Lecture 4: Polynomial Identity Testing

Rafael Oliveira

University of Waterloo
Cheriton School of Computer Science

rafael.oliveira.teaching@gmail.com

January 20, 2021

Overview

- Word Problems and Polynomial Identity Testing
- Why is PIT so fundamental?
- PIT for restricted circuit classes
- Conclusion
- Acknowledgements

Word Problems

(1) G : multiplication table size $|G|^2$

$g_{i_1} \cdots g_{i_t} (g_{a_1} \cdots g_{a_n})^{-1} = id_G$

1. **Setting:** a group is given *succinctly* via generators and relations
2. **Input:** given a sequence of generators and operations among them forming a *word*, is this word the identity element in the group?

(2) G : generators $g_1, \ldots, g_n$

relations of G $\quad g_i g_j = g_j g_i$ (abelian)

representation 2 is more succinct than 1

$g_{i_1} g_{i_2} \cdots g_{i_t} \stackrel{?}{=} id_G$

Word Problems

1. **Setting:** a group is given *succinctly* via generators and relations
2. **Input:** given a sequence of generators and operations among them forming a *word*, is this word the identity element in the group?
3. For general finitely presented groups, this is undecidable

finitely many generators

finitely generated relations

Word Problems

1. **Setting:** a group is given *succinctly* via generators and relations
2. **Input:** given a sequence of generators and operations among them forming a *word*, is this word the identity element in the group?
3. For general finitely presented groups, this is undecidable
4. For hyperbolic groups, it is in P given Gromov's geometric techniques

Word Problems

1. **Setting:** a group is given *succinctly* via generators and relations
2. **Input:** given a sequence of generators and operations among them forming a *word*, is this word the identity element in the group?
3. For general finitely presented groups, this is undecidable
4. For hyperbolic groups, it is in P given Gromov's geometric techniques
5. what other word problems appear in TCS?

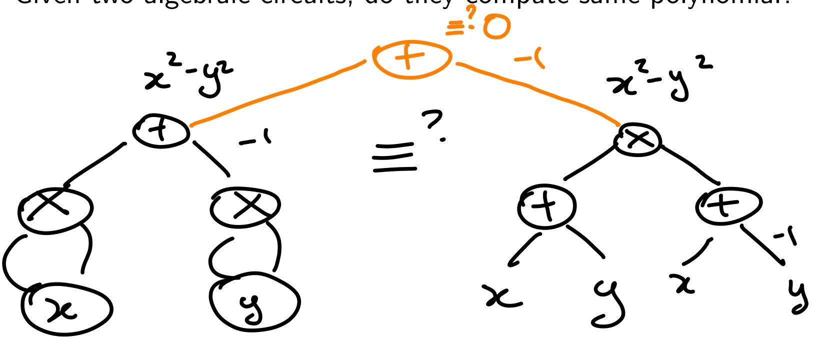
Polynomial Identity Testing (PIT)

word problem in algebraic complexity

1. Polynomials are given *succinctly* via algebraic circuits

Polynomial Identity Testing (PIT)

1. Polynomials are given *succinctly* via algebraic circuits
2. Given two algebraic circuits, do they compute same polynomial?



Polynomial Identity Testing (PIT)

1. Polynomials are given *succinctly* via algebraic circuits
2. Given two algebraic circuits, do they compute same polynomial?
3. Can be reduced to the question: given an algebraic circuit, does it compute the *zero polynomial*?

Polynomial Identity Testing

Polynomial Identity Testing (PIT)

1. Polynomials are given *succinctly* via algebraic circuits
2. Given two algebraic circuits, do they compute same polynomial?
3. Can be reduced to the question: given an algebraic circuit, does it compute the *zero polynomial*?

Polynomial Identity Testing

4. Two ways in which input can be given:
   1. **White-box model:** circuit is given as an input, with bound on the degree of the polynomial being computed
   2. **Black-box model:** one is given a bound on the degree of the polynomial, and one has only "oracle access" via evaluation

white-box : can see the entire ckt (we can modify the ckt in ways we want)

$$\bar{x} \equiv \boxed{\Phi} \longrightarrow \Phi(\bar{x})$$

Polynomial Identity Testing (PIT)

1. Polynomials are given *succinctly* via algebraic circuits
2. Given two algebraic circuits, do they compute same polynomial?
3. Can be reduced to the question: given an algebraic circuit, does it compute the *zero polynomial*?

