Arithmetic circuits:
Depth reductions, chasms and escalators

Ramprasad Saptharishi

Based on joint work with

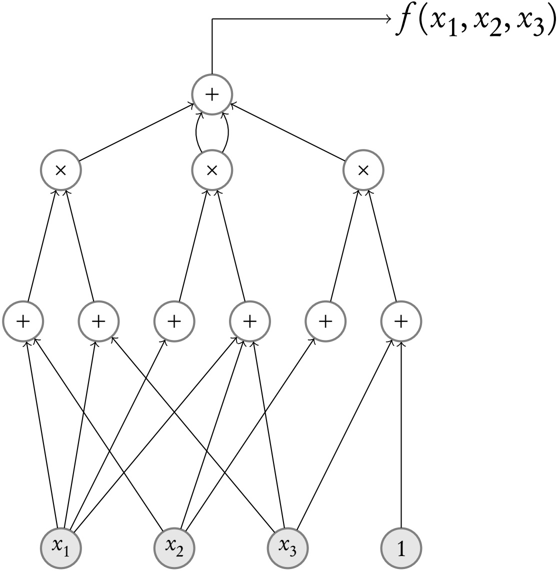
Ankit Gupta	Pritish Kamath	Neeraj Kayal

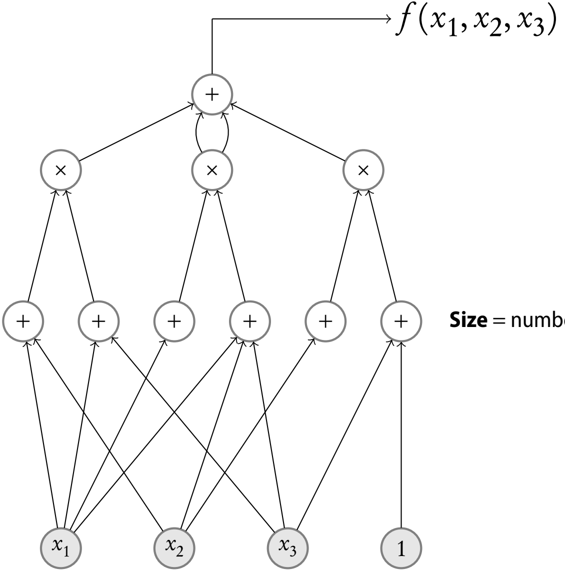
TCS+

September 2013

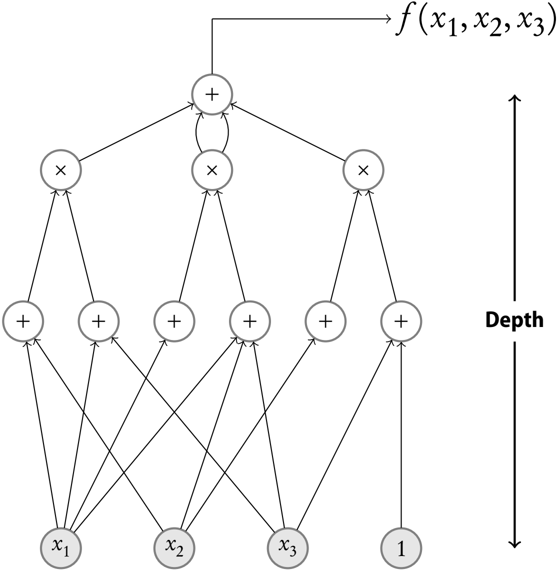
http://www.cmi.ac.in/~ramprasad/chasm_at_3.pdf

Arithmetic Circuits



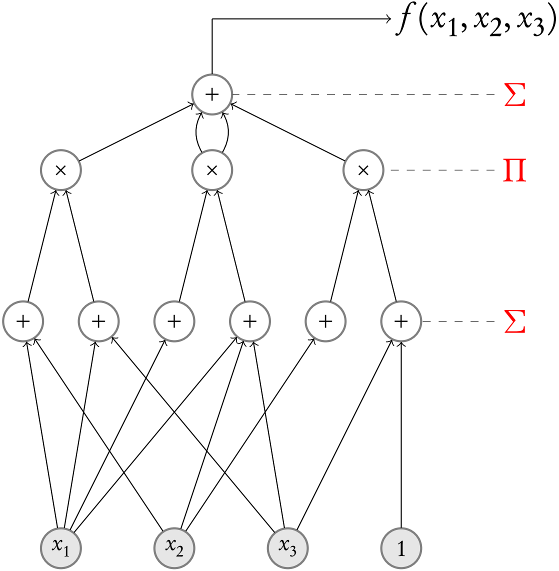


Size = number of gates

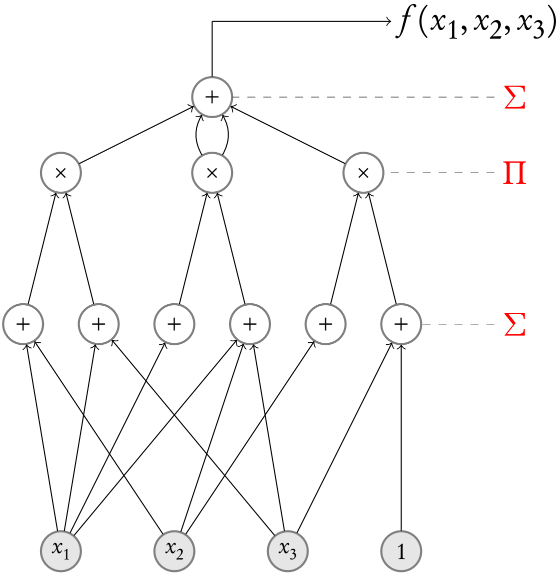


$f(x_1, x_2, x_3)$
+
×
×
×
+
+
+
+
+
+
Depth
x_1
x_2
x_3
1

Arithmetic Circuits



Arithmetic Circuits



Root is always (+)

"Two" questions about arithmetic circuits

About the poly:

Given a circuit C
does the poly ...

About circuits:

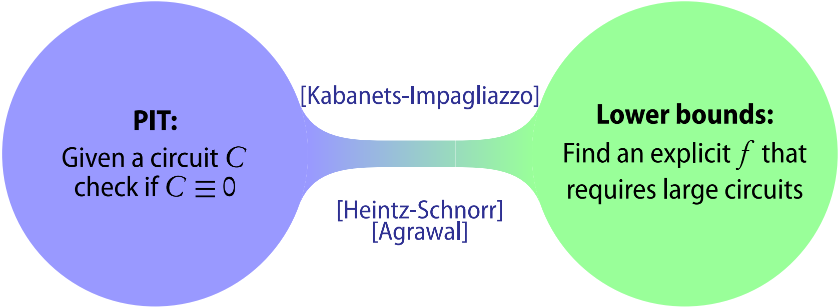
For a polynomial f
does the circuit ...

"Two" questions about arithmetic circuits

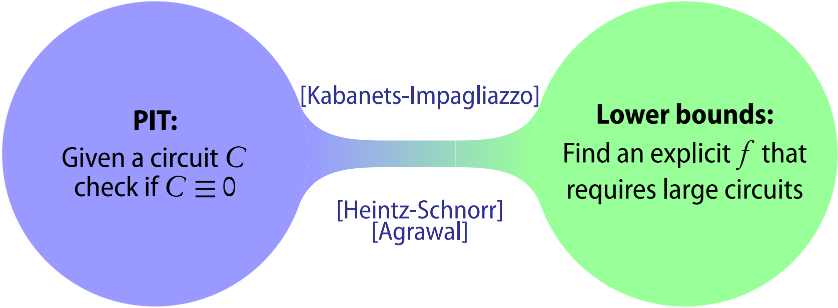
PIT:
Given a circuit C
check if $C \equiv 0$

Lower bounds:
Find an explicit f that
requires large circuits

"Two" questions about arithmetic circuits



"Two" questions about arithmetic circuits



"For the pessimist, this indicates that derandomizing identity testing is a hopeless problem. For the optimist, this means on the contrary that to obtain an arithmetic circuit lower bound, we 'simply' have to prove a good upper bound on identity testing." - *[Kayal-Saraf09]*

An algebraic analogue of "P vs NP"

$$\mathsf{VP} \stackrel{\text{def}}{=} \left\{ \begin{array}{l} P(x_1,\ldots,x_n) \text{ that are computable} \\ \text{by poly}(n)\text{-sized circuits} \end{array} \right\}$$

$$\mathsf{VNP} \stackrel{\text{def}}{=} \left\{ \begin{array}{l} P(x_1,\ldots,x_n) \;=\; \sum_{\mathbf{e}} c_{\mathbf{e}} \cdot x_1^{e_1} \ldots x_n^{e_n} \\ \text{where } \mathrm{Coeff}_P(e_1,\ldots,e_n) \in \mathsf{P} \end{array} \right\}$$

