Algebraic Complexity Theory

Lecture 5: Determinant vs. Permanent; Circuit lower bounds

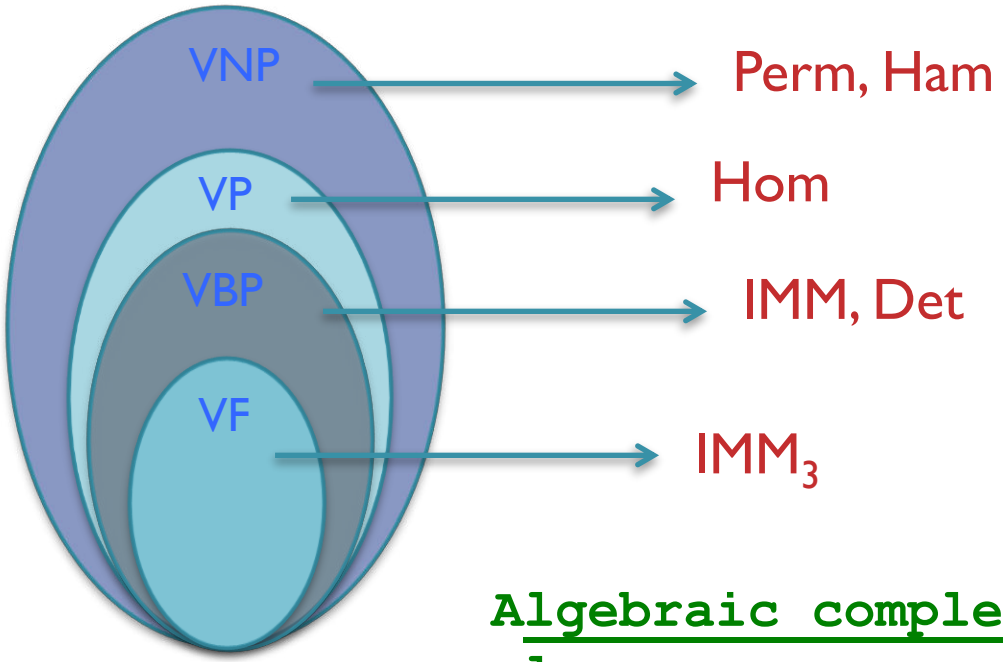
Department of Computer Science,
Indian Institute of Science

Recap

- In the last two lectures, we defined the complexity classes VNP, VP, VBP, VF, and observed that VF $\subseteq$ VBP $\subseteq$ VP $\subseteq$ VNP. Whether or not any of these containments is proper is an open problem.

- We also defined "complete" families of polynomials for the above-mentioned classes using *p-projections* and saw that IMM_3 is VF-complete, Det and IMM are VBP-complete, Hom is VP-complete, and Perm and Ham are VNP-complete.

Recap



Algebraic complexity classes

The VBP vs. VNP problem can be equivalently stated as follows: Prove that if $Perm_n$ is a projection of Det_m then $m = n^{\omega(1)}$. Naturally, it is also known as the ***Permanent versus Determinant*** problem.

Permanent versus Determinant

Perm versus Det

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?

Perm versus Det: Upper bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Recall the following from the previous lectures.
- Obs. Ryser's formula gives a layered ABP of size $n2^n$ for Perm_n.
- Claim. *(Valiant 1979)* If f is computable by a layered ABP of size s then f is an affine projection of $\text{Det}_{O(s)}$.

- Thus, $m = O(n2^n)$.

Perm versus Det: Upper bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Recall the following from the previous lectures.
- Obs. Ryser's formula gives a layered ABP of size $n2^n$ for Perm_n.
- Claim. *(Valiant 1979)* If f is computable by a layered ABP of size s then f is an affine projection of $\text{Det}_{O(s)}$.

- Thus, $m = O(n2^n)$. There's a better upper bound!

Perm versus Det: Upper bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Theorem. *(Grenet 2012, Nisan 1991)* $m \leq 2^n - 1$.

- Thus, for $n = 3$, $m \leq 7$, which is known to be optimal (*Ikenmeyer, Hüttenhain 2016*; *Alper, Bogart, Velasco 2017*).
- It's easy to see that for $n = 2$, $m = 2$.

$$\mathrm{perm} \begin{bmatrix} x_{11} & x_{12} \\ x_{21} & x_{22} \end{bmatrix} = \det \begin{bmatrix} x_{11} & -x_{12} \\ x_{21} & x_{22} \end{bmatrix}$$

Perm versus Det: Upper bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Theorem. *(Grenet 2012, Nisan 1991)* $m \leq 2^n - 1$.
- *Proof sketch.* We'll create a layered ABP A with 2^n nodes that computes Perm_n. Then, we'll derive a matrix M from A such that $\det(M) = \text{Perm}_n$ (as in Valiant's proof of VBP-hardness of Det).

Perm versus Det: Upper bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Theorem. *(Grenet 2012, Nisan 1991)* $m \leq 2^n - 1$.
- *Proof sketch.* We'll create a layered ABP A with 2^n nodes that computes Perm_n. Then, we'll derive a matrix M from A such that $\det(M) = \mathrm{Perm}_n$.
- The ABP A has n+1 layers of nodes $V_0, \ldots, V_n$. The nodes of V_i are labelled by all subsets of [n] of size i.

