A Near-Optimal Depth-Hierarchy Theorem for Small-Depth Multilinear Circuits

Nutan Limaye
Compuer Science and Engineering Department,
Indian Institute of Technology, Bombay, (IITB) India.
Joint work with

Suryajith Chillara, Christian Engels, Srikanth Srinivasan.
IIT Bombay, India.

Seminar 18391 – Algebraic Methods in Computational Complexity
Dagstuhl, September 2018.

More Resources

More Resources

More Resources ⇒ More power?

Power of resources

Turing machines

Power of resources

Turing machines

With more time, can Turing machines compute more languages?

Power of resources

Turing machines

With more time, can Turing machines compute more languages?

With more space, can Turing machines compute more languages?

Power of resources

Turing machines

With more time, can Turing machines compute more languages?

With more space, can Turing machines compute more languages?

Theorem (Time Hierarchy Theorem, [Hartmanis and Stearns, 65])

There exists a language L that is computed by a TM in time $O(t(n) \log t(n))$ such that no TM running in time $o(t(n))$ can compute it.

Power of resources

Turing machines

With more time, can Turing machines compute more languages?

With more space, can Turing machines compute more languages?

Theorem (Time Hierarchy Theorem, [Hartmanis and Stearns, 65])

There exists a language L that is computed by a TM in time $O(t(n) \log t(n))$ such that no TM running in time $o(t(n))$ can compute it.

Theorem (Space Hierarchy Theorem, [Stearns, Hartmanis, Lewis, 65])

There exists a language L that is computed by a TM in space $O(s(n))$ such that no TM running in space $o(s(n))$ can compute it.

Power of resources

Turing machines

With more time, can Turing machines compute more languages?

With more space, can Turing machines compute more languages?

Theorem (Time Hierarchy Theorem, [Hartmanis and Stearns, 65])

There exists a language L that is computed by a TM in time $O(t(n) \log t(n))$ such that no TM running in time $o(t(n))$ can compute it.

Theorem (Space Hierarchy Theorem, [Stearns, Hartmanis, Lewis, 65])

There exists a language L that is computed by a TM in space $O(s(n))$ such that no TM running in space $o(s(n))$ can compute it.

Non-explicit separations.

Power of resources

We can ask a similar question for any model of computation and a resource.

Power of resources

We can ask a similar question for any model of computation and a resource.

Model of computation	Resources of interest

Power of resources

We can ask a similar question for any model of computation and a resource.

Model of computation	Resources of interest
Turing machines	time, space, number of random bits, non-determinism, advice

Power of resources

We can ask a similar question for any model of computation and a resource.

Model of computation	Resources of interest
Turing machines	time, space, number of random bits, non-determinism, advice
Boolean circuits	size, depth

Power of resources

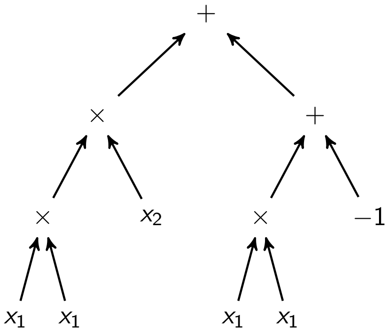
We can ask a similar question for any model of computation and a resource.

Model of computation	Resources of interest
Turing machines	time, space, number of random bits, non-determinism, advice
Boolean circuits	size, depth

Today we will focus on arithmetic formulas as a model of computation.

A model of computation for polynomials

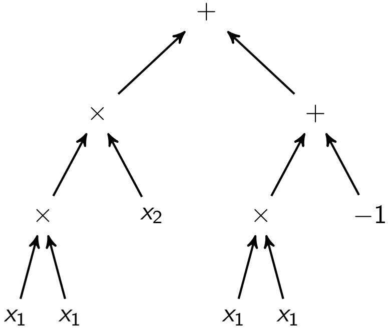
Arithmetic formula



Definition: An arithmetic formula

A model of computation for polynomials

Arithmetic formula

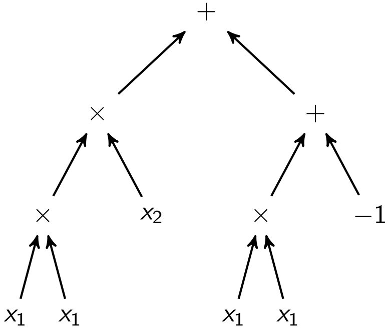


Definition: An arithmetic formula

a directed tree

A model of computation for polynomials

Arithmetic formula



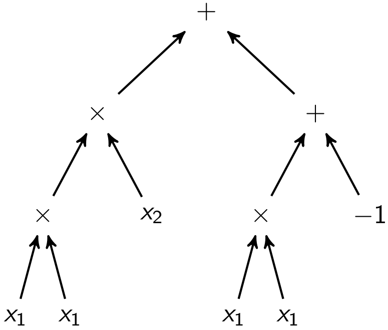
Definition: An arithmetic formula

a directed tree

with nodes labeled by $+$, $\times$, $x_1, \ldots, x_n$ or constants .

A model of computation for polynomials

Arithmetic formula



Definition: An arithmetic formula

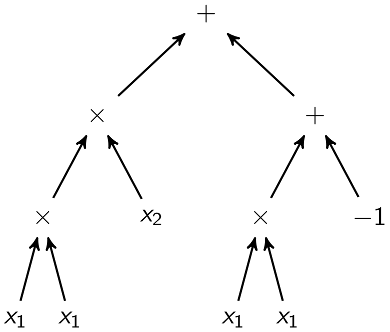
a directed tree

with nodes labeled by $+$, $\times$, $x_1, \ldots, x_n$ or constants .

indegree 0 nodes : input gates

A model of computation for polynomials

Arithmetic formula



Definition: An arithmetic formula

a directed tree

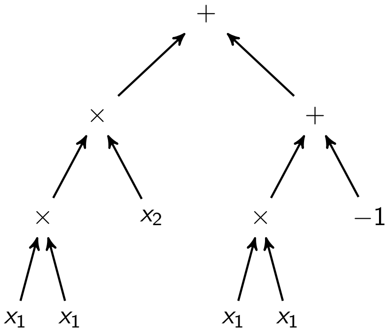
with nodes labeled by $+$, $\times$, $x_1, \ldots, x_n$ or constants .

indegree 0 nodes : input gates

outdegree 0 nodes: output gates

A model of computation for polynomials

Arithmetic formula



Definition: An arithmetic formula

a directed tree

with nodes labeled by $+$, $\times$, $x_1, \ldots, x_n$ or constants .

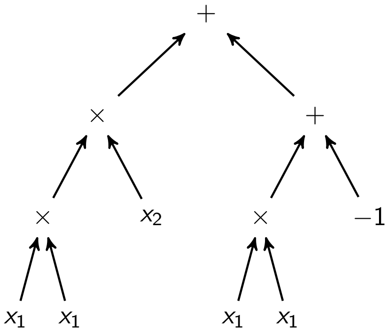
indegree 0 nodes : input gates

outdegree 0 nodes: output gates

without loss of generality alternate $+$, $\times$ layers.

A model of computation for polynomials

Arithmetic formula



Definition: An arithmetic formula

a directed tree

with nodes labeled by $+$, $\times$, $x_1, \ldots, x_n$ or constants (say over $\mathbb{F}$).

indegree 0 nodes : input gates

outdegree 0 nodes: output gates

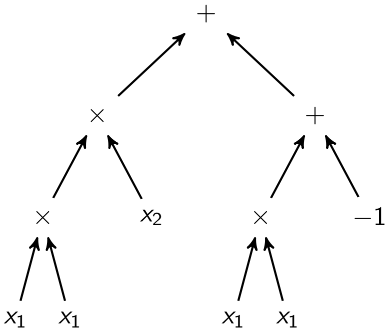
without loss of generality alternate $+$, $\times$ layers.

Size and product-depth

The number of nodes in the tree is the size of the formula.

A model of computation for polynomials

Arithmetic formula



Definition: An arithmetic formula

a directed tree

with nodes labeled by $+$, $\times$, $x_1, \ldots, x_n$ or constants (say over $\mathbb{F}$).

indegree 0 nodes : input gates

outdegree 0 nodes: output gates

without loss of generality alternate $+$, $\times$ layers.