An algebraic analogue of "P vs NP"

$$\mathsf{VP} \stackrel{\text{def}}{=} \left\{ \begin{array}{l} P(x_1,\ldots,x_n) \text{ that are computable} \\ \text{by poly}(n)\text{-sized circuits} \end{array} \right\}$$

$$\mathsf{VNP} \stackrel{\text{def}}{=} \left\{ \begin{array}{l} P(x_1,\ldots,x_n) \;=\; \sum_{\mathbf{e}} c_{\mathbf{e}} \cdot x_1^{e_1} \ldots x_n^{e_n} \\ \text{where } \mathrm{Coeff}_P(e_1,\ldots,e_n) \in \mathsf{P} \end{array} \right\}$$

Det_n		VP
vs	[Valiant-79] ⟷	vs
Perm_n		VNP

Known lower bounds (before 2012)

Known lower bounds (before 2012)

Model	Lower bound	
General circuits	$\Omega(n \log n)$	[Baur-Strassen-83]
General formulas	$\Omega(n^3)$	[Kalorkoti-85]

Known lower bounds (before 2012)

Model	Lower bound	
General circuits	$\Omega(n \log n)$	[Baur-Strassen-83]
General formulas	$\Omega(n^3)$	[Kalorkoti-85]
Depth-2 circuits	$2^{\Omega(n \log n)}$	(Trivial!)
Homogeneous Depth-3 circuits	$2^{\Omega(n)}$	[Nisan-Wigderson-97]
Depth-3 circuits over finite fields	$2^{\Omega(n)}$	[Grigoriev-Karpinski-98]
Depth-3 circuits	$\Omega(n^2)$	[Shpilka-Wigderson-01]

Known lower bounds (before 2012)

Model	Lower bound	
Multilinear formula	$n^{\Omega(\log n)}$	[Raz-09]
Constant depth, *multilinear* formula	$2^{\tilde{\Omega}(n^{1/d})}$	[Raz-Yehudayoff-09]
Monotone Circuits	$2^{\Omega(n)}$	[Jerrum-Snir-82]

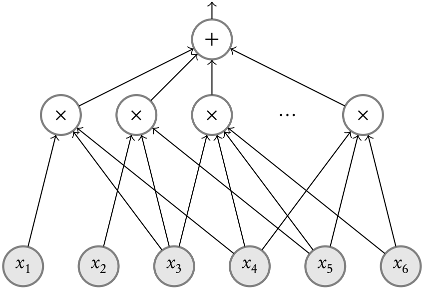
Known lower bounds (before 2012)

Model	Lower bound	
Multilinear formula	$n^{\Omega(\log n)}$	[Raz-09]
Constant depth, *multilinear* formula	$2^{\tilde{\Omega}(n^{1/d})}$	[Raz-Yehudayoff-09]
Monotone Circuits	$2^{\Omega(n)}$	[Jerrum-Snir-82]

Summary: No lower bounds known beyond depth-3, unless other restrictions are imposed.
Similar state for PITs as well...

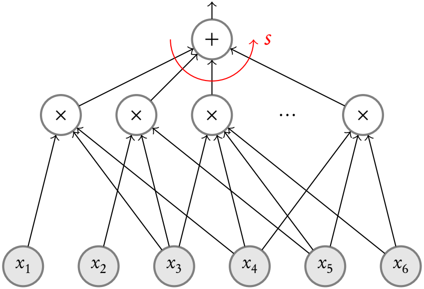
How powerful are shallow circuits?

Depth-2 circuits



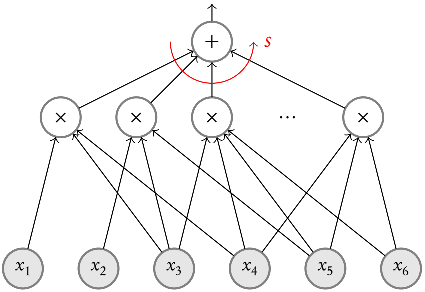
Depth-2 circuits

$$f(\mathbf{x}) \quad = \quad m_1 + \cdots + m_s$$



Depth-2 circuits

$$f(\mathbf{x}) \quad = \quad m_1 + \cdots + m_s$$

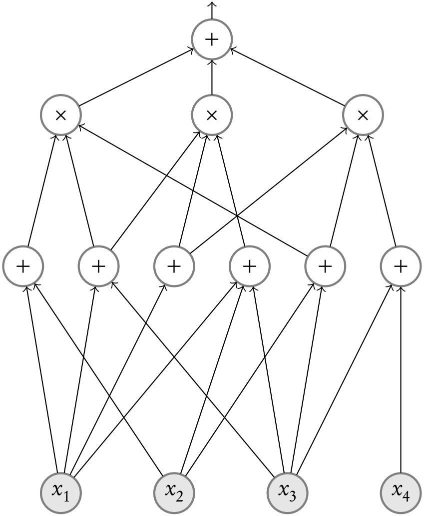


Polynomials with few monomials

Best depth-2 circuit is just the monomial representation.

Can't really do anything clever.

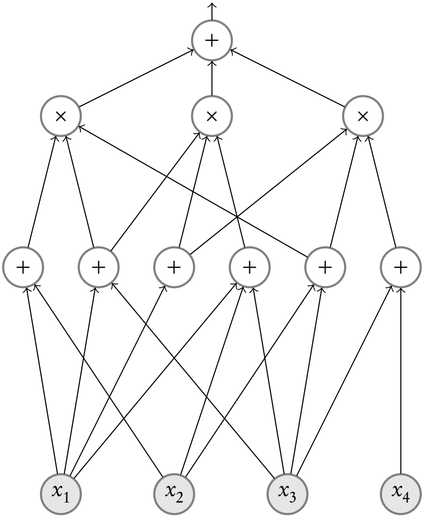
Depth-3 Circuits



Depth-3 Circuits

$$
\begin{aligned}
f(\mathbf{x}) &= T_1 + \cdots + T_m \\
\text{where each } T_i &= L_{i1} \ldots L_{id}
\end{aligned}
$$

Sum of products of linear polynomials.



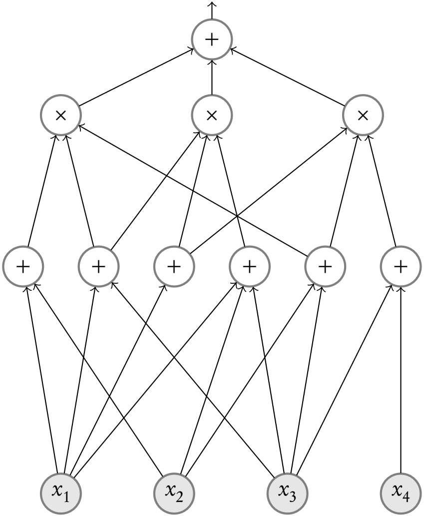
Depth-3 Circuits

$$f(\mathbf{x}) = T_1 + \cdots + T_m$$

where each $T_i = L_{i1} \ldots L_{id}$

Sum of products of linear polynomials.

Surprisingly powerful!



[Power of depth-3 circuits] Example 1

$$\mathrm{MAJ}(x_1,\ldots,x_n) \quad = \quad \bigvee_{1\le i_1<\cdots<i_{n/2}\le n} x_{i_1}\ldots x_{i_{n/2}}$$

$$\mathrm{SYM}_{n/2}(x_1,\ldots,x_n) \quad = \quad \sum_{1\le i_1<\cdots<i_{n/2}\le n} x_{i_1}\ldots x_{i_{n/2}}$$

$$\mathrm{SYM}_{n/2}(x_1, \ldots, x_n) \quad = \quad \sum_{1 \le i_1 < \cdots < i_{n/2} \le n} x_{i_1} \ldots x_{i_{n/2}}$$

Ben-Or's trick:

$$P_{\mathbf{x}}(t) \quad = \quad (1 + t x_1) \ldots (1 + t x_n) \quad = \quad \sum_{i=0}^{n} t^i \cdot \mathrm{SYM}_i$$

$$\mathrm{SYM}_{n/2}(x_1,\ldots,x_n) = \sum_{1\le i_1<\cdots<i_{n/2}\le n} x_{i_1}\ldots x_{i_{n/2}}$$