Polynomial Identity Testing

4. Two ways in which input can be given:
    1. **White-box model:** circuit is given as an input, with bound on the degree of the polynomial being computed
    2. **Black-box model:** one is given a bound on the degree of the polynomial, and one has only "oracle access" via evaluation
5. Central question in TCS
    - best parallel algorithms for finding perfect matchings
    - Primes is in P
    - used in IP = PSPACE
    - proof of PCP theorem
    - structure of algebraic proof systems

Ore-Schwartz-Zippel-deMillo-Lipton Folklore Lemma

Lemma

If $p(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ is a non-zero polynomial of degree $\leq d$ and $S \subset \mathbb{F}$ is a finite set, then

$$\Pr_{\underbrace{a_i \in S}}[p(a_1, \ldots, a_n) = 0] \leq \frac{d}{|S|}$$

random point $(a_1, a_2, \ldots, a_n) \in S^n$

evaluate $p(x_1, \ldots, x_n)$

p can be computed by small algebraic ckts

Gives us randomized algorithm for PIT! coRP $\subset$ BPP

Ore-Schwartz-Zippel-deMillo-Lipton Folklore Lemma

Lemma

If $p(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ is a non-zero polynomial of degree $\leq d$ and $S \subset \mathbb{F}$ is a finite set, then

$$\Pr_{a_i \in S}[p(a_1, \ldots, a_n) = 0] \leq \frac{d}{|S|}$$

- Proof idea: in a domain $R[x]$, any polynomial $f(x)$ of degree $\leq d$ has at most d roots in R.

Ore-Schwartz-Zippel-deMillo-Lipton Folklore Lemma

Lemma

If $p(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ is a non-zero polynomial of degree $\leq d$ and $S \subset \mathbb{F}$ is a finite set, then

$$\Pr_{a_i \in S}[p(a_1, \ldots, a_n) = 0] \leq \frac{d}{|S|}$$

- Proof idea: in a domain $R[x]$, any polynomial $f(x)$ of degree $\leq d$ has at most d roots in R. base case
- Induction on number of variables: write

$$p(x_1, \ldots, x_n) = \sum_{e=1}^{k} \underline{p_e(x_1, \ldots, x_{n-1})} x_n^e \qquad p_k \neq 0$$

$\deg_n(p) = k$

Ore-Schwartz-Zippel-deMillo-Lipton Folklore Lemma

Lemma

If $p(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$ is a non-zero polynomial of degree $\leq d$ and $S \subset \mathbb{F}$ is a finite set, then

$$\Pr_{a_i \in S}[p(a_1, \ldots, a_n) = 0] \leq \frac{d}{|S|}$$

- Proof idea: in a domain $R[x]$, any polynomial $f(x)$ of degree $\leq d$ has at most d roots in R.
- Induction on number of variables: write

$\deg(p) \leq d$

$\deg(p_k \cdot x_n^k) \leq d$

$$p(x_1, \ldots, x_n) = \sum_{e=1}^{k} p_e(x_1, \ldots, x_{n-1}) x_n^e \qquad p_k \neq 0$$

$\deg(p_k) \leq d-k$

- By induction hypothesis $\Pr_{a_i \in S}[p_k(a_1, \ldots, a_{n-1}) = 0] \leq \dfrac{d-k}{|S|}$
- If $p_k(a_1, \ldots, a_{n-1}) \neq 0$ then $\leq k$ values of x_n will make $\underbrace{p(a_1, \ldots, a_{n-1}, x_n)}_{\deg k}$ zero, as it has degree k.

Black-Box Setting: Hitting Sets & Generators

$P(x_1, \ldots, x_n) \in \mathbb{F}[x_1, \ldots, x_n]$

- In black-box setting, given a circuit class $\mathcal{C}$, all we can do is to come up with a set $\mathcal{H} \subset \mathbb{F}^n$ (*hitting set*) such that

$$\Phi \in \mathcal{C}, \quad \Phi \not\equiv 0 \quad \Rightarrow \quad \exists \alpha \in \mathcal{H} \ \text{ s.t. } \ \Phi(\alpha) \neq 0$$

$\mathcal{H}$ hitting set ckt class $\mathcal{C}$

if $\Phi \in \mathcal{C}$ $\Phi \not\equiv 0$

then there is pt $\alpha \in \mathcal{H}$

s.t. $\Phi(\alpha) \neq 0$

(witnesses non-zeroness)

$\alpha_1 \to \Phi(\alpha_1)$

"

0

$\Phi(\alpha_2) = 0$

$\Phi(\alpha_3) = 0$

"non-adaptive setting"

Black-Box Setting: Hitting Sets & Generators

- In black-box setting, given a circuit class $\mathcal{C}$, all we can do is to come up with a set $\mathcal{H} \subset \mathbb{F}^n$ (*hitting set*) such that

$$\Phi \in \mathcal{C}, \quad \Phi \not\equiv 0 \quad \Rightarrow \quad \exists \alpha \in \mathcal{H} \text{ s.t. } \Phi(\alpha) \neq 0$$

- A polynomial map $\mathcal{G} = (\overbrace{g_1, \ldots, g_n}^{\text{polynomials}}) : \mathbb{F}^t \to \mathbb{F}^n$ is a *hitting set generator* for a circuit class $\mathcal{C}$ if

$$\Phi(x_1, \ldots, x_n) \in \mathcal{C}, \quad \Phi \not\equiv 0 \quad \Rightarrow \quad \underbrace{[\Phi \circ (g_1, \ldots, g_n)]}_{\text{polynomial}}(y_1, \ldots, y_t) \not\equiv 0$$

non zero

Black-Box Setting: Hitting Sets & Generators

- In black-box setting, given a circuit class $\mathcal{C}$, all we can do is to come up with a set $\mathcal{H} \subset \mathbb{F}^n$ (*hitting set*) such that

$$\Phi \in \mathcal{C}, \quad \Phi \not\equiv 0 \quad \Rightarrow \quad \exists \alpha \in \mathcal{H} \text{ s.t. } \Phi(\alpha) \neq 0$$

