Arithmetic circuits: lower bounds by partial derivatives and structural results

Hervé Fournier (Université Paris Cité)

July 10, 2023 Algebraic Complexity Workshop – ICALP

Polynomials

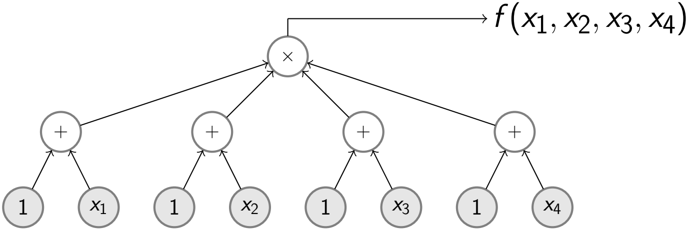
$$
\begin{aligned}
f(x_1, x_2, x_3, x_4) \quad = \quad & 1 + x_1 + x_2 + x_3 + x_4 \\
& + x_1 x_2 + x_1 x_3 + x_1 x_4 + x_2 x_3 + x_2 x_4 + x_3 x_4 \\
& + x_2 x_3 x_4 + x_1 x_3 x_4 + x_1 x_2 x_4 + x_1 x_2 x_3 \\
& + x_1 x_2 x_3 x_4
\end{aligned}
$$

Polynomials

$$\begin{aligned} f(x_1, x_2, x_3, x_4) \quad &= \quad 1 + x_1 + x_2 + x_3 + x_4 \\ &\quad\;\; + x_1x_2 + x_1x_3 + x_1x_4 + x_2x_3 + x_2x_4 + x_3x_4 \\ &\quad\;\; + x_2x_3x_4 + x_1x_3x_4 + x_1x_2x_4 + x_1x_2x_3 \\ &\quad\;\; + x_1x_2x_3x_4 \\ \\ &= \quad (1 + x_1)(1 + x_2)(1 + x_3)(1 + x_4) \end{aligned}$$

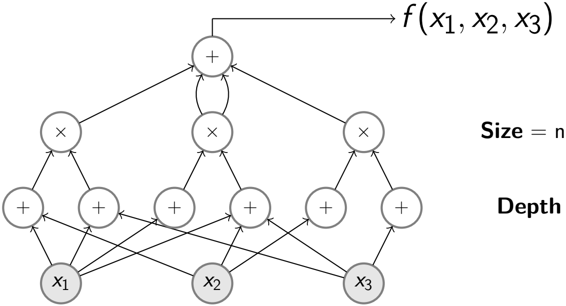
more compact representation!

Arithmetic Formulas



- Tree
- Leaves containing variables or constants

Arithmetic Circuits



Size = number of gates

Depth = longest path

Algebraic classes VP vs VNP

VP $\ni (P_n)$ if

- P_n computed by circuits of size $n^{O(1)}$
- P_n has degree $n^{O(1)}$

VNP : exponential sum in front of VP
$(Q_n) \in$ VNP if there exists $P_n(\bar{x}, \bar{y}) \in$ VP s.t.

$$Q_n(\bar{x}) = \sum_{\bar{y} \in \{0,1\}^{|\bar{y}|}} P_n(\bar{x}, \bar{y})$$

[Valiant-79] VP = VNP?

What does VNP contain?

- polynomials with coefficients computable in polynomial time

What does VNP contain?

- polynomials with coefficients computable in polynomial time
- in particular: generating functions of graph properties
  - Permanent: perfect matchings of $K_{n,n}$ (cycle covers of K_n)

$$\sum_{\{(i_1,j_1),\ldots,(i_n,j_n)\} \text{ perfect matching}} X_{i_1,j_1} \ldots X_{i_n,j_n}$$

What does VNP contain?

- polynomials with coefficients computable in polynomial time
- in particular: generating functions of graph properties
  - Permanent: perfect matchings of $K_{n,n}$ (cycle covers of K_n)

$$\sum_{\{(i_1,j_1),\dots,(i_n,j_n)\} \text{ perfect matching}} X_{i_1,j_1} \dots X_{i_n,j_n}$$

  - Hamiltonian cycles of K_n

$$\sum_{C \text{ Ham. cycle}} \prod_{e \in C} X_e$$

Determinant vs. Permanent

$$\mathrm{Det}_n(x_{11}, \ldots, x_{nn}) = \sum_{\sigma \in S_n} \mathrm{sign}(\sigma) \cdot x_{1\sigma(1)} \ldots x_{n\sigma(n)}$$

$$\mathrm{Perm}_n(x_{11}, \ldots, x_{nn}) = \sum_{\sigma \in S_n} x_{1\sigma(1)} \ldots x_{n\sigma(n)}$$

Determinant vs. Permanent

$$\begin{aligned} \mathrm{Det}_n(x_{11},\ldots,x_{nn}) &= \sum_{\sigma\in S_n} \mathrm{sign}(\sigma)\cdot x_{1\sigma(1)}\cdots x_{n\sigma(n)} \\ \mathrm{Perm}_n(x_{11},\ldots,x_{nn}) &= \sum_{\sigma\in S_n} x_{1\sigma(1)}\cdots x_{n\sigma(n)} \end{aligned}$$

$(\mathrm{Det}_n) \in \mathrm{VP}$, $(\mathrm{Perm}_n) \in \mathrm{VNP}$

Variant of VP vs. VNP:
Is the Permanent a projection of a "not too large" Determinant?