Perm versus Det: Upper bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Theorem. *(Grenet 2012, Nisan 1991)* $m \leq 2^n - 1$.
- *Proof sketch.* We'll create a layered ABP A with 2^n nodes that computes Perm_n. Then, we'll derive a matrix M from A such that $\det(M) = \mathrm{Perm}_n$.
- The ABP A has $n+1$ layers of nodes $V_0, \ldots, V_n$. The nodes of V_i are labelled by all subsets of $[n]$ of size i.
- There's an edge, labelled by x_{ij}, from a node S in V_{i-1} to a node $S \cup \{j\}$ in V_i if $j \notin S$.

Perm versus Det: Upper bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Theorem. *(Grenet 2012, Nisan 1991)* $m \leq 2^n - 1$.
- *Proof sketch.* Observe that the ABP A has 2^n nodes and it computes Perm_n.

Perm versus Det: Upper bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Theorem. *(Grenet 2012, Nisan 1991)* $m \leq 2^n - 1$.
- *Proof sketch.* Observe that the ABP A has 2^n nodes and it computes Perm_n.
- Merge the nodes in V_0 and V_n, and add a self-loop to every other node to obtain a digraph G on $2^n - 1$ vertices. Let M be the adjacency matrix of G.
- Observe that $\det(M) = \mathrm{Perm}_n$ if n is odd.

Perm versus Det: Upper bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Theorem. *(Grenet 2012, Nisan 1991)* $m \leq 2^n - 1$.
- *Proof sketch.* Observe that the ABP A has 2^n nodes and it computes Perm_n.
- Merge the nodes in V_0 and V_n, and add a self-loop to every other node to obtain a digraph G on $2^n - 1$ vertices. Let M be the adjacency matrix of G.
- Observe that $\det(M) = \mathrm{Perm}_n$ if n is odd.
- If n is even, alter G slightly. (*Homework*: how?)

Perm versus Det: Lower bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- It is conjectured that $m = 2^{\Omega(n)}$.

- Obs. If we show $m = n^{\omega(1)}$, then VBP $\neq$ VNP. If we show $m = n^{\omega(\log n)}$, then by the "depth reduction" results we can infer that VP $\neq$ VNP.

- Degree comparison gives $m \geq n$. There's a significantly better lower bound!

Perm versus Det: Lower bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Theorem. *(Mignon & Ressayre 2004)* $m \geq n^2/2$ over any field of characteristic zero.

- The lower bound can be extended to arbitrary fields of characteristic $\neq 2$. (*Cai, Chen and Li 2008*).

- Theorem. *(Yabe 2015)* $m \geq n^2 - 2n + 2$ over $\mathbb{R}$.

Perm versus Det: Lower bound

- Question. How large is the *smallest* m s.t. Perm_n is an *affine projection* of Det_m?
- Theorem. *(Mignon & Ressayre 2004)* $m \geq n^2/2$ over any field of characteristic zero.

- Unfortunately, the above theorem doesn't imply a superlinear (in the number of variables) lower bound for circuits, or even ABPs, as n^2 is the number of variables of Perm_n.

Univariate circuit lower bounds

Univariate polynomials and circuits

- Obs. A polynomial $f(x) = a_D x^D + \ldots + a_0$, where $a_i \in \mathbb{F}$, can be easily computed by a circuit over $\mathbb{F}$ of size $O(D \log D)$ using repeated squaring.

Univariate polynomials and circuits

- Obs. A polynomial $f(x) = a_D x^D + \ldots + a_0$, where $a_i \in \mathbb{F}$, can be easily computed by a circuit over $\mathbb{F}$ of size $O(D \log D)$ using repeated squaring.

- Horner's rule. *(1819)* Polynomial f can be computed by a formula that uses D additions and D multiplications as $f = a_0 + x(a_1 + x(a_2 + x(a_3 + \ldots)))$.

Univariate polynomials and circuits

- Obs. A polynomial $f(x) = a_D x^D + \ldots + a_0$, where $a_i \in \mathbb{F}$, can be easily computed by a circuit over $\mathbb{F}$ of size $O(D \log D)$ using repeated squaring.

- Horner's rule. *(1819)* Polynomial f can be computed by a formula that uses D additions and D multiplications as $f = a_0 + x(a_1 + x(a_2 + x(a_3 + \ldots)))$.

- Question. *(Ostrowski 1954)* Is Horner's rule optimal?
- Ref. "*On two problems in abstract algebra connected to Horner's rule*", by Ostrowski (1954).

Univariate polynomials and circuits

- Definition. The number of × and ÷ gates with at least two children *not* labelled by field constants, is called the **non-scalar complexity** of a circuit. If the circuit has no ÷ gates, then non-scalar complexity is also called the **multiplicative complexity**.

Univariate polynomials and circuits

- Definition. The number of × and ÷ gates with at least two children *not* labelled by field constants, is called the **non-scalar complexity** of a circuit. If the circuit has no ÷ gates, then non-scalar complexity is also called the **multiplicative complexity**.

- Notations. $S(f)$:= complexity of f = the size of the smallest circuit computing f. Similarly, $S_m(f)$:= the multiplicative complexity of f, and $S_{ns}(f)$:= the non-scalar complexity of f.

Univariate polynomials and circuits

- Let $f(x) = a_D x^D + \ldots + a_0$, where $a_0, \ldots, a_D$ and x are *variables*. Horner's rule implies $S_{ns}(f) \leq D$.
- Theorem. *(Pan 1966)* $S_{ns}(f) = D$.

- Ref. "*Methods of computing values of polynomials*" by Pan (1966).

Univariate polynomials and circuits

- Let $f(x) = a_D x^D + \ldots + a_0$, where $a_0, \ldots, a_D$ and x are *variables*. Horner's rule implies $S_{ns}(f) \leq D$.
- Theorem. *(Pan 1966)* $S_{ns}(f) = D$.