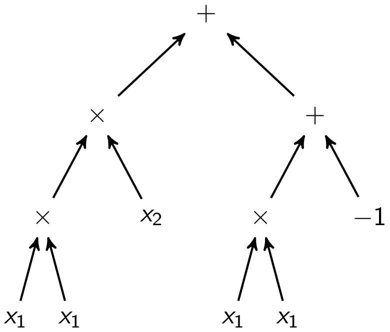
Size and product-depth

The number of nodes in the tree is the size of the formula.

The maximum number of product gates along any leaf to root path is its product-depth.

A model of computation for polynomials

Arithmetic formula



Definition: An arithmetic formula

a directed tree

with nodes labeled by $+$, $\times$, $x_1, \ldots, x_n$ or constants (say over $\mathbb{F}$).

indegree 0 nodes : input gates

outdegree 0 nodes: output gates

without loss of generality alternate $+$, $\times$ layers.

Size and product-depth

The number of nodes in the tree is the size of the formula.

The maximum number of product gates along any leaf to root path is its product-depth. (Closely related to the depth.)

Small depth formulas

We will focus on small product-depth formulas.

Small depth formulas

We will focus on small product-depth formulas.

Product-depth $\Delta = 1$: $\sum \prod$ formulas

Small depth formulas

We will focus on small product-depth formulas.

Product-depth $\Delta = 1$: $\sum \prod$ formulas

The model is complete.

Small depth formulas

We will focus on small product-depth formulas.
Product-depth $\Delta = 1$: $\sum \prod$ formulas

The model is complete.

Let $X = \{x_1, \ldots, x_n\}$.

Small depth formulas

We will focus on small product-depth formulas.
Product-depth $\Delta = 1$: $\sum\prod$ formulas

The model is complete.

Let $X = \{x_1, \ldots, x_n\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d polynomial.

Small depth formulas

We will focus on small product-depth formulas.

Product-depth $\Delta = 1$: $\sum\prod$ formulas

The model is complete.

Let $X = \{x_1, \ldots, x_n\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d polynomial.

$$p(x) = \sum_{m \in \mathcal{M}} c_m \cdot m,$$

Small depth formulas

We will focus on small product-depth formulas.
Product-depth $\Delta = 1$: $\sum\prod$ formulas

The model is complete.

Let $X = \{x_1, \ldots, x_n\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d polynomial.

$$p(x) = \sum_{m \in \mathcal{M}} c_m \cdot m,$$

where $\mathcal{M}$ set of all monomials in n variables of degree at most d.

Small depth formulas

We will focus on small product-depth formulas.

Product-depth $\Delta = 1$: $\sum\prod$ formulas

The model is complete.

Let $X = \{x_1, \ldots, x_n\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d polynomial.

$$p(x) = \sum_{m \in \mathcal{M}} c_m \cdot m,$$

where $\mathcal{M}$ set of all monomials in n variables of degree at most d.

It may not always be a succinct representation.

Examples of polynomials

$$\text{Let } p(X) = \sum_{S \subseteq [n]} \prod_{i \in S} x_i$$

Examples of polynomials

$$\text{Let } p(X) = \sum_{S \subseteq [n]} \prod_{i \in S} x_i$$

This has size $O(2^n)$.

Examples of polynomials

$$\text{Let } p(X) = \sum_{S \subseteq [n]} \prod_{i \in S} x_i$$

This has size $O(2^n)$.

However, here is its succinct representation.

Examples of polynomials

$$\text{Let } p(X) = \sum_{S \subseteq [n]} \prod_{i \in S} x_i$$

This has size $O(2^n)$.

However, here is its succinct representation.

$p(X) = \prod_{i \in [n]} (1 + x_i)$

Examples of polynomials

$$\text{Let } p(X) = \sum_{S \subseteq [n]} \prod_{i \in S} x_i$$

This has size $O(2^n)$.

However, here is its succinct representation.

$p(X) = \prod_{i \in [n]} (1 + x_i)$ of size $O(n)$.

Examples of polynomials

$$\text{Let } p(X) = \sum_{S \subseteq [n]} \prod_{i \in S} x_i$$

This has size $O(2^n)$.

However, here is its succinct representation.

$p(X) = \prod_{i \in [n]} (1 + x_i)$ of size $O(n)$.

This is a $\prod \sum$ formula for the same polynomial.

Multilinear polynomials

Let $X = \{x_1, \ldots, x_N\}$.

Multilinear polynomials

Let $X = \{x_1, \ldots, x_N\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d multilinear polynomial.

Multilinear polynomials

Let $X = \{x_1, \ldots, x_N\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d multilinear polynomial.

$$p(x) = \sum_{S \in [n] : |S| \leq d} c_S \cdot \prod_{i \in S} x_i,$$

Multilinear polynomials

Let $X = \{x_1, \ldots, x_N\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d multilinear polynomial.

$$p(x) = \sum_{S \in [n] : |S| \leq d} c_S \cdot \prod_{i \in S} x_i,$$

Many interesting polynomials are multilinear.

Multilinear polynomials

Let $X = \{x_1, \ldots, x_N\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d multilinear polynomial.

$$p(x) = \sum_{S \in [n]:|S| \leq d} c_S \cdot \prod_{i \in S} x_i,$$

Many interesting polynomials are multilinear.

Determinant: $\mathrm{Det}(X) = \sum_{\sigma \in S_n} sgn(\sigma) \prod_{i=1}^{n} x_{i\sigma(i)}$

Multilinear polynomials

Let $X = \{x_1, \ldots, x_N\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d multilinear polynomial.

$$p(x) = \sum_{S \in [n] : |S| \leq d} c_S \cdot \prod_{i \in S} x_i,$$

Many interesting polynomials are multilinear.

Determinant: $\mathrm{Det}(X) = \sum_{\sigma \in S_n} sgn(\sigma) \prod_{i=1}^{n} x_{i\sigma(i)}$

Permanent: $\mathrm{Perm}(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i\sigma(i)}$

Multilinear polynomials

Let $X = \{x_1, \ldots, x_N\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d multilinear polynomial.

$$p(x) = \sum_{S \in [n] : |S| \leq d} c_S \cdot \prod_{i \in S} x_i,$$

Many interesting polynomials are multilinear.

Determinant: $\mathrm{Det}(X) = \sum_{\sigma \in S_n} sgn(\sigma) \prod_{i=1}^{n} x_{i\sigma(i)}$

Permanent: $\mathrm{Perm}(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i\sigma(i)}$

Matrix Multiplication: $(X \times Y)_{i,j} = \sum_{k=1}^{n} x_{ik} \times y_{kj}$

Multilinear polynomials

Let $X = \{x_1, \ldots, x_N\}$. Let $p(X) \in \mathbb{F}[X]$ be a degree d multilinear polynomial.

$$p(x) = \sum_{S \in [n]:|S| \leq d} c_S \cdot \prod_{i \in S} x_i,$$

Many interesting polynomials are multilinear.

Determinant: $\mathrm{Det}(X) = \sum_{\sigma \in S_n} sgn(\sigma) \prod_{i=1}^{n} x_{i\sigma(i)}$

Permanent: $\mathrm{Perm}(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} x_{i\sigma(i)}$

Matrix Multiplication: $(X \times Y)_{i,j} = \sum_{k=1}^{n} x_{ik} \times y_{kj}$

Multilinear formulas

A formula is multilinear if every gate in it computes a multilinear polynomial.

Multilinear formulas

A formula is multilinear if every gate in it computes a multilinear polynomial.

Many tools and techniques

A breakthrough result of Raz [Raz04] gave a strong lower bound.

Multilinear formulas

A formula is multilinear if every gate in it computes a multilinear polynomial.

Many tools and techniques

A breakthrough result of Raz [Raz04] gave a strong lower bound.

Multilinear formulas for Det/Perm must have superpolynomial size.

Multilinear formulas

A formula is multilinear if every gate in it computes a multilinear polynomial.

Many tools and techniques

A breakthrough result of Raz [Raz04] gave a strong lower bound.

Multilinear formulas for Det/Perm must have superpolynomial size.

A set of tools introduced in [Raz04].

Multilinear formulas

A formula is multilinear if every gate in it computes a multilinear polynomial.

Many tools and techniques

- A breakthrough result of Raz [Raz04] gave a strong lower bound.
- Multilinear formulas for Det/Perm must have superpolynomial size.
- A set of tools introduced in [Raz04].
- Extended and appended by a line of work. [Raz06,RSY07,RY09,DMPY12,KV17].

Small depth formulas

We will focus on small product-depth multilinear formulas.

