

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g'_i \cdot h'_i \qquad\qquad P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g'_1, \ldots, g'_{t_k}, h'_1, \ldots, h'_{t_k})$$

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g'_i \cdot h'_i \qquad P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g'_1, \ldots, g'_{t_k}, h'_1, \ldots, h'_{t_k}) \qquad \mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$$

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g_i' \cdot h_i'$$

$$P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g_1', \ldots, g_{t_k}', h_1', \ldots, h_{t_k}')$$

$$\mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$$

$$\mathcal{V} \subseteq \mathcal{V}'$$

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g_i' \cdot h_i' \qquad P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g_1', \ldots, g_{t_k}', h_1', \ldots, h_{t_k}') \qquad \mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$$

$$\mathcal{V} \subseteq \mathcal{V}' \implies \dim(\mathcal{V}) \leq \dim(\mathcal{V}')$$

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g_i' \cdot h_i' \qquad\qquad P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g_1', \ldots, g_{t_k}', h_1', \ldots, h_{t_k}') \qquad\qquad \mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$$

$$\mathcal{V} \subseteq \mathcal{V}' \implies \dim(\mathcal{V}) \leq \dim(\mathcal{V}')$$

$$R = 0$$

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g'_i \cdot h'_i \qquad\qquad P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g'_1, \ldots, g'_{t_k}, h'_1, \ldots, h'_{t_k}) \qquad\qquad \mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$$

$$\mathcal{V} \subseteq \mathcal{V}' \implies \dim(\mathcal{V}) \leq \dim(\mathcal{V}')$$

$$R = 0 \quad \Rightarrow \quad \mathcal{V}' = \left\{d \cdot x_1^{d-1}, \ldots, d \cdot x_n^{d-1}\right\}$$

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g_i' \cdot h_i' \qquad\qquad P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g_1', \ldots, g_{t_k}', h_1', \ldots, h_{t_k}') \qquad\qquad \mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$$

$$\mathcal{V} \subseteq \mathcal{V}' \implies \dim(\mathcal{V}) \leq \dim(\mathcal{V}')$$

$$R = 0 \quad \Rightarrow \quad \mathcal{V}' = \left\{d \cdot x_1^{d-1}, \ldots, d \cdot x_n^{d-1}\right\} \quad \Rightarrow \quad \dim(\mathcal{V}') = 0$$

Base Case: Proof Overview (contd.)

$P_{n,d} - R = \sum_{i=1}^{t_k} g'_i \cdot h'_i$

$P_{n,d} = \sum_{i=1}^{n} x_i^d$

$\mathcal{V} = \mathbb{V}(g'_1, \ldots, g'_{t_k}, h'_1, \ldots, h'_{t_k})$

$\mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$

$$\mathcal{V} \subseteq \mathcal{V}' \implies \dim(\mathcal{V}) \leq \dim(\mathcal{V}')$$

$$\deg(R) \leq d - 1 \quad \implies \quad \dim(\mathcal{V}') = 0$$

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g_i' \cdot h_i' \qquad\qquad P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g_1', \ldots, g_{t_k}', h_1', \ldots, h_{t_k}') \qquad\qquad \mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$$

$$\mathcal{V} \subseteq \mathcal{V}' \implies \dim(\mathcal{V}) \leq \dim(\mathcal{V}')$$

$$\deg(R) \leq d - 1 \implies \dim(\mathcal{V}') = 0$$

$$\dim(\mathcal{V}) \geq n - 2t_k$$

Base Case: Proof Overview (contd.)

$$P_{n,d} - R = \sum_{i=1}^{t_k} g'_i \cdot h'_i \qquad\qquad P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g'_1, \ldots, g'_{t_k}, h'_1, \ldots, h'_{t_k}) \qquad\qquad \mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$$

$$\mathcal{V} \subseteq \mathcal{V}' \implies \dim(\mathcal{V}) \leq \dim(\mathcal{V}')$$

$$\deg(R) \leq d - 1 \implies \dim(\mathcal{V}') = 0$$

$$\dim(\mathcal{V}) \geq n - 2t_k \implies n - 2t_k \leq 0$$

$$P_{n,d} - R = \sum_{i=1}^{t_k} g'_i \cdot h'_i \qquad\qquad P_{n,d} = \sum_{i=1}^{n} x_i^d$$