Algebraic vs. Boolean lower bounds

[Bürgisser-99] (Under GRH) If $\mathrm{VP} = \mathrm{VNP}$ over $\mathbb{C}$, then $\#\mathrm{P} \subseteq \mathrm{FNC}$ (non-uniform)

Main goal of algebraic complexity

Lower bounds for the size of circuits computing polynomials

Main goal of algebraic complexity

Lower bounds for the size of circuits computing polynomials

[Baur-Strassen-83] Computing

$$x_1^n + x_2^n + \ldots + x_n^n$$

requires $\Omega(n \log n)$ arithmetic operations.

Main goal of algebraic complexity

Lower bounds for the size of circuits computing polynomials

[Baur-Strassen-83] Computing

$$x_1^n + x_2^n + \ldots + x_n^n$$

requires $\Omega(n \log n)$ arithmetic operations.

Strong lower bounds for *restricted* models:

- branching program, formulas, bounded-depth formulas
- non-commutative, monotone, multilinear models

Restricted circuits

⋆ A circuit C is homogeneous if every gate computes a homogeneous polynomial.

⋆ A circuit C is multilinear if every gate computes a multilinear polynomial.

Restricted circuits

⋆ A circuit C is homogeneous if every gate computes a homogeneous polynomial.

⋆ A circuit C is multilinear if every gate computes a multilinear polynomial.

A circuit computing a homogeneous or multilinear polynomial may not be homogeneous or multilinear

Many lower bounds hold for such restricted models of computation

Complexity of the elementary symmetric polynomials S_n^d

Elementary symmetric polynomials of degree d on $X_1, \ldots, X_n$:

$$S_n^d = \sum_{T \in \binom{[n]}{d}} X_T \qquad \text{where } X_T := \prod_{i \in T} X_i$$

Complexity of the elementary symmetric polynomials S_n^d

Elementary symmetric polynomials of degree d on $X_1, \ldots, X_n$:

$$S_n^d = \sum_{T \in \binom{[n]}{d}} X_T \qquad \text{where } X_T := \prod_{i \in T} X_i$$

Upper bound by **interpolation**

S_n^d is the coefficent of T^{n-d} in

$$(T + X_1)(T + X_2) \ldots (T + X_n)$$

so it is equal to a linear combination of this polynomial (in T) evaluated at n distinct points

Complexity of the elementary symmetric polynomials S_n^d

Elementary symmetric polynomials of degree d on $X_1, \ldots, X_n$:

$$S_n^d = \sum_{T \in \binom{[n]}{d}} X_T \qquad \text{where } X_T := \prod_{i \in T} X_i$$

Upper bound by **interpolation**

S_n^d is the coefficent of T^{n-d} in

$$(T + X_1)(T + X_2) \ldots (T + X_n)$$

so it is equal to a linear combination of this polynomial (in T) evaluated at n distinct points

$\Rightarrow \Sigma\Pi\Sigma$ formula of size $O(n^2)$

Remark: the formula obtained is not multilinear and not homogeneous

Lower bounds for *restricted* models

Model	**Lower bound**	
General circuits	$\Omega(n \log n)$	[Baur-Strassen-83]
Monotone Formula	$2^{\Omega(n)}$	[Nisan-91]
Homogeneous Depth-3 circuits	$2^{\Omega(n)}$	[Nisan-Wigderson-97]
Multilinear formula	$2^{\Omega(n \log n)}$	[Raz-09]
Constant-depth circuits (poly of small degree d)	$n^{d^{\Omega(1)}}$	[Limaye-Srinivasan-Tavenas-21]

Outline

- Some lower bounds based on partial derivatives
  - ⋆ Partial derivatives of order 1
  - ⋆ Dimension of partial derivatives of all order
  - ⋆ Partial derivatives w.r.t. a subset of variables
- Structural results
  - ⋆ homogenization
  - ⋆ depth-reduction

Computing all partial derivatives of degree 1

Lemma (Baur and Strassen)

If $P(x_1, \ldots, x_n)$ is computed by a circuit of size s, there is a circuit of size $O(s)$ computing

$$\left\{ \frac{\partial P}{\partial x_1}, \ldots, \frac{\partial P}{\partial x_n} \right\}.$$

Proof. By induction on the size on the size of the circuit, using chain rule for partial derivatives.

Computing all partial derivatives of degree 1

Lemma (Baur and Strassen)

If $P(x_1, \ldots, x_n)$ is computed by a circuit of size s, there is a circuit of size $O(s)$ computing

$$\left\{\frac{\partial P}{\partial x_1}, \ldots, \frac{\partial P}{\partial x_n}\right\}.$$

Proof. By induction on the size on the size of the circuit, using chain rule for partial derivatives.

