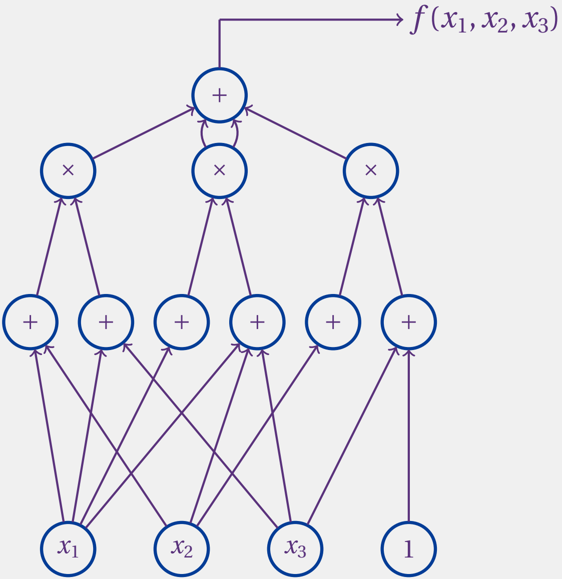
SUBEXPONENTIAL SIZE HITTING SETS FOR BOUNDED DEPTH MULTILINEAR FORMULAS

Ben Lee Volk (Tel Aviv University)

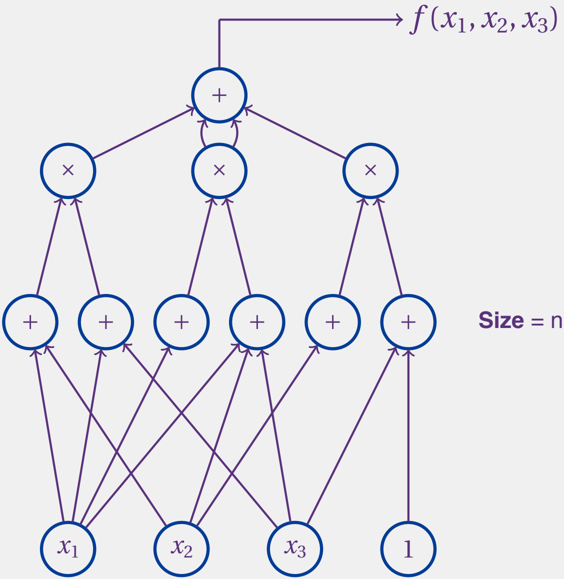
Joint work with

Rafael Oliveira (Princeton University)
Amir Shpilka (Tel Aviv University)

ARITHMETIC CIRCUITS

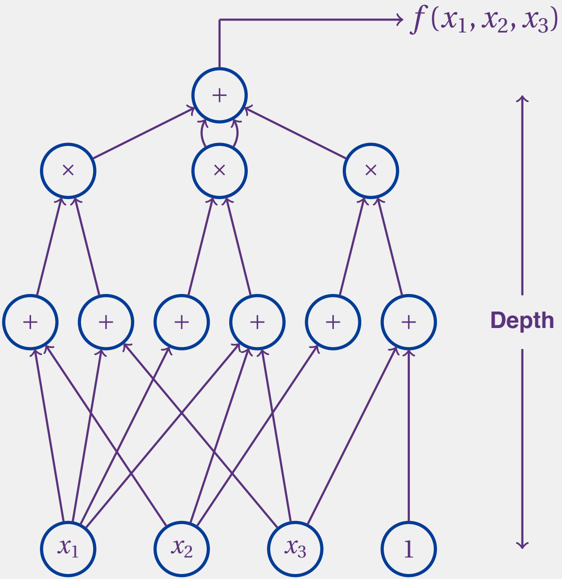


ARITHMETIC CIRCUITS

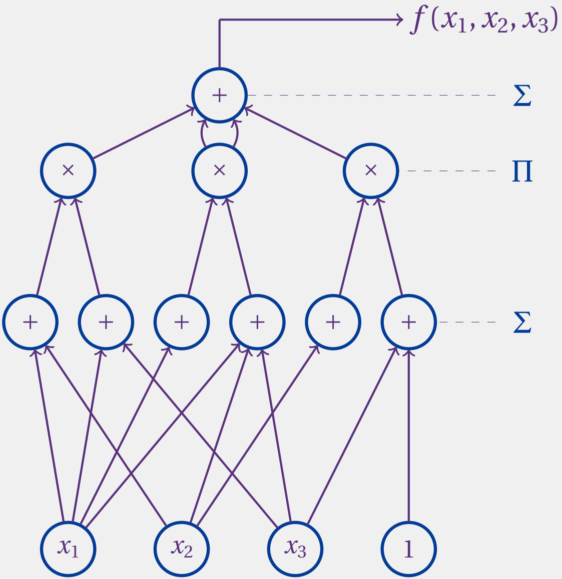


Size = number of gates

ARITHMETIC CIRCUITS

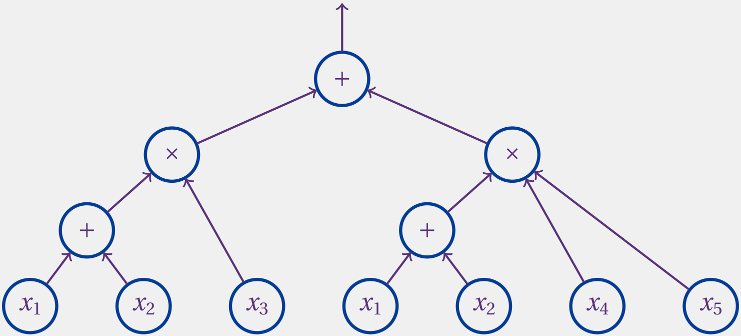


Arithmetic Circuits



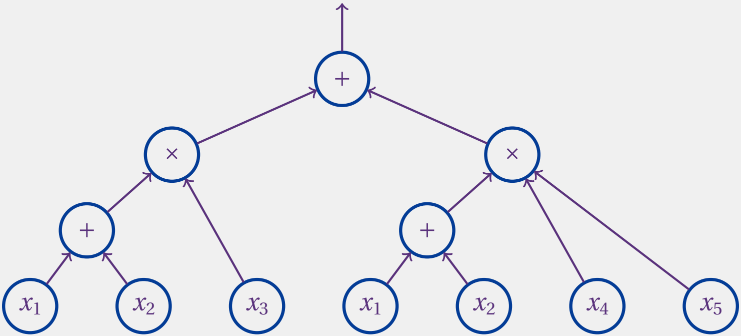
Arithmetic Formulas

Underlying graph is a tree.



