Recent Results on Polynomial Identity Testing

Amir Shpilka
Technion

Goal of talk

- Survey known results
- Explain proof techniques
- Give an interesting set of `accessible' open questions

Talk outline

- Definition of the problem
- Connection to lower bounds (hardness)
  - Kabanets-Implagliazzo
  - Dvir-S-Yehudayoff
  - Heintz-Schnorr, Agrawal
  - Agrawal-Vinay
- Survey of positive results
- Some proofs:
  - Sparse polynomials
  - Partial derivatives technique
  - Depth-3 circuits
  - Depth-4 circuits
  - Read-Once formulas
- Connection to polynomial factorization

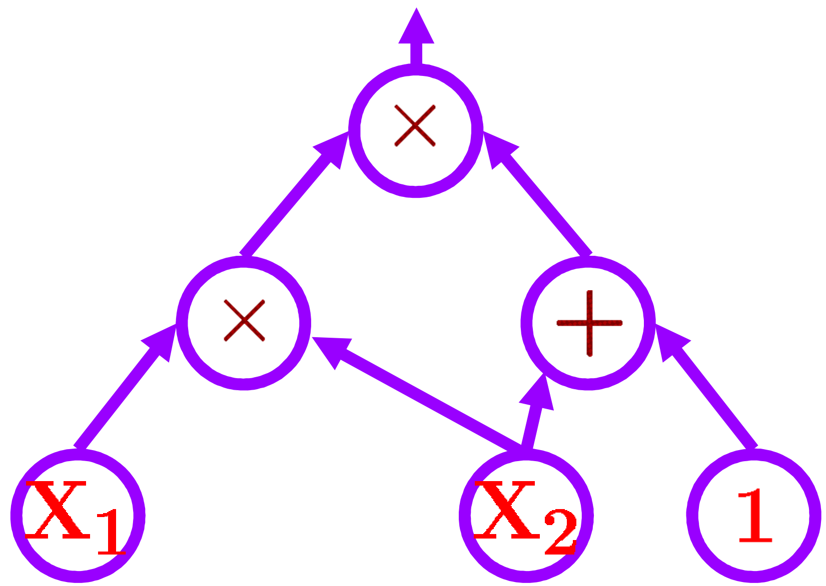
Arithmetic Circuits

Field: $\mathbb{F}$

Variables: $X_1,\ldots,X_n$

Gates: $+$, $\times$

Every gate in the circuit computes a polynomial in $\mathbb{F}[X_1,\ldots,X_n]$



Example: $(X_1 \cdot X_2) \cdot (X_2 + 1)$

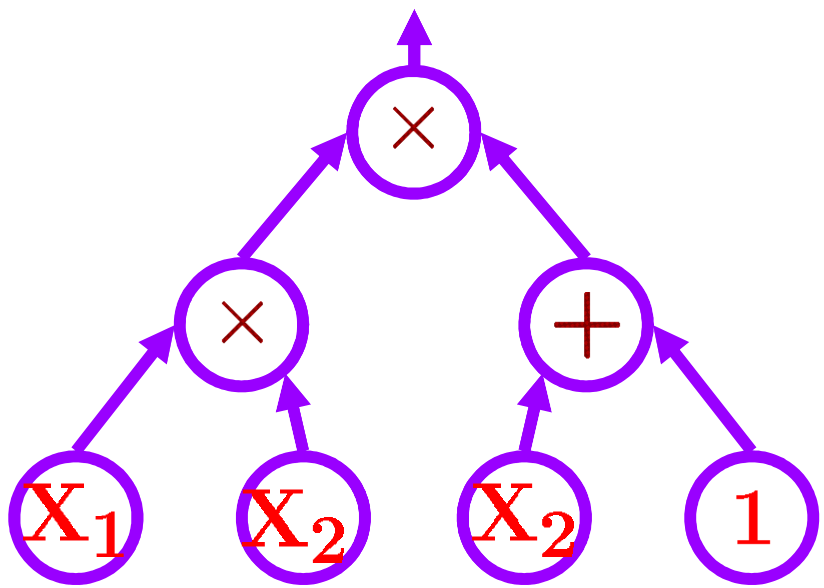
Size = number of gates

Depth = length of longest input-output path

Degree = max degree of internal gates

Arithmetic Formulas

Same, except underlying graph is a tree



Bounded depth circuits

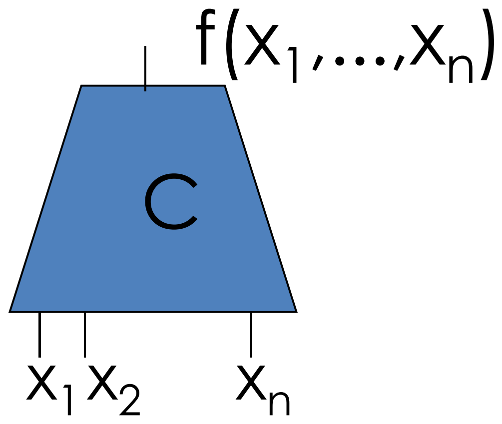
- $\Sigma\Pi$ circuits: depth-2 circuits with $+$ at the top and $\times$ at the bottom. Size s circuits compute s-sparse polynomials.