- A polynomial map $\mathcal{G} = (g_1, \ldots, g_n) : \mathbb{F}^t \to \mathbb{F}^n$ is a *hitting set generator* for a circuit class $\mathcal{C}$ if $t << n$

$$\Phi(x_1, \ldots, x_n) \in \mathcal{C}, \quad \Phi \not\equiv 0 \quad \Rightarrow \quad \underbrace{[\Phi \circ (g_1, \ldots, g_n)]}_{D \leftarrow \text{poly}}(y_1, \ldots, y_t) \not\equiv 0$$

- Hitting set generator decreases number of variables, and we can use brute-force to find non-zero

$|S| = D+1$ evaluate on S^t

$(D+1)^t \sim (D+1)^{\log n} \sim n^{\log n}$

$t = \log n$ quasi-P

Black-Box Setting: Hitting Sets & Generators

- In black-box setting, given a circuit class $\mathcal{C}$, all we can do is to come up with a set $\mathcal{H} \subset \mathbb{F}^n$ (*hitting set*) such that

$$\Phi \in \mathcal{C}, \quad \Phi \not\equiv 0 \quad \Rightarrow \quad \exists \alpha \in \mathcal{H} \ \text{ s.t. } \ \Phi(\alpha) \neq 0$$

- A polynomial map $\mathcal{G} = (g_1, \ldots, g_n) : \mathbb{F}^t \to \mathbb{F}^n$ is a *hitting set generator* for a circuit class $\mathcal{C}$ if

$$\Phi(x_1, \ldots, x_n) \in \mathcal{C}, \quad \Phi \not\equiv 0 \quad \Rightarrow \quad [\Phi \circ (g_1, \ldots, g_n)](y_1, \ldots, y_t) \not\equiv 0$$

- Hitting set generator decreases number of variables, and we can use brute-force to find non-zero
- In algebraic complexity, hitting set generators are also pseudorandom generators (decreased the number of "random seeds" needed)

$t \ll n$ means reducing randomness needed in Schwartz-Zippel

- Word Problems and Polynomial Identity Testing
- Why is PIT so fundamental?
- PIT for restricted circuit classes
- Conclusion
- Acknowledgements

Why do we want to derandomize PIT?

- PIT is an outstanding open question in derandomization (understand whether randomness is needed in design of efficient algorithms)

Why do we want to derandomize PIT?

- PIT is an outstanding open question in derandomization (understand whether randomness is needed in design of efficient algorithms)
- Hardness-Randomness tradeoff:

Theorem ([Kabanets & Impagliazzo 2004])

The following three assumptions cannot be simultaneously true:

1. $NEXP \subseteq P_{/poly}$
2. *Permanent is computable by polynomial size arithmetic circuits over* $\mathbb{Z}$
3. $PIT \in SUBEXP$

if have efficient PIT algorithm then we proved a lower bd that is way beyond current reach!

Why do we want to derandomize PIT?

- PIT is an outstanding open question in derandomization (understand whether randomness is needed in design of efficient algorithms)
- Hardness-Randomness tradeoff:

Theorem ([Kabanets & Impagliazzo 2004])

The following three assumptions cannot be simultaneously true:

1. $NEXP \subseteq P_{/poly}$
2. *Permanent is computable by polynomial size arithmetic circuits over* $\mathbb{Z}$
3. $PIT \in SUBEXP$

- Today we will show that (a strong version of) $\neg 2 \Rightarrow 3$

$$\text{Exponential lower bound on Permanent} \Rightarrow PIT \in \text{quasi-P}$$

hardness $\Rightarrow$ derandomization

"replace randomness by hard function"

Why do we want to derandomize PIT?

- PIT is an outstanding open question in derandomization (understand whether randomness is needed in design of efficient algorithms)
- Hardness-Randomness tradeoff:

Theorem ([Kabanets & Impagliazzo 2004])

The following three assumptions cannot be simultaneously true:

1. $NEXP \subseteq P_{/poly}$
2. *Permanent is computable by polynomial size arithmetic circuits over* $\mathbb{Z}$
3. $PIT \in SUBEXP$

- Today we will show that (a strong version of) $\neg 2 \Rightarrow 3$

$$\text{Exponential lower bound on Permanent} \;\Rightarrow\; PIT \in \text{quasi-P}$$

-

Lower Bounds imply Derandomization

NW'97 hardness vs randomness
in Boolean comp.