- Ref. "*Methods of computing values of polynomials*" by Pan (1966).

- Question. Are there *explicit* degree-D univariate polynomials with circuit complexity $\Omega(D)$?

Univariate polynomials and circuits

- Theorem. *(Strassen 1974)* Let $f(x) = \sum_{i \in [0,D]} 2^{2^{iD^2}} x^i$. Then, the number of operations in any circuit over $\mathbb{C}$ computing f is $\Omega(D)$, i.e., $S(f) = \Omega(D)$ over $\mathbb{C}$.

- Ref. "*Polynomial with rational coefficients which are hard to compute*" by Strassen (1974).

Univariate polynomials and circuits

- Theorem. *(Strassen 1974)* Let $f(x) = \sum_{i \in [0,D]} 2^{2^{iD^2}} x^i$. Then, the number of operations in any circuit over $\mathbb{C}$ computing f is $\Omega(D)$, i.e., $S(f) = \Omega(D)$ over $\mathbb{C}$.

- Theorem. *(Bürgisser, Clausen, Shokrollahi 1997)* Let $f(x) = \sum_{i \in [1,D]} \sqrt{p_i} \cdot x^i$, where p_i is the i^{th} prime. Then, $S(f) = \Omega(D/\log D)$ and $S_m(f) = \Omega(\sqrt{(D/\log D)})$ over $\mathbb{C}$.

- Ref. "*Algebraic Complexity Theory*" (Ch-9, Cor 9.4) by *Bürgisser, Clausen, Shokrollahi (1997)*.

Univariate polynomials and circuits

- Theorem. *(Strassen 1974)* Let $f(x) = \sum_{i\in[0,D]} 2^{2^{iD^2}} x^i$. Then, the number of operations in any circuit over $\mathbb{C}$ computing f is $\Omega(D)$, i.e., $S(f) = \Omega(D)$ over $\mathbb{C}$.

- Theorem. *(Bürgisser, Clausen, Shokrollahi 1997)* Let $f(x) = \sum_{i\in[1,D]} \sqrt{p_i} \cdot x^i$, where p_i is the i^{th} prime. Then, $S(f) = \Omega(D/\log D)$ and $S_m(f) = \Omega(\sqrt{(D/\log D)})$ over $\mathbb{C}$.

- However, the f in the above two theorems are *not sufficiently explicit*. Unless the bit complexity of the coefficients is $\text{poly}(D)$, f can't be evaluated efficiently.

Univariate polynomials and circuits

- Theorem. *(Strassen 1974)* Let $f(x) = \sum_{i\in[0,D]} 2^{2^{iD^2}} x^i$. Then, the number of operations in any circuit over $\mathbb{C}$ computing f is $\Omega(D)$, i.e., $S(f) = \Omega(D)$ over $\mathbb{C}$.

- Theorem. *(Bürgisser, Clausen, Shokrollahi 1997)* Let $f(x) = \sum_{i\in[1,D]} \sqrt{p_i} \cdot x^i$, where p_i is the i^{th} prime. Then, $S(f) = \Omega(D/\log D)$ and $S_m(f) = \Omega(\sqrt{(D/\log D)})$ over $\mathbb{C}$.

- Another reason to look for an f with low coefficient complexity comes from the connection between univariate and *multivariate circuit lower bounds*.

Univariate lb ⇒ Multivariate lb

- Let $f(x) = \sum_{i \in [0,D]} a_i\, x^i$, where $a_i \in \mathbb{F}$. Let $n = \lfloor \log D \rfloor + 1$, $\mathbf{y} = \{y_0, \ldots, y_{n-1}\}$, and $\mathrm{bin}(i) = (i_{n-1}, \ldots, i_0)$ be the bits in the binary representation of i.
- Define $\mathbf{y}^{\mathrm{bin}(i)} = y_0^{i_0} \cdot \ldots \cdot y_{n-1}^{i_{n-1}}$ and $f(\mathbf{y}) = \sum_{i \in [0,D]} a_i\, \mathbf{y}^{\mathrm{bin}(i)}$. Observe that $f(\mathbf{y})$ is a multilinear polynomial.

Univariate lb ⇒ Multivariate lb

- Let $f(x) = \sum_{i\in[0,D]} a_i\, x^i$, where $a_i \in \mathbb{F}$. Let $n = \lfloor \log D \rfloor + 1$, $\mathbf{y} = \{y_0, \ldots, y_{n-1}\}$, and $\mathrm{bin}(i) = (i_{n-1}, \ldots, i_0)$ be the bits in the binary representation of i.
- Define $\mathbf{y}^{\mathrm{bin}(i)} = y_0^{i_0} \cdot \ldots \cdot y_{n-1}^{i_{n-1}}$ and $f(\mathbf{y}) = \sum_{i\in[0,D]} a_i\, \mathbf{y}^{\mathrm{bin}(i)}$. Observe that $f(\mathbf{y})$ is a multilinear polynomial.
- Lemma. If *any* circuit computing f has size $\geq s$, then *any* circuit computing f has size $\mathcal{s} \geq s - O(\log D)$.
- Corollary. If $S(f) = \Omega(D)$, then $S(f) = \Omega(2^n)$.