Ben-Or's trick:

$$P_{\mathbf{x}}(t) = (1+tx_1)\ldots(1+tx_n) = \sum_{i=0}^{n} t^i \cdot \mathrm{SYM}_i$$

$$\begin{bmatrix} 1 & \alpha_0 & \ldots & \alpha_0^n \\ 1 & \alpha_1 & \ldots & \alpha_1^n \\ \vdots & \vdots & \ddots & \vdots \\ 1 & \alpha_n & \ldots & \alpha_n^n \end{bmatrix} \begin{bmatrix} \mathrm{SYM}_0 \\ \mathrm{SYM}_1 \\ \vdots \\ \mathrm{SYM}_n \end{bmatrix} = \begin{bmatrix} P_{\mathbf{x}}(\alpha_0) \\ P_{\mathbf{x}}(\alpha_1) \\ \vdots \\ P_{\mathbf{x}}(\alpha_n) \end{bmatrix}$$

$$\mathrm{SYM}_{n/2}(x_1,\ldots,x_n) = \sum_{1\le i_1<\cdots<i_{n/2}\le n} x_{i_1}\ldots x_{i_{n/2}}$$

Ben-Or's trick:

$$P_{\mathbf{x}}(t) = (1+tx_1)\ldots(1+tx_n) = \sum_{i=0}^{n} t^i\cdot \mathrm{SYM}_i$$

$$\begin{bmatrix} \mathrm{SYM}_0 \\ \mathrm{SYM}_1 \\ \vdots \\ \mathrm{SYM}_n \end{bmatrix} = \begin{bmatrix} * & * & \ldots & * \\ * & * & \ldots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \ldots & * \end{bmatrix} \begin{bmatrix} P_{\mathbf{x}}(\alpha_0) \\ P_{\mathbf{x}}(\alpha_1) \\ \vdots \\ P_{\mathbf{x}}(\alpha_n) \end{bmatrix}$$

$$\mathrm{SYM}_{n/2}(x_1,\ldots,x_n) = \sum_{1\le i_1<\cdots<i_{n/2}\le n} x_{i_1}\ldots x_{i_{n/2}}$$

Ben-Or's trick:

$$P_{\mathbf{x}}(t) = (1+tx_1)\ldots(1+tx_n) = \sum_{i=0}^{n} t^i\cdot \mathrm{SYM}_i$$

$$\begin{bmatrix} \mathrm{SYM}_0 \\ \mathrm{SYM}_1 \\ \vdots \\ \mathrm{SYM}_n \end{bmatrix} = \begin{bmatrix} * & * & \ldots & * \\ * & * & \ldots & * \\ \vdots & \vdots & \ddots & \vdots \\ * & * & \ldots & * \end{bmatrix} \begin{bmatrix} P_{\mathbf{x}}(\alpha_0) \\ P_{\mathbf{x}}(\alpha_1) \\ \vdots \\ P_{\mathbf{x}}(\alpha_n) \end{bmatrix}$$

$$\mathrm{SYM}_{n/2} = \sum_{i=0}^{n} \beta_i\cdot P_{\mathbf{x}}(\alpha_i)$$

$$\mathrm{SYM}_{n/2}(x_1,\ldots,x_n) \quad = \quad \sum_{1\le i_1<\cdots<i_{n/2}\le n} x_{i_1}\ldots x_{i_{n/2}}$$

Ben-Or's trick:

$$P_{\mathbf{x}}(t) \quad = \quad (1+t\,x_1)\ldots(1+t\,x_n) \quad = \quad \sum_{i=0}^{n} t^i\cdot \mathrm{SYM}_i$$

$$\mathrm{SYM}_{n/2} \quad = \quad \sum_{i=0}^{n} \beta_i \cdot P_{\mathbf{x}}(\alpha_i)$$

$$\mathrm{SYM}_{n/2}(x_1,\ldots,x_n) \quad = \quad \sum_{1\le i_1<\cdots<i_{n/2}\le n} x_{i_1}\ldots x_{i_{n/2}}$$

Ben-Or's trick:

$$P_{\mathbf{x}}(t) \quad = \quad (1+tx_1)\ldots(1+tx_n) \quad = \quad \sum_{i=0}^{n} t^i \cdot \mathrm{SYM}_i$$

$$\mathrm{SYM}_{n/2} \quad = \quad \sum_{i=0}^{n} \beta_i \cdot (1+\alpha_i x_1)\ldots(1+\alpha_i x_n)$$

$$\mathrm{SYM}_{n/2}(x_1,\ldots,x_n) \quad = \quad \sum_{1\le i_1<\cdots<i_{n/2}\le n} x_{i_1}\ldots x_{i_{n/2}}$$

Ben-Or's trick:

$$P_{\mathbf{x}}(t) \quad = \quad (1+tx_1)\ldots(1+tx_n) \quad = \quad \sum_{i=0}^{n} t^i \cdot \mathrm{SYM}_i$$

$$\mathrm{SYM}_{n/2} \quad = \quad \sum_{i=0}^{n} \beta_i \cdot (1+\alpha_i x_1)\ldots(1+\alpha_i x_n)$$

... a depth-3 circuit of $O(n^2)$ size!

[Power of depth-3 circuits] Example 1

$$\mathrm{SYM}_{n/2}(x_1,\ldots,x_n) \quad = \quad \sum_{1\le i_1<\cdots<i_{n/2}\le n} x_{i_1}\ldots x_{i_{n/2}}$$

Ben-Or's trick:

$$P_{\mathbf{x}}(t) \quad = \quad (1+tx_1)\ldots(1+tx_n) \quad = \quad \sum_{i=0}^{n} t^i \cdot \mathrm{SYM}_i$$

$$\mathrm{SYM}_{n/2} \quad = \quad \sum_{i=0}^{n} \beta_i \cdot (1+\alpha_i x_1)\ldots(1+\alpha_i x_n)$$

... a depth-3 non-homogeneous circuit of $O(n^2)$ size!

- Formal degree of a leaf labelled by a variable is 1, and a leaf labelled by a constant is 0.
- Formal degree of a $(+)$ gate is the **maximum** of the formal degrees of the children.
- Formal degree of a $(\times)$ gate is the **sum** of the formal degrees of the children.

- Formal degree of a leaf labelled by a variable is 1, and a leaf labelled by a constant is 0.
- Formal degree of a $(+)$ gate is the **maximum** of the formal degrees of the children.
- Formal degree of a $(\times)$ gate is the **sum** of the formal degrees of the children.

Clearly an upper bound for the degree of the polynomial computed.

- Formal degree of a leaf labelled by a variable is 1, and a leaf labelled by a constant is 0.
- Formal degree of a $(+)$ gate is the **maximum** of the formal degrees of the children.
- Formal degree of a $(\times)$ gate is the **sum** of the formal degrees of the children.

Clearly an upper bound for the degree of the polynomial computed.

General Circuits: **formal degree** $=$ **degree** (wlog). [Strassen]
General Formulas: **formal degree** $\leq$ **size**.

[Power of depth-3 circuits] Example 2

$$\text{Perm}\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} = \sum_{\sigma \in S_n} x_{1\sigma(1)} \cdots x_{n\sigma(n)}$$

$$\mathsf{Perm}\begin{bmatrix} x_{11} & \dots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \dots & x_{nn} \end{bmatrix} = \sum_{\sigma\in S_n} x_{1\sigma(1)}\dots x_{n\sigma(n)}$$

$$(x_{11}+\cdots+x_{1n})\dots(x_{n1}+\cdots+x_{nn})$$

[Power of depth-3 circuits] Example 2

$$\text{Perm}\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} = \sum_{\sigma \in S_n} x_{1\sigma(1)} \cdots x_{n\sigma(n)}$$

Perm_n +

mons involving $(n-1)$
or fewer columns

$$= (x_{11} + \cdots + x_{1n}) \ldots (x_{n1} + \cdots + x_{nn})$$

[Power of depth-3 circuits] Example 2

$$\text{Perm}\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} = \sum_{\sigma \in S_n} x_{1\sigma(1)} \cdots x_{n\sigma(n)}$$

Perm_n $+$

mons involving $(n-2)$ or fewer columns

$$= (x_{11} + \cdots + x_{1n}) \ldots (x_{n1} + \cdots + x_{nn}) - \sum_{|S|=(n-1)} \prod_{i=1}^{n} \sum_{j \in S} x_{ij}$$

$$\text{Perm}\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} = \sum_{\sigma \in S_n} x_{1\sigma(1)} \dots x_{n\sigma(n)}$$

Perm_n +

mons involving $(n-3)$ or fewer columns

$$= \begin{aligned} &(x_{11} + \cdots + x_{1n}) \dots (x_{n1} + \cdots + x_{nn}) \\ &- \sum_{|S|=(n-1)} \prod_{i=1}^{n} \sum_{j \in S} x_{ij} \\ &+ \sum_{|S|=(n-2)} \prod_{i=1}^{n} \sum_{j \in S} x_{ij} \end{aligned}$$

$$\text{Perm}\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} = \sum_{\sigma \in S_n} x_{1\sigma(1)} \cdots x_{n\sigma(n)}$$

Perm$_n$ +

mons involving $(n-3)$ or fewer columns

$$= (x_{11} + \cdots + x_{1n}) \ldots (x_{n1} + \cdots + x_{nn}) - \sum_{|S|=(n-1)} \prod_{i=1}^{n} \sum_{j \in S} x_{ij} + \sum_{|S|=(n-2)} \prod_{i=1}^{n} \sum_{j \in S} x_{ij}$$

... (inclusion-exclusion)

[Power of depth-3 circuits] Example 2

$$\text{Perm}\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} = \sum_{\sigma \in S_n} x_{1\sigma(1)} \cdots x_{n\sigma(n)}$$

Ryser's formula:

$$\text{Perm}_n = \sum_{S \subseteq [n]} (-1)^{n-|S|} \prod_{i=1}^{n} \sum_{j \in S} x_{ij}$$

...a $2^{O(n)}$-sized depth-3 circuit for Perm_n.