- Use Nisan-Wigderson designs:
  - $n \leq 2^m$ integers
  - There exist $S_1, \ldots, S_n \subset [m^2]$ such that
  - $|S_i| = m$, for all $1 \leq i \leq k$
  - $i \neq j \Rightarrow |S_i \cap S_j| \leq \log(n)$

Lower Bounds imply Derandomization

- Use Nisan-Wigderson designs:
    - $n \leq 2^m$ integers
    - There exist $S_1, \ldots, S_n \subset [m^2]$ such that
    - $|S_i| = m$, for all $1 \leq i \leq k$
    - $i \neq j \Rightarrow |S_i \cap S_j| \leq \log(n)$
- relaxed notion of combinatorial designs

S_i large enough but have small pairwise intersection

(q, k, t)-design

$[m] \supset S_1, \ldots, S_n$

$|S_i| = q$

$i \neq j \quad |S_i \cap S_j| = k \quad (\leq k)$

drop $\{$ $e \in [m]$ appears in t subsets $\}$

Lower Bounds imply Derandomization

- Use Nisan-Wigderson designs:
  - $n \leq 2^m$ integers
  - There exist $S_1, \ldots, S_n \subset [m^2]$ such that
  - $|S_i| = m$, for all $1 \leq i \leq k$
  - $i \neq j \Rightarrow |S_i \cap S_j| \leq \log(n)$
- relaxed notion of combinatorial designs
- Construction:
  1. Assume that $m = p$ is a prime

Lower Bounds imply Derandomization

- Use Nisan-Wigderson designs:
  - $n \leq 2^m$ integers
  - There exist $S_1, \ldots, S_n \subset [m^2]$ such that
  - $|S_i| = m$, for all $1 \leq i \leq k$
  - $i \neq j \Rightarrow |S_i \cap S_j| \leq \log(n)$
- relaxed notion of combinatorial designs
- Construction:
  1. Assume that $m = p$ is a prime
  2. Then, $\mathbb{F}_p^2 \sim [m^2]$

$(a, b) \in \mathbb{F}_p^2$

$\mathbb{F}_p$ finite field

p elements

Lower Bounds imply Derandomization

- Use Nisan-Wigderson designs:
    - $n \leq 2^m$ integers
    - There exist $S_1, \dots, S_n \subset [m^2]$ such that
    - $|S_i| = m$, for all $1 \leq i \leq k$
    - $i \neq j \Rightarrow |S_i \cap S_j| \leq \log(n)$
- relaxed notion of combinatorial designs
- Construction:
    1. Assume that $m = p$ is a prime
    2. Then, $\mathbb{F}_p^2 \sim [m^2]$
    3. Let $q_1, \dots, q_n \in \mathbb{F}_p^{\log(n)}$ be polynomials of degree $< \log(n)$.

$p^{\log n} > n$

$q_i(x) = q_{i0} + q_{i1}x + \dots + q_{i(\log n - 1)} x^{\log n - 1}$

$(q_{i0}, q_{i1}, \dots, q_{i(\log n - 1)})$

Lower Bounds imply Derandomization

- Use Nisan-Wigderson designs:
  - $n \leq 2^m$ integers
  - There exist $S_1, \ldots, S_n \subset [m^2]$ such that
  - $|S_i| = m$, for all $1 \leq i \leq k$ ✓
  - $i \neq j \Rightarrow |S_i \cap S_j| \leq \log(n)$
- relaxed notion of combinatorial designs
- Construction:
  1. Assume that $m = p$ is a prime
  2. Then, $\mathbb{F}_p^2 \sim [m^2]$
  3. Let $q_1, \ldots, q_n \in \mathbb{F}_p^{\log(n)}$ be polynomials of degree $< \log(n)$.
  4. $S_i = \{(a, q_i(a)) \mid a \in \mathbb{F}_p\}$ $\Rightarrow |S_i| = |\mathbb{F}_p| = p$

Lower Bounds imply Derandomization

- Use Nisan-Wigderson designs:
  - $n \leq 2^m$ integers
  - There exist $S_1, \ldots, S_n \subset [m^2]$ such that
  - $|S_i| = m$, for all $1 \leq i \leq k$
  - $i \neq j \Rightarrow |S_i \cap S_j| \leq \log(n)$
- relaxed notion of combinatorial designs
- Construction:
  1. Assume that $m = p$ is a prime
  2. Then, $\mathbb{F}_p^2 \sim [m^2]$
  3. Let $q_1, \ldots, q_n \in \mathbb{F}_p^{\log(n)}$ be polynomials of degree $< \log(n)$.
  4. $S_i = \{(a, q_i(a)) \mid a \in \mathbb{F}_p\}$
  5. $(a, y) \in \underbrace{S_i \cap S_j}_{} \Leftrightarrow \underbrace{q_i(a)}_{} = \underbrace{q_j(a)}_{} = y$