Small depth formulas

We will focus on small product-depth multilinear formulas.
Product-depth $\Delta = 1$: $\sum\prod$ or $\sum\prod\sum$ formulas

Small depth formulas

We will focus on small product-depth multilinear formulas.

Product-depth $\Delta = 1$: $\sum\prod$ or $\sum\prod\sum$ formulas

$\sum\prod$ formulas are not succinct.

Small depth formulas

We will focus on small product-depth multilinear formulas.

Product-depth $\Delta = 1$: $\sum\prod$ or $\sum\prod\sum$ formulas

$\sum\prod$ formulas are not succinct.

What about $\sum\prod\sum$ formulas?

Small depth formulas

We will focus on small product-depth multilinear formulas.

Product-depth $\Delta = 1$: $\sum\prod$ or $\sum\prod\sum$ formulas

$\sum\prod$ formulas are not succinct.

What about $\sum\prod\sum$ formulas?

$$p(x) = \sum_{i\in[s]} \prod_{j\in[s']} L_{i,j}, \text{where, } L_{i,j} \text{ are linear polynomials in } X.$$

The model is surprisingly powerful!

Small depth formulas

We will focus on small product-depth multilinear formulas.

Product-depth $\Delta = 1$: $\sum\prod$ or $\sum\prod\sum$ formulas

$\sum\prod$ formulas are not succinct.

What about $\sum\prod\sum$ formulas?

$$p(x) = \sum_{i\in[s]} \prod_{j\in[s']} L_{i,j}, \text{where, } L_{i,j} \text{ are linear polynomials in } X.$$

The model is surprisingly powerful! [AV08,Koi09,Tav10,GKKS12]

Any polynomial on n variables of degree d computable by a size s circuit can be computed by

Small depth formulas

We will focus on small product-depth multilinear formulas.

Product-depth $\Delta = 1$: $\sum\prod$ or $\sum\prod\sum$ formulas

$\sum\prod$ formulas are not succinct.

What about $\sum\prod\sum$ formulas?

$$p(x) = \sum_{i \in [s]} \prod_{j \in [s']} L_{i,j}, \text{where, } L_{i,j} \text{ are linear polynomials in } X.$$

The model is surprisingly powerful! [AV08,Koi09,Tav10,GKKS12]

Any polynomial on n variables of degree d computable by a size s circuit can be computed by $\sum\prod\sum$ formula of size $s^{O(\sqrt{d})}$.

Small depth formulas

We will focus on small product-depth multilinear formulas.
Product-depth $\Delta = 1$: $\sum\prod$ or $\sum\prod\sum$ formulas

$\sum\prod$ formulas are not succinct.

What about $\sum\prod\sum$ formulas?

$$p(x) = \sum_{i\in[s]}\prod_{j\in[s']} L_{i,j}, \text{where, } L_{i,j} \text{ are linear polynomials in } X.$$

The model is surprisingly powerful! [AV08,Koi09,Tav10,GKKS12]
Any polynomial on n variables of degree d computable by a size s circuit can be computed by $\sum\prod\sum$ formula of size $s^{O(\sqrt{d})}$.
(Assume characteristic 0.)

Small depth formulas

We will focus on small product-depth multilinear formulas.

Product-depth $\Delta = 1$: $\sum\prod$ or $\sum\prod\sum$ formulas

$\sum\prod$ formulas are not succinct.

What about $\sum\prod\sum$ formulas?

$$p(x) = \sum_{i \in [s]} \prod_{j \in [s']} L_{i,j}, \text{where, } L_{i,j} \text{ are linear polynomials in } X.$$

The model is surprisingly powerful! [AV08,Koi09,Tav10,GKKS12]

Any polynomial on n variables of degree d computable by a size s circuit can be computed by $\sum\prod\sum$ formula of size $s^{O(\sqrt{d})}$.

(Assume characteristic 0.)

This $\sum\prod\sum$ realization is non-multilinear!

Product-depth $\Delta = 1$

Non-multilinear to multilinear formula conversion.

Product-depth $\Delta = 1$

Non-multilinear to multilinear formula conversion.

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum$ formula of size s.

Product-depth $\Delta = 1$

Non-multilinear to multilinear formula conversion.

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum$ formula of size s.

Does $p(X)$ have a $\sum\prod\sum$ multilinear formula of size $s^{O(1)}$?

Product-depth $\Delta = 1$

Non-multilinear to multilinear formula conversion.

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum$ formula of size s.

Does $p(X)$ have a $\sum\prod\sum$ multilinear formula of size $s^{O(1)}$?

[Chillara, **L**, Srinivasan, 18] prove that the answer is no.

Product-depth $\Delta = 1$

Non-multilinear to multilinear formula conversion.

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum$ formula of size s.

Does $p(X)$ have a $\sum\prod\sum$ multilinear formula of size $s^{O(1)}$?

[Chillara, **L**, Srinivasan, 18] prove that the answer is no.

Product-depth $\Delta = 2$ to $\Delta = 1$ conversion

Product-depth $\Delta = 1$

Non-multilinear to multilinear formula conversion.

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum$ formula of size s.

Does $p(X)$ have a $\sum\prod\sum$ multilinear formula of size $s^{O(1)}$?

[Chillara, **L**, Srinivasan, 18] prove that the answer is no.

Product-depth $\Delta = 2$ to $\Delta = 1$ conversion

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum\prod$ multilinear formula of size s.

Product-depth $\Delta = 1$

Non-multilinear to multilinear formula conversion.

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum$ formula of size s.

Does $p(X)$ have a $\sum\prod\sum$ multilinear formula of size $s^{O(1)}$?

[Chillara, **L**, Srinivasan, 18] prove that the answer is no.

Product-depth $\Delta = 2$ to $\Delta = 1$ conversion

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum\prod$ multilinear formula of size s.

Does $p(X)$ have a $\sum\prod\sum$ multilinear formula of size $s^{O(1)}$?

Product-depth $\Delta = 1$

Non-multilinear to multilinear formula conversion.

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum$ formula of size s.

Does $p(X)$ have a $\sum\prod\sum$ multilinear formula of size $s^{O(1)}$?

[Chillara, **L**, Srinivasan, 18] prove that the answer is no.

Product-depth $\Delta = 2$ to $\Delta = 1$ conversion

Let $p(X)$ be a multilinear polynomial computable by a $\sum\prod\sum\prod$ multilinear formula of size s.

Does $p(X)$ have a $\sum\prod\sum$ multilinear formula of size $s^{O(1)}$?

[Kayal, Nair, Saha, 15] show that this is not possible.

How expensive $\Delta = 2 \longrightarrow \Delta = 1$?

Consider $\sum\prod\sum\prod$ formula of size s.

How expensive $\Delta = 2 \longrightarrow \Delta = 1$?

Consider $\sum\prod\sum\prod$ formula of size s.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$.

How expensive $\Delta = 2 \longrightarrow \Delta = 1$?

Consider $\sum\prod\sum\prod$ formula of size s.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$.

That is, $\sum\left(\prod^{[t]}\sum\right)\prod$.

How expensive $\Delta = 2 \longrightarrow \Delta = 1$?

Consider $\sum\prod\sum\prod$ formula of size s.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$.

That is, $\sum\left(\prod^{[t]}\sum\right)\prod$.

Open up the multiplication of summands as a sum of multiplications.

How expensive $\Delta = 2 \longrightarrow \Delta = 1$?

Consider $\sum\prod\sum\prod$ formula of size s.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$.

That is, $\sum\left(\prod^{[t]}\sum\right)\prod$.

Open up the multiplication of summands as a sum of multiplications.

$$\sum\left(\prod^{[t]}\sum\right)\prod \longrightarrow \sum\left(\sum^{[\exp(t)]}\prod\right)\prod$$

How expensive $\Delta = 2 \longrightarrow \Delta = 1$?

Consider $\sum \prod \sum \prod$ formula of size s.

Consider the $\prod \sum$ layer $\prod_{i \in [t]} Q_i$.

That is, $\sum \left(\prod^{[t]} \sum \right) \prod$.

Open up the multiplication of summands as a sum of multiplications.

$$\sum \left(\prod^{[t]} \sum \right) \prod \longrightarrow \sum \left(\sum^{[\exp(t)]} \prod \right) \prod \longrightarrow \sum^{[\exp(t)]} \prod$$

How expensive $\Delta = 2 \longrightarrow \Delta = 1$?

Consider $\sum \prod \sum \prod$ formula of size s.

Consider the $\prod \sum$ layer $\prod_{i \in [t]} Q_i$.