[Power of depth-3 circuits] Example 2

$$\text{Perm}\begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} = \sum_{\sigma \in S_n} x_{1\sigma(1)} \cdots x_{n\sigma(n)}$$

Ryser's formula:

$$\text{Perm}_n = \sum_{S \subseteq [n]} (-1)^{n-|S|} \prod_{i=1}^{n} \sum_{j \in S} x_{ij}$$

...a $2^{O(n)}$-sized depth-3 circuit for Perm_n.

Strangely, no such depth-3 circuit was known for Det_n.
In fact, nothing significantly better than $n! = n^{O(n)}$

Is there a conceptually simpler class of circuits with the expressive power of general arithmetic circuits?

Is there a conceptually simpler class of circuits with the expressive power of general arithmetic circuits?

Would suffice to solve PIT on the *conceptually simpler class*.

Would suffice to prove lower bounds on the *conceptually simpler class*.

Is there a conceptually simpler class of circuits with the expressive power of general arithmetic circuits?

Would suffice to solve PIT on the *conceptually simpler class*.

Would suffice to prove lower bounds on the *conceptually simpler class*.

Are *shallow circuits* a candidate for this?

Depth reduction

Theorem ([Valiant-Skyum-Berkowitz-Rackoff-83])

Can be computed by		Can be computed by
arithmetic circuits	$\longrightarrow$	log-depth circuits
of "small" size		of "small" size

Reduction to log-depth

Theorem ([Valiant-Skyum-Berkowitz-Rackoff-83])

Can be computed by		*Can be computed by*
arithmetic circuits	$\longrightarrow$	*log-depth circuits*
of $\text{poly}(n, d)$ *size*		*of* $\text{poly}(n, d)$ *size*

Reduction to depth-4

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Can be computed by		*Can be computed by*
arithmetic circuits	$\longrightarrow$	*depth-4 circuits*
of "small" size		*of "not-too-large" size*

Reduction to depth-4

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Can be computed by		*Can be computed by*
arithmetic circuits	$\longrightarrow$	*depth-4 circuits*
of $\text{poly}(n,d)$ *size*		*of* $n^{O(\sqrt{d})}$ *size*

Reduction to depth-4

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Can be computed by		Can be computed by
arithmetic circuits	$\longrightarrow$	depth-4 circuits*
of poly(n, d) size		of $n^{O(\sqrt{d})}$ size

Reduction to depth-4

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Cannot be computed by		*Cannot be computed by*
arithmetic circuits	⟵	*depth-4 circuits**
of poly(n, d) *size*		*of* $n^{O(\sqrt{d})}$ *size*

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Cannot be computed by		*Cannot* be computed by
arithmetic circuits	$\longleftarrow$	*depth-4 circuits**
of $\text{poly}(n,d)$ *size*		*of* $n^{O(\sqrt{d})}$ *size*

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Cannot be computed by		*Cannot be computed by*
arithmetic circuits	$\longleftarrow$	*depth-4 circuits**
of $\mathrm{poly}(n, d)$ *size*		*of* $n^{O(\sqrt{d})}$ *size*

And last year ...

Chasm at depth-4

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Cannot be computed by arithmetic circuits of poly(n,d) size ⟵ *Cannot be computed by* depth-4 circuits* of $n^{O(\sqrt{d})}$ size

Theorem ([Gupta-Kamath-Kayal-Saptharishi])

Perm_n (or Det_n) requires

depth-4 circuits*

of $2^{\Omega(\sqrt{n})}$ size

Chasm at depth-4

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Cannot be computed by		*Cannot be computed by*
arithmetic circuits	⟵	*depth-4 circuits**
of $\mathrm{poly}(n, d)$ *size*		*of* $n^{O(\sqrt{d})}$ *size*

Theorem ([Kayal-Saha-Saptharishi])

An explicit poly in VNP *requires*

*depth-4 circuits**

of $n^{\Omega(\sqrt{d})}$ *size*

Chasm at depth-4

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Cannot be computed by		*Cannot be computed by*
arithmetic circuits	⟵	*depth-4 circuits**
of poly(n, d) *size*		*of* $n^{O(\sqrt{d})}$ *size*

Theorem ([Fournier-Limaye-Malod-Srinivasan])

An explicit. poly in VP *requires*

*depth-4 circuits**

of $n^{\Omega(\sqrt{d})}$ *size*

Escalator at depth-4

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Cannot be computed by		*Cannot be computed by*
arithmetic circuits	⟵	*depth-4 circuits**
of $\mathrm{poly}(n, d)$ *size*		*of* $n^{O(\sqrt{d})}$ *size*

Theorem ([Fournier-Limaye-Malod-Srinivasan])

An explicit. poly in VP *requires*

*depth-4 circuits**

of $n^{\Omega(\sqrt{d})}$ *size*

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Can be computed by		Can be computed by
arithmetic circuits	$\longrightarrow$	depth-4 circuits
poly(n, d) size		of $n^{O(\sqrt{d})}$ size

Squashing it further

Theorem ([Agrawal-Vinay-08, Koiran-12, Tavenas-13])

Can be computed by		*Can be computed by*
arithmetic circuits	$\longrightarrow$	*depth-4 circuits*
poly(n, d) *size*		*of* $n^{O(\sqrt{d})}$ *size*

Is a similar depth-reduction possible to depth-3?

But...

- $2^{\Omega(n)}$ lower bound for depth-3 circuits computing Det_n over **finite fields** [Grigoriev-Karpinski].

But...

- $2^{\Omega(n)}$ lower bound for depth-3 circuits computing Det_n over **finite fields** [Grigoriev-Karpinski].
  Hence, such a reduction must necessarily be field dependent.

But...

- $2^{\Omega(n)}$ lower bound for depth-3 circuits computing Det_n over **finite fields** [Grigoriev-Karpinski].
  Hence, such a reduction must necessarily be field dependent.

- $2^{\Omega(n)}$ lower bound known for **homogeneous** depth-3 computing Det_n or $\mathrm{SYM}_{n/2}$ [Nisan-Wigderson].

But...

- $2^{\Omega(n)}$ lower bound for depth-3 circuits computing Det_n over **finite fields** [Grigoriev-Karpinski].
  Hence, such a reduction must necessarily be field dependent.

- $2^{\Omega(n)}$ lower bound known for **homogeneous** depth-3 computing Det_n or $\mathrm{SYM}_{n/2}$ [Nisan-Wigderson].
  Hence, such a reduction must compute intermediate polynomials of *high* degree and cleverly cancel them off in the end.

But...

- $2^{\Omega(n)}$ lower bound for depth-3 circuits computing Det_n over **finite fields** [Grigoriev-Karpinski].
  Hence, such a reduction must necessarily be field dependent.

- $2^{\Omega(n)}$ lower bound known for **homogeneous** depth-3 computing Det_n or $\text{SYM}_{n/2}$ [Nisan-Wigderson].
  Hence, such a reduction must compute intermediate polynomials of *high* degree and cleverly cancel them off in the end.

- No depth-3 circuit for determinant of size $2^{O(n)}$ was known.
  The permanent however has *Ryser's formula*.

$$\text{Perm}_n = \sum_{S \in [n]} (-1)^{n-|S|} \prod_{i=1}^{n} \sum_{j \in S} x_{ij}$$

... *it's true!*

Theorem ([Gupta-Kamath-Kayal-Saptharishi])

Can be computed by		*Can be computed by*
arithmetic circuits	$\xrightarrow{\text{over } \mathbb{Q}}$	*depth-3 circuits*
poly(n, d) *size*		*of* $n^{O(\sqrt{d})}$ *size*

... it's true!