$$\mathcal{V} = \mathbb{V}(g'_1, \ldots, g'_{t_k}, h'_1, \ldots, h'_{t_k}) \qquad\qquad \mathcal{V}' = \mathbb{V}(\{\partial_{x_i}(P_{n,d} - R)\})$$

$$\mathcal{V} \subseteq \mathcal{V}' \implies \dim(\mathcal{V}) \leq \dim(\mathcal{V}')$$

$$\deg(R) \leq d - 1 \implies \dim(\mathcal{V}') = 0$$

$$\dim(\mathcal{V}) \geq n - 2t_k \implies n - 2t_k \leq 0 \implies t_k \geq n/2$$

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

General Case: Proof Idea

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

- Iteratively reduce the number of layers of the ABP by a constant fraction till the number of layers become n

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

- Iteratively reduce the number of layers of the ABP by a constant fraction till the number of layers become n
- Size does not change in this process

General Case: Proof Idea

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

- Iteratively reduce the number of layers of the ABP by a constant fraction till the number of layers become n
- Size does not change in this process
- However, the polynomial being computed changes

General Case: Proof Idea

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

- Iteratively reduce the number of layers of the ABP by a constant fraction till the number of layers become n
- Size does not change in this process
- However, the polynomial being computed changes

$$f = \sum_{i=1}^{n} x_i^d + \sum_{i=1}^{r} A_i(\mathbf{x}) \cdot B_i(\mathbf{x}) + \delta \quad \text{where} \quad A_i(0) = 0 = B_i(0)$$

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

- Iteratively reduce the number of layers of the ABP by a constant fraction till the number of layers become n
- Size does not change in this process
- However, the polynomial being computed changes

$$f = \sum_{i=1}^{n} x_i^d + \sum_{i=1}^{r} A_i(\mathbf{x}) \cdot B_i(\mathbf{x}) + \delta \quad \text{where} \quad A_i(0) = 0 = B_i(0)$$

- **Updated Base Case:** $V_k = \{v_1, \ldots, v_{t_k}\}$ — k-th layer vertices

$$\text{For every } k \in [d-1], \quad t_k \geq n/2 - r$$

The Main Claim

ℓ-th step

ℓ-th step

Given: ABP $\mathcal{A}_\ell$ of size = s_ℓ
no. of layers = d_ℓ
no. of error terms = r_ℓ

THE MAIN CLAIM

ℓ-th step

Given: ABP $\mathcal{A}_\ell$ of size = s_ℓ
no. of layers = d_ℓ
no. of error terms = r_ℓ

Want to construct: ABP $\mathcal{A}_{\ell+1}$ of size = $s_{\ell+1}$

The Main Claim

ℓ-th step

Given: ABP $\mathcal{A}_\ell$ of size = s_ℓ
no. of layers = d_ℓ
no. of error terms = r_ℓ

Want to construct: ABP $\mathcal{A}_{\ell+1}$ of size = $s_{\ell+1} \leq s_\ell$

The Main Claim

ℓ-th step

Given: ABP $\mathcal{A}_\ell$ of size = s_ℓ
no. of layers = d_ℓ
no. of error terms = r_ℓ

Want to construct: ABP $\mathcal{A}_{\ell+1}$ of size = $s_{\ell+1} \leq s_\ell$
no. of layers = $d_{\ell+1}$

ℓ-th step

Given: ABP $\mathcal{A}_\ell$ of size = s_ℓ
no. of layers = d_ℓ
no. of error terms = r_ℓ

Want to construct: ABP $\mathcal{A}_{\ell+1}$ of size = $s_{\ell+1} \leq s_\ell$
no. of layers = $d_{\ell+1} \leq \frac{2}{3} d_\ell$

ℓ-th step

Given: ABP $\mathcal{A}_\ell$ of size = s_ℓ
no. of layers = d_ℓ
no. of error terms = r_ℓ