Univariate lb ⇒ Multivariate lb

- Let $f(x) = \sum_{i\in[0,D]} a_i\, x^i$, where $a_i \in \mathbb{F}$. Let $n = \lfloor \log D \rfloor + 1$, $\mathbf{y} = \{y_0, \ldots, y_{n-1}\}$, and $\mathrm{bin}(i) = (i_{n-1}, \ldots, i_0)$ be the bits in the binary representation of i.
- Define $\mathbf{y}^{\mathrm{bin}(i)} = y_0^{i_0} \cdot \ldots \cdot y_{n-1}^{i_{n-1}}$ and $f(\mathbf{y}) = \sum_{i\in[0,D]} a_i\, \mathbf{y}^{\mathrm{bin}(i)}$. Observe that $f(\mathbf{y})$ is a multilinear polynomial.
- Lemma. If *any* circuit computing f has size $\geq s$, then *any* circuit computing f has size $\mathcal{s} \geq s - O(\log D)$.
- Corollary. If $S(f) = \omega(\log D)$, then $S(f) = \omega(n)$.

Univariate lb ⇒ Multivariate lb

- Let $f(x) = \sum_{i\in[0,D]} a_i\, x^i$, where $a_i \in \mathbb{F}$. Let $n = \lfloor \log D \rfloor + 1$, $\mathbf{y} = \{y_0, \ldots, y_{n-1}\}$, and $\mathrm{bin}(i) = (i_{n-1}, \ldots, i_0)$ be the bits in the binary representation of i.
- Define $\mathbf{y}^{\mathrm{bin}(i)} = y_0^{i_0} \cdot \ldots \cdot y_{n-1}^{i_{n-1}}$ and $f(\mathbf{y}) = \sum_{i\in[0,D]} a_i\, \mathbf{y}^{\mathrm{bin}(i)}$. Observe that $f(\mathbf{y})$ is a multilinear polynomial.
- Lemma. If *any* circuit computing f has size $\geq s$, then *any* circuit computing f has size $\mathcal{s} \geq s - O(\log D)$.
- Corollary. If $S(f) = \omega(\log D)$, then $S(f) = \omega(n)$.
- If the coefficients of f are computable in #P/poly then f defines a family in VNP. For this to happen the a_i's must necessarily have bit complexity poly(log D).

Univariate lb ⇒ Multivariate lb

- Let $f(x) = \sum_{i\in[0,D]} a_i\, x^i$, where $a_i \in \mathbb{F}$. Let $n = \lfloor \log D \rfloor + 1$, $\mathbf{y} = \{y_0, \ldots, y_{n-1}\}$, and $\mathrm{bin}(i) = (i_{n-1}, \ldots, i_0)$ be the bits in the binary representation of i.
- Define $\mathbf{y}^{\mathrm{bin}(i)} = y_0^{i_0} \cdot \ldots \cdot y_{n-1}^{i_{n-1}}$ and $f(\mathbf{y}) = \sum_{i\in[0,D]} a_i\, \mathbf{y}^{\mathrm{bin}(i)}$. Observe that $f(\mathbf{y})$ is a multilinear polynomial.
- Lemma. If *any* circuit computing f has size $\geq s$, then *any* circuit computing f has size $\mathcal{s} \geq s - O(\log D)$.
- *Proof.* Let C be a circuit of size $\mathcal{s}$ computing f. By replacing y_k by x^{2^k} in C we get a circuit for f.

Univariate lb ⇒ Multivariate lb

- Let $f(x) = \sum_{i\in[0,D]} a_i\, x^i$, where $a_i \in \mathbb{F}$. Let $n = \lfloor \log D \rfloor + 1$, $\mathbf{y} = \{y_0, \ldots, y_{n-1}\}$, and $\mathrm{bin}(i) = (i_{n-1}, \ldots, i_0)$ be the bits in the binary representation of i.
- Define $\mathbf{y}^{\mathrm{bin}(i)} = y_0^{i_0} \cdot \ldots \cdot y_{n-1}^{i_{n-1}}$ and $f(\mathbf{y}) = \sum_{i\in[0,D]} a_i\, \mathbf{y}^{\mathrm{bin}(i)}$. Observe that $f(\mathbf{y})$ is a multilinear polynomial.
- Lemma. If *any* circuit computing f has size $\geq s$, then *any* circuit computing f has size $\mathcal{s} \geq s - O(\log D)$.
- *Proof.* Let C be a circuit of size $\mathcal{s}$ computing f. By replacing y_k by x^{2^k} in C we get a circuit for f. As $x, x^2, \ldots, x^{2^{n-1}}$ can be computed using repeated squaring, $\mathcal{s} + 2(n-1) \geq s$ implying $\mathcal{s} \geq s - O(\log D)$.

Univariate lb ⇒ Multivariate lb

- Let $f(x) = \sum_{i \in [0,D]} a_i\, x^i$, where $a_i \in \mathbb{F}$. Let $n = \lfloor \log D \rfloor + 1$, $\mathbf{y} = \{y_0, \ldots, y_{n-1}\}$, and $\mathrm{bin}(i) = (i_{n-1}, \ldots, i_0)$ be the bits in the binary representation of i.
- Define $\mathbf{y}^{\mathrm{bin}(i)} = y_0^{i_0} \cdot \ldots \cdot y_{n-1}^{i_{n-1}}$ and $f(\mathbf{y}) = \sum_{i \in [0,D]} a_i\, \mathbf{y}^{\mathrm{bin}(i)}$. Observe that $f(\mathbf{y})$ is a multilinear polynomial.
- Lemma. If *any* circuit computing f has size $\geq s$, then *any* circuit computing f has size $\mathcal{s} \geq s - O(\log D)$.

- Proving a univariate circuit lower bound is "harder" than proving a multivariate circuit lower bound.