Theorem ([Gupta-Kamath-Kayal-Saptharishi])

Can be computed by		Can be computed by
arithmetic circuits	$\xrightarrow{\text{over } \mathbb{Q}}$	*depth-3 circuits*
$\mathrm{poly}(n,d)$ *size*		*of* $n^{O(\sqrt{d})}$ *size*

Corollary

A depth-3 circuit for Det_n *of size* $n^{O(\sqrt{n})}$ *over* $\mathbb{Q}$.

... *it's true!*

Theorem ([Gupta-Kamath-Kayal-Saptharishi])

Can be computed by		Can be computed by
arithmetic circuits	$\xrightarrow{\text{over } \mathbb{Q}}$	depth-3 circuits
poly(n,d) size		of $n^{O(\sqrt{d})}$ size

Corollary

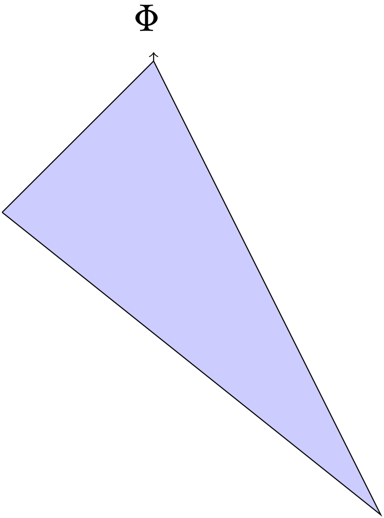
A depth-3 circuit for Det_n *of size* $n^{O(\sqrt{n})}$ *over* $\mathbb{Q}$.

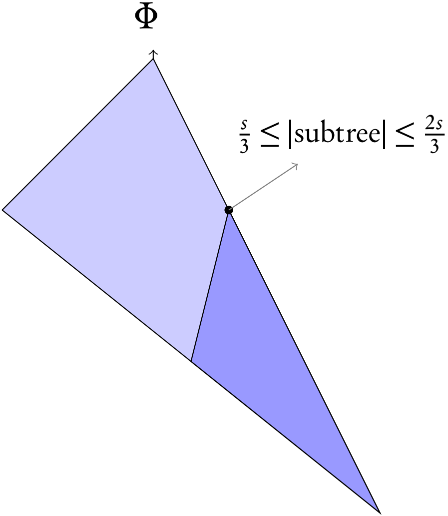
Note: Resulting depth-3 circuit is *heavily non-homogeneous*, with degrees going up to $n^{O(\sqrt{d})}$.

Plan for rest of the talk

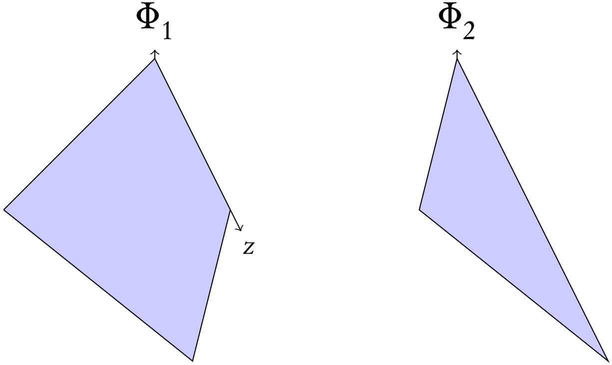
1. Depth reduction to $O(\log n)$ *for formulas*
2. Sketch of depth reduction to depth-4
3. Depth reduction to depth-3 (over $\mathbb{C}$)

Depth reducing formulas

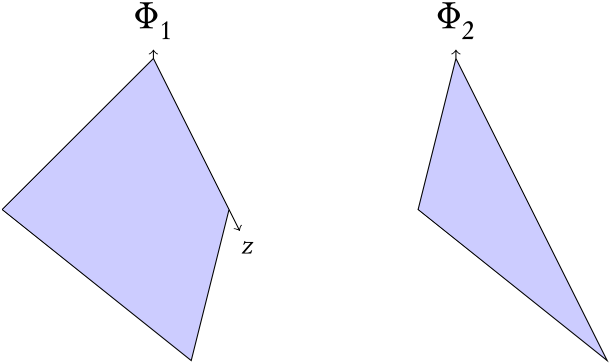



Φ
s/3 ≤ |subtree| ≤ 2s/3

Depth reducing formulas

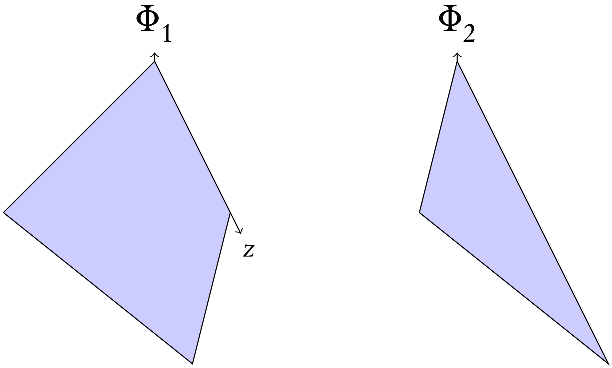


Depth reducing formulas



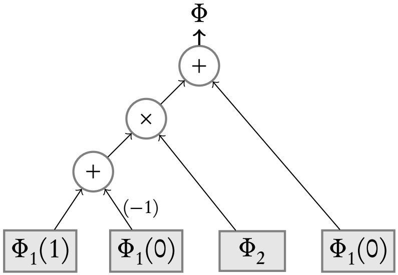
$$\Phi_1(z) = A \cdot z + B$$
$$\Phi = A \cdot \Phi_2 + B$$

Depth reducing formulas

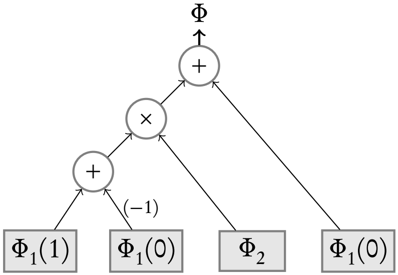


$$
\begin{aligned}
\Phi_1(z) &= A \cdot z + B \\
\Phi &= A \cdot \Phi_2 + B = (\Phi_1(1) - \Phi_1(0)) \cdot \Phi_2 + \Phi_1(0)
\end{aligned}
$$

Depth reducing formulas

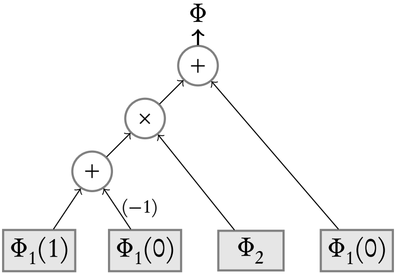


$$\Phi_1(z) = A \cdot z + B$$
$$\Phi = A \cdot \Phi_2 + B = (\Phi_1(1) - \Phi_1(0)) \cdot \Phi_2 + \Phi_1(0)$$



$$\begin{aligned}\Phi_1(z) &= A\cdot z + B\\ \Phi &= A\cdot\Phi_2 + B = (\Phi_1(1)-\Phi_1(0))\cdot\Phi_2 + \Phi_1(0)\end{aligned}$$

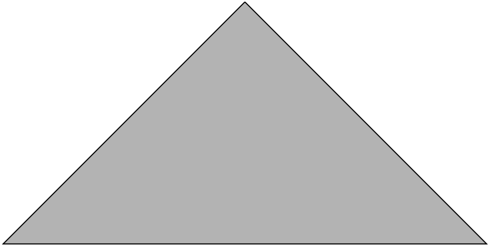
Recursive application yields $O(\log s)$-depth and poly(s) size. [Brent]



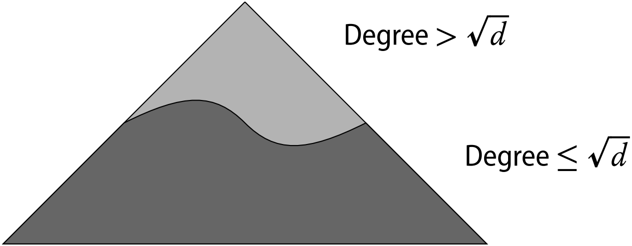
$$
\begin{aligned}
\Phi_1(z) &= A \cdot z + B \\
\Phi &= A \cdot \Phi_2 + B = (\Phi_1(1) - \Phi_1(0)) \cdot \Phi_2 + \Phi_1(0)
\end{aligned}
$$

Recursive application yields $O(\log s)$-depth and poly(s) size. [Brent] [Valiant-Skyum-Berkowitz-Rackoff] achieve the same for circuits.