Want to construct: ABP $\mathcal{A}_{\ell+1}$ of size = $s_{\ell+1} \leq s_\ell$
no. of layers = $d_{\ell+1} \leq \frac{2}{3} d_\ell$
no. of error terms = $r_{\ell+1}$

ℓ-th step

Given: ABP $\mathcal{A}_\ell$ of size = s_ℓ
no. of layers = d_ℓ
no. of error terms = r_ℓ

Want to construct: ABP $\mathcal{A}_{\ell+1}$ of size = $s_{\ell+1} \leq s_\ell$
no. of layers = $d_{\ell+1} \leq \frac{2}{3} d_\ell$
no. of error terms = $r_{\ell+1} \leq r_\ell + \frac{s_\ell}{d_\ell/3}$

Completing the Proof

No. of steps needed = L

No. of steps needed = L $L = \Theta(\log n)$

Completing the Proof

No. of steps needed = L $L = \Theta(\log n)$ $d_{L-1} > d$

Completing the Proof

No. of steps needed = L $\quad$ $L = \Theta(\log n)$ $\quad$ $d_{L-1} > d$

$s_0 \leq c \cdot nd$

Completing the Proof

No. of steps needed = L $\qquad L = \Theta(\log n)$ $\qquad d_{L-1} > d$

$$s_0 \leq c \cdot nd \qquad d_{\ell+1} \leq \frac{2}{3} d_\ell \implies \frac{1}{d_\ell} \leq \frac{2}{3} \cdot \frac{1}{d_{\ell+1}}$$

No. of steps needed = L $\quad L = \Theta(\log n) \quad d_{L-1} > d$

$$s_0 \leq c \cdot nd \qquad d_{\ell+1} \leq \frac{2}{3} d_\ell \implies \frac{1}{d_\ell} \leq \frac{2}{3} \cdot \frac{1}{d_{\ell+1}}$$

$$r_L = 3 \cdot s_0 \left(\frac{1}{d_0} + \frac{1}{d_1} \cdots + \frac{1}{d_{L-1}} \right)$$

No. of steps needed = L $\qquad L = \Theta(\log n)$ $\qquad d_{L-1} > d$

$$s_0 \leq c \cdot nd \qquad d_{\ell+1} \leq \frac{2}{3} d_\ell \implies \frac{1}{d_\ell} \leq \frac{2}{3} \cdot \frac{1}{d_{\ell+1}}$$

$$\begin{aligned} r_L &= 3 \cdot s_0 \left(\frac{1}{d_0} + \frac{1}{d_1} \cdots + \frac{1}{d_{L-1}} \right) \\ &\leq \frac{3 \cdot c \cdot nd}{d_{L-1}} \left(\left(\frac{2}{3}\right)^{L-1} + \left(\frac{2}{3}\right)^{L-2} + \cdots + 1 \right) \end{aligned}$$

Completing the Proof

No. of steps needed = L $\quad L = \Theta(\log n)$ $\quad d_{L-1} > d$

$$s_0 \le c \cdot nd \qquad d_{\ell+1} \le \frac{2}{3} d_\ell \implies \frac{1}{d_\ell} \le \frac{2}{3} \cdot \frac{1}{d_{\ell+1}}$$

$$\begin{aligned} r_L &= 3 \cdot s_0 \left(\frac{1}{d_0} + \frac{1}{d_1} \cdots + \frac{1}{d_{L-1}} \right) \\ &\le \frac{3 \cdot c \cdot nd}{d_{L-1}} \left(\left(\frac{2}{3}\right)^{L-1} + \left(\frac{2}{3}\right)^{L-2} + \cdots + 1 \right) < 9 \cdot c \cdot n \end{aligned}$$