That is, $\sum \left(\prod^{[t]} \sum \right) \prod$.

Open up the multiplication of summands as a sum of multiplications.

$$\sum \left(\prod^{[t]} \sum \right) \prod \longrightarrow \sum \left(\sum^{[\exp(t)]} \prod \right) \prod \longrightarrow \sum^{[\exp(t)]} \prod$$

The conversion incurs an exponential blow-up.

How expensive $\Delta = 2 \longrightarrow \Delta = 1$?

Consider $\sum\prod\sum\prod$ formula of size s.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$.

That is, $\sum\left(\prod^{[t]}\sum\right)\prod$.

Open up the multiplication of summands as a sum of multiplications.

$$\sum\left(\prod^{[t]}\sum\right)\prod \longrightarrow \sum\left(\sum^{[\exp(t)]}\prod\right)\prod \longrightarrow \sum^{[\exp(t)]}\prod$$

The conversion incurs an exponential blow-up.

[Kayal, Nair, Saha, 15] show that this exponential blow-up is essential while going from $\Delta = 2$ to $\Delta = 1$.

Larger product depth $\Delta + 1 \longrightarrow \Delta$

Consider $\sum\prod\sum\prod\ldots\sum\prod\sum$ formula of size s and product depth $\Delta + 1$.

Larger product depth $\Delta + 1 \longrightarrow \Delta$

Consider $\sum\prod\sum\prod\ldots\sum\prod\sum$ formula of size s and product depth $\Delta + 1$.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$, such that $t \leq s^{O(1/\Delta)}$.

Larger product depth $\Delta + 1 \longrightarrow \Delta$

Consider $\sum\prod\sum\prod\ldots\sum\prod\sum$ formula of size s and product depth $\Delta + 1$.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$, such that $t \leq s^{O(1/\Delta)}$.

That is, $\sum\prod\ldots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\ldots\sum\prod\sum$.

Larger product depth $\Delta + 1 \longrightarrow \Delta$

Consider $\sum\prod\sum\prod\ldots\sum\prod\sum$ formula of size s and product depth $\Delta + 1$.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$, such that $t \leq s^{O(1/\Delta)}$.

That is, $\sum\prod\ldots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\ldots\sum\prod\sum$.

Open up the multiplication of summands as a sum of multiplications.

Larger product depth $\Delta + 1 \longrightarrow \Delta$

Consider $\sum\prod\sum\prod\ldots\sum\prod\sum$ formula of size s and product depth $\Delta + 1$.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$, such that $t \leq s^{O(1/\Delta)}$.

That is, $\sum\prod\ldots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\ldots\sum\prod\sum$.

Open up the multiplication of summands as a sum of multiplications.

$$\sum\prod\ldots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\ldots\sum\prod\sum \longrightarrow$$
$$\sum\prod\ldots\sum\left(\sum^{[\exp((s^{O(1/\Delta)}))]}\prod\right)\prod\ldots\sum\prod\sum$$

Larger product depth $\Delta + 1 \longrightarrow \Delta$

Consider $\sum\prod\sum\prod\ldots\sum\prod\sum$ formula of size s and product depth $\Delta + 1$.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$, such that $t \leq s^{O(1/\Delta)}$.

That is, $\sum\prod\ldots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\ldots\sum\prod\sum$.

Open up the multiplication of summands as a sum of multiplications.

$$\sum\prod\cdots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\cdots\sum\prod\sum \longrightarrow$$
$$\sum\prod\cdots\sum\left(\sum^{[\exp((s^{O(1/\Delta)}))]}\prod\right)\prod\cdots\sum\prod\sum \longrightarrow$$
$$\sum\prod\cdots\left(\sum^{[\exp((s^{O(1/\Delta)}))]}\prod\right)\cdots\sum\prod\sum$$

Larger product depth $\Delta + 1 \longrightarrow \Delta$

Consider $\sum\prod\sum\prod\ldots\sum\prod\sum$ formula of size s and product depth $\Delta + 1$.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$, such that $t \leq s^{O(1/\Delta)}$.

That is, $\sum\prod\ldots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\ldots\sum\prod\sum$.

Open up the multiplication of summands as a sum of multiplications.

$$\sum\prod\cdots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\cdots\sum\prod\sum \longrightarrow$$
$$\sum\prod\cdots\sum\left(\sum^{[\exp((s^{O(1/\Delta)}))]}\prod\right)\prod\cdots\sum\prod\sum \longrightarrow$$
$$\sum\prod\cdots\left(\sum^{[\exp((s^{O(1/\Delta)}))]}\prod\right)\cdots\sum\prod\sum$$

Careful analysis shows a blow-up of $\exp(s^{1/\Delta+o(1)})$.

Larger product depth $\Delta + 1 \longrightarrow \Delta$

Consider $\sum\prod\sum\prod\ldots\sum\prod\sum$ formula of size s and product depth $\Delta + 1$.

Consider the $\prod\sum$ layer $\prod_{i\in[t]} Q_i$, such that $t \leq s^{O(1/\Delta)}$.

That is, $\sum\prod\ldots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\ldots\sum\prod\sum$.

Open up the multiplication of summands as a sum of multiplications.

$$\sum\prod\cdots\sum\left(\prod^{[(s^{O(1/\Delta)})]}\sum\right)\prod\cdots\sum\prod\sum \longrightarrow$$
$$\sum\prod\cdots\sum\left(\sum^{[\exp((s^{O(1/\Delta)}))]}\prod\right)\prod\cdots\sum\prod\sum \longrightarrow$$
$$\sum\prod\cdots\left(\sum^{[\exp((s^{O(1/\Delta)}))]}\prod\right)\cdots\sum\prod\sum$$

Careful analysis shows a blow-up of $\exp(s^{1/\Delta+o(1)})$.

Is the blow-up essential?

More Resources

Depth hierarchy theorem

More Resources ⇒

Depth hierarchy theorem

More Resources ⇒ More power?

Depth hierarchy theorem

More Resources ⟹ More power?

More Product-depth

Depth hierarchy theorem

More Resources ⟹ More power?

More Product-depth ⟹

Depth hierarchy theorem

More Resources ⟹ More power?

More Product-depth ⟹ More power?

Depth hierarchy theorems

Arithmetic circuit complexity world

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Boolean circuit complexity world

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Boolean circuit complexity world

[Ajtai,Frust et al.,Yao, Håstad, 1980s] proved quasipolynomial depth-hierarchy theorem.

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Boolean circuit complexity world

[Ajtai,Frust et al.,Yao, Håstad, 1980s] proved quasipolynomial depth-hierarchy theorem.

[Håstad, 1986] proved exponential depth-hierarchy theorem.

Depth hierarchy theorems

Arithmetic circuit complexity world

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Our result: Near-optimal Depth Hierarchy Theorem

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Our result: Near-optimal Depth Hierarchy Theorem

For any constant Δ, there is an explicit polynomial $P_{\Delta+1}(x_1, \ldots, x_n)$ such that

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Our result: Near-optimal Depth Hierarchy Theorem

For any constant Δ, there is an explicit polynomial $P_{\Delta+1}(x_1, \ldots, x_n)$ such that

> $P_{\Delta+1}(X)$ is computed by multilinear formula $F_{\Delta+1}$ of product-depth $\Delta + 1$ and size $O(n)$.

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Our result: Near-optimal Depth Hierarchy Theorem

For any constant Δ, there is an explicit polynomial $P_{\Delta+1}(x_1, \ldots, x_n)$ such that

- $P_{\Delta+1}(X)$ is computed by multilinear formula $F_{\Delta+1}$ of product-depth $\Delta + 1$ and size $O(n)$.
- However, any multilinear formula of product-depth $\leq \Delta$ for $P_{\Delta+1}(X)$ must have size $\exp(n^{\alpha_\Delta})$

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Our result: Near-optimal Depth Hierarchy Theorem

For any constant Δ, there is an explicit polynomial $P_{\Delta+1}(x_1, \ldots, x_n)$ such that

- $P_{\Delta+1}(X)$ is computed by multilinear formula $F_{\Delta+1}$ of product-depth $\Delta + 1$ and size $O(n)$.
- However, any multilinear formula of product-depth $\leq \Delta$ for $P_{\Delta+1}(X)$ must have size $\exp(n^{\alpha_\Delta})$, where $\alpha_\Delta = \Omega(1/\Delta)$.