(Proof of lower bound.) Let

$$P = x_1^d + x_2^d + \ldots + x_n^d$$

computed by a circuit of size s. There is a circuit of size $O(s)$ computing simultaneously $x_1^d, \ldots, x_n^d$.
Using Bezout this requires $n \log d$ products: $s = \Omega(n \log d)$.
(tight by doing fast exponentiation)

Computing all partial derivatives of degree 1

Multinear setting

A circuit is *syntactically multilinear* if for any product gate $P \times Q$, the polynomials P and Q are over disjoint sets of variables

Lemma

If a polynomial is computed by a syntactically multilinear circuit of size s, all its first order partial derivatives are computed by a syntactically multilinear circuit of size $O(s)$.

Applications in the mutilinear setting:

- $\star$ $\mathrm{NC}_1 \neq \mathrm{NC}_2$ (formulas $\subsetneq$ circuits)
- $\star$ $\Omega(n^2/\log^2 n)$ lower bound

Computing all partial derivatives of degree 1

Non-commutative setting [P.Chaterjee-Hrubes-23]

Partial derivative with respect to the first position:
$\partial_x(xu) = u$ (where u non-commutative monomial)
$\partial_x(yu) = 0$ (y variable, $y \neq x$)

Lemma

If $P \in \mathbb{C}\langle x_1, \ldots, x_n \rangle$ is computed by a homogeneous non-commutative circuit of size s, all $\partial_{x_i} P$ ($i \in [n]$) can be simultaneously computed by a homogeneous circuit of size $O(s)$.

Application:
$\Omega(nd)$ lower bound for the size of homogeneous non-commutative circuits (for some polynomial of degree d over n variables)

Computing all partial derivatives of degree 1

Non-commutative setting [P.Chaterjee-Hrubes-23]

Partial derivative with respect to the first position:
$\partial_x(xu) = u$ (where u non-commutative monomial)
$\partial_x(yu) = 0$ (y variable, $y \neq x$)

Lemma
If $P \in \mathbb{C}\langle x_1, \ldots, x_n \rangle$ is computed by a homogeneous non-commutative circuit of size s, all $\partial_{x_i} P$ ($i \in [n]$) can be simultaneously computed by a homogeneous circuit of size $O(s)$.

Application:
$\Omega(nd)$ lower bound for the size of homogeneous non-commutative circuits (for some polynomial of degree d over n variables)

Question: can homogeneity assumption be removed?

Complexity measure Γ

Construct a map $\Gamma : \mathbb{F}[x_1, \dots, x_n] \to \mathbb{N}$, that assigns a number to every polynomial such that:

1. If f is computable by "small" circuits, then $\Gamma(f)$ is "small".
2. For the polynomial f for which we wish to show a lower bound, $\Gamma(f)$ is "large".

Measure based on Partial Derivative

[Nisan-Wigderson-97]

$$\partial(f) \stackrel{\text{def}}{=} \text{Set of partial derivatives (of all orders) of } f$$

$$\Gamma(f) \stackrel{\text{def}}{=} \dim\{\partial(f)\}$$

Measure based on Partial Derivative

[Nisan-Wigderson-97]

$$\partial(f) \stackrel{\text{def}}{=} \text{Set of partial derivatives (of all orders) of } f$$
$$\Gamma(f) \stackrel{\text{def}}{=} \dim\{\partial(f)\}$$

Properties:

- $\Gamma(f+g) \leqslant \Gamma(f) + \Gamma(g)$ (sub-additivity)
- $\Gamma(fg) \leqslant \Gamma(f)\Gamma(g)$

Measure based on Partial Derivative

[Nisan-Wigderson-97]

$$\partial(f) \stackrel{\text{def}}{=} \text{Set of partial derivatives (of all orders) of } f$$

$$\Gamma(f) \stackrel{\text{def}}{=} \dim\{\partial(f)\}$$

Properties:

- $\Gamma(f+g) \leqslant \Gamma(f) + \Gamma(g)$ (sub-additivity)
- $\Gamma(fg) \leqslant \Gamma(f)\Gamma(g)$

Application: Lower bounds on the elementary symmetric polynomials

Lower bounds on elementary symmetric polynomials (1/3)

Elementary symmetric polynomials of degree d on $X_1, \ldots, X_n$:

$$S_n^d = \sum_{T \in \binom{[n]}{d}} X_T \qquad \text{where } X_T := \prod_{i \in T} X_i$$

Lower bounds on elementary symmetric polynomials (1/3)

Elementary symmetric polynomials of degree d on $X_1, \ldots, X_n$:

$$S_n^d = \sum_{T \in \binom{[n]}{d}} X_T \qquad \text{where } X_T := \prod_{i \in T} X_i$$

Step 1: $\Gamma(f)$ is small for f computed by $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuits

g of the form $\Pi^{[d]}\Sigma$: $g = \ell_1 \ell_2 \ldots \ell_d$ with ℓ_i affine

$$\partial(g) \subseteq \text{span}\{\prod_{i \in I} \ell_i \mid I \subset [d]\}$$

Hence $\Gamma(g) \leqslant 2^d$

Lower bounds on elementary symmetric polynomials (1/3)