Completing the Proof

No. of steps needed = L $\qquad L = \Theta(\log n) \qquad d_{L-1} > d$

$$s_0 \le c \cdot nd \qquad d_{\ell+1} \le \frac{2}{3} d_\ell \implies \frac{1}{d_\ell} \le \frac{2}{3} \cdot \frac{1}{d_{\ell+1}}$$

$$\begin{aligned} r_L &= 3 \cdot s_0 \left(\frac{1}{d_0} + \frac{1}{d_1} \cdots + \frac{1}{d_{L-1}} \right) \\ &\le \frac{3 \cdot c \cdot nd}{d_{L-1}} \left(\left(\frac{2}{3}\right)^{L-1} + \left(\frac{2}{3}\right)^{L-2} + \cdots + 1 \right) < 9 \cdot c \cdot n \end{aligned}$$

$$c = 0.01 \implies t_k \ge n/2 - 0.09 \cdot n$$

Completing the Proof

No. of steps needed = L $\quad L = \Theta(\log n) \quad d_{L-1} > d$

$$s_0 \leq c \cdot nd \qquad d_{\ell+1} \leq \frac{2}{3} d_\ell \implies \frac{1}{d_\ell} \leq \frac{2}{3} \cdot \frac{1}{d_{\ell+1}}$$

$$\begin{aligned} r_L &= 3 \cdot s_0 \left(\frac{1}{d_0} + \frac{1}{d_1} \cdots + \frac{1}{d_{L-1}} \right) \\ &\leq \frac{3 \cdot c \cdot nd}{d_{L-1}} \left(\left(\frac{2}{3}\right)^{L-1} + \left(\frac{2}{3}\right)^{L-2} + \cdots + 1 \right) < 9 \cdot c \cdot n \end{aligned}$$

$$c = 0.01 \implies t_k \geq n/2 - 0.09 \cdot n = \Omega(n)$$

Completing the Proof

No. of steps needed = L $\quad L = \Theta(\log n) \quad d_{L-1} > d$

$$s_0 \leq c \cdot nd \qquad d_{\ell+1} \leq \frac{2}{3} d_\ell \implies \frac{1}{d_\ell} \leq \frac{2}{3} \cdot \frac{1}{d_{\ell+1}}$$

$$\begin{aligned} r_L &= 3 \cdot s_0 \left(\frac{1}{d_0} + \frac{1}{d_1} \cdots + \frac{1}{d_{L-1}} \right) \\ &\leq \frac{3 \cdot c \cdot nd}{d_{L-1}} \left(\left(\frac{2}{3}\right)^{L-1} + \left(\frac{2}{3}\right)^{L-2} + \cdots + 1 \right) < 9 \cdot c \cdot n \end{aligned}$$

$$c = 0.01 \implies t_k \geq n/2 - 0.09 \cdot n = \Omega(n) \implies\impliedby$$

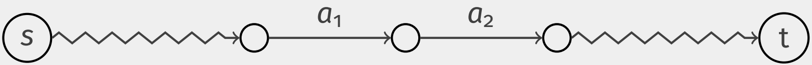
The Main Claim (once again...)

ℓ-th step

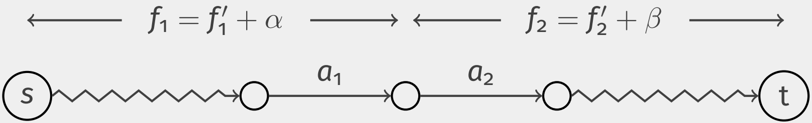
Given: ABP $\mathcal{A}_\ell$ of size = s_ℓ
no. of layers = d_ℓ
no. of error terms = r_ℓ

Want to construct: ABP $\mathcal{A}_{\ell+1}$ of size = $s_{\ell+1} \leq s_\ell$
no. of layers = $d_{\ell+1} \leq \frac{2}{3} d_\ell$
no. of error terms = $r_{\ell+1} \leq r_\ell + \frac{s_\ell}{d_\ell/3}$