Depth hierarchy theorems

Arithmetic circuit complexity world

[Raz and Yehudayoff, 2009] prove quasipolynomial depth-hierarchy theorem, i.e. they show quasipolynomial size lower bound for converting product-depth $\Delta + 1$ multilinear formula into product-depth Δ formula, as long as Δ is *small*.

[Kayal, Nair, Saha, 2015] show an exponential size lower bound for converting product-depth 2 multilinear formulas into product-depth 1 formulas.

Our result: Near-optimal Depth Hierarchy Theorem

For any constant Δ(can be slowly growing function of n), there is an explicit polynomial $P_{\Delta+1}(x_1, \ldots, x_n)$ such that

- $P_{\Delta+1}(X)$ is computed by multilinear formula $F_{\Delta+1}$ of product-depth $\Delta + 1$ and size $O(n)$.
- However, any multilinear formula of product-depth $\leq \Delta$ for $P_{\Delta+1}(X)$ must have size $\exp(n^{\alpha_\Delta})$, where $\alpha_\Delta = \Omega(1/\Delta)$.

Hard Polynomial

Construction of the hard polynomial

Hard Polynomial

Construction of the hard polynomial

Polynomial P_{Δ} is constructed inductively.

Hard Polynomial

Construction of the hard polynomial

Polynomial P_Δ is constructed inductively.

This polynomial is inspired by the construction of [Chen, Oliviera, Servedio, Tan, 2016] who prove near optimal Boolean circuit lower bounds for checking *graph connectivity* at small depth.

Hard Polynomial

Construction of the hard polynomial

Polynomial P_Δ is constructed inductively.

This polynomial is inspired by the construction of [Chen, Oliviera, Servedio, Tan, 2016] who prove near optimal Boolean circuit lower bounds for checking *graph connectivity* at small depth.

Hard Polynomial

$P^{(0)}$ is a 4 layered *Algebraic Branching Program* defined by $G^{(0)}$.

Hard Polynomial

$P^{(0)}$ is a 4 layered *Algebraic Branching Program* defined by $G^{(0)}$.

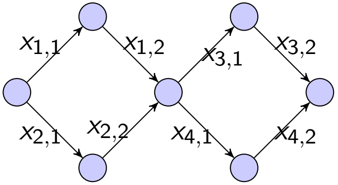


Figure: Definition of $G^{(0)}$

Hard Polynomial

$P^{(0)}$ is a 4 layered *Algebraic Branching Program* defined by $G^{(0)}$.

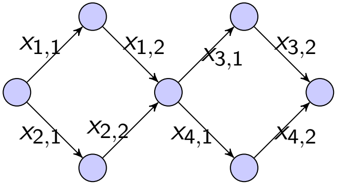


Figure: Definition of $G^{(0)}$

$$P^{(0)} = x_{1,1}x_{1,2}x_{3,1}x_{3,2} + x_{1,1}x_{1,2}x_{4,1}x_{4,2} + x_{2,1}x_{2,2}x_{3,1}x_{3,2} + x_{2,1}x_{2,2}x_{4,1}x_{4,2}.$$

Hard Polynomial

$P^{(0)}$ is a 4 layered *Algebraic Branching Program* defined by $G^{(0)}$.

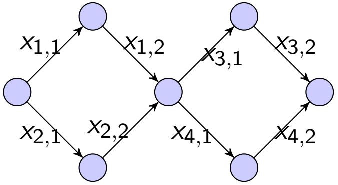


Figure: Definition of $G^{(0)}$

$$P^{(0)} = x_{1,1}x_{1,2}x_{3,1}x_{3,2} + x_{1,1}x_{1,2}x_{4,1}x_{4,2} + x_{2,1}x_{2,2}x_{3,1}x_{3,2} + x_{2,1}x_{2,2}x_{4,1}x_{4,2}.$$

This is a succint expression of the form $\Sigma\Pi$, i.e., product-depth 1

Hard Polynomial

$P^{(0)}$ is a 4 layered *Algebraic Branching Program* defined by $G^{(0)}$.

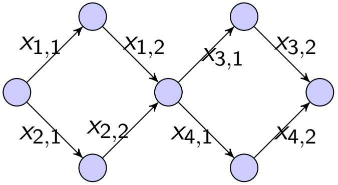


Figure: Definition of $G^{(0)}$

$$P^{(0)} = x_{1,1}x_{1,2}x_{3,1}x_{3,2} + x_{1,1}x_{1,2}x_{4,1}x_{4,2} + x_{2,1}x_{2,2}x_{3,1}x_{3,2} + x_{2,1}x_{2,2}x_{4,1}x_{4,2}.$$

This is a succint expression of the form $\Sigma\Pi$, i.e., product-depth 1 and of size $O(1)$.

Hard Polynomial

$H^{(1)}$ is obtained by composing two copies of $G^{(0)}$ in parallel.

Hard Polynomial

$H^{(1)}$ is obtained by composing two copies of $G^{(0)}$ in parallel.

$G^{(1)}$ is obtained by composing m series of $H^{(1)}$ in series.

Hard Polynomial

$H^{(1)}$ is obtained by composing two copies of $G^{(0)}$ in parallel.

$G^{(1)}$ is obtained by composing m series of $H^{(1)}$ in series.

$P^{(1)}$ is the sum of weights of all the source to sink paths in $G^{(1)}$.

Hard Polynomial

$H^{(1)}$ is obtained by composing two copies of $G^{(0)}$ in parallel.

$G^{(1)}$ is obtained by composing m series of $H^{(1)}$ in series.

$P^{(1)}$ is the sum of weights of all the source to sink paths in $G^{(1)}$.

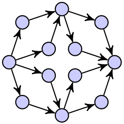


Figure: $H^{(1)}$.

Hard Polynomial

$H^{(1)}$ is obtained by composing two copies of $G^{(0)}$ in parallel.

$G^{(1)}$ is obtained by composing m series of $H^{(1)}$ in series.

$P^{(1)}$ is the sum of weights of all the source to sink paths in $G^{(1)}$.

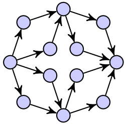
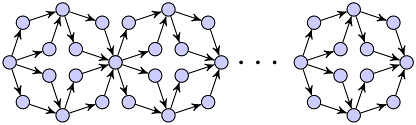


Figure: $H^{(1)}$.



m copies

Figure: $G^{(1)}$

Hard Polynomial

$H^{(1)}$ is obtained by composing two copies of $G^{(0)}$ in parallel.

$G^{(1)}$ is obtained by composing m series of $H^{(1)}$ in series.

$P^{(1)}$ is the sum of weights of all the source to sink paths in $G^{(1)}$.

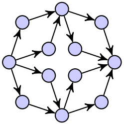
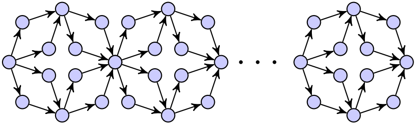


Figure: $H^{(1)}$.



m copies

Figure: $G^{(1)}$

$$P^{(1)} = \prod_{i=1}^{m} (P^{(0)}_{(i,1)} + P^{(0)}_{(i,2)}).$$

Hard Polynomial

$H^{(1)}$ is obtained by composing two copies of $G^{(0)}$ in parallel.

$G^{(1)}$ is obtained by composing m series of $H^{(1)}$ in series.

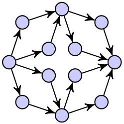
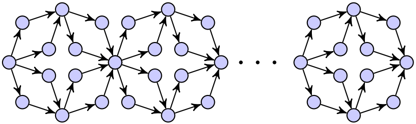


Figure: $H^{(1)}$.



m copies

Figure: $G^{(1)}$

$P^{(1)}$ is the sum of weights of all the source to sink paths in $G^{(1)}$.

$P^{(1)}$ is a polynomial over $n_1 := 8(2m)$ many variables and has product-depth 2, size $O(m) = O(n_1)$ formula.

$$P^{(1)} = \prod_{i=1}^{m}(P^{(0)}_{(i,1)} + P^{(0)}_{(i,2)}).$$

Hard Polynomial

$H^{(\Delta)}$ is obtained by composing two copies of $G^{(\Delta-1)}$ in parallel.

Hard Polynomial

$H^{(\Delta)}$ is obtained by composing two copies of $G^{(\Delta-1)}$ in parallel.

$G^{(\Delta)}$ is obtained by composing m series of $H^{(\Delta)}$ in series.

Hard Polynomial

$H^{(\Delta)}$ is obtained by composing two copies of $G^{(\Delta-1)}$ in parallel.