Univariate lb ⇒ Multivariate lb

- Let $f(x) = \sum_{i\in[0,D]} a_i\, x^i$, where $a_i \in \mathbb{F}$. Let $n = \lfloor \log D \rfloor + 1$, $\mathbf{y} = \{y_0, \ldots, y_{n-1}\}$, and $\mathrm{bin}(i) = (i_{n-1}, \ldots, i_0)$ be the bits in the binary representation of i.
- Define $\mathbf{y}^{\mathrm{bin}(i)} = y_0^{i_0} \cdot \ldots \cdot y_{n-1}^{i_{n-1}}$ and $f(\mathbf{y}) = \sum_{i\in[0,D]} a_i\, \mathbf{y}^{\mathrm{bin}(i)}$. Observe that $f(\mathbf{y})$ is a multilinear polynomial.
- Lemma. If *any* circuit computing f has size $\geq s$, then *any* circuit computing f has size $\mathcal{s} \geq s - O(\log D)$.

- Open problem. Describe an *explicit* univariate polynomial of degree D and having coefficient complexity $\mathrm{poly}(D)$ such that $S(f) = \omega(\log D)$.

Univariate lb ⇒ Multivariate lb

- Let $f(x) = \sum_{i\in[0,D]} a_i\, x^i$, where $a_i \in \mathbb{F}$. Let $n = \lfloor \log D \rfloor + 1$, $\mathbf{y} = \{y_0, \ldots, y_{n-1}\}$, and $\mathrm{bin}(i) = (i_{n-1}, \ldots, i_0)$ be the bits in the binary representation of i.
- Define $\mathbf{y}^{\mathrm{bin}(i)} = y_0^{i_0} \cdot \ldots \cdot y_{n-1}^{i_{n-1}}$ and $f(\mathbf{y}) = \sum_{i\in[0,D]} a_i\, \mathbf{y}^{\mathrm{bin}(i)}$. Observe that $f(\mathbf{y})$ is a multilinear polynomial.
- Lemma. If *any* circuit computing f has size $\geq s$, then *any* circuit computing f has size $\mathcal{s} \geq s - O(\log D)$.

- Remark. Proving a $\Omega(\log D)$ univariate lower bound is easy -- think of computing x^D.

A candidate "hard" univariate

- Wilkinson's polynomial. $w_D(x) := \prod_{i \in [1,D]} (x - i)$.
- Conjecture. $S(w_D) = \omega(\log D)$ over rationals.

- Remarks.
  - The bit complexity of every coefficient of w_D is *poly(D)*. So, w_D is more explicit than the two univariate polynomials mentioned before.
  - w_D is computable by a circuit over $\mathbb{Z}$ of size $O(D)$. The constants appearing in the circuit have bit complexity $O(\log D)$.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.

- Refs.

1. "*Factoring numbers in O(log n) arithmetic steps*" by Shamir (1979).
2. "*Straight-line complexity and integer factorization*" by Lipton (1994).

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Assume w.l.o.g. that $w_D(x) := \prod_{i \in [1,D]} (x + i)$.
- Goal. Design a poly-time TM M (with polynomial bits of advice) that takes input integer N and outputs a non-trivial factor of N, provided N is composite.
- Input size is $\lfloor \log N \rfloor + 1$.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Let n be such that $2^{n-1} < N \leq 2^n$.
- Advice. The circuits for $w_1, w_2, w_4, \ldots, w_{2^n}$.
- Observe that the size of the advice string is *poly(n)*, by the condition given in the theorem statement.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Let n be such that $2^{n-1} < N \leq 2^n$.
- Advice. The circuits for $w_1, w_2, w_4, \ldots, w_{2^n}$.
- Fact. (N-1)! = 0 mod N if and only if N is composite.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Let n be such that $2^{n-1} < N \leq 2^n$.
- Advice. The circuits for $w_1, w_2, w_4, \ldots, w_{2^n}$.
- Fact. $(N-1)! = 0 \bmod N$ if and only if N is composite.
- Our TM M tries to find the smallest $\ell < N$ such that $\ell! = 0 \bmod N$. Then, it computes $\gcd(\ell, N)$. As $(\ell-1)! \neq 0 \bmod N$, $\gcd(\ell, N)$ must be nontrivial.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Let n be such that $2^{n-1} < N \leq 2^n$.
- Advice. The circuits for $w_1, w_2, w_4, \ldots, w_{2^n}$.
- Fact. $(N-1)! = 0 \bmod N$ if and only if N is composite.
- Our TM M tries to find the smallest $\ell < N$ such that $\ell! = 0 \bmod N$. Then, it computes $\gcd(\ell, N)$.
- Observe, if $m! = 0 \bmod N$ then $(m+1)! = 0 \bmod N$.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Use "binary search" to compute ℓ.
  - Find the smallest i s.t. $2^i! = w_{2^i}(0) = 0 \bmod N$.

    This is done by evaluating the circuit for w_{2^i} at 0 and computing the output of every gate modulo N.

    As $2^{(i-1)}! \neq 0 \bmod N$, $\ell \in [2^{i-1}, 2^i]$.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Use "binary search" to compute ℓ.
  - Find the smallest i s.t. $2^i! = w_{2^i}(0) = 0 \bmod N$.

    Also, compute $2^{(i-1)}! = w_{2^{i-1}}(0) \bmod N$.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Use "binary search" to compute ℓ.
  - Find the smallest i s.t. $2^i! = w_{2^i}(0) = 0 \bmod N$.