Proving the Claim

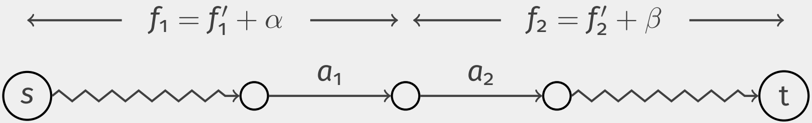


Proving the Claim



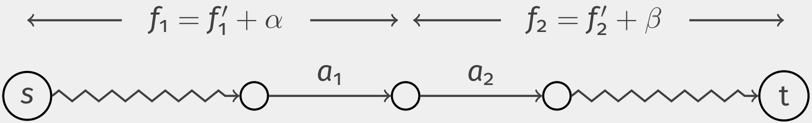
Proving the Claim

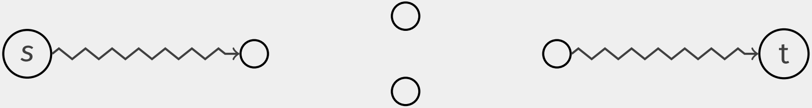
$\mathcal{A} = f_1 \cdot f_2$



Proving the Claim

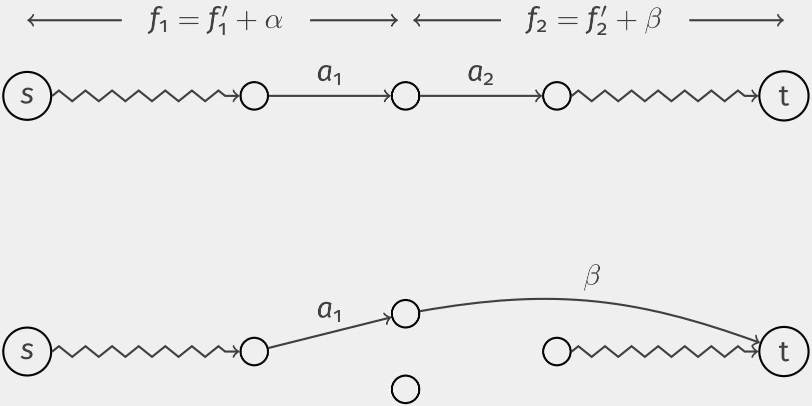
$\mathcal{A} = f_1 \cdot f_2$



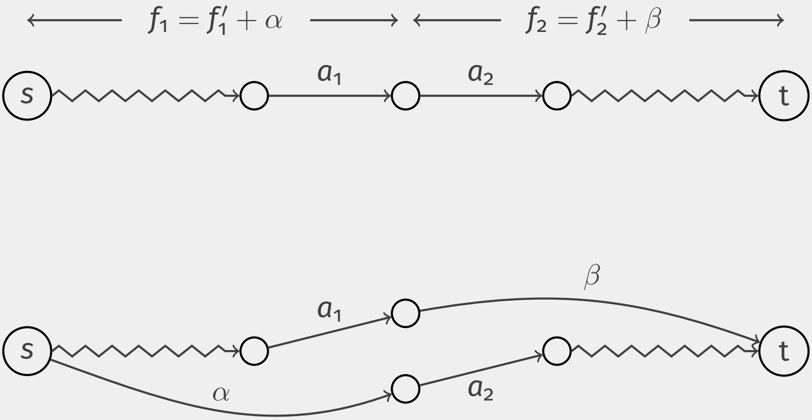


Proving the Claim

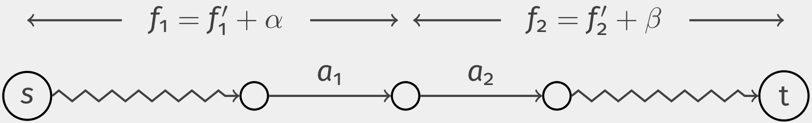
$\mathcal{A} = f_1 \cdot f_2$



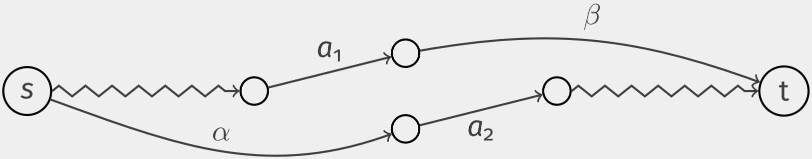
$\mathcal{A} = f_1 \cdot f_2$



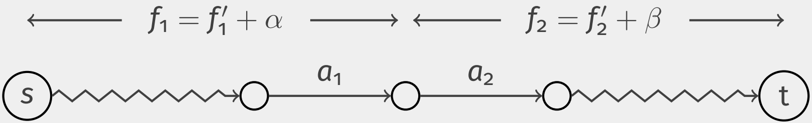
$\mathcal{A} = f_1 \cdot f_2$



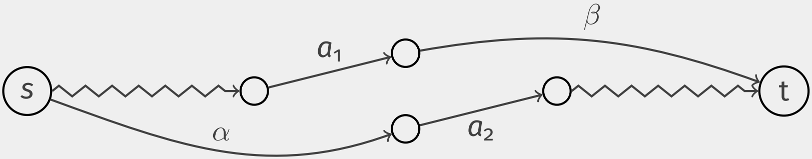
$\mathcal{A}' = \beta \cdot f_1 + \alpha \cdot f_2$