$G^{(\Delta)}$ is obtained by composing m series of $H^{(\Delta)}$ in series.

$P^{(\Delta)}$ is the sum of weights of all the source to sink paths in $G^{(\Delta)}$.

Hard Polynomial

$H^{(\Delta)}$ is obtained by composing two copies of $G^{(\Delta-1)}$ in parallel.

$G^{(\Delta)}$ is obtained by composing m series of $H^{(\Delta)}$ in series.

$P^{(\Delta)}$ is the sum of weights of all the source to sink paths in $G^{(\Delta)}$.

Hard Polynomial

$H^{(\Delta)}$ is obtained by composing two copies of $G^{(\Delta-1)}$ in parallel.

$G^{(\Delta)}$ is obtained by composing m series of $H^{(\Delta)}$ in series.

$P^{(\Delta)}$ is the sum of weights of all the source to sink paths in $G^{(\Delta)}$.

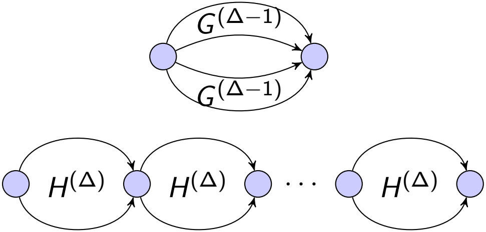


Figure: $H^{(\Delta)}$ (above) and $G^{(\Delta)}$(*below*).

Hard Polynomial

- $H^{(\Delta)}$ is obtained by composing two copies of $G^{(\Delta-1)}$ in parallel.
- $G^{(\Delta)}$ is obtained by composing m series of $H^{(\Delta)}$ in series.
- $P^{(\Delta)}$ is the sum of weights of all the source to sink paths in $G^{(\Delta)}$.
- $P^{(\Delta)}$ is a polynomial over $n_\Delta := 2m \cdot n_{\Delta-1} = 8(2m)^\Delta$ many variables and has a product-depth of $\Delta + 1$.

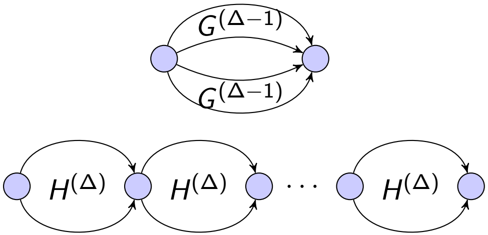


Figure: $H^{(\Delta)}$ (above) and $G^{(\Delta)}$(*below*).

$$P^{(\Delta)} = \prod_{i=1}^{m} (P^{(\Delta-1)}_{(i,1)} + P^{(\Delta-1)}_{(i,1)}).$$

Hard Polynomial

- $H^{(\Delta)}$ is obtained by composing two copies of $G^{(\Delta-1)}$ in parallel.
- $G^{(\Delta)}$ is obtained by composing m series of $H^{(\Delta)}$ in series.
- $P^{(\Delta)}$ is the sum of weights of all the source to sink paths in $G^{(\Delta)}$.
- $P^{(\Delta)}$ is a polynomial over $n_\Delta := 2m \cdot n_{\Delta-1} = 8(2m)^\Delta$ many variables and has a product-depth of $\Delta + 1$.

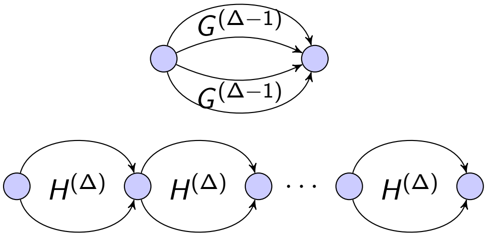


Figure: $H^{(\Delta)}$ (above) and $G^{(\Delta)}$(*below*).

$$P^{(\Delta)} = \prod_{i=1}^{m} (P_{(i,1)}^{(\Delta-1)} + P_{(i,1)}^{(\Delta-1)}).$$

Our result restated

Any product-depth Δ formula for P_Δ has size $\exp(\Omega(m^{\Omega(1)}))$

Hard Polynomial

- $H^{(\Delta)}$ is obtained by composing two copies of $G^{(\Delta-1)}$ in parallel.
- $G^{(\Delta)}$ is obtained by composing m series of $H^{(\Delta)}$ in series.
- $P^{(\Delta)}$ is the sum of weights of all the source to sink paths in $G^{(\Delta)}$.
- $P^{(\Delta)}$ is a polynomial over $n_\Delta := 2m \cdot n_{\Delta-1} = 8(2m)^\Delta$ many variables and has a product-depth of $\Delta + 1$.

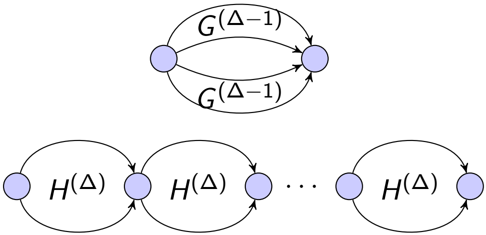


Figure: $H^{(\Delta)}$ (above) and $G^{(\Delta)}$(*below*).

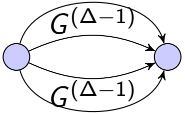
$$P^{(\Delta)} = \prod_{i=1}^{m} (P^{(\Delta-1)}_{(i,1)} + P^{(\Delta-1)}_{(i,1)}).$$

Our result restated

Any product-depth Δ formula for P_Δ has size $\exp(\Omega(m^{\Omega(1)}))$ $= \exp(n_{\Delta+1}^{\Omega(1/\Delta)})$.

Hard Polynomial

- $H^{(\Delta)}$ is obtained by composing two copies of $G^{(\Delta-1)}$ in parallel.
- $G^{(\Delta)}$ is obtained by composing m series of $H^{(\Delta)}$ in series.
- $P^{(\Delta)}$ is the sum of weights of all the source to sink paths in $G^{(\Delta)}$.
- $P^{(\Delta)}$ is a polynomial over $n_\Delta := 2m \cdot n_{\Delta-1} = 8(2m)^\Delta$ many variables and has a product-depth of $\Delta + 1$.



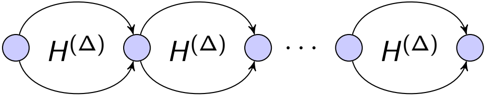


Figure: $H^{(\Delta)}$ (above) and $G^{(\Delta)}$(*below*).

$$P^{(\Delta)} = \prod_{i=1}^{m} (P^{(\Delta-1)}_{(i,1)} + P^{(\Delta-1)}_{(i,1)}).$$

Our result restated

Any product-depth Δ formula for P_Δ has size $\exp(\Omega(m^{\Omega(1)}))$ $= \exp(n_{\Delta+1}^{\Omega(1/\Delta)})$.

Conclusion

Conclusion

In the multilinear world there is a strict depth-hierarchy.

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$$

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

The lower bound we prove is near-optimal.

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

The lower bound we prove is near-optimal.

What about general constant depth formuals?

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

The lower bound we prove is near-optimal.

What about general constant depth formuals? Open!

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

The lower bound we prove is near-optimal.

What about general constant depth formuals? Open!

Do similar techniques yield a non-commutative formula depth-hierarchy theorem?

Proof details

Designing a measure

Designing a measure

The measure must satisfy

Designing a measure

The measure must satisfy

If $f(X)$ is computable by a product-depth $\Delta = 1$ multilinear formula of size s then $\mu(f) \leq s \times \mathcal{U}$.

Designing a measure

The measure must satisfy

If $f(X)$ is computable by a product-depth $\Delta = 1$ multilinear formula of size s then $\mu(f) \leq s \times \mathcal{U}$.

There is a polynomial $P(X)$ computable by product-depth $\Delta = 2$ multilinear formula such that $\mu(P) \geq \mathcal{L}$.

Designing a measure

The measure must satisfy

If $f(X)$ is computable by a product-depth $\Delta = 1$ multilinear formula of size s then $\mu(f) \leq s \times \mathcal{U}$.

There is a polynomial $P(X)$ computable by product-depth $\Delta = 2$ multilinear formula such that $\mu(P) \geq \mathcal{L}$.

Conclude that $s \geq \mathcal{L}/\mathcal{U}$.

Designing a measure

The measure must satisfy

If $f(X)$ is computable by a product-depth $\Delta = 1$ multilinear formula of size s then $\mu(f) \leq s \times \mathcal{U}$.