    Also, compute $2^{(i-1)}! = w_{2^{i-1}}(0) \bmod N$.
  - Find the smallest $j \le i-1$ s.t. $\underbrace{2^{i-1}! \cdot w_{2^j}(2^{i-1})}_{= (2^{i-1} + 2^j)!} = 0 \bmod N$.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Use "binary search" to compute ℓ.
  - Find the smallest i s.t. $2^i! = w_{2^i}(0) = 0 \bmod N$.

    Also, compute $2^{(i-1)}! = w_{2^{i-1}}(0) \bmod N$.
  - Find the smallest $j \le i-1$ s.t. $2^{i-1}! \cdot w_{2^j}(2^{i-1}) = 0 \bmod N$.

    Then, $\ell \in [2^{i-1} + 2^{j-1}, 2^{i-1} + 2^j]$.

A candidate "hard" univariate

- Theorem. (*Shamir '79, Lipton '94*) If w_D is computable by a circuit over $\mathbb{Z}$ of size *poly(log D)* and the integers labelling the edges of the circuit have bit complexity *poly(log D)*, then *integer factoring* is in P/poly.
- *Proof.* Use "binary search" to compute ℓ.
  - Find the smallest i s.t. $2^i! = w_{2^i}(0) = 0 \bmod N$.

    Also, compute $2^{(i-1)}! = w_{2^{i-1}}(0) \bmod N$.
  - Find the smallest $j \le i-1$ s.t. $2^{i-1}! \cdot w_{2^j}(2^{i-1}) = 0 \bmod N$.

    Then, $\ell \in [2^{i-1} + 2^{j-1}, 2^{i-1} + 2^j]$.
  - Continue the "binary search" as above to find ℓ.

Multivariate circuit lower bounds

Existence of "hard" multivariates

- Obs. Every n-variate polynomial of degree d can be computed by a circuit of size $d \cdot {}^{n+d}C_d$.
- Question. Is there a polynomial with *low bit complexity of the coefficients* that requires circuit size $\Omega({}^{n+d}C_d)$?

Existence of "hard" multivariates

- Obs. Every n-variate polynomial of degree d can be computed by a circuit of size $d \cdot {}^{n+d}C_d$.
- Question. Is there a polynomial with *low bit complexity of the coefficients* that requires circuit size $\Omega({}^{n+d}C_d)$?

- As mentioned before, univariate lower bounds imply multivariate lower bounds. But, the univariates for which we know good lower bounds don't have low bit complexity of the coefficients. (Think about the univariate in Strassen's theorem.)

Existence of "hard" multivariates

- Obs. Every n-variate polynomial of degree d can be computed by a circuit of size $d \cdot {}^{n+d}C_d$.
- Question. Is there a polynomial with *low bit complexity of the coefficients* that requires circuit size $\Omega({}^{n+d}C_d)$?

- Unlike the case for Boolean circuits, a simple counting argument doesn't work here as there are *infinitely many circuits* (over infinite fields) even if the underlying digraph is fixed.

Existence of "hard" multivariates

- Obs. Every n-variate polynomial of degree d can be computed by a circuit of size $d \cdot {}^{n+d}C_d$.
- Question. Is there a polynomial with *low bit complexity of the coefficients* that requires circuit size $\Omega({}^{n+d}C_d)$? **Yes!**

- Unlike the case for Boolean circuits, a simple counting argument doesn't work here as there are *infinitely many circuits* (over infinite fields) even if the underlying digraph is fixed.

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.

- Algebraic independence is a generalization of the notion of linear independence.

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.
- Lemma *. Let $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg f_i \leq d$, and $m > n$. Then, there's a nonzero $h \in \mathbb{F}[y_1, \ldots, y_m]$ of degree $d^{O(n)}$ such that $h(f_1, \ldots, f_m) = 0$.

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.
- Lemma *. Let $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg f_i \leq d$, and $m > n$. Then, there's a nonzero $h \in \mathbb{F}[y_1, \ldots, y_m]$ of degree $d^{O(n)}$ such that $h(f_1, \ldots, f_m) = 0$.
- *Proof sketch.* Let h be a polynomial of degree D (to be fixed later in the analysis). Pretend that the coefficients of h are variables; call these $\mathbf{z}$ variables. So, $|\mathbf{z}| = {}^{m+D}C_m$.

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.
- Lemma *. Let $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg f_i \leq d$, and $m > n$. Then, there's a nonzero $h \in \mathbb{F}[y_1, \ldots, y_m]$ of degree $d^{O(n)}$ such that $h(f_1, \ldots, f_m) = 0$.
- *Proof sketch.* The polynomial $h(f_1, \ldots, f_m) \in \mathbb{F}[x_1, \ldots, x_n]$ has degree at most dD. So, there are at most ${}^{n+dD}C_n$ monomials in $h(f_1, \ldots, f_m)$.

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.
- Lemma *. Let $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg f_i \leq d$, and $m > n$. Then, there's a nonzero $h \in \mathbb{F}[y_1, \ldots, y_m]$ of degree $d^{O(n)}$ such that $h(f_1, \ldots, f_m) = 0$.
- *Proof sketch.* The polynomial $h(f_1, \ldots, f_m) \in \mathbb{F}[x_1, \ldots, x_n]$ has degree at most dD. So, there are at most ${}^{n+dD}C_n$ monomials in $h(f_1, \ldots, f_m)$. The coefficients of these monomials are *linear forms* in the $\mathbf{z}$ variables.