Elementary symmetric polynomials of degree d on $X_1, \ldots, X_n$:

$$S_n^d = \sum_{T \in \binom{[n]}{d}} X_T \qquad \text{where } X_T := \prod_{i \in T} X_i$$

Step 1: $\Gamma(f)$ is small for f computed by $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuits

g of the form $\Pi^{[d]}\Sigma$: $g = \ell_1 \ell_2 \ldots \ell_d$ with ℓ_i affine

$$\partial(g) \subseteq \text{span}\{\prod_{i \in I} \ell_i \mid I \subset [d]\}$$

Hence $\Gamma(g) \leqslant 2^d$

f sum of s polynomials computed by $\Pi^{[d]}\Sigma$ circuits
$\Gamma(f) \leqslant s \cdot 2^d$ by sub-additivity

Lower bounds on elementary symmetric polynomials (2/3)

Step 2: $\Gamma(S_n^d)$ is large

Lower bounds on elementary symmetric polynomials (2/3)

Step 2: $\Gamma(S_n^d)$ is large

Consider the matrix M:
– Rows indexed by subsets $A \in \binom{[n]}{d/2}$
– Columns indexed by subsets $B \in \binom{[n]}{d/2}$
– Column B is the polynomial $\frac{\partial S_n^d}{\partial X_B}$

Element in row A and column B is the coefficient of X_A in $\frac{\partial S_n^d}{\partial X_B}$

Lower bounds on elementary symmetric polynomials (2/3)

Step 2: $\Gamma(S_n^d)$ is large

Consider the matrix M:
– Rows indexed by subsets $A \in \binom{[n]}{d/2}$
– Columns indexed by subsets $B \in \binom{[n]}{d/2}$
– Column B is the polynomial $\frac{\partial S_n^d}{\partial X_B}$
Element in row A and column B is the coefficient of X_A in $\frac{\partial S_n^d}{\partial X_B}$

$M_{A,B} = 1$ if $A \cap B = \emptyset$ and 0 otherwise
M is a disjointness matrix, known to be full-rank

Hence, $\Gamma(S_n^d) \geqslant \binom{n}{d/2}$

Lower bounds on elementary symmetric polynomials (3/3)

Step 1: $\Gamma(f) \leqslant s \cdot 2^d$ is small for f computed by $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuit

Step 2: $\binom{n}{d/2} \leqslant \Gamma(S_n^d)$

Conclusion: if a $\Sigma^{[s]}\Pi^{[d]}\Sigma$ circuit computes S_n^d:

$$\binom{n}{d/2} \leqslant \Gamma(S_n^d) \leqslant s2^d$$

Hence $s = \Omega((\frac{n}{4d})^d)$

Rank of the coefficient matrix

[Raz-09] Multilinear polynomial f over variables X

Partition of the variables $X = Y \dot{\cup} Z$

Matrix M of coefficients:

$$\text{Z monomials}\left\{ \overbrace{\begin{pmatrix} \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & & & & & & & \\ \cdot & & & & \text{(coef)} & & & \cdot \\ \cdot & & & & & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \end{pmatrix}}^{\text{Y monomials}} \right.$$

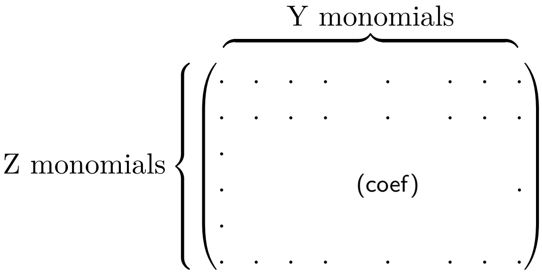
$\Gamma_{Y,Z}(f) =$ rank of M

Rank of the coefficient matrix

[Raz-09] Multilinear polynomial f over variables X

Partition of the variables $X = Y \dot\cup Z$

Matrix M of coefficients:



$\Gamma_{Y,Z}(f) =$ rank of M

Remark. $\Gamma_{Y,Z}(f)$ is the rank of partial derivatives of all orders w.r.t. Y variables

Rank of the coefficient matrix: properties

⋆ Subadditivity:

$$\Gamma(f+g) \leqslant \Gamma(f) + \Gamma(g)$$

(because $M_{f+g} = M_f + M_g$)

⋆ If f and g are polynomials over disjoint variables:

$$\Gamma(fg) = \Gamma(f)\Gamma(g)$$

(because $M_{fg} = M_f \otimes M_g$)

Proof sketch of separation in the multilinear setting

$\star$ The formula

$$(y_1 + z_1)(y_2 + z_2) \dots (y_n + z_n)$$

has rank 2^n with respect to the partition $Y \cup Z$
(maximum possible rank for $2n$ variables)

Proof sketch of separation in the multilinear setting

⋆ The formula

$$(y_1 + z_1)(y_2 + z_2)\dots(y_n + z_n)$$

has rank 2^n with respect to the partition $Y \cup Z$
(maximum possible rank for $2n$ variables)