There is a polynomial $P(X)$ computable by product-depth $\Delta = 2$ multilinear formula such that $\mu(P) \geq \mathcal{L}$.

Conclude that $s \geq \mathcal{L}/\mathcal{U}$.

Partial Derivative Matrix & Complexity Measure

Rank measure defined by [Raz 2004]

Let $\rho : X \mapsto Y \sqcup Z$ be a partitioning function such that $|Y| = |Z|$.

Partial Derivative Matrix & Complexity Measure

Rank measure defined by [Raz 2004]

Let $\rho : X \mapsto Y \sqcup Z$ be a partitioning function such that $|Y| = |Z|$.

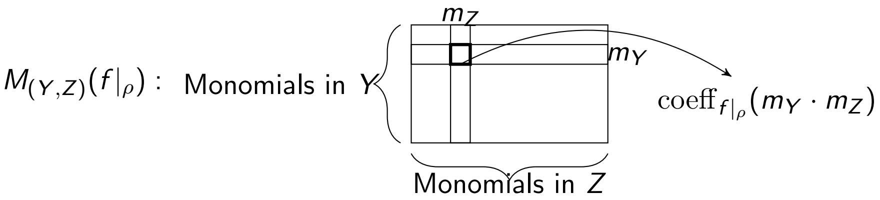
$$f = \sum_{i=1}^{2^n} c_i \cdot m_i \quad \mapsto \quad f|_\rho = \sum_{i=1}^{2^n} c_i \cdot m_{i,Y} \cdot m_{i,Z}$$

Partial Derivative Matrix & Complexity Measure

Rank measure defined by [Raz 2004]

Let $\rho : X \mapsto Y \sqcup Z$ be a partitioning function such that $|Y| = |Z|$.

$$f = \sum_{i=1}^{2^n} c_i \cdot m_i \quad \mapsto \quad f|_\rho = \sum_{i=1}^{2^n} c_i \cdot m_{i,Y} \cdot m_{i,Z}$$



Complexity measure: $\mu(f)$ w.r.t. ρ is $\mathrm{rank}(M_{(Y,Z)}(f|_\rho))$.

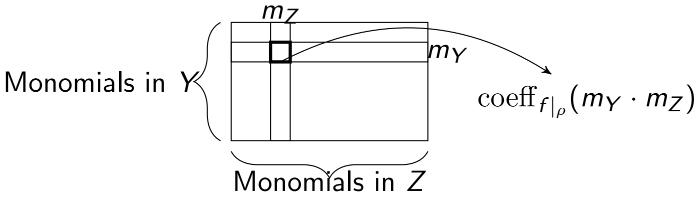
Partial Derivative Matrix & Complexity Measure

Rank measure defined by [Raz 2004]

Let $\rho : X \mapsto Y \sqcup Z$ be a partitioning function such that $|Y| = |Z|$.

$$f = \sum_{i=1}^{2^n} c_i \cdot m_i \quad \mapsto \quad f|_\rho = \sum_{i=1}^{2^n} c_i \cdot m_{i,Y} \cdot m_{i,Z}$$

$M_{(Y,Z)}(f|_\rho)$:



Complexity measure: $\mu(f)$ w.r.t. ρ is $\mathrm{rank}(M_{(Y,Z)}(f|_\rho))$.

Understanding the measure

Example

Understanding the measure

Example

Let $f(x_1, \ldots x_{2n}) = \prod_{i=1}^{n}(x_i + x_{n+i})$.

Understanding the measure

Example

Let $f(x_1, \ldots x_{2n}) = \prod_{i=1}^{n}(x_i + x_{n+i})$.

Let $\rho(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n \\ & \end{cases}$

Understanding the measure

Example

Let $f(x_1, \ldots x_{2n}) = \prod_{i=1}^{n}(x_i + x_{n+i})$.

Let $\rho(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n \\ z_{i-n} & \text{if } n+1 \leq i \leq 2n \end{cases}$

Understanding the measure

Example

Let $f(x_1, \ldots x_{2n}) = \prod_{i=1}^{n}(x_i + x_{n+i})$.

Let $\rho(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n \\ z_{i-n} & \text{if } n+1 \leq i \leq 2n \end{cases}$

Therefore, $f|_\rho(Y, Z) = \sum_{S \subseteq [n]} Y_S Z_{[n] \setminus S}$

Understanding the measure

Example

Let $f(x_1, \ldots x_{2n}) = \prod_{i=1}^{n}(x_i + x_{n+i})$.

Let $\rho(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n \\ z_{i-n} & \text{if } n+1 \leq i \leq 2n \end{cases}$

Therefore, $f|_{\rho}(Y, Z) = \sum_{S \subseteq [n]} Y_S Z_{[n] \setminus S}$

That is, $M_{(Y,Z)}(f|_{\rho})$ is a disjointness matrix.

Understanding the measure

Example

Let $f(x_1, \ldots x_{2n}) = \prod_{i=1}^{n}(x_i + x_{n+i})$. $(\Delta = 1)$

Let $\rho(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n \\ z_{i-n} & \text{if } n+1 \leq i \leq 2n \end{cases}$

Therefore, $f|_\rho(Y, Z) = \sum_{S \subseteq [n]} Y_S Z_{[n] \setminus S}$

That is, $M_{(Y,Z)}(f|_\rho)$ is a disjointness matrix.

Therefore, $\mu(f)$ w.r.t. the above ρ is 2^n.

Understanding the measure

Example

Let $f(x_1, \ldots x_{2n}) = \prod_{i=1}^{n}(x_i + x_{n+i})$.

Let $\rho(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n \\ z_{i-n} & \text{if } n+1 \leq i \leq 2n \end{cases}$

Therefore, $f|_\rho(Y, Z) = \sum_{S \subseteq [n]} Y_S Z_{[n] \setminus S}$

That is, $M_{(Y,Z)}(f|_\rho)$ is a disjointness matrix.

Therefore, $\mu(f)$ w.r.t. the above ρ is 2^n.

Let $\rho'(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n/2 \text{ or } n+1 \leq i \leq 3n/2 \\ & \end{cases}$

Understanding the measure

Example

Let $f(x_1, \ldots x_{2n}) = \prod_{i=1}^{n}(x_i + x_{n+i})$.

Let $\rho(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n \\ z_{i-n} & \text{if } n+1 \leq i \leq 2n \end{cases}$

Therefore, $f|_\rho(Y, Z) = \sum_{S \subseteq [n]} Y_S Z_{[n] \setminus S}$

That is, $M_{(Y,Z)}(f|_\rho)$ is a disjointness matrix.

Therefore, $\mu(f)$ w.r.t. the above ρ is 2^n.

Let $\rho'(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n/2 \text{ or } n+1 \leq i \leq 3n/2 \\ z_{i-n} & \text{if } n/2+1 \leq i \leq n \text{ or } 3n/2+2 \leq i \leq 2n \end{cases}$

Understanding the measure

Example

Let $f(x_1, \ldots x_{2n}) = \prod_{i=1}^{n}(x_i + x_{n+i})$.

Let $\rho(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n \\ z_{i-n} & \text{if } n+1 \leq i \leq 2n \end{cases}$

Therefore, $f|_\rho(Y, Z) = \sum_{S \subseteq [n]} Y_S Z_{[n] \setminus S}$

That is, $M_{(Y,Z)}(f|_\rho)$ is a disjointness matrix.

Therefore, $\mu(f)$ w.r.t. the above ρ is 2^n.

Let $\rho'(x_i) = \begin{cases} y_i & \text{if } 1 \leq i \leq n/2 \text{ or } n+1 \leq i \leq 3n/2 \\ z_{i-n} & \text{if } n/2+1 \leq i \leq n \text{ or } 3n/2+2 \leq i \leq 2n \end{cases}$

$\mu(f)$ w.r.t. $\rho' \ll 2^n$.

Designing μ and ρ

Designing μ and ρ

The measure must satisfy

Designing μ and ρ

The measure must satisfy

If $f(X)$ is computable by a product-depth $\Delta = 1$ multilinear formula of size s then $\mu(f) \leq s \times \mathcal{U}$ w.r.t ρ.

Designing μ and ρ

The measure must satisfy

If $f(X)$ is computable by a product-depth $\Delta = 1$ multilinear formula of size s then $\mu(f) \leq s \times \mathcal{U}$ w.r.t ρ.

There is a polynomial $P(X)$ computable by product-depth $\Delta = 2$ multilinear formula such that $\mu(P) \geq \mathcal{L}$ w.r.t the same ρ.