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.
- Lemma *. Let $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg f_i \leq d$, and $m > n$. Then, there's a nonzero $h \in \mathbb{F}[y_1, \ldots, y_m]$ of degree $d^{O(n)}$ such that $h(f_1, \ldots, f_m) = 0$.
- *Proof sketch.* If $|\mathbf{z}| = {}^{m+D}C_m > {}^{n+dD}C_n$, it's possible to set the $\mathbf{z}$ variables to $\mathbb{F}$ elements (not all 0) s.t. all the previously mentioned linear forms vanish, implying $h(f_1, \ldots, f_m) = 0$. Now choose D s.t. ${}^{m+D}C_m > {}^{n+dD}C_n$.

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.
- Lemma *. Let $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg f_i \leq d$, and $m > n$. Then, there's a nonzero $h \in \mathbb{F}[y_1, \ldots, y_m]$ of degree $d^{O(n)}$ such that $h(f_1, \ldots, f_m) = 0$.
- *Proof sketch.* ${}^{n+dD}C_n \leq (e(n+dD)/n)^n \leq (2edD/n)^n$ (assuming $dD \geq n$).

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.
- Lemma *. Let $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg f_i \leq d$, and $m > n$. Then, there's a nonzero $h \in \mathbb{F}[y_1, \ldots, y_m]$ of degree $d^{O(n)}$ such that $h(f_1, \ldots, f_m) = 0$.
- *Proof sketch.* ${}^{n+dD}C_n \leq (e(n+dD)/n)^n \leq (2edD/n)^n$ (assuming $dD \geq n$). ${}^{m+D}C_m \geq {}^{n+1+D}C_{n+1} \geq (D/(n+1))^{n+1}$.

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.
- Lemma *. Let $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg f_i \leq d$, and $m > n$. Then, there's a nonzero $h \in \mathbb{F}[y_1, \ldots, y_m]$ of degree $d^{O(n)}$ such that $h(f_1, \ldots, f_m) = 0$.
- *Proof sketch.* ${}^{n+dD}C_n \leq (e(n+dD)/n)^n \leq (2edD/n)^n$ (assuming $dD \geq n$). ${}^{m+D}C_m \geq {}^{n+1+D}C_{n+1} \geq (D/(n+1))^{n+1}$. If we choose D s.t. $(D/(n+1))^{n+1} \geq (2edD/n)^n$, we're done. Set $D = d^{O(n)}$.

Algebraic independence: A detour

- Definition. Polynomials $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$ are *algebraically dependent* if there's a nonzero polynomial $h \in \mathbb{F}[y_1, \ldots, y_m]$ such that $h(f_1, \ldots, f_m) = 0$. Such an h is called an *annihilating polynomial* for $f_1, \ldots, f_m$.
- Lemma *. Let $f_1, \ldots, f_m \in \mathbb{F}[x_1, \ldots, x_n]$, $\deg f_i \leq d$, and $m > n$. Then, there's a nonzero $h \in \mathbb{F}[y_1, \ldots, y_m]$ of degree $d^{O(n)}$ such that $h(f_1, \ldots, f_m) = 0$.

- Theorem. *(Perron 1927)* There's an annihilating polynomial h for $f_1, \ldots, f_m$ of degree $\leq d^n$.

Existence of "hard" multivariates

- Theorem. Let $n \leq s < {}^{n+d}C_n$ & $|\mathbb{F}| > 2^{O(s \log s)}$. There's a polynomial f_{hard} with bit complexity of the coefficients $O(s \log s)$ s.t. **no** circuit of size s computes f_{hard}.

- If $s = poly(n)$, the bit complexity of the coefficients of f_{hard} are polynomially bounded.

Existence of "hard" multivariates

- Theorem. Let $n \leq s < {}^{n+d}C_n$ & $|\mathbb{F}| > 2^{O(s \log s)}$. There's a polynomial f_{hard} with bit complexity of the coefficients $O(s \log s)$ s.t. **no** circuit of size s computes f_{hard}.
- *Proof sketch.* W.l.o.g a circuit of size s has at most s nodes. So, the number of distinct digraphs with s edges is $s^{O(s)}$. The nodes of such a digraph can be labelled in $n^{O(s)}$ ways using $+$, $\times$ or one of the n variables.

Existence of "hard" multivariates

- Theorem. Let $n \leq s < {}^{n+d}C_n$ & $|\mathbb{F}| > 2^{O(s \log s)}$. There's a polynomial f_{hard} with bit complexity of the coefficients $O(s \log s)$ s.t. **no** circuit of size s computes f_{hard}.
- *Proof sketch.* W.l.o.g a circuit of size s has at most s nodes. So, the number of distinct digraphs with s edges is $s^{O(s)}$. The nodes of such a digraph can be labelled in $n^{O(s)}$ ways using $+$, $\times$ or one of the n variables. Pick one of these $s^{O(s)}$ many digraphs, call it C, and label it's s edges by distinct variables $z_1, \ldots, z_s$.

Existence of "hard" multivariates

- Theorem. Let $n \leq s < {}^{n+d}C_n$ & $|\mathbb{F}| > 2^{O(s \log s)}$. There's a polynomial f_{hard} with bit complexity of the coefficients $O(s \log s)$ s.t. **no** circuit of size s computes f_{hard}.
- *Proof sketch.* W.l.o.g a circuit of size s has at most s nodes. So, the number of distinct digraphs with s edges is $s^{O(s)}$. The nodes of such a digraph can be labelled in $n^{O(s)}$ ways using $+$, $\times$ or one of the n variables. Pick one of these $s^{O(s)}$ many digraphs, call it C, and label it's s edges by distinct variables $z_1, \ldots, z_s$.
- The output of C is a polynomial whose ${}^{n+d}C_n$ coefficients are polynomials in $z_1, \ldots, z_s$ of degree $\leq s$.