ARITHMETIC FORMULAS

Underlying graph is a tree.



Multilinear Formula: every node computes a multilinear polynomial.

POLYNOMIAL IDENTITY TESTING

Given $C(x_1, \ldots, x_n)$,
decide *deterministically* whether $C \equiv 0$.

POLYNOMIAL IDENTITY TESTING

Given $C(x_1, \ldots, x_n)$,
decide *deterministically* whether $C \equiv 0$.

White Box

Can "see" the circuit.

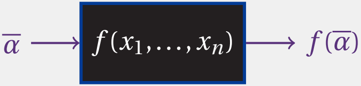
POLYNOMIAL IDENTITY TESTING

Given $C(x_1, \ldots, x_n)$,
decide *deterministically* whether $C \equiv 0$.

White Box

Can "see" the circuit.

Black Box



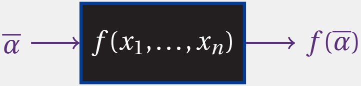
POLYNOMIAL IDENTITY TESTING

Given $C(x_1, \ldots, x_n)$ from a class $\mathcal{C}$
decide *deterministically* whether $C \equiv 0$.

White Box

Can "see" the circuit.

Black Box



POLYNOMIAL IDENTITY TESTING

Given $C(x_1, \ldots, x_n)$ from a class $\mathcal{C}$
decide *deterministically* whether $C \equiv 0$.

White Box

Can "see" the circuit.

Black Box

$$\overline{\alpha} \longrightarrow \boxed{f(x_1, \ldots, x_n)} \longrightarrow f(\overline{\alpha})$$

Black Box PIT $\equiv$ explicit hitting set.
Hitting Set for class $\mathcal{C}$: A set $\mathcal{H} \subseteq \mathbb{F}^n$ such that for every non-zero $f \in \mathcal{C}$ there exists $\overline{\alpha} \in \mathcal{H}$ such that $f(\overline{\alpha}) \neq 0$.

RELATED WORK

PIT for bounded depth circuits:

RELATED WORK

PIT for bounded depth circuits:

Model	Running time

RELATED WORK

PIT for bounded depth circuits:

Model	**Running time**	
Depth 3 circuits Bounded top fan-in	Poly time, BB	**[Kayal-Saraf], [Saxena-Seshadri]**

Related Work

PIT for bounded depth circuits:

Model	Running time	
Depth 3 circuits Bounded top fan-in	Poly time, BB	**[Kayal-Saraf], [Saxena-Seshadri]**
Multilinear Depth 4, Bounded top fan-in	Poly time, BB	**[Saraf-Volkovich]**

RELATED WORK

PIT for bounded depth circuits:

Model	Running time	
Depth 3 circuits Bounded top fan-in	Poly time, BB	**[Kayal-Saraf], [Saxena-Seshadri]**
Multilinear Depth 4, Bounded top fan-in	Poly time, BB	**[Saraf-Volkovich]**
Multilinear Depth 3, Bounded *distance*	Quasi-poly, BB	**[Agrawal-Gurjar-Korwar-Saxena]**

RELATED WORK

PIT for bounded depth circuits:

Model	**Running time**	
Depth 3 circuits Bounded top fan-in	Poly time, BB	**[Kayal-Saraf], [Saxena-Seshadri]**
Multilinear Depth 4, Bounded top fan-in	Poly time, BB	**[Saraf-Volkovich]**
Multilinear Depth 3, Bounded *distance*	Quasi-poly, BB	**[Agrawal-Gurjar-Korwar-Saxena]**

(and many others)

THIS TALK

Hitting sets for bounded depth multilinear formulas, with

THIS TALK

Hitting sets for bounded depth multilinear formulas, with

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+2\delta/3})}$ for depth 3 formulas of size $2^{n^{\delta}}$

THIS TALK

Hitting sets for bounded depth multilinear formulas, with

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+2\delta/3})}$ for depth 3 formulas of size 2^{n^δ}

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+4\delta/3})}$ for depth 4 formulas of size 2^{n^δ}

THIS TALK

Hitting sets for bounded depth multilinear formulas, with

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+2\delta/3})}$ for depth 3 formulas of size 2^{n^δ}

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+4\delta/3})}$ for depth 4 formulas of size 2^{n^δ}

$|\mathcal{H}| = 2^{\tilde{O}(n^{1-1/\exp(d)})}$ for *regular** depth d formulas of size $2^{n^{1/\exp(d)}}$

*All nodes in the same layer have the same fan-in.

THIS TALK

Hitting sets for bounded depth multilinear formulas, with

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+2\delta/3})}$ for depth 3 formulas of size 2^{n^δ}
$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+4\delta/3})}$ for depth 4 formulas of size 2^{n^δ}
$|\mathcal{H}| = 2^{\tilde{O}(n^{1-1/\exp(d)})}$ for *regular** depth d formulas of size $2^{n^{1/\exp(d)}}$

*All nodes in the same layer have the same fan-in.

All constructible in time $\mathrm{poly}(|\mathcal{H}|)$.

THIS TALK

Hitting sets for bounded depth multilinear formulas, with

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+2\delta/3})}$ for depth 3 formulas of size 2^{n^δ}

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+4\delta/3})}$ for depth 4 formulas of size 2^{n^δ}

$|\mathcal{H}| = 2^{\tilde{O}(n^{1-1/\exp(d)})}$ for *regular** depth d formulas of size $2^{n^{1/\exp(d)}}$

*All nodes in the same layer have the same fan-in.

All constructible in time $\mathsf{poly}(|\mathcal{H}|)$.

$\implies$ lower bounds of $2^{\tilde{\Omega}(n^{1/2})}$ (depth 3), $2^{\tilde{\Omega}(n^{1/4})}$ (depth 4) and $2^{\tilde{\Omega}(n^{1/\exp(d)})}$ (depth d) for polynomials in $\mathsf{DTIME}(2^{O(n)})$.