$(a, y) = (a, q_i(a)) = (a, q_j(a))$

Lower Bounds imply Derandomization

- Use Nisan-Wigderson designs:
  - $n \leq 2^m$ integers
  - There exist $S_1, \ldots, S_n \subset [m^2]$ such that
  - $|S_i| = m$, for all $1 \leq i \leq k$ ✓
  - $i \neq j \Rightarrow |S_i \cap S_j| \leq \log(n)$ ✓
- relaxed notion of combinatorial designs
- Construction:
  1. Assume that $m = p$ is a prime
  2. Then, $\mathbb{F}_p^2 \sim [m^2]$
  3. Let $q_1, \ldots, q_n \in \mathbb{F}_p^{\log(n)}$ be polynomials of degree $< \log(n)$.
  4. $S_i = \{(a, q_i(a)) \mid a \in \mathbb{F}_p\}$
  5. $(a, y) \in S_i \cap S_j \Leftrightarrow q_i(a) = q_j(a) = y$
  6. $|S_i \cap S_j| \leq \deg(q_i) < \log(n)$

$\deg(q_i - q_j) < \log n$

at most $\log n$ roots

$(q_i - q_j)(a) = 0$

Lower Bounds imply Derandomization

- Assume Per_n cannot be computed by circuits of size $\leq 2^{cn}$

Lower Bounds imply Derandomization

- Assume Per_n cannot be computed by circuits of size $\leq 2^{cn}$
- Take NW-design with $m = \log^4 n$ $(n < 2^m)$
  - $S_1, \ldots, S_n \subset [m^2]$
  - $|S_i| = m$ and $|S_i \cap S_j| \leq \log n$

Lower Bounds imply Derandomization

- Assume Per_n cannot be computed by circuits of size $\leq 2^{cn}$
- Take NW-design with $m = \log^4 n$
  - $S_1, \ldots, S_n \subset [m^2]$
  - $|S_i| = m$ and $|S_i \cap S_j| \leq \log n$
- Hitting set generator: $\mathcal{G} = (g_1, \ldots, g_n) : \mathbb{F}^{m^2} \to \mathbb{F}^n$

poly log(n) ↓

quasiP alg. PIT

$\log^2 n = \sqrt{m}$

$$g_i(y_{S_i}) = \boxed{\mathrm{Per}_{\log^2 n}(y_{S_i})}$$

depends only on the variables in S_i

$g_i(y_{S_i})$ poly in $\log^4 n$ variables
$\log^2 n$ deg.

Can have quasi-P many monomials
(write it in sparse representation)

Lower Bounds imply Derandomization

- Assume Per_n cannot be computed by circuits of size $\leq 2^{cn}$
- Take NW-design with $m = \log^4 n$
  - $S_1, \dots, S_n \subset [m^2]$
  - $|S_i| = m$ and $|S_i \cap S_j| \leq \log n$
- Hitting set generator: $\mathcal{G} = (g_1, \dots, g_n) : \mathbb{F}^{m^2} \to \mathbb{F}^n$

$$g_i(y_{S_i}) = \mathrm{Per}_{\log^2 n}(y_{S_i}) \qquad S(\Phi) = n^c$$

$\Rightarrow$

- For any $\Phi \in \mathrm{VP}$ we have $\Phi \equiv 0 \Leftrightarrow \Phi \circ \mathcal{G} \equiv 0$

($\Leftarrow$)

1. Suppose $\Phi \not\equiv 0$ but $\Phi \circ \mathcal{G} \equiv 0$ — Hybrid argument
2. There is index $k \in [n]$ such that $\Phi(g_1, \dots, g_k, x_{k+1}, \dots, x_n) \not\equiv 0$ but $\Phi(g_1, \dots, g_k, g_{k+1}, x_{k+2}, \dots, x_n) \equiv 0$

$x_{k+1} \leftarrow g_{k+1}$

$\boxed{x_{k+1} - g_{k+1}}$ root of $\Phi(g_1, \dots, g_k, x_{k+1}, \dots, x_n)$

Lower Bounds imply Derandomization

- Assume Per_n cannot be computed by circuits of size $\leq 2^{cn}$
- Take NW-design with $m = \log^4 n$
  - $S_1, \ldots, S_n \subset [m^2]$
  - $|S_i| = m$ and $|S_i \cap S_j| \leq \log n$
- Hitting set generator: $\mathcal{G} = (g_1, \ldots, g_n) : \mathbb{F}^{m^2} \to \mathbb{F}^n$

$$g_i(y_{S_i}) = \mathrm{Per}_{\log^2 n}(y_{S_i})$$

- For any $\Phi \in \mathrm{VP}$ we have $\Phi \equiv 0 \Leftrightarrow \Phi \circ \mathcal{G} \equiv 0$
  1. Suppose $\Phi \not\equiv 0$ but $\Phi \circ \mathcal{G} \equiv 0$
  2. There is index $k \in [n]$ such that $\Phi(g_1, \ldots, g_k, x_{k+1}, \ldots, x_n) \not\equiv 0$ but $\Phi(g_1, \ldots, g_k, g_{k+1}, x_{k+2}, \ldots, x_n) \equiv 0$
  3. $x_{k+1} - g_{k+1}$ divides $\Phi(g_1, \ldots, g_k, \cancel{g_{k+1}}, x_{k+2}, \ldots, x_n)$

x_{k+1}

because g_{k+1} root .