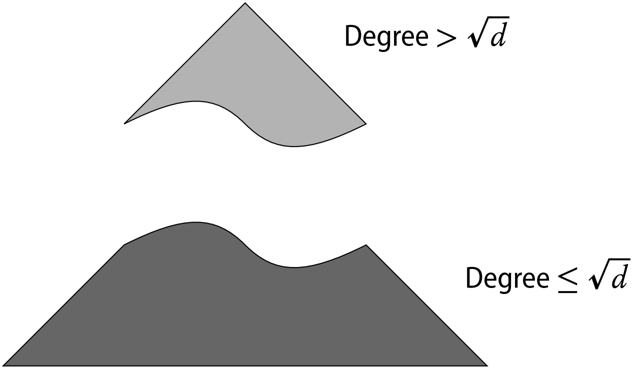
Depth reduction to depth-4



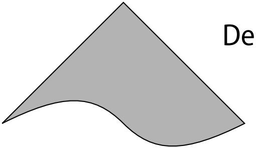
Depth reduction to depth-4

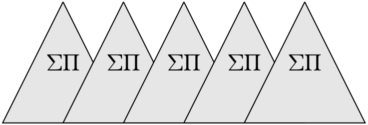


Depth reduction to depth-4

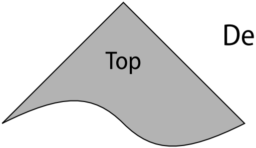


Depth reduction to depth-4

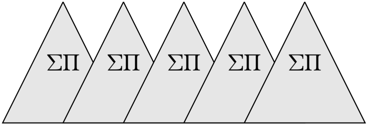




Depth reduction to depth-4



Degree $> \sqrt{d}$



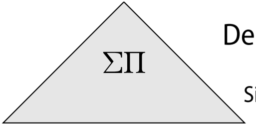
Degree $\leq \sqrt{d}$

Size $\binom{n+\sqrt{d}}{\sqrt{d}}$ each

Lemma ([Tavenas13])

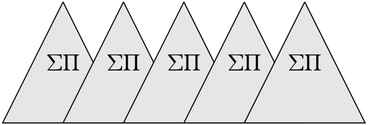
$$\deg(\mathrm{Top}(z_1, \ldots, z_s)) \quad \leq \quad 15\sqrt{d}$$

Depth reduction to depth-4



Degree $> \sqrt{d}$

Size $\binom{s+15\sqrt{d}}{15\sqrt{d}}$



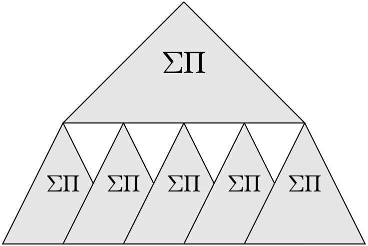
Degree $\leq \sqrt{d}$

Size $\binom{n+\sqrt{d}}{\sqrt{d}}$ each

Lemma ([Tavenas13])

$$\deg(\mathrm{Top}(z_1, \ldots, z_s)) \quad \leq \quad 15\sqrt{d}$$

Depth reduction to depth-4

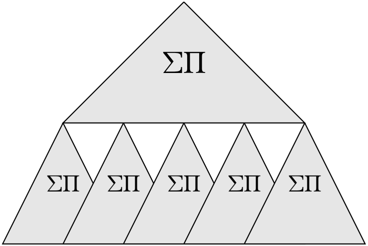


Theorem

Equivalent depth-4 circuit of size

$$s\binom{n+\sqrt{d}}{n} \quad + \quad \binom{s+15\sqrt{d}}{s} \quad = \quad s^{O(\sqrt{d})}$$

Depth reduction to depth-4



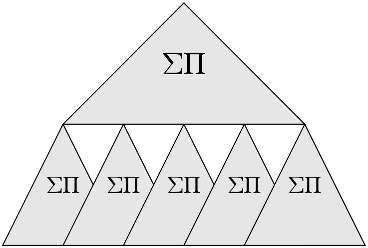
Theorem

Equivalent depth-4 circuit of size

$$s\binom{n+\sqrt{d}}{n} \quad + \quad \binom{s+15\sqrt{d}}{s} \quad = \quad s^{O(\sqrt{d})}$$

□

Depth reduction to depth-4



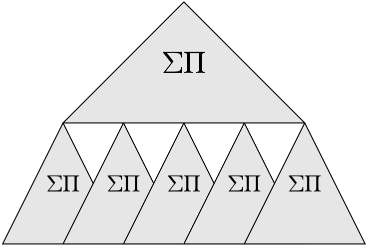
Theorem

Equivalent depth-4 circuit with multiplication fan-in at most $15\sqrt{d}$ of size

$$s\binom{n+\sqrt{d}}{n} \quad + \quad \binom{s+15\sqrt{d}}{s} \quad = \quad s^{O(\sqrt{d})}$$

□

Depth reduction to depth-4



Theorem

Equivalent depth-4 circuit of size*

$$s\binom{n+\sqrt{d}}{n} \quad + \quad \binom{s+15\sqrt{d}}{s} \quad = \quad s^{O(\sqrt{d})}$$

□

Reduction to Depth-3 Circuits

Road map

$$\Sigma\overset{\sqrt{d}}{\Pi}\Sigma\overset{\sqrt{d}}{\Pi}$$

circuits

$$\downarrow$$

$$\Sigma\overset{\sqrt{d}}{\bigwedge}\Sigma\overset{\sqrt{d}}{\bigwedge}\Sigma$$

circuits

$$\downarrow$$

$$\Sigma\Pi\Sigma$$

circuits

Road map

$$\Sigma\Pi^{\sqrt{d}}\Sigma\Pi^{\sqrt{d}}$$

circuits

$\downarrow$ App. of Ryser's formula

$$\Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma$$

circuits

$\downarrow$

$$\Sigma\Pi\Sigma$$

circuits

Road map

$$\Sigma\overset{\sqrt{d}}{\Pi}\Sigma\overset{\sqrt{d}}{\Pi}$$

circuits

↓ App. of Ryser's formula

$$\Sigma\overset{\sqrt{d}}{\bigwedge}\Sigma\overset{\sqrt{d}}{\bigwedge}\Sigma$$

circuits

↓ [Saxena]'s duality trick

$$\Sigma\Pi\Sigma$$

circuits

Road map

$$\Sigma \overset{\sqrt{d}}{\Pi} \Sigma \overset{\sqrt{d}}{\Pi}$$

circuits

↓

Only over $\mathbb{Q}$,$\mathbb{R}$ etc. — App. of Ryser's formula

$$\Sigma \overset{\sqrt{d}}{\bigwedge} \Sigma \overset{\sqrt{d}}{\bigwedge} \Sigma$$

circuits

↓

Heavily non-homogeneous — [Saxena]'s duality trick

$$\Sigma\Pi\Sigma$$

circuits

Step 1*:* $\Sigma\Pi\Sigma\Pi$ *to* $\Sigma\wedge\Sigma\wedge\Sigma$

Step 1: $\Sigma\Pi\Sigma\Pi$ *to* $\Sigma\wedge\Sigma\wedge\Sigma$

Recall Ryser's formula:

$$\mathrm{Perm}_n \begin{bmatrix} x_{11} & \cdots & x_{1n} \\ \vdots & \ddots & \vdots \\ x_{n1} & \cdots & x_{nn} \end{bmatrix} = \sum_{S\subseteq[n]} (-1)^{n-|S|} \prod_{i=1}^{n} \sum_{j\in S} x_{ij}$$

Step 1: $\Sigma\Pi\Sigma\Pi$ *to* $\Sigma\wedge\Sigma\wedge\Sigma$

Recall Ryser's formula:

$$\mathrm{Perm}_n \begin{bmatrix} x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_1 & \cdots & x_n \end{bmatrix} = \sum_{S \subseteq [n]} (-1)^{n-|S|} \prod_{i=1}^{n} \sum_{j \in S} x_j$$

Step 1*:* $\Sigma\Pi\Sigma\Pi$ *to* $\Sigma\wedge\Sigma\wedge\Sigma$

Recall Ryser's formula:

$$\mathsf{Perm}_n \begin{bmatrix} x_1 & \cdots & x_n \\ \vdots & \ddots & \vdots \\ x_1 & \cdots & x_n \end{bmatrix} = \sum_{S \subseteq [n]} (-1)^{n-|S|} \left(\sum_{j \in S} x_j \right)^n$$

Step 1: $\Sigma\Pi\Sigma\Pi$ *to* $\Sigma\wedge\Sigma\wedge\Sigma$

Recall Ryser's formula:

$$n! \cdot x_1 \dots x_n \quad = \quad \sum_{S \subseteq [n]} (-1)^{n-|S|} \left(\sum_{j \in S} x_j \right)^n$$