THIS TALK

Hitting sets for bounded depth multilinear formulas, with

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+2\delta/3})}$ for depth 3 formulas of size 2^{n^δ}

$|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+4\delta/3})}$ for depth 4 formulas of size 2^{n^δ}

$|\mathcal{H}| = 2^{\tilde{O}(n^{1-1/\exp(d)})}$ for *regular** depth d formulas of size $2^{n^{1/\exp(d)}}$

*All nodes in the same layer have the same fan-in.

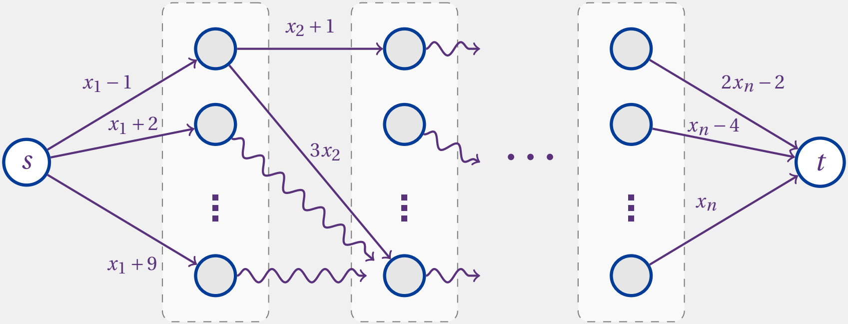
All constructible in time $\mathrm{poly}(|\mathcal{H}|)$.

$\implies$ lower bounds of $2^{\tilde{\Omega}(n^{1/2})}$ (depth 3), $2^{\tilde{\Omega}(n^{1/4})}$ (depth 4) and $2^{\tilde{\Omega}(n^{1/\exp(d)})}$ (depth d) for polynomials in $\mathrm{DTIME}(2^{O(n)})$.

(slightly better lower bounds known but not via hitting sets)

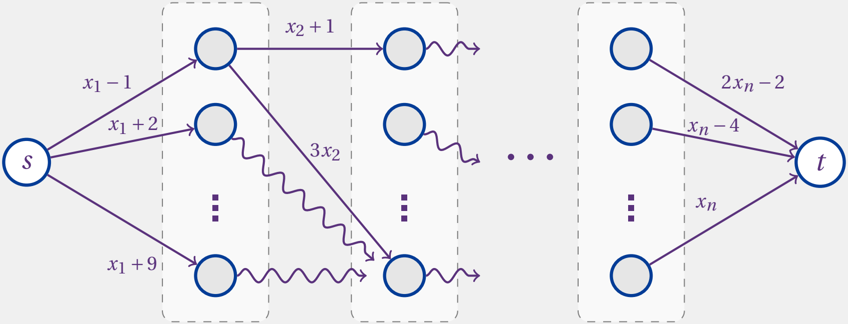
PROOF TECHNIQUE

"Reduction" to *Read Once Algebraic Branching Program*:



PROOF TECHNIQUE

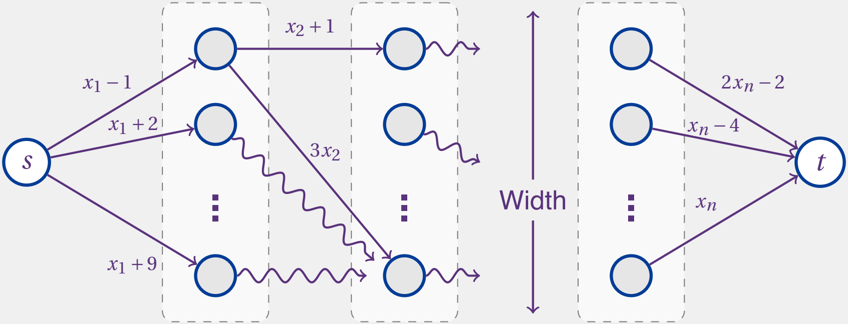
"Reduction" to *Read Once Algebraic Branching Program*:



- Each $s \to t$ path computes multiplication of edge labels
- Program computes the sum of those over all $s \to t$ paths
- *Read Once*: Variable x_i read only at level i

PROOF TECHNIQUE

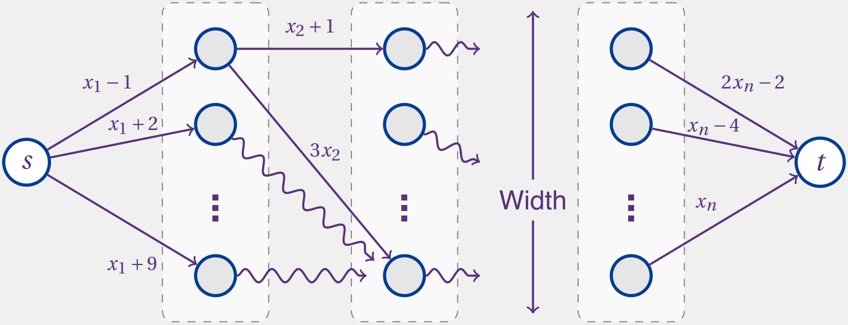
"Reduction" to *Read Once Algebraic Branching Program*:



- Each $s \to t$ path computes multiplication of edge labels
- Program computes the sum of those over all $s \to t$ paths
- *Read Once*: Variable x_i read only at level i

PROOF TECHNIQUE

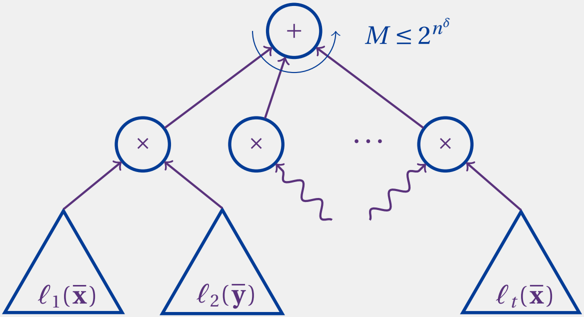
"Reduction" to *Read Once Algebraic Branching Program*:



Theorem [FS13, FSS14, AGKS14]: $\exists$ explicit hitting set for ROABPs of width w of size $\text{poly}(n, w)^{O(\log n)}$.