x_{k+1} , $y_{S_{k+1}}$ important variables

Lower Bounds imply Derandomization

- Assume Per_n cannot be computed by circuits of size $\leq 2^{cn}$
- Take NW-design with $m = \log^4 n$
  - $S_1, \dots, S_n \subset [m^2]$
  - $|S_i| = m$ and $|S_i \cap S_j| \leq \log n$
- Hitting set generator: $\mathcal{G} = (g_1, \dots, g_n) : \mathbb{F}^{m^2} \to \mathbb{F}^n$

$$g_i(y_{S_i}) = \mathrm{Per}_{\log^2 n}(y_{S_i})$$

- For any $\Phi \in \mathrm{VP}$ we have $\Phi \equiv 0 \Leftrightarrow \Phi \circ \mathcal{G} \equiv 0$
  1. Suppose $\Phi \not\equiv 0$ but $\Phi \circ \mathcal{G} \equiv 0$
  2. There is index $k \in [n]$ such that $\Phi(g_1, \dots, g_k, x_{k+1}, \dots, x_n) \not\equiv 0$ but $\Phi(g_1, \dots, g_k, g_{k+1}, x_{k+2}, \dots, x_n) \equiv 0$
  3. $x_{k+1} - g_{k+1}$ divides $\Phi(g_1, \dots, g_k, g_{k+1}, x_{k+2}, \dots, x_n)$
  4. Set variables $\underbrace{x_{k+2}, \dots, x_n}_{\text{other x vars}}$, and $\underbrace{y_j \in [m^2] \setminus S_{k+1}}_{\text{other y variables}}$ to random values

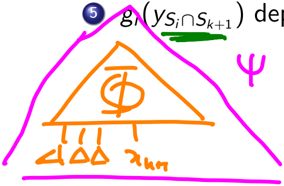
$x_{k+1} - g_{k+1}$ unchanged Φ remain nonzero (still divisible by $x_{k+1} - g_{k+1}$)

Lower Bounds imply Derandomization

- Assume Per_n cannot be computed by circuits of size $\leq 2^{cn}$
- Take NW-design with $m = \log^4 n$
  - $S_1, \ldots, S_n \subset [m^2]$
  - $|S_i| = m$ and $|S_i \cap S_j| \leq \log n$
- Hitting set generator: $\mathcal{G} = (g_1, \ldots, g_n) : \mathbb{F}^{m^2} \to \mathbb{F}^n$

$$g_i(y_{S_i}) = \mathrm{Per}_{\log^2 n}(y_{S_i})$$

- For any $\Phi \in \mathrm{VP}$ we have $\Phi \equiv 0 \Leftrightarrow \Phi \circ \mathcal{G} \equiv 0$
  1. Suppose $\Phi \not\equiv 0$ but $\Phi \circ \mathcal{G} \equiv 0$
  2. There is index $k \in [n]$ such that $\Phi(g_1, \ldots, g_k, x_{k+1}, \ldots, x_n) \not\equiv 0$ but $\Phi(g_1, \ldots, g_k, g_{k+1}, x_{k+2}, \ldots, x_n) \equiv 0$
  3. $x_{k+1} - g_{k+1}$ divides $\Phi(\underline{g_1}, \ldots, \underline{g_k}, \overset{x_{k+1}}{\cancel{g_{k+1}}}, x_{k+2}, \ldots, x_n)$
  4. Set variables $x_{k+2}, \ldots, x_n$, and $y_j \in [m^2] \setminus S_{k+1}$ to random values
  5. $g_i(\underline{y_{S_i \cap S_{k+1}}})$ depends only on $\log n$ variables, so poly-size circuit!



$|S_i \cap S_{k+1}|$

$\log^2 n$ deg.

$s(\Psi) = n^c$

Lower Bounds imply Derandomization

- Assume Per_n cannot be computed by circuits of size $\leq 2^{cn}$
- Take NW-design with $m = \log^4 n$
  - $S_1, \ldots, S_n \subset [m^2]$
  - $|S_i| = m$ and $|S_i \cap S_j| \leq \log n$
- Hitting set generator: $\mathcal{G} = (g_1, \ldots, g_n) : \mathbb{F}^{m^2} \to \mathbb{F}^n$

$$g_i(y_{S_i}) = \mathrm{Per}_{\log^2 n}(y_{S_i})$$

any factor of Φ has ckt size $n^{\tilde{c}}$

- For any $\Phi \in \mathrm{VP}$ we have $\Phi \equiv 0 \Leftrightarrow \Phi \circ \mathcal{G} \equiv 0$
  1. Suppose $\Phi \not\equiv 0$ but $\Phi \circ \mathcal{G} \equiv 0$
  2. There is index $k \in [n]$ such that $\Phi(g_1, \ldots, g_k, x_{k+1}, \ldots, x_n) \not\equiv 0$ but $\Phi(g_1, \ldots, g_k, g_{k+1}, x_{k+2}, \ldots, x_n) \equiv 0$
  3. $x_{k+1} - g_{k+1}$ divides $\Phi(g_1, \ldots, g_k, g_{k+1}, x_{k+2}, \ldots, x_n)$
  4. Set variables $x_{k+2}, \ldots, x_n$, and $y_j \in [m^2] \setminus S_{k+1}$ to random values
  5. $g_i(y_{S_i \cap S_{k+1}})$ depends only on $\log n$ variables, so poly-size circuit!
  6. By Kaltofen, VP is closed under taking factors
  7. Implies $x_{k+1} - g_{k+1}$ has poly size circuit!