⋆ But formulas have the following weakness [Raz-05]

no small formula can be full rank for any balanced partition

Consider $f_1 f_2$ over $2n$ variables (f_1, f_2 over disjoint sets of var.)
f_i is over variables X_i, $n_i := |X_i|$, $n_1 + n_2 = 2n$
Consider a balanced partition of the variables $X = Y \cup Z$
$\rightarrow X_i = Y_i \cup Z_i$. Let $\delta := \frac{1}{2}||Y_i| - |Z_i||$
Then

$$\Gamma_{f_1 f_2} \leqslant 2^{(n_1-\delta)/2} 2^{(n_2-\delta)/2} = \frac{1}{2^\delta} \cdot 2^n$$

Proof sketch of separation of multilinear formulas and circuits

⋆ The is a polynomial size circuit computing a polynomial P which is full rank w.r.t. any balanced partition $X = Y \cup Z$ (dynamic programming)

⋆ Consider a formula of $n^{O(1)}$-size computing f
One can write

$$f = \sum_{i=1}^{s} f_{i,1} f_{i_2} \ldots f_{i,\log n}$$

Proof sketch of separation of multilinear formulas and circuits

$\star$ The is a polynomial size circuit computing a polynomial P which is full rank w.r.t. any balanced partition $X = Y \cup Z$ (dynamic programming)

$\star$ Consider a formula of $n^{O(1)}$-size computing f
One can write

$$f = \sum_{i=1}^{s} f_{i,1} f_{i_2} \dots f_{i,\log n}$$

For a random balanced partition $X = Y \cup Z$, with positive probability, the rank defect in each term is enough so that f not full rank for the partition $Y \cup Z$

Proof sketch of separation of multilinear formulas and circuits

⋆ The is a polynomial size circuit computing a polynomial P which is full rank w.r.t. any balanced partition $X = Y \cup Z$ (dynamic programming)

⋆ Consider a formula of $n^{O(1)}$-size computing f
One can write

$$f = \sum_{i=1}^{s} f_{i,1} f_{i_2} \dots f_{i,\log n}$$

For a random balanced partition $X = Y \cup Z$, with positive probability, the rank defect in each term is enough so that f not full rank for the partition $Y \cup Z$

[Raz-06] Any multilinear formula computing P has size $n^{\Omega(\log n)}$

Formulas with small individual degree

[Raz-05] Any multilinear formula computing $\det_n$ or per_n has size $n^{\Omega(\log n)}$

Question: Lower bound for the size of **multiquadratic** formula computing $\det_n$ or per_n.

Formulas with small individual degree

[Raz-05] Any multilinear formula computing $\det_n$ or per_n has size $n^{\Omega(\log n)}$

Question: Lower bound for the size of **multiquadratic** formula computing $\det_n$ or per_n.

Related work: Lower bounds for homogeneous multi-r-ic formulas [Kayal-Saha-Tavenas-18]

Homogenization of circuits

Consider a circuit C computing a homogeneous polynomial of degree d: we will construct C' homogeneous circuit computing P

Each node u of the circuit C is replaced with $u_0, \dots, u_d$ in C' computing the homogeneous components of the polynomial computed at u in P.

– Addition gate: if $u = v + w$ in C, $u_k = v_k + w_k$ in C'

– Product gate: if $u = v \times w$ in C, in C':

$$u_k = \sum_{i+j=k} v_i \times w_j$$

in C'.

Homogenization of circuits

Consider a circuit C computing a homogeneous polynomial of degree d: we will construct C' homogeneous circuit computing P

Each node u of the circuit C is replaced with $u_0, \dots, u_d$ in C' computing the homogeneous components of the polynomial computed at u in P.

– Addition gate: if $u = v + w$ in C, $u_k = v_k + w_k$ in C'
– Product gate: if $u = v \times w$ in C, in C':

$$u_k = \sum_{i+j=k} v_i \times w_j$$

in C'.

If C has size s, C' has size $O(sd^2)$.

Homogenization of formulas

Given F formula of size s computing a polynomial of degree d:

– Do the circuit homogenization on F to get C' homogeneous circuit
– Duplicate gates in C' to get a homogeneous formula F'

F' has size $s^{\log d}$

Homogenization of formulas

Given F formula of size s computing a polynomial of degree d:

– Do the circuit homogenization on F to get C' homogeneous circuit
– Duplicate gates in C' to get a homogeneous formula F'

F' has size $s^{\log d}$

[Raz-10] A formula of size d and degree $d = O(\log s)$ can be homogenized in size $s^{O(1)}$.