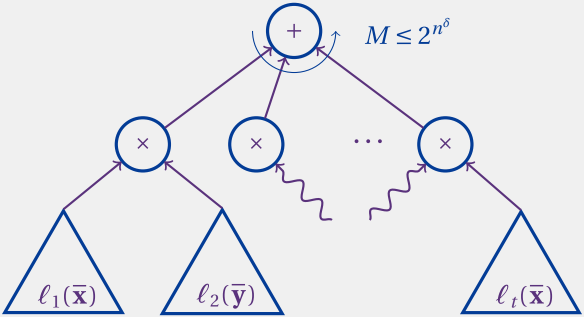
DEPTH 3 MULTILINEAR FORMULAS: $\Sigma\Pi\Sigma$

Sum of products of linear functions with disjoint support:



Depth 3 Multilinear Formulas: $\Sigma\Pi\Sigma$

Sum of products of linear functions with disjoint support:



... if every linear function has only 1 variable in its support, the polynomial is computed by width M ROABP.

Depth 3 Multilinear Formulas: $\Sigma\Pi\Sigma$

What if we had a partition $S_1, \ldots, S_k$ such that all the vars in S_i appear in different linear functions?
(for every $\times$ gate and for every i)

Depth 3 multilinear formulas: $\Sigma\Pi\Sigma$

What if we had a partition $S_1, \ldots, S_k$ such that all the vars in S_i appear in different linear functions?
(for every $\times$ gate and for every i)

Think of the polynomial in the vars of S_1 over $\mathbb{F}(S_2 \cup \cdots \cup S_k)$.

DEPTH 3 MULTILINEAR FORMULAS: $\Sigma\Pi\Sigma$

What if we had a partition $S_1, \ldots, S_k$ such that all the vars in S_i appear in different linear functions?
(for every $\times$ gate and for every i)

Think of the polynomial in the vars of S_1 over $\mathbb{F}(S_2 \cup \cdots \cup S_k)$.
It has ROABP of width M. Plug in hitting set on the vars of S_1.

DEPTH 3 MULTILINEAR FORMULAS: $\Sigma\Pi\Sigma$

What if we had a partition $S_1, \ldots, S_k$ such that all the vars in S_i appear in different linear functions?
(for every $\times$ gate and for every i)

Think of the polynomial in the vars of S_1 over $\mathbb{F}(S_2 \cup \cdots \cup S_k)$.
It has ROABP of width M. Plug in hitting set on the vars of S_1.

$\exists$ assignment which keeps the polynomial non-zero.

Depth 3 Multilinear Formulas: $\Sigma\Pi\Sigma$

What if we had a partition $S_1, \ldots, S_k$ such that all the vars in S_i appear in different linear functions?
(for every $\times$ gate and for every i)

Think of the polynomial in the vars of S_1 over $\mathbb{F}(S_2 \cup \cdots \cup S_k)$.
It has ROABP of width M. Plug in hitting set on the vars of S_1.

$\exists$ assignment which keeps the polynomial non-zero.

Now do the same for $S_2, S_3, \ldots$

Depth 3 Multilinear Formulas: $\Sigma\Pi\Sigma$

Plan:

- Partition the variables into $n^{1-\varepsilon}$ disjoint sets $S_1, \ldots, S_{n^{1-\varepsilon}}$, and hope that the intersection of every linear function with every S_i contains at most 1 variable

Depth 3 Multilinear Formulas: $\Sigma\Pi\Sigma$

Plan:

- Partition the variables into $n^{1-\varepsilon}$ disjoint sets $S_1, \ldots, S_{n^{1-\varepsilon}}$, and hope that the intersection of every linear function with every S_i contains at most 1 variable
- Plug in hitting sets for width M ROABPs on the variables of each S_i

Depth 3 Multilinear Formulas: $\Sigma\Pi\Sigma$

Plan:

- Partition the variables into $n^{1-\varepsilon}$ disjoint sets $S_1, \ldots, S_{n^{1-\varepsilon}}$, and hope that the intersection of every linear function with every S_i contains at most 1 variable
- Plug in hitting sets for width M ROABPs on the variables of each S_i

If all linear functions in the circuit had small support we might expect a random partition to work.

Depth 3 Multilinear Formulas: $\Sigma\Pi\Sigma$

Plan:

- Partition the variables into $n^{1-\varepsilon}$ disjoint sets $S_1, \dots, S_{n^{1-\varepsilon}}$, and hope that the intersection of every linear function with every S_i contains at most 1 variable
- Plug in hitting sets for width M ROABPs on the variables of each S_i

If all linear functions in the circuit had small support we might expect a random partition to work.

Problems:

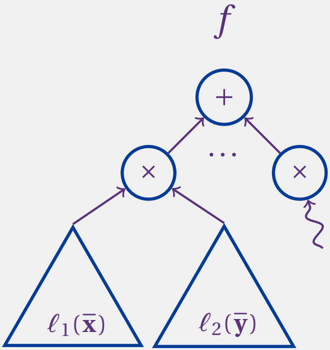
- What about linear functions which contain a lot of variables?
- How to find the partition?

Reducing Bottom Support

Solution to Problem #1: get rid of linear functions with large support (more than n^{ε} is a problem)

REDUCING BOTTOM SUPPORT

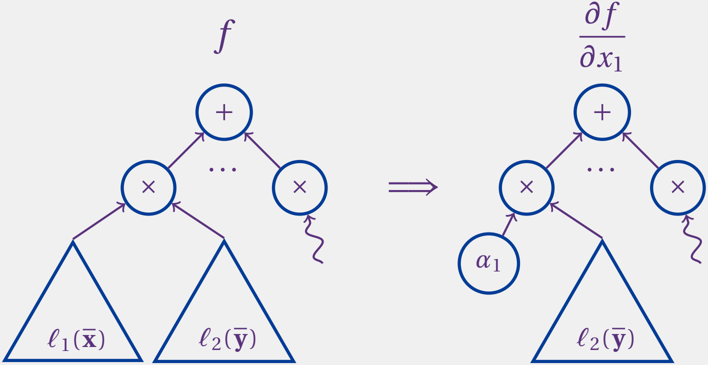
Solution to Problem #1: get rid of linear functions with large support (more than n^{ε} is a problem)



$\ell_1(\bar{\mathbf{x}}) = \sum_i \alpha_i x_i$

Reducing Bottom Support

Solution to Problem #1: get rid of linear functions with large support (more than n^{ε} is a problem)



$\ell_1(\bar{\mathbf{x}}) = \sum_i \alpha_i x_i$

REDUCING BOTTOM SUPPORT

- Only care about linear functions with $\geq n^{\varepsilon}$ vars, so $\exists$ var x_i which works for $1/n^{1-\varepsilon}$ of them.