Step 1: $\Sigma\Pi\Sigma\Pi$ *to* $\Sigma\wedge\Sigma\wedge\Sigma$

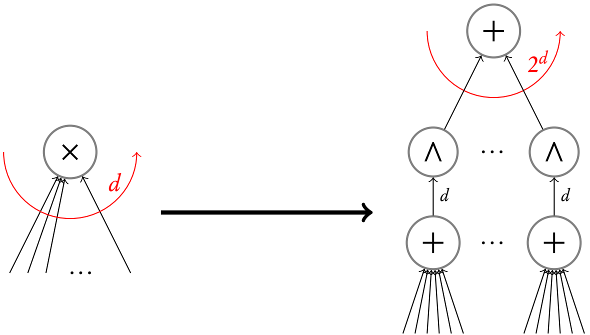
[Fischer]:

$$n! \cdot x_1 \ldots x_n = \sum_{S \subseteq [n]} (-1)^{n-|S|} \left(\sum_{j \in S} x_j \right)^n$$

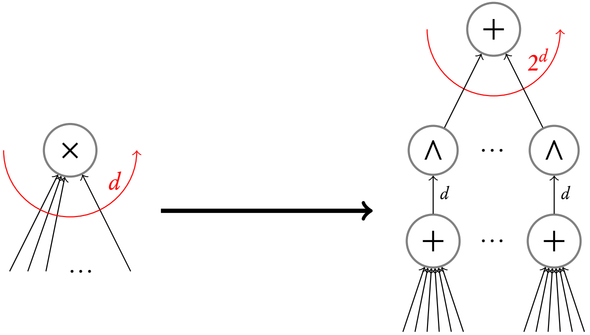
Step 1: $\Sigma\Pi\Sigma\Pi$ *to* $\Sigma\wedge\Sigma\wedge\Sigma$

[Fischer]:

$$n! \cdot x_1 \dots x_n \quad = \quad \sum_{S \subseteq [n]} (-1)^{n-|S|} \left(\sum_{j \in S} x_j \right)^n$$

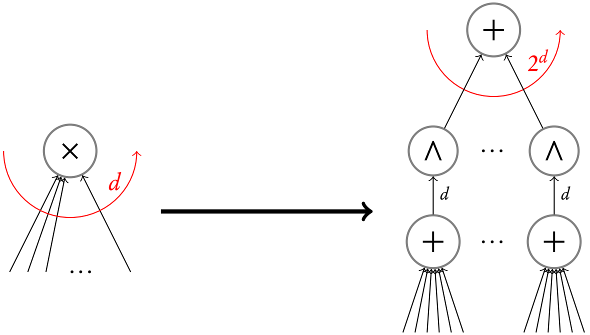


Step 1: $\Sigma\Pi\Sigma\Pi$ *to* $\Sigma\wedge\Sigma\wedge\Sigma$



$$\prod^{d} \longrightarrow \sum^{2^d}\bigwedge^{d}\sum^{d}$$

Step 1: $\Sigma\Pi\Sigma\Pi$ *to* $\Sigma\wedge\Sigma\wedge\Sigma$



$$\prod^{d} \longrightarrow \sum^{2^d} \bigwedge^{d} \sum^{d}$$

$$\Sigma\prod^{\sqrt{d}}\Sigma\prod^{\sqrt{d}} \text{ of size } s \longrightarrow \Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma \text{ of size } 2^{O(\sqrt{d})} \cdot s$$

Road map

$$\Sigma\prod^{\sqrt{d}}\Sigma\prod^{\sqrt{d}}$$

circuits

$$\downarrow$$

$$\Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma$$

circuits

$$\downarrow$$

$$\Sigma\Pi\Sigma$$

circuits

Road map

$$\Sigma\Pi^{\sqrt{d}}\Sigma\Pi^{\sqrt{d}}$$

circuits

$$\downarrow \checkmark$$

$$\Sigma\wedge^{\sqrt{d}}\Sigma\wedge^{\sqrt{d}}\Sigma$$

circuits

$$\downarrow$$

$$\Sigma\Pi\Sigma$$

circuits

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$\Sigma \overset{\sqrt{d}}{\wedge} \Sigma \overset{\sqrt{d}}{\wedge} \Sigma$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$\Sigma \overset{\sqrt{d}}{\wedge} \Sigma \overset{\sqrt{d}}{\wedge} \Sigma$$

$$\ell$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$\Sigma \overset{\sqrt{d}}{\wedge} \Sigma \overset{\sqrt{d}}{\wedge} \Sigma$$

$$\ell^{\sqrt{d}}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$\sum \bigwedge^{\sqrt{d}} \sum \bigwedge^{\sqrt{d}} \sum$$

$$\ell_1^{\sqrt{d}} + \ldots + \ell_s^{\sqrt{d}}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$\Sigma \overset{\sqrt{d}}{\wedge} \Sigma \overset{\sqrt{d}}{\wedge} \Sigma$$

$$\left(\ell_1^{\sqrt{d}} + \ldots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$\Sigma\overset{\sqrt{d}}{\wedge}\Sigma\overset{\sqrt{d}}{\wedge}\Sigma$$

$$\sum_i \left(\ell_{i1}^{\sqrt{d}} + \ldots + \ell_{is}^{\sqrt{d}}\right)^{\sqrt{d}}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$C = \sum_i \left(\ell_{i1}^{\sqrt{d}} + \cdots + \ell_{is}^{\sqrt{d}}\right)^{\sqrt{d}}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ to $\Sigma\Pi\Sigma$

$$T \quad = \quad \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

Lemma ([Saxena])

There exists univariate polynomials f_{ij}'s of degree at most d such that

$$(x_1 + \cdots + x_s)^d \quad = \quad \sum_{i=1}^{sd+1} \prod_{j=1}^{s} f_{ij}(x_j)$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ to $\Sigma\Pi\Sigma$

$$T \quad = \quad \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

Lemma ([Saxena])

There exists univariate polynomials f_{ij}'s of degree at most d such that

$$(x_1 + \cdots + x_s)^d \quad = \quad \sum_{i=1}^{sd+1} \prod_{j=1}^{s} f_{ij}(x_j)$$

Sketch of a proof by Gupta-Forbes-Shpilka

$$P_{\mathbf{x}}(t) = (1 + x_1 t)\ldots(1 + x_s t) \quad = \quad 1 + \ell t + \text{(higher degree terms)}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ to $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

Lemma ([Saxena])

There exists univariate polynomials f_{ij}'s of degree at most d such that

$$(x_1 + \cdots + x_s)^d = \sum_{i=1}^{sd+1} \prod_{j=1}^{s} f_{ij}(x_j)$$

Sketch of a proof by Gupta-Forbes-Shpilka

$$P_{\mathbf{x}}(t) - 1 = \ell t + \text{(higher degree terms)}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ to $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

Lemma ([Saxena])

There exists univariate polynomials f_{ij}'s of degree at most d such that

$$(x_1 + \cdots + x_s)^d = \sum_{i=1}^{sd+1} \prod_{j=1}^{s} f_{ij}(x_j)$$

Sketch of a proof by Gupta-Forbes-Shpilka

$$(P_{\mathbf{x}}(t) - 1)^d = \ell^d t^d + \text{(higher degree terms)}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T \quad = \quad \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

Lemma ([Saxena])

There exists univariate polynomials f_{ij}'s of degree at most d such that

$$(x_1 + \cdots + x_s)^d \quad = \quad \sum_{i=1}^{sd+1} \prod_{j=1}^{s} f_{ij}(x_j)$$

Sketch of a proof by Gupta-Forbes-Shpilka

$$(P_{\mathbf{x}}(t) - 1)^d \quad = \quad \ell^d t^d \quad + \quad \text{(higher degree terms)}$$

Interpolate! □

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

$$\left(x_1 + \cdots + x_s\right)^{\sqrt{d}} = \sum_{i}^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} f_{ij}\left(x_j\right)$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T \quad = \quad \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

$$\left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}} \quad = \quad \sum_{i}^{\mathsf{poly}(s,d)} \prod_{j=1}^{s} f_{ij}\left(\ell_j^{\sqrt{d}}\right)$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

$$\left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}} = \sum_{i}^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} f_{ij}\left(\ell_j^{\sqrt{d}}\right) = \sum_{i}^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} \tilde{f}_{ij}(\ell_j)$$

where $\tilde{f}_{ij}(t) := f_{ij}(t^{\sqrt{d}})$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

$$\left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}} = \sum_{i}^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} f_{ij}\left(\ell_j^{\sqrt{d}}\right)$$

$$= \sum_{i}^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} \tilde{f}_{ij}(\ell_j)$$

Note that $\tilde{f}_{ij}(t)$ is a univariate polynomial

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

$$\begin{aligned}\left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}} &= \sum_i^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} f_{ij}\left(\ell_j^{\sqrt{d}}\right) \\ &= \sum_i^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} \tilde{f}_{ij}(\ell_j)\end{aligned}$$