Depth-reduction (parallelization)

- With polynomial blow-up of size
  - ⋆ Formulas: reduction to depth $O(\log s)$ (Brent, Kuck and Maruyama)
  - ⋆ Circuits: reduction de depth $O(\log d)$ (Valiant, Skyum, Berkowitz and Rackoff)

Depth-reduction (parallelization)

- With polynomial blow-up of size
  - ⋆ Formulas: reduction to depth $O(\log s)$ (Brent, Kuck and Maruyama)
  - ⋆ Circuits: reduction de depth $O(\log d)$ (Valiant, Skyum, Berkowitz and Rackoff)
- With subexponential blow-up
  - ⋆ Reduction to depth 4 (Agrawal and Vinay ; Koiran ; Tavenas)
  - ⋆ Reduction to depth 3 (Gupta, Kamath, Kayal and Saptharishi)

Reduction to depth $O(\log s)$ for formulas

For a formula F of size s:

- Find a subformula G of size $\approx s/2$
- The polynomial computed by F can be written as

$$F = G \times H_1 + H_2$$

  where H_1 and H_2 are also computed by formulas of size $\approx s/2$
- Apply induction to these three subformulas G, H_1, H_2

Depth-reduction for formulas

[Fournier-Limaye-Malod-Srinivasan-Tavenas-23] Let F be a homogeneous algebraic formula of size s and syntactic degree d computing a polynomial P. Then P is also computed by a formula F' of size $s^{O(1)}$ and depth $O(\log d)$.

Depth-reduction for formulas

[Fournier-Limaye-Malod-Srinivasan-Tavenas-23] Let F be a homogeneous algebraic formula of size s and syntactic degree d computing a polynomial P. Then P is also computed by a formula F' of size $s^{O(1)}$ and depth $O(\log d)$.

Moreover, the construction preserves

- monotonicity
- non-commutativity
- (set-)multilinearity

Reducing the size blow-up

Depth-reduction with near-linear size [Bshouty-Cleve-Eberly-95], [Bonnet-Buss-94]

$\varepsilon > 0$, F be a algebraic formula of size s computing P.
Then there is an algebraic formula F' of
– size at most $s^{1+\varepsilon}$
– depth $\Delta = 2^{O(1/\varepsilon)} \cdot \log s$
computing P.

Reducing the size blow-up

Depth-reduction with near-linear size [Bshouty-Cleve-Eberly-95], [Bonnet-Buss-94]

$\varepsilon > 0$, F be a algebraic formula of size s computing P.
Then there is an algebraic formula F' of
– size at most $s^{1+\varepsilon}$
– depth $\Delta = 2^{O(1/\varepsilon)} \cdot \log s$
computing P.

The construction preserves

- homogeneity
- monotonicity

Depth-reduction with small size blow-up

Using the above result, we can prove the following improved version of our depth-reduction:

Assume that P is computed by a formula of size s and syntactic degree $d \geq 1$. Then P is also computed by a formula of size at most $s^{1+\varepsilon}$ and depth $\Delta = 2^{O(1/\varepsilon)} \cdot \log d$.

Works also in the non-commutative case.
Preserves homogeneous and/or monotonicity.

Depth-reduction: optimality in the monotone setting

Let n and $d = d(n)$ be growing parameters such that $d(n) \leq \sqrt{n}$.

Then there is a monotone algebraic formula F of size at most n and depth $O(\log d)$ computing a polynomial $P \in \mathbb{F}[x_1, \ldots, x_n]$ of degree at most d such that:

any monotone formula of depth $o(\log d)$ computing P must have size $n^{\omega(1)}$.

Optimality of $O(\log d)$ depth-reduction: the hard polynomial

Parameters $k \geq 1$ and $r \geq 2$. The polynomial $H = H^{(k,r)}$ is:

k-nested inner products, each one of size r

H is computed by a monotone formula M of size $(2r)^k$ and depth $2k$, with a $+$-gate at the top, alternating layers of $+$-gates and $\times$-gates, with $+$-gates of fan-in r and $\times$-gates of fan-in 2, and leaves labelled with distinct variables.

H is a polynomial of degree $d = 2^k$ and has r^{d-1} monomials.

Optimality of $O(\log d)$ depth-reduction: the hard polynomial

Parameters $k \geq 1$ and $r \geq 2$. The polynomial $H = H^{(k,r)}$ is:

k-nested inner products, each one of size r

H is computed by a monotone formula M of size $(2r)^k$ and depth $2k$, with a $+$-gate at the top, alternating layers of $+$-gates and $\times$-gates, with $+$-gates of fan-in r and $\times$-gates of fan-in 2, and leaves labelled with distinct variables.

H is a polynomial of degree $d = 2^k$ and has r^{d-1} monomials.

Lower bound: A monotone formula of product-depth $\Delta \leq \log d$ which computes H has size at least $r^{\Omega(\Delta d^{1/\Delta})}$.

Depth-reduction: proof (1/5)

Formula G of syntactic degree $d_G \geq 1$ and sum-depth $\Delta(G)$
Potential function $\phi_\delta(G)$:

$$\begin{cases} \phi_{\delta,1}(G) = \lceil \log(d_G) \rceil \\ \phi_{\delta,2}(G) = \lceil \Delta(G)/\delta \rceil \end{cases}$$

and let

$$\phi_\delta(G) = \phi_{\delta,1}(G) + \phi_{\delta,2}(G).$$

(δ : positive integer to be chosen)

Depth-reduction: proof (2/5)

Formula F of

- size s
- syntactic degree d (not necessarily homogeneous)
- depth $O(\log s)$, fanin 2

(after classical depth-reduction).