REDUCING BOTTOM SUPPORT

- Only care about linear functions with $\geq n^{\varepsilon}$ vars, so $\exists$ var x_i which works for $1/n^{1-\varepsilon}$ of them.
- Derive w.r.t $\leq n^{1-\varepsilon} \log M$ vars $\implies$ no linear function remaining with more than n^{ε} vars.

REDUCING BOTTOM SUPPORT

- Only care about linear functions with $\geq n^{\varepsilon}$ vars, so $\exists$ var x_i which works for $1/n^{1-\varepsilon}$ of them.
- Derive w.r.t $\leq n^{1-\varepsilon} \log M$ vars $\implies$ no linear function remaining with more than n^{ε} vars.

How to get black box access to $\frac{\partial f}{\partial x_{i_1} \cdots \partial x_{i_t}}$?

REDUCING BOTTOM SUPPORT

- Only care about linear functions with $\geq n^{\varepsilon}$ vars, so $\exists$ var x_i which works for $1/n^{1-\varepsilon}$ of them.
- Derive w.r.t $\leq n^{1-\varepsilon} \log M$ vars $\implies$ no linear function remaining with more than n^{ε} vars.

How to get black box access to $\dfrac{\partial f}{\partial x_{i_1} \cdots \partial x_{i_t}}$?

$$\frac{\partial f}{\partial x_i} = f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n).$$

REDUCING BOTTOM SUPPORT

- Only care about linear functions with $\geq n^{\varepsilon}$ vars, so $\exists$ var x_i which works for $1/n^{1-\varepsilon}$ of them.
- Derive w.r.t $\leq n^{1-\varepsilon} \log M$ vars $\implies$ no linear function remaining with more than n^{ε} vars.

How to get black box access to $\frac{\partial f}{\partial x_{i_1} \cdots \partial x_{i_t}}$?

$$\frac{\partial f}{\partial x_i} = f(x_1, \ldots, x_{i-1}, 1, x_{i+1}, \ldots, x_n) - f(x_1, \ldots, x_{i-1}, 0, x_{i+1}, \ldots, x_n).$$

Applied repeatedly, each query to the derivative is simulated by 2^t queries to f.

VARIABLE PARTITION

Solution to Problem #2: Settle for less.

VARIABLE PARTITION

Solution to Problem #2: Settle for less.

Assume all linear functions have support $\leq n^{\varepsilon}$.
If we partition the vars randomly to $n^{1-\varepsilon}$ sets $S_1, \ldots, S_{n^{1-\varepsilon}}$, then with high probability, for every set S_i:

1. No linear function intersects S_i in more than n^{δ} vars.

Variable Partition

Solution to Problem #2: Settle for less.

Assume all linear functions have support $\leq n^{\varepsilon}$.
If we partition the vars randomly to $n^{1-\varepsilon}$ sets $S_1, \ldots, S_{n^{1-\varepsilon}}$, then with high probability, for every set S_i:

1. No linear function intersects S_i in more than n^{δ} vars.
2. On every $\times$ gate, the # of linear functions which intersect S_i in at least 2 vars is at most n^{δ}.

Variable Partition

Solution to Problem #2: Settle for less.

Assume all linear functions have support $\leq n^{\varepsilon}$.
If we partition the vars randomly to $n^{1-\varepsilon}$ sets $S_1, \ldots, S_{n^{1-\varepsilon}}$, then with high probability, for every set S_i:

1. No linear function intersects S_i in more than n^{δ} vars.
2. On every $\times$ gate, the # of linear functions which intersect S_i in at least 2 vars is at most n^{δ}.

Brute-force expand linear functions of the second type to get ROABP of width $M \cdot n^{n^{\delta}}$ in vars of S_i.

Variable Partition

Solution to Problem #2: Settle for less.

Assume all linear functions have support $\leq n^{\varepsilon}$.
If we partition the vars randomly to $n^{1-\varepsilon}$ sets $S_1, \ldots, S_{n^{1-\varepsilon}}$, then with high probability, for every set S_i:

1. No linear function intersects S_i in more than n^{δ} vars.
2. On every $\times$ gate, the # of linear functions which intersect S_i in at least 2 vars is at most n^{δ}.

Brute-force expand linear functions of the second type to get ROABP of width $M \cdot n^{n^{\delta}}$ in vars of S_i.

Deterministic version: Partition vars according to n^{δ}-wise independent family of hash functions.