$\mathcal{A} = f_1 \cdot f_2$

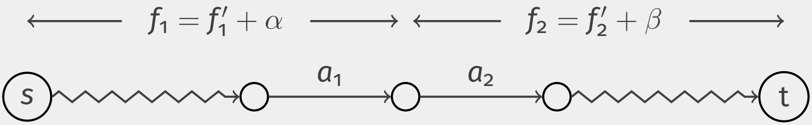


$\mathcal{A}' = \beta \cdot f_1 + \alpha \cdot f_2 = \mathcal{A} - f_1' \cdot f_2' + \alpha \cdot \beta$



Proving the Claim

$\mathcal{A} = f_1 \cdot f_2$



$\mathcal{A}' = \beta \cdot f_1 + \alpha \cdot f_2 = \mathcal{A} - f_1' \cdot f_2' + \alpha \cdot \beta$

Other Results

- If the edge labels on the ABP are allowed to have degree Δ, then the lowerbound we get is $\Omega(n^2/\Delta)$.

Other Results

- If the edge labels on the ABP are allowed to have degree Δ, then the lowerbound we get is $\Omega(n^2/\Delta)$.
- For unlayered ABPs with edge labels of degree $\leq \Delta$, the lowerbound we get is $\Omega(n \log n/\Delta \log\log n)$.

Other Results

- If the edge labels on the ABP are allowed to have degree Δ, then the lowerbound we get is $\Omega(n^2/\Delta)$.
- For unlayered ABPs with edge labels of degree $\leq \Delta$, the lowerbound we get is $\Omega(n \log n/\Delta \log\log n)$.
- The lowerbound is also true for a multilinear polynomial

$$\mathrm{ESym}(n, 0.1n) = \sum_{i_1 < \cdots < i_{0.1n} \in [n]} \prod_{j=1}^{0.1n} x_{i_j}.$$

- If the edge labels on the ABP are allowed to have degree Δ, then the lowerbound we get is $\Omega(n^2/\Delta)$.
- For unlayered ABPs with edge labels of degree $\leq \Delta$, the lowerbound we get is $\Omega(n \log n / \Delta \log \log n)$.
- The lowerbound is also true for a multilinear polynomial

$$\text{ESym}(n, 0.1n) = \sum_{i_1 < \cdots < i_{0.1n} \in [n]} \prod_{j=1}^{0.1n} x_{i_j}.$$

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

Open Threads

1. Prove a quadratic lowerbound (on wires?) for un-layered Algebraic Branching Programs

Open Threads

1. Prove a quadratic lowerbound (on wires?) for un-layered Algebraic Branching Programs
2. Prove a super-quadratic lowerbound on (homogeneous?) formulas (of constant depth?)

Open Threads

1. Prove a quadratic lowerbound (on wires?) for un-layered Algebraic Branching Programs
2. Prove a super-quadratic lowerbound on (homogeneous?) formulas (of constant depth?)
3. Reprove the $\Omega(n \log n)$ lowerbound for general circuits

Open Threads

1. Prove a quadratic lowerbound (on wires?) for un-layered Algebraic Branching Programs
2. Prove a super-quadratic lowerbound on (homogeneous?) formulas (of constant depth?)
3. Reprove the $\Omega(n \log n)$ lowerbound for general circuits
4. Prove a better (than n/2) determinantal complexity lowerbound

1. Prove a quadratic lowerbound (on wires?) for un-layered Algebraic Branching Programs
2. Prove a super-quadratic lowerbound on (homogeneous?) formulas (of constant depth?)
3. Reprove the $\Omega(n \log n)$ lowerbound for general circuits
4. Prove a better (than n/2) determinantal complexity lowerbound

Thank you!