Φ poly deg poly size ckts $\Rightarrow$ any factor of Φ also has poly-size ckts

Lower Bounds imply Derandomization

- Assume Per_n cannot be computed by circuits of size $\leq 2^{cn}$
- Take NW-design with $m = \log^4 n$
  - $S_1, \ldots, S_n \subset [m^2]$
  - $|S_i| = m$ and $|S_i \cap S_j| \leq \log n$
- Hitting set generator: $\mathcal{G} = (g_1, \ldots, g_n) : \mathbb{F}^{m^2} \to \mathbb{F}^n$

$$g_i(y_{S_i}) = \boxed{\mathrm{Per}_{\log^2 n}(y_{S_i})}$$

- For any $\Phi \in \mathrm{VP}$ we have $\boxed{\Phi \equiv 0 \Leftrightarrow \Phi \circ \mathcal{G} \equiv 0}$
  1. Suppose $\Phi \not\equiv 0$ but $\Phi \circ \mathcal{G} \equiv 0$
  2. There is index $k \in [n]$ such that $\Phi(g_1, \ldots, g_k, x_{k+1}, \ldots, x_n) \not\equiv 0$ but $\Phi(g_1, \ldots, g_k, g_{k+1}, x_{k+2}, \ldots, x_n) \equiv 0$
  3. $x_{k+1} - g_{k+1}$ divides $\Phi(g_1, \ldots, g_k, g_{k+1}, x_{k+2}, \ldots, x_n)$
  4. Set variables $x_{k+2}, \ldots, x_n$, and $y_j \in [m^2] \setminus S_{k+1}$ to random values
  5. $g_i(y_{S_i \cap S_{k+1}})$ depends only on $\log n$ variables, so poly-size circuit!
  6. By Kaltofen, VP is closed under taking factors
  7. Implies $\underline{y - g_{k+1}}$ has poly size circuit! $S(g_{k+1}) \leq n^c = 2^{c \log n}$
  8. Contradicts fact that $\mathrm{Per}_{\log^2 n}$ cannot be computed by $2^{c \log^{2} n} = n^{c \log n}$ size

Lower Bounds imply Derandomization

$x_{k+1} - g_{k+1} \mid \Phi(g_1, \ldots, g_k, x_{k+1}, \ldots, x_n)$

set $x_{k+2}, \ldots, x_n$ $\left.\begin{matrix} \\ \\ y_i \in [m^2] \setminus S_{k+1} \end{matrix}\right\}$ random field elements

$g_i(y_{S_i}) \xrightarrow[\text{restriction}]{\text{after}} g_i(\underbrace{y_{S_i \cap S_{k+1}}}_{\substack{\text{alive vars} \\ \leq \log n}})$

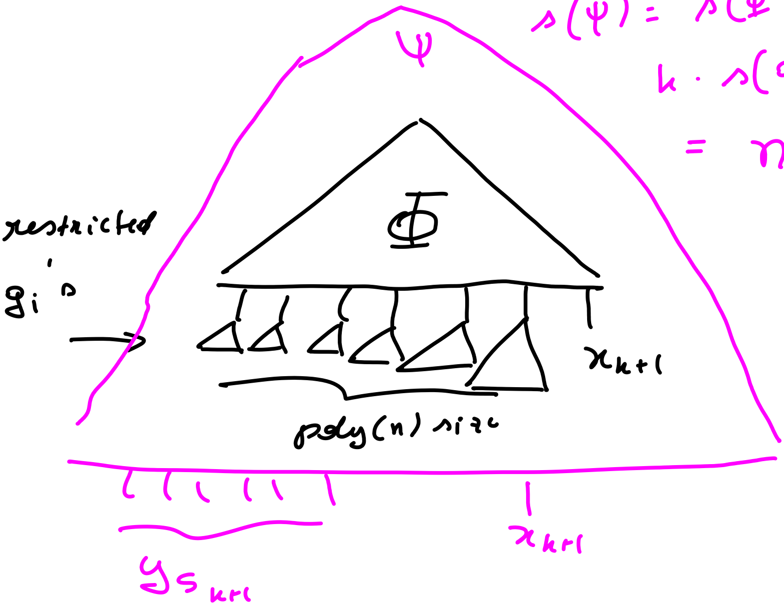
g_i poly in $\log n$ vars
~~deg $\log^2 n$~~
multilinear

only monomials that are left $\boxed{\prod_{\ell \in T} y_\ell}$ $T \subset [S_i \cap S_{k+1}]$

have $\leq 2^{c \log n} = n^c$ monomials

Lower Bounds imply Derandomization

$$s(\Psi) = s(\Phi) + k \cdot s(g_i) = n^c$$



Fast Parallel Algorithms for Matching

Fast Parallel Algorithms for Matching

Sparse Polynomials - Klivans-Spielman

- **Input:** oracle (black-box) access to a polynomial $p(x_1, \ldots, x_n)$ with $\leq s$ monomials and degree d $(n, s, d$ given to you)
- **Output:** is $p(x_1, \ldots, x_n) \equiv 0$?