WRAP-UP: DEPTH 3 CASE

WRAP-UP: DEPTH 3 CASE

1. Pick $n^{1-\varepsilon}\log M$ vars and compute a black box for the derivative of f with respect to those vars

WRAP-UP: DEPTH 3 CASE

1. Pick $n^{1-\varepsilon}\log M$ vars and compute a black box for the derivative of f with respect to those vars

   Cost: $\binom{n}{n^{1-\varepsilon}\log M}\cdot 2^{n^{1-\varepsilon}\log M}$

WRAP-UP: DEPTH 3 CASE

1. Pick $n^{1-\varepsilon}\log M$ vars and compute a black box for the derivative of f with respect to those vars

   Cost: $\binom{n}{n^{1-\varepsilon}\log M}\cdot 2^{n^{1-\varepsilon}\log M}$

2. Partition remaining vars to $n^{1-\varepsilon}$ sets using a n^{δ}-wise indep. family of hash functions

WRAP-UP: DEPTH 3 CASE

1. Pick $n^{1-\varepsilon}\log M$ vars and compute a black box for the derivative of f with respect to those vars

   Cost: $\binom{n}{n^{1-\varepsilon}\log M}\cdot 2^{n^{1-\varepsilon}\log M}$

2. Partition remaining vars to $n^{1-\varepsilon}$ sets using a n^{δ}-wise indep. family of hash functions

   Cost: $n^{n^{\delta}}$

WRAP-UP: DEPTH 3 CASE

1. Pick $n^{1-\varepsilon}\log M$ vars and compute a black box for the derivative of f with respect to those vars

   Cost: $\binom{n}{n^{1-\varepsilon}\log M}\cdot 2^{n^{1-\varepsilon}\log M}$

2. Partition remaining vars to $n^{1-\varepsilon}$ sets using a n^{δ}-wise indep. family of hash functions

   Cost: $n^{n^{\delta}}$

3. Plug in a copy of the ROABP hitting set for each set S_i

WRAP-UP: DEPTH 3 CASE

1. Pick $n^{1-\varepsilon}\log M$ vars and compute a black box for the derivative of f with respect to those vars

   Cost: $\binom{n}{n^{1-\varepsilon}\log M}\cdot 2^{n^{1-\varepsilon}\log M}$

2. Partition remaining vars to $n^{1-\varepsilon}$ sets using a n^{δ}-wise indep. family of hash functions

   Cost: $n^{n^{\delta}}$

3. Plug in a copy of the ROABP hitting set for each set S_i

   Cost: $(M\cdot n^{n^{\delta}})^{\log n\cdot n^{1-\varepsilon}}$

WRAP-UP: DEPTH 3 CASE

1. Pick $n^{1-\varepsilon}\log M$ vars and compute a black box for the derivative of f with respect to those vars

   Cost: $\binom{n}{n^{1-\varepsilon}\log M} \cdot 2^{n^{1-\varepsilon}\log M}$

2. Partition remaining vars to $n^{1-\varepsilon}$ sets using a n^{δ}-wise indep. family of hash functions

   Cost: $n^{n^{\delta}}$

3. Plug in a copy of the ROABP hitting set for each set S_i

   Cost: $(M \cdot n^{n^{\delta}})^{\log n \cdot n^{1-\varepsilon}}$

Optimizing parameters: $|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+2\delta/3})}$.

WRAP-UP: DEPTH 3 CASE

1. Pick $n^{1-\varepsilon}\log M$ vars and compute a black box for the derivative of f with respect to those vars

   Cost: $\binom{n}{n^{1-\varepsilon}\log M}\cdot 2^{n^{1-\varepsilon}\log M}$

2. Partition remaining vars to $n^{1-\varepsilon}$ sets using a n^{δ}-wise indep. family of hash functions

   Cost: $n^{n^{\delta}}$

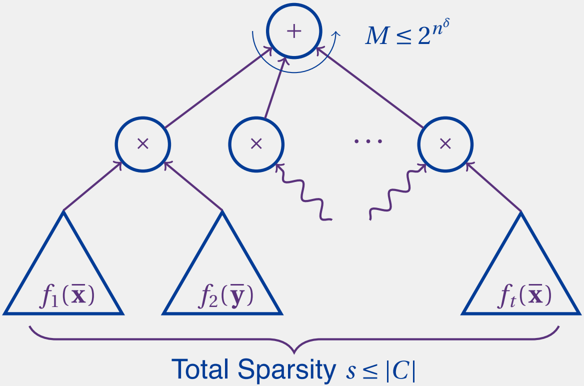
3. Plug in a copy of the ROABP hitting set for each set S_i

   Cost: $(M\cdot n^{n^{\delta}})^{\log n\cdot n^{1-\varepsilon}}$

Optimizing parameters: $|\mathcal{H}| = 2^{\tilde{O}(n^{2/3+2\delta/3})}$.
Lower bound: set $\delta = 1/2 - O(\log\log n/\log n)$. Find non-zero polynomial which vanishes over $\mathcal{H}$.

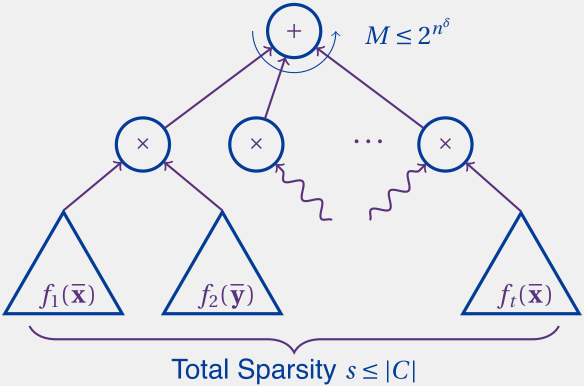
DEPTH 4 MULTILINEAR FORMULAS: $\Sigma\Pi\Sigma\Pi$

Sum of products of sparse polynomials with disjoint support:



DEPTH 4 MULTILINEAR FORMULAS: $\Sigma\Pi\Sigma\Pi$

Sum of products of sparse polynomials with disjoint support:



... what has changed?

REDUCING BOTTOM SUPPORT: DEPTH 4 VERSION

- Can't get rid f_1 by taking a derivative according to x_1

Reducing bottom support: depth 4 version

- Can't get rid f_1 by taking a derivative according to x_1
- But: for every multilinear polynomial f and variable x, either setting $x = 0$ or taking derivative with respect to x reduces the sparsity by at least half

REDUCING BOTTOM SUPPORT: DEPTH 4 VERSION

- Can't get rid f_1 by taking a derivative according to x_1
- But: for every multilinear polynomial f and variable x, either setting $x = 0$ or taking derivative with respect to x reduces the sparsity by at least half
- Similar argument as before: we only care about *high-support* polynomials. $\exists$ var x_i such that either in $f|_{x_i=0}$ or $\frac{\partial f}{\partial x_i}$ the total sparsity of bad polynomials is reduced by a factor of $1 - \frac{1}{2n^{1-\varepsilon}}$

REDUCING BOTTOM SUPPORT: DEPTH 4 VERSION

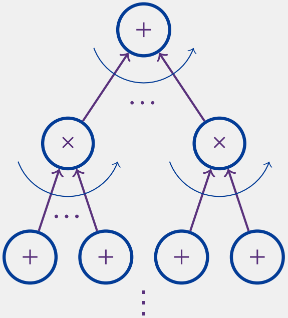
- Can't get rid f_1 by taking a derivative according to x_1
- But: for every multilinear polynomial f and variable x, either setting $x = 0$ or taking derivative with respect to x reduces the sparsity by at least half
- Similar argument as before: we only care about *high-support* polynomials. $\exists$ var x_i such that either in $f|_{x_i=0}$ or $\frac{\partial f}{\partial x_i}$ the total sparsity of bad polynomials is reduced by a factor of $1 - \frac{1}{2n^{1-\varepsilon}}$
- Repeat $O(n^{1-\varepsilon} \log s)$ times to eliminate all high-support polynomials