Existence of "hard" multivariates

- Theorem. Let $n \leq s < {}^{n+d}C_n$ & $|\mathbb{F}| > 2^{O(s \log s)}$. There's a polynomial f_{hard} with bit complexity of the coefficients $O(s \log s)$ s.t. **no** circuit of size s computes f_{hard}.
- *Proof sketch.* Let $M(n,d) := \{\mathbf{e} \in \mathbb{Z}^n_{\geq 0} : ||\mathbf{e}||_1 \leq d\}$, and $\mathbf{y} := \{y_{\mathbf{e}} : \mathbf{e} \in M(n,d)\}$. Observe, $|M(n,d)| = |\mathbf{y}| = {}^{n+d}C_n$.
- By Lemma *, there's a *non-zero* annihilating polynomial $h_C(\mathbf{y})$ of degree $s^{O(s)}$ for the ${}^{n+d}C_n$ many coefficients polynomials in the $\mathbf{z}$ variables. (as $|\mathbf{z}| = s < {}^{n+d}C_n$)

Existence of "hard" multivariates

- Theorem. Let $n \leq s < {}^{n+d}C_n$ & $|\mathbb{F}| > 2^{O(s \log s)}$. There's a polynomial f_{hard} with bit complexity of the coefficients $O(s \log s)$ s.t. **no** circuit of size s computes f_{hard}.
- *Proof sketch.* Let $M(n,d) := \{\mathbf{e} \in \mathbb{Z}^n_{\geq 0} : ||\mathbf{e}||_1 \leq d\}$, and $\mathbf{y} := \{y_\mathbf{e} : \mathbf{e} \in M(n,d)\}$. Observe, $|M(n,d)| = |\mathbf{y}| = {}^{n+d}C_n$.
- By Lemma *, there's a *non-zero* annihilating polynomial $h_C(\mathbf{y})$ of degree $s^{O(s)}$ for the ${}^{n+d}C_n$ many coefficients polynomials in the $\mathbf{z}$ variables.
- Define $P(\mathbf{y}) := \prod_{\text{Circuit } C} h_C(\mathbf{y})$. Note that $\deg P = s^{O(s)}$.

Existence of "hard" multivariates

- Theorem. Let $n \leq s < {}^{n+d}C_n$ & $|\mathbb{F}| > 2^{O(s \log s)}$. There's a polynomial f_{hard} with bit complexity of the coefficients $O(s \log s)$ s.t. **no** circuit of size s computes f_{hard}.
- *Proof sketch.* Let $M(n,d) := \{\mathbf{e} \in \mathbb{Z}^n_{\geq 0} : ||\mathbf{e}||_1 \leq d\}$, and $\mathbf{y} := \{y_\mathbf{e} : \mathbf{e} \in M(n,d)\}$. Observe, $|M(n,d)| = |\mathbf{y}| = {}^{n+d}C_n$.
- By Lemma *, there's a *non-zero* annihilating polynomial $h_C(\mathbf{y})$ of degree $s^{O(s)}$ for the ${}^{n+d}C_n$ many coefficients polynomials in the $\mathbf{z}$ variables.
- Define $P(\mathbf{y}) := \prod_{\text{Circuit } C} h_C(\mathbf{y})$. Note that $\deg P = s^{O(s)}$.
- By SZ lemma, there's an $\mathbf{a} := (a_\mathbf{e} : \mathbf{e} \in M(n,d))$ s.t. $P(\mathbf{a}) \neq 0$ & bit complexity of $a_\mathbf{e}$ is $O(s \log s)$.

Existence of "hard" multivariates

- Theorem. Let $n \leq s < {}^{n+d}C_n$ & $|\mathbb{F}| > 2^{O(s \log s)}$. There's a polynomial f_{hard} with bit complexity of the coefficients $O(s \log s)$ s.t. **no** circuit of size s computes f_{hard}.
- *Proof sketch.* This mean, the hard polynomial

$$f_{hard} := \sum_{e \in M(n,d)} a_e \, x^e$$

is not computable by *any* circuit of size s.

Existence of "hard" multivariates

- Theorem. Let $n \leq s < {}^{n+d}C_n$ & $|\mathbb{F}| > 2^{O(s \log s)}$. There's a polynomial f_{hard} with bit complexity of the coefficients $O(s \log s)$ s.t. **no** circuit of size s computes f_{hard}.

- Remarks.
  - The application of the SZ lemma in the proof implies that *most* polynomials with coefficient bit complexity $O(s \log s)$ require circuits of size s.
  - The coefficient bit complexity of $O(s \log s)$ is not quite satisfactory if $s = n^{\omega(1)}$.

Existence of "hard" multivariates

- Theorem. *(Hrubes & Yehudayoff 2011)* Almost all n-variate multilinear polynomials with 0/1 coefficients require circuits of size $2^{\Omega(n)}$ over any field.
- Ref. "*Arithmetic complexity of algebraic extensions*" *by* Hrubes & Yehudayoff (2011).

Existence of "hard" multivariates

- Theorem. *(Hrubes & Yehudayoff 2011)* Almost all n-variate multilinear polynomials with 0/1 coefficients require circuits of size $2^{\Omega(n)}$ over any field.
- Ref. "*Arithmetic complexity of algebraic extensions*" *by* Hrubes & Yehudayoff (2011).

- Question. Is there an family of multilinear polynomials with 0/1 coefficients in VNP that has super-polynomial circuit complexity?
- Conjecture. Perm is such a family over fields of char $\neq 2$.