Designing μ and ρ

The measure must satisfy

If $f(X)$ is computable by a product-depth $\Delta = 1$ multilinear formula of size s then $\mu(f) \leq s \times \mathcal{U}$ w.r.t ρ.

There is a polynomial $P(X)$ computable by product-depth $\Delta = 2$ multilinear formula such that $\mu(P) \geq \mathcal{L}$ w.r.t the same ρ.

Conclude that $s \geq \mathcal{L}/\mathcal{U}$.

Designing μ and ρ

The measure must satisfy

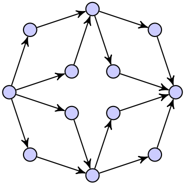
If $f(X)$ is computable by a product-depth $\Delta = 1$ multilinear formula of size s then $\mu(f) \leq s \times \mathcal{U}$ w.r.t ρ.

There is a polynomial $P(X)$ computable by product-depth $\Delta = 2$ multilinear formula such that $\mu(P) \geq \mathcal{L}$ w.r.t the same ρ.

Conclude that $s \geq \mathcal{L}/\mathcal{U}$.

A random $\rho : X \to Y \cup Z \cup \mathbb{F}$

Map every copy of $H^{(1)}$ uniformly at random to one of the three possibilities.



A random $\rho : X \to Y \cup Z \cup \mathbb{F}$

Map every copy of $H^{(1)}$ uniformly at random to one of the three possibilities.

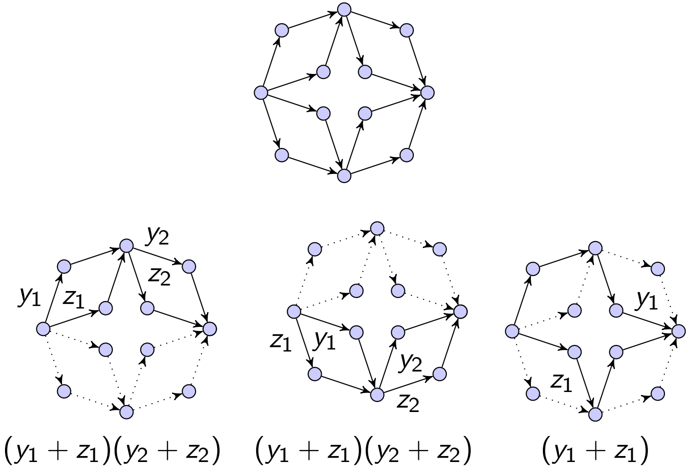
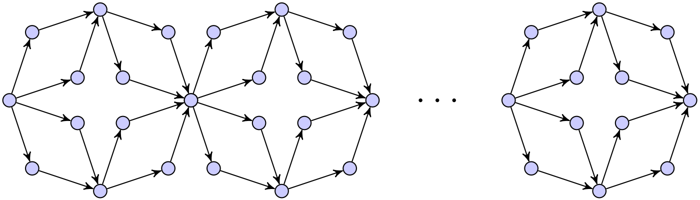


Figure: Map ρ applied to each copy of $H^{(1)}$. Edges that are not labelled have their variables set to 1. Dotted edges have their variables set to 0.

Random map ρ

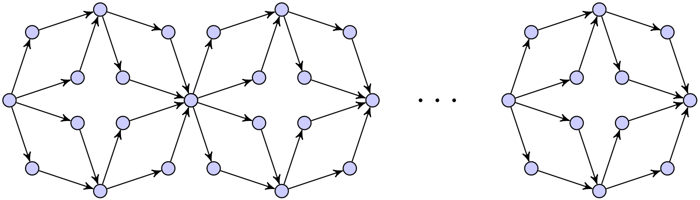
Recall that $G^{(1)}$ is m copies of $H^{(1)}$.



Under the above choice of random ρ, the resulting polynomial will be $P^{(1)}|_\rho = \prod_{i \in [t]}(y_i + z_i)$

Random map ρ

Recall that $G^{(1)}$ is m copies of $H^{(1)}$.

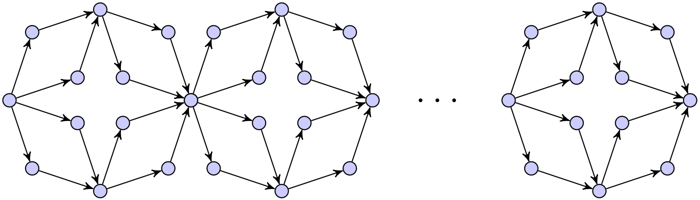


Under the above choice of random ρ, the resulting polynomial will be $P^{(1)}|_\rho = \prod_{i \in [t]}(y_i + z_i)$, where $t = \Omega(m)$ in expectation.

Therefore, $\mu(P^{(1)}) = 2^{\Omega(m)}$

Random map ρ

Recall that $G^{(1)}$ is m copies of $H^{(1)}$.



Under the above choice of random ρ, the resulting polynomial will be $P^{(1)}|_\rho = \prod_{i\in[t]}(y_i + z_i)$, where $t = \Omega(m)$ in expectation.

Therefore, $\mu(P^{(1)}) = 2^{\Omega(m)}$ w.h.p. over the distribution defined by these random partitions.

Effect of ρ on $\sum\prod\sum$

Effect of ρ on $\sum\prod\sum$

$$P(x) = \sum_{i=1}^{s} \prod_{j=1}^{s'} L_{i,j}$$

Effect of ρ on $\sum\prod\sum$

$$P(x) = \sum_{i=1}^{s} \prod_{j=1}^{s'} L_{i,j}$$

Easy to see that μ of linear polynomials is small with constant probability.

Effect of ρ on $\sum\prod\sum$

$$P(x) = \sum_{i=1}^{s} \prod_{j=1}^{s'} L_{i,j}$$

Easy to see that μ of linear polynomials is small with constant probability.

Also, μ of each product term is low, say $\mathcal{U}$, w.h.p.

Effect of ρ on $\sum\prod\sum$

$$P(x) = \sum_{i=1}^{s} \prod_{j=1}^{s'} L_{i,j}$$

Easy to see that μ of linear polynomials is small with constant probability.

Also, μ of each product term is low, say $\mathcal{U}$, w.h.p.

By subadditivity of ranks, $\mu(P)$ is at most $s \cdot \mathcal{U}$

Effect of ρ on $\sum\prod\sum$

$$P(x) = \sum_{i=1}^{s} \prod_{j=1}^{s'} L_{i,j}$$

Easy to see that μ of linear polynomials is small with constant probability.

Also, μ of each product term is low, say $\mathcal{U}$, w.h.p.

By subadditivity of ranks, $\mu(P)$ is at most $s \cdot \mathcal{U}$

Hence, $s \geq 2^{\Omega(m)}/\mathcal{U}$.

Effect of ρ on $\sum\prod\sum$

$$P(x) = \sum_{i=1}^{s}\prod_{j=1}^{s'} L_{i,j}$$

Easy to see that μ of linear polynomials is small with constant probability.

Also, μ of each product term is low, say $\mathcal{U}$, w.h.p.

By subadditivity of ranks, $\mu(P)$ is at most $s \cdot \mathcal{U}$

Hence, $s \geq 2^{\Omega(m)}/\mathcal{U}$.

At larger depths ...

Effect of ρ on $\sum\prod\sum$

$$P(x) = \sum_{i=1}^{s}\prod_{j=1}^{s'} L_{i,j}$$

Easy to see that μ of linear polynomials is small with constant probability.

Also, μ of each product term is low, say $\mathcal{U}$, w.h.p.

By subadditivity of ranks, $\mu(P)$ is at most $s \cdot \mathcal{U}$

Hence, $s \geq 2^{\Omega(m)}/\mathcal{U}$.

At larger depths ...

A carefully chosen ρ at each *level* of the polynomial.

Conclusion

Conclusion

In the multilinear world there is a strict depth-hierarchy.

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$$

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

The lower bound we prove is near-optimal.

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

The lower bound we prove is near-optimal.

What about general constant depth formuals?

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

The lower bound we prove is near-optimal.

What about general constant depth formuals? Open!

Conclusion

In the multilinear world there is a strict depth-hierarchy.

$(\sum\prod)^{\Delta}\sum \subsetneq (\sum\prod)^{\Delta+1}\sum$, while $\Delta = O(1)$.

The classes are exponentially separated.

The lower bound we prove is near-optimal.

What about general constant depth formuals? Open!

Do similar techniques yield a non-commutative formula depth-hierarchy theorem?

Thank You!