Reducing Bottom Support: Depth 4 Version

- Can't get rid f_1 by taking a derivative according to x_1
- But: for every multilinear polynomial f and variable x, either setting $x = 0$ or taking derivative with respect to x reduces the sparsity by at least half
- Similar argument as before: we only care about *high-support* polynomials. $\exists$ var x_i such that either in $f|_{x_i=0}$ or $\frac{\partial f}{\partial x_i}$ the total sparsity of bad polynomials is reduced by a factor of $1 - \frac{1}{2n^{1-\varepsilon}}$
- Repeat $O(n^{1-\varepsilon} \log s)$ times to eliminate all high-support polynomials
- Continue as before

REGULAR BOUNDED-DEPTH FORMULAS

[KSS14]

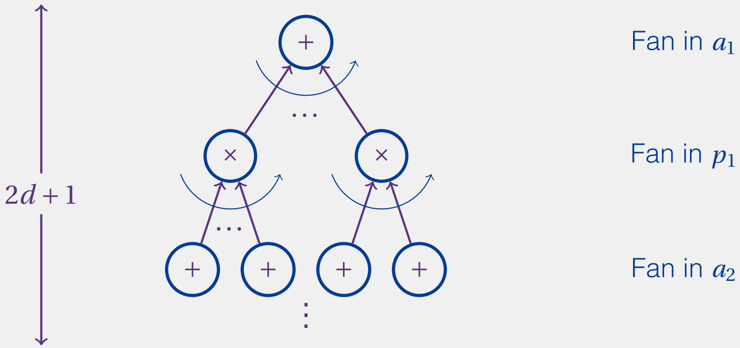


Fan in a_1

Fan in p_1

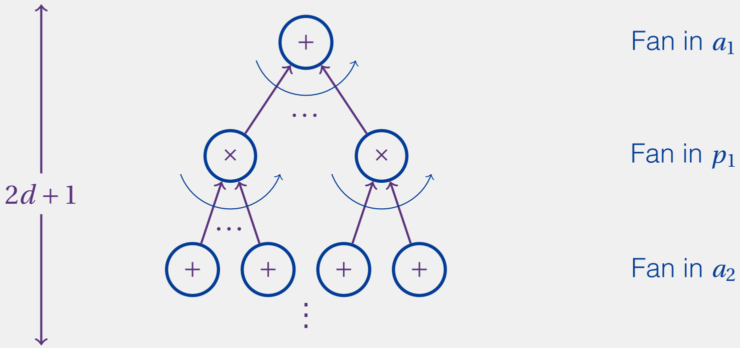
Fan in a_2

Regular Bounded-Depth Formulas



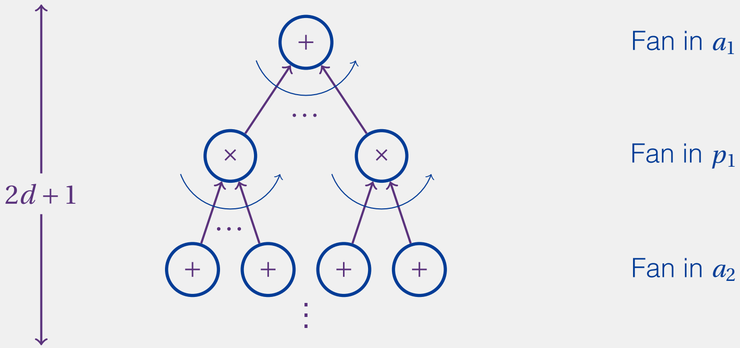
- $d+1$ levels labeled '+', d labeled '×'

REGULAR BOUNDED-DEPTH FORMULAS



- $d+1$ levels labeled '+', d labeled '×'
- Total degree: $\prod_{i=1}^{d} p_i$

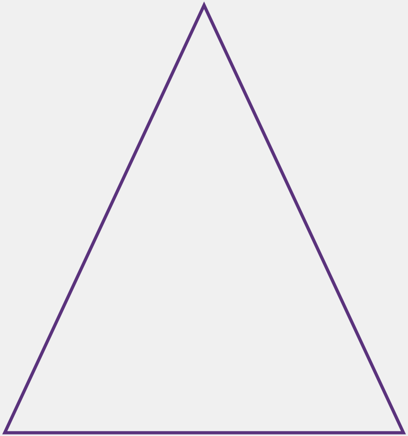
REGULAR BOUNDED-DEPTH FORMULAS



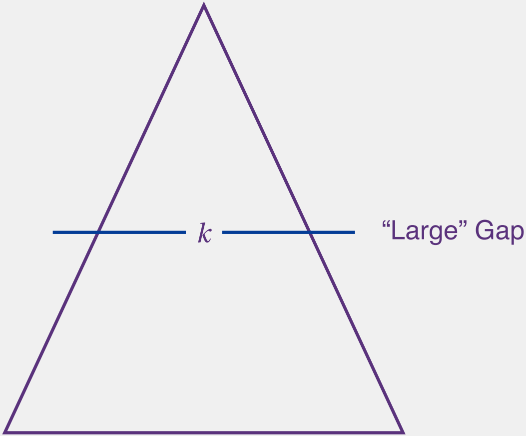
- $d+1$ levels labeled '+', d labeled '×'
- Total degree: $\prod_{i=1}^{d} p_i$

(actually fan-in of + gates is not that important)