Note that $\tilde{f}_{ij}(t)$ is a univariate polynomial that can be factorized over $\mathbb{C}$:

$$\tilde{f}_{ij}(t) = \prod_{k=1}^{d}(t - \zeta_{ijk})$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

$$\begin{aligned}\left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}} &= \sum_i^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} f_{ij}\left(\ell_j^{\sqrt{d}}\right) \\ &= \sum_i^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} \tilde{f}_{ij}(\ell_j)\end{aligned}$$

Note that $\tilde{f}_{ij}(t)$ is a univariate polynomial that can be factorized over $\mathbb{C}$:

$$\tilde{f}_{ij}(\ell_j) = \prod_{k=1}^{d}(\ell_j - \zeta_{ijk})$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

$$\begin{aligned}\left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}} &= \sum_i^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} f_{ij}\left(\ell_j^{\sqrt{d}}\right) \\ &= \sum_i^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} \tilde{f}_{ij}(\ell_j) \\ &= \sum_i^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} \prod_{k=1}^{d} \left(\ell_j - \zeta_{ijk}\right)\end{aligned}$$

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T = \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

$$\begin{aligned}\left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}} &= \sum_i^{\text{poly}(s,d)} \prod_{j=1}^{s} f_{ij}\left(\ell_j^{\sqrt{d}}\right) \\ &= \sum_i^{\text{poly}(s,d)} \prod_{j=1}^{s} \tilde{f}_{ij}(\ell_j) \\ &= \sum_i^{\text{poly}(s,d)} \prod_{j=1}^{s} \prod_{k=1}^{d} \left(\ell_j - \zeta_{ijk}\right)\end{aligned}$$

... a $\Sigma\Pi\Sigma$ circuit of poly(s,d) size.

Step 2: $\Sigma\wedge\Sigma\wedge\Sigma$ *to* $\Sigma\Pi\Sigma$

$$T \quad = \quad \left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}}$$

$$\begin{aligned}
\left(\ell_1^{\sqrt{d}} + \cdots + \ell_s^{\sqrt{d}}\right)^{\sqrt{d}} &= \sum_{i}^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} f_{ij}\left(\ell_j^{\sqrt{d}}\right) \\
&= \sum_{i}^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} \tilde{f}_{ij}(\ell_j) \\
&= \sum_{i}^{\mathrm{poly}(s,d)} \prod_{j=1}^{s} \prod_{k=1}^{d} \left(\ell_j - \zeta_{ijk}\right)
\end{aligned}$$

... a $\Sigma\Pi\Sigma$ circuit of poly(s,d) size and degree sd.

Putting it together

general circuit
of size s

Putting it together

general circuit
of size s $\longrightarrow$ $\Sigma\prod^{\sqrt{d}}\Sigma\prod^{\sqrt{d}}$ circuit
of size $s^{O(\sqrt{d})}$

Putting it together

general circuit of size s $\longrightarrow$ $\Sigma\Pi^{\sqrt{d}}\Sigma\Pi^{\sqrt{d}}$ circuit of size $s^{O(\sqrt{d})}$

$\downarrow$

$\Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma$ circuits of size $s^{O(\sqrt{d})} \cdot 2^{O(\sqrt{d})}$

Putting it together

general circuit of size s $\longrightarrow$ $\Sigma\prod^{\sqrt{d}}\Sigma\prod^{\sqrt{d}}$ circuit of size $s^{O(\sqrt{d})}$

$\downarrow$

$\Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma$ circuits of size $s^{O(\sqrt{d})}$

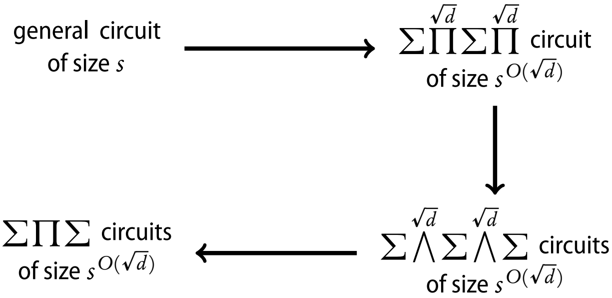
Putting it together

general circuit of size s $\longrightarrow$ $\Sigma\Pi^{\sqrt{d}}\Sigma\Pi^{\sqrt{d}}$ circuit of size $s^{O(\sqrt{d})}$

$\downarrow$

$\Sigma\wedge^{\sqrt{d}}\Sigma\wedge^{\sqrt{d}}\Sigma$ circuits of size $s^{O(\sqrt{d})}$ $\longrightarrow$ $\Sigma\Pi\Sigma$ circuits of size $s^{O(\sqrt{d})}$

Putting it together



Question: Where should one try to prove lower bounds?

Putting it together

general hom. circuit of size s $\longrightarrow$ $\Sigma\prod^{\sqrt{d}}\Sigma\prod^{\sqrt{d}}$ circuit of size $s^{O(\sqrt{d})}$

$\downarrow$

$\Sigma\prod\Sigma$ circuits of size $s^{O(\sqrt{d})}$ $\longleftarrow$ $\Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma$ circuits of size $s^{O(\sqrt{d})}$

Question: Where should one try to prove lower bounds?

general hom. circuit of size s $\longrightarrow$ $\Sigma\Pi^{\sqrt{d}}\Sigma\Pi^{\sqrt{d}}$ hom. circuit of size $s^{O(\sqrt{d})}$

$\downarrow$

$\Sigma\Pi\Sigma$ circuits of size $s^{O(\sqrt{d})}$ $\longleftarrow$ $\Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma$ circuits of size $s^{O(\sqrt{d})}$

Question: Where should one try to prove lower bounds?

Putting it together

general hom. circuit of size s → $\Sigma\prod^{\sqrt{d}}\Sigma\prod^{\sqrt{d}}$ hom. circuit of size $s^{O(\sqrt{d})}$

↓

$\Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma$ hom. circuits of size $s^{O(\sqrt{d})}$ → $\Sigma\prod\Sigma$ circuits of size $s^{O(\sqrt{d})}$

Question: Where should one try to prove lower bounds?

Putting it together

general hom. circuit of size s $\longrightarrow$ $\Sigma\Pi^{\sqrt{d}}\Sigma\Pi^{\sqrt{d}}$ hom. circuit of size $s^{O(\sqrt{d})}$

$\downarrow$

$\Sigma\Pi\Sigma$ **non-hom.** circuits of size $s^{O(\sqrt{d})}$ $\longleftarrow$ $\Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma$ hom. circuits of size $s^{O(\sqrt{d})}$

Question: Where should one try to prove lower bounds?

Putting it together

general hom. circuit of size s $\longrightarrow$ $\Sigma\Pi^{\sqrt{d}}\Sigma\Pi^{\sqrt{d}}$ hom. circuit of size $s^{O(\sqrt{d})}$

$\downarrow$

$\Sigma\Pi\Sigma$ **non-hom.** circuits of size $s^{O(\sqrt{d})}$ $\longleftarrow$ $\Sigma\bigwedge^{\sqrt{d}}\Sigma\bigwedge^{\sqrt{d}}\Sigma$ hom. circuits of size $s^{O(\sqrt{d})}$

Question: Where should one try to prove lower bounds?

Suffices to show this:

Find an explicit $f(x_1, \ldots, x_n)$ such that if

$$\begin{aligned} f(x_1, \ldots, x_n) &= Q_1^{\sqrt{d}} + \ldots + Q_s^{\sqrt{d}} \\ \text{where} \quad \deg Q_i &\leq \sqrt{d} \quad \text{for all } i \end{aligned}$$

then $s = n^{\omega(\sqrt{d})}$.

Suffices to show this:

Find an explicit $f(x_1, \ldots, x_n)$ such that if

$$\begin{aligned} f(x_1,\ldots,x_n) &= Q_1^{\sqrt{d}} + \ldots + Q_s^{\sqrt{d}} \\ \text{where} \quad \deg Q_i &\leq \sqrt{d} \quad \text{for all } i \end{aligned}$$

then $s = n^{\omega(\sqrt{d})}$.

How hard can this be !?

Suffices to show this:

Find an explicit $f(x_1, \ldots, x_n)$ such that if

$$\begin{aligned} f(x_1, \ldots, x_n) &= Q_1^{\sqrt{d}} + \ldots + Q_s^{\sqrt{d}} \\ \text{where} \quad \deg Q_i &\leq \sqrt{d} \quad \text{for all } i \end{aligned}$$

then $s = n^{\omega(\sqrt{d})}$.

How hard can this be !?

Thank you!