Depth-reduction: proof (2/5)

Formula F of

- size s
- syntactic degree d (not necessarily homogeneous)
- depth $O(\log s)$, fanin 2

(after classical depth-reduction).

We prove that F can be parallelized into a formula of arbitrary fan-in with

- product-depth at most $\phi_\delta(F)$
- size at most $s \cdot 2^{\delta \log(d)}$

Taking $\delta = \frac{\log s}{\log d}$ gives the result
Potential function: $\phi_\delta(H) = \lceil \log(d_H) \rceil + \lceil \Delta(H)/\delta \rceil$

Depth-reduction: proof (3/5)

Potential function: $\phi_\delta(H) = \lceil \log(d_H) \rceil + \lceil \Delta(H)/\delta \rceil$

Consider the following set of gates of F:

$$\mathcal{A} = \left\{ \alpha \mid \phi_\delta(F_\alpha) < \phi_\delta(F) = \phi_\delta(F_{\text{parent}(\alpha)}) \right\}.$$

G: formula obtained from F by replacing gates from $\mathcal{A}$ by variables

Depth-reduction: proof (3/5)

Potential function: $\phi_\delta(H) = \lceil \log(d_H) \rceil + \lceil \Delta(H)/\delta \rceil$

Consider the following set of gates of F:

$$\mathcal{A} = \left\{ \alpha \mid \phi_\delta(F_\alpha) < \phi_\delta(F) = \phi_\delta(F_{\mathsf{parent}(\alpha)}) \right\}.$$

G: formula obtained from F by replacing gates from $\mathcal{A}$ by variables

- G is skew
- G has sum-depth at most δ

Depth-reduction: proof (4/5)

Lemma] The polynomial computed by G is a multilinear polynomial with at most 2^δ monomials. Moreover any variable labelling in G

- son a $+$-leaf,
- or son a $\times$-leaf, and whose sibling is a leaf

appears in exactly one monomial (*non-duplicable* leaves).

Depth-reduction: proof (4/5)

Lemma] The polynomial computed by G is a multilinear polynomial with at most 2^{δ} monomials. Moreover any variable labelling in G

- son a $+$-leaf,
- or son a $\times$-leaf, and whose sibling is a leaf

appears in exactly one monomial (*non-duplicable* leaves).

Proof: parse trees

Depth-reduction: proof (5/5)

Start with F of fanin 2 and depth reduced to $O(\log s)$
Potential function $\phi_\delta(H) = \lceil \log(d_H) \rceil + \lceil \Delta(H)/\delta \rceil$ with $\delta = \frac{\log s}{\log d}$
$F = G(F_1, \ldots, F_\ell)$ where F_i are highest gates where ϕ decreases

- Write G as a $\sum\prod$-formula
- Recurse on each F_i, where $F = G(F_1, \ldots, F_\ell)$

Depth-reduction: proof (5/5)

Start with F of fanin 2 and depth reduced to $O(\log s)$
Potential function $\phi_\delta(H) = \lceil \log(d_H) \rceil + \lceil \Delta(H)/\delta \rceil$ with $\delta = \frac{\log s}{\log d}$
$F = G(F_1, \ldots, F_\ell)$ where F_i are highest gates where ϕ decreases

- Write G as a $\sum\prod$-formula
- Recurse on each F_i, where $F = G(F_1, \ldots, F_\ell)$

Depth of the formula obtained is $\phi_\delta(F) = O(\log d)$
Size of the resulting formula is bounded by

$$\sum_{\alpha \text{ non-duplicable}} \left(s_\alpha \cdot 2^{\delta \log(d_\alpha)}\right) + \left(2^\delta \cdot \sum_{\alpha \text{ duplicable}} \left(s_\alpha \cdot 2^{\delta \log(d_\alpha)}\right)\right).$$

Depth-reduction: proof (5/5)

Start with F of fanin 2 and depth reduced to $O(\log s)$
Potential function $\phi_\delta(H) = \lceil \log(d_H) \rceil + \lceil \Delta(H)/\delta \rceil$ with $\delta = \frac{\log s}{\log d}$
$F = G(F_1, \dots, F_\ell)$ where F_i are highest gates where ϕ decreases

- Write G as a $\sum\prod$-formula
- Recurse on each F_i, where $F = G(F_1, \dots, F_\ell)$

Depth of the formula obtained is $\phi_\delta(F) = O(\log d)$
Size of the resulting formula is bounded by

$$\sum_{\alpha \text{ non-duplicable}} \left(s_\alpha \cdot 2^{\delta \log(d_\alpha)}\right) + \left(2^\delta \cdot \sum_{\alpha \text{ duplicable}} \left(s_\alpha \cdot 2^{\delta \log(d_\alpha)}\right)\right).$$

$$\leq s \cdot 2^{\delta \log d} + \sum_{\alpha \text{ duplicable}} 2^\delta \left(s_\alpha \cdot 2^{\delta(\log(d)-1)}\right)$$

Depth-reduction: proof (5/5)