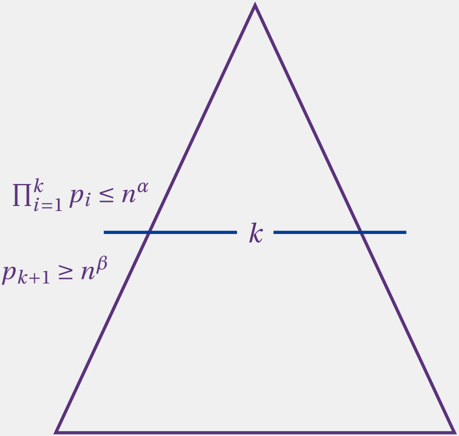
REDUCTION TO DEPTH 4



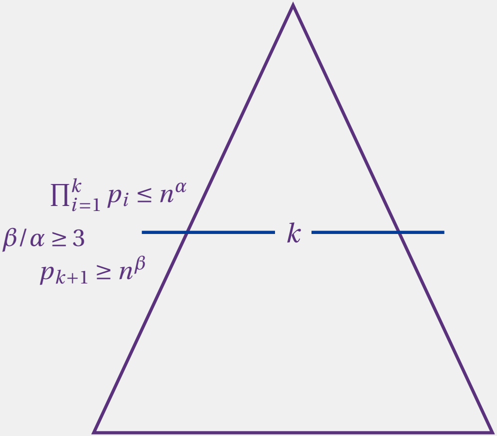
REDUCTION TO DEPTH 4



REDUCTION TO DEPTH 4



Reduction to Depth 4

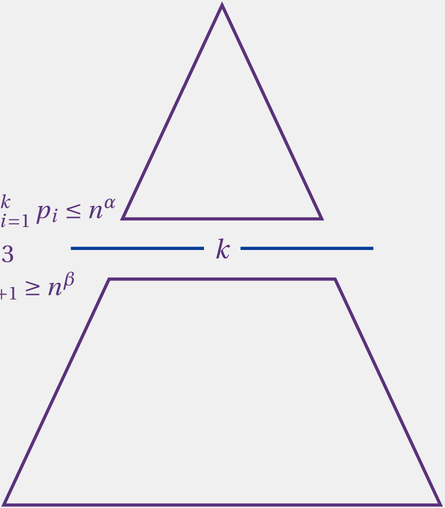


Reduction to Depth 4

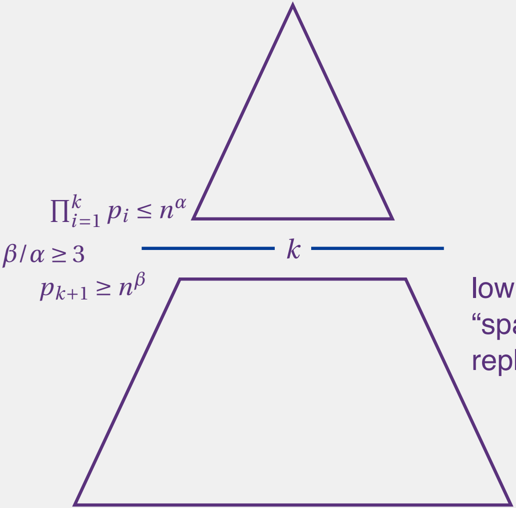
$\prod_{i=1}^{k} p_i \leq n^{\alpha}$

$\beta / \alpha \geq 3$

$p_{k+1} \geq n^{\beta}$



Reduction to depth 4



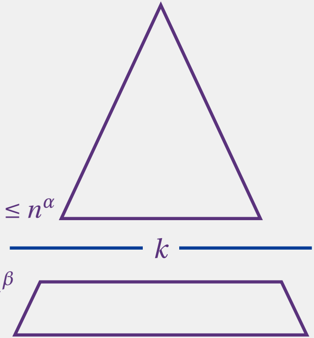
lower part: $\deg \leq n^{1-1/\exp(d)}$
"sparse" polynomial
replace w/ subexp. $\Sigma\Pi$ ckt

REDUCTION TO DEPTH 4

$\prod_{i=1}^{k} p_i \leq n^{\alpha}$

$\beta / \alpha \geq 3$

$p_{k+1} \geq n^{\beta}$

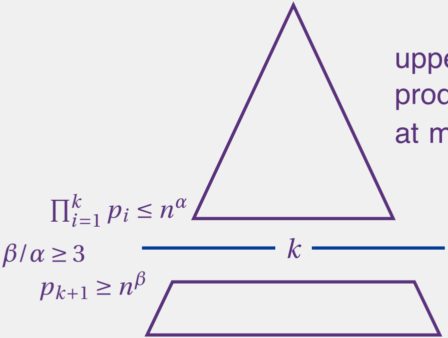


lower part: $\deg \leq n^{1-1/\exp(d)}$

“sparse” polynomial

replace w/ subexp. $\Sigma\Pi$ ckt

REDUCTION TO DEPTH 4



upper part: expand all products from level k upwards, at most $|C|^{\prod_{i=1}^{k} p_i}$ summands

lower part: $\deg \leq n^{1-1/\exp(d)}$
"sparse" polynomial
replace w/ subexp. $\Sigma\Pi$ ckt

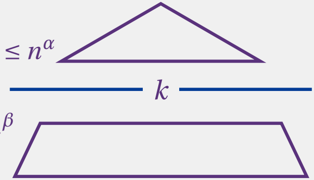
REDUCTION TO DEPTH 4

upper part: expand all products from level k upwards, at most $|C|^{\prod_{i=1}^{k} p_i}$ summands

$\prod_{i=1}^{k} p_i \leq n^{\alpha}$

$\beta / \alpha \geq 3$

$p_{k+1} \geq n^{\beta}$



lower part: $\deg \leq n^{1-1/\exp(d)}$
"sparse" polynomial
replace w/ subexp. $\Sigma\Pi$ ckt

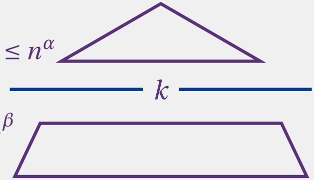
REDUCTION TO DEPTH 4

upper part: expand all products from level k upwards, at most $|C|^{\prod_{i=1}^{k} p_i}$ summands

$\prod_{i=1}^{k} p_i \leq n^{\alpha}$

$\beta / \alpha \geq 3$

$p_{k+1} \geq n^{\beta}$



lower part: $\deg \leq n^{1-1/\exp(d)}$
"sparse" polynomial
replace w/ subexp. $\Sigma\Pi$ ckt

such a large gap is required to match the depth 4 parameters. will be nice to improve.

OPEN PROBLEMS

OPEN PROBLEMS

- Smaller hitting sets for depth 3 and depth 4 formulas.

OPEN PROBLEMS

- Smaller hitting sets for depth 3 and depth 4 formulas.
- Improving the depth d case

OPEN PROBLEMS

- Smaller hitting sets for depth 3 and depth 4 formulas.
- Improving the depth d case

  Either by improving the depth 4 parameters,
  or the reduction to depth 4 formulas

Open Problems

- Smaller hitting sets for depth 3 and depth 4 formulas.
- Improving the depth d case

  Either by improving the depth 4 parameters,
  or the reduction to depth 4 formulas
- non-reg depth d formulas?

OPEN PROBLEMS

- Smaller hitting sets for depth 3 and depth 4 formulas.
- Improving the depth d case

  Either by improving the depth 4 parameters,
  or the reduction to depth 4 formulas

- non-reg depth d formulas?

THANK YOU