poly(n, s, d)

$\Sigma \Pi$

Sparse Polynomials - Klivans-Spielman

- **Input:** oracle (black-box) access to a polynomial $p(x_1, \ldots, x_n)$ with $\leq s$ monomials and degree d (n, s, d given to you)
- **Output:** is $p(x_1, \ldots, x_n) \equiv 0$?
- First idea: Kronecker substitution

preserves all monomials

$$P\left(y^{d+1}, y^{(d+1)^2}, \ldots, y^{(d+1)^n}\right) \not\equiv 0$$

$$\Updownarrow$$

$$P(x_1, \ldots, x_n) \not\equiv 0$$

"base $d+1$"

$e_i \leq d$

$x_1^{e_1} \cdots x_n^{e_n}$

$$P(\bar{x}) = \sum p_{\bar{e}} \cdot \bar{x}^{\bar{e}}$$

$$x_i \mapsto y^{(d+1)^i}$$

base $d+1$

$$(e_1, \ldots, e_n) \longrightarrow \sum_{i=1}^{n} e_i (d+1)^i$$

$(a_1, \ldots, a_n)$

Sparse Polynomials - Klivans-Spielman

- **Input:** oracle (black-box) access to a polynomial $p(x_1, \ldots, x_n)$ with $\leq s$ monomials and degree d (n, s, d given to you)
- **Output:** is $p(x_1, \ldots, x_n) \equiv 0$?
- First idea: Kronecker substitution
- Problem is that the degree is really high. How to fix it?

Sparse Polynomials - Klivans-Spielman

- **Input:** oracle (black-box) access to a polynomial $p(x_1, \ldots, x_n)$ with $\leq s$ monomials and degree d (n, s, d given to you)
- **Output:** is $p(x_1, \ldots, x_n) \equiv 0$?
- First idea: Kronecker substitution
- Problem is that the degree is really high. How to fix it?
- Let $p \in \mathbb{Z}$ be a prime. Make substitution:

$$x_i \to y^{\underline{(d+1)^i \mod p}}$$

deg var $\leq p$

- Now degrees are under control. But how to preserve non-zeroness?

$\exists$ monomials can be mapped to the same univariate monomial y

Sparse Polynomials - Klivans-Spielman

- **Input:** oracle (black-box) access to a polynomial $p(x_1, \ldots, x_n)$ with $\leq s$ monomials and degree d (n, s, d given to you)
- **Output:** is $p(x_1, \ldots, x_n) \equiv 0$?
- First idea: Kronecker substitution
- Problem is that the degree is really high. How to fix it?
- Let $p \in \mathbb{Z}$ be a prime. Make substitution:

$$x_i \to y^{(d+1)^i} \mod p$$

- Now degrees are under control. But how to preserve non-zeroness?
- Chinese Remaindering Theorem!
  1. If two monomials $(a_1, \ldots, a_n)$ and $(b_1, \ldots, b_n)$ are distinct and degree $\leq d$, then

     $$a_1 + a_2(d+1) + \cdots + a_n(d+1)^n \neq b_1 + b_2(d+1) + \cdots + b_n(d+1)^n$$

  2. Thus if we take $p_1, \ldots, p_{nd}$ primes, one of the differences $\mod p_i$ will be non-zero

poly many primes

Pick enough primes and do union bound

Sparse Polynomials - Klivans-Spielman

$p_1, \ldots, p_m$ $\quad m = \text{poly}(n, s, d)$

there is one which preserves all s monomials

$P\left(y^{d+1 \bmod p_i}, \ldots, y^{(d+1)^n \bmod p_i}\right) \not\equiv 0$

and has $\leq p_i d$ roots

test univariate poly over $\{0, \ldots, p_i d\}$

Sparse Polynomials - Klivans-Spielman

Conclusion

- Today we learned about word problems and their importance
- Polynomial Identity Testing (PIT)
- Hardness versus randomness
- Application of PIT in TCS (parallel algorithms for matching)
- deterministic PIT algorithm for sparse polynomials

Acknowledgement

- Lecture based largely on:
  - Survey [Shpilka & Yehudayoff 2010, Chapter 4]

References I

Shpilka, Amir and Yehudayoff, Amir 2010.
Arithmetic circuits: a survey of recent results and open questions
Foundations and Trends in Theoretical Computer Science

Kabanets, Valentine and Impagliazzo, Russel 2004.
Derandomizing polynomial identity tests means proving circuit lower bounds
Computational Complexity