Start with F of fanin 2 and depth reduced to $O(\log s)$
Potential function $\phi_\delta(H) = \lceil \log(d_H) \rceil + \lceil \Delta(H)/\delta \rceil$ with $\delta = \frac{\log s}{\log d}$
$F = G(F_1, \ldots, F_\ell)$ where F_i are highest gates where ϕ decreases

- Write G as a $\sum\prod$-formula
- Recurse on each F_i, where $F = G(F_1, \ldots, F_\ell)$

Depth of the formula obtained is $\phi_\delta(F) = O(\log d)$
Size of the resulting formula is bounded by

$$\sum_{\alpha \text{ non-duplicable}} \left(s_\alpha \cdot 2^{\delta \log(d_\alpha)}\right) + \left(2^\delta \cdot \sum_{\alpha \text{ duplicable}} \left(s_\alpha \cdot 2^{\delta \log(d_\alpha)}\right)\right).$$

$$\leq s \cdot 2^{\delta \log d} + \sum_{\alpha \text{ duplicable}} 2^\delta \left(s_\alpha \cdot 2^{\delta(\log(d)-1)}\right)$$

$$\leq s \cdot 2^{\delta \log d} = s \cdot 2^{\log s} = s^{O(1)}$$

Structure inside VF

Consider these three classes

- homF[$s(n)$]: (f_n) computed by a homogeneous formula of size $\mathrm{poly}(s(n))$,
- lowSynDegF[$s(n)$]: (f_n) computed by a formula of size $\mathrm{poly}(s(n))$ and of *syntactic degree* $\mathrm{poly}(\deg(f_n))$
- lowDepthF[$s(n)$]: computed by a formula of size $\mathrm{poly}(s(n))$ and of *depth* $O(\log \deg(f_n))$.

Structure inside VF

Consider these three classes

- homF[$s(n)$]: (f_n) computed by a homogeneous formula of size $\mathrm{poly}(s(n))$,
- lowSynDegF[$s(n)$]: (f_n) computed by a formula of size $\mathrm{poly}(s(n))$ and of *syntactic degree* $\mathrm{poly}(\deg(f_n))$
- lowDepthF[$s(n)$]: computed by a formula of size $\mathrm{poly}(s(n))$ and of *depth* $O(\log \deg(f_n))$.

homF[$\mathrm{poly}(n)$] $\subseteq$ lowSynDegF[$\mathrm{poly}(n)$] $\subseteq$ lowDepthF[$\mathrm{poly}(n)$] $\subseteq$ VF,

Question: which inclusions are strict?

Homegenous vs. Low syntactic degree

Elementary Symmetric Polynomials $S_n^d(x_1, \ldots, x_n)$
$\star$ Computed by *inhomogeneous* formula of depth-3 and size $O(n^2)$
[Ben-Or]
S_n^d is in lowDepthF$[\mathrm{poly}(n)]$.
$\star$ S_n^d has depth-6 formulas of syntactic degree at most $\mathrm{poly}(d)$
[Shpilka-Wigderson]
S_n^d in lowSynDegF$[\mathrm{poly}(n)]$.
$\star$ S_n^d is not known to have $\mathrm{poly}(n)$-sized homogeneous formulas

Homegenous vs. Low syntactic degree

Elementary Symmetric Polynomials $S_n^d(x_1, \ldots, x_n)$
$\star$ Computed by *inhomogeneous* formula of depth-3 and size $O(n^2)$ [Ben-Or]
S_n^d is in $\text{lowDepthF}[\mathrm{poly}(n)]$.
$\star$ S_n^d has depth-6 formulas of syntactic degree at most $\mathrm{poly}(d)$ [Shpilka-Wigderson]
S_n^d in $\text{lowSynDegF}[\mathrm{poly}(n)]$.
$\star$ S_n^d is not known to have $\mathrm{poly}(n)$-sized homogeneous formulas

If S_n^d does not have $\mathrm{poly}(n)$-sized homogeneous formulas

$$\text{homF}[\mathrm{poly}(n)] \subsetneq \text{lowSynDegF}[\mathrm{poly}(n)]$$

Homegenous vs. Low syntactic degree

Elementary Symmetric Polynomials $S_n^d(x_1, \ldots, x_n)$
$\star$ Computed by *inhomogeneous* formula of depth-3 and size $O(n^2)$ [Ben-Or]
S_n^d is in lowDepthF$[\mathrm{poly}(n)]$.
$\star$ S_n^d has depth-6 formulas of syntactic degree at most $\mathrm{poly}(d)$ [Shpilka-Wigderson]
S_n^d in lowSynDegF$[\mathrm{poly}(n)]$.
$\star$ S_n^d is not known to have $\mathrm{poly}(n)$-sized homogeneous formulas

If S_n^d does not have $\mathrm{poly}(n)$-sized homogeneous formulas

$$\text{homF}[\mathrm{poly}(n)] \subsetneq \text{lowSynDegF}[\mathrm{poly}(n)]$$

If S_n^d is computed by $\mathrm{poly}(n)$-sized homogeneous formulas, any depth-3 formula of polynomial size and low syntactic degree can be homogenized : $\sum[c \cdot \prod_i (1 + \ell_i)]$

Thank you!