- $\Sigma\Pi\Sigma$ circuits: depth-3 circuits with $+$ at the top, $\times$ at the middle and $+$ at the bottom. Compute sums of products of linear functions. I.e. a sparse polynomial composed with a linear transformation.
- $\Sigma\Pi\Sigma\Pi$ circuits: depth-4 circuits. Compute sums of products of sparse polynomials.

Why Arithmetic Circuits?

- Most natural model for computing polynomials
- For many problems (e.g. Matrix Multiplication, Det) best algorithm is an arithmetic circuit
- Great algorithmic achievements:
  - Fourier Transform
  - Matrix Multiplication
  - Polynomial Factorization
- Structured model (compared to Boolean circuits) $\mathbb{P}$ vs. $\mathbb{NP}$ may be easier

Polynomial Identity Testing

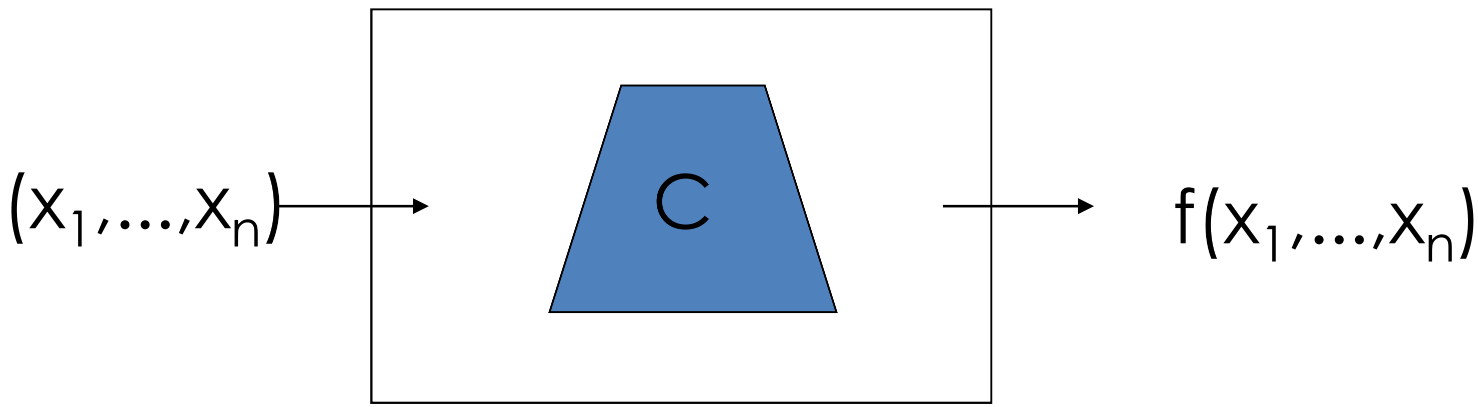
Input: Arithmetic circuit computing f
Problem: Does $f \equiv 0$?



Randomized algorithm [Schwartz, Zippel, DeMillo-Lipton]: evaluate f at a random point
Goal: deterministic algorithm

Black Box PIT ≡ Explicit Hitting Set

Input: A Black-Box circuit computing f.
Problem: Does f=0 ?



Goal: deterministic algorithm (a.k.a. Hitting Set)
S,Z,DM-L: ∃ small Hitting Set (not explicit)

Motivation

- Natural and fundamental problem
- Strong connection to circuit lower bounds
- Algorithmic importance:
  - Primality testing [Agrawal-Kayal-Saxena]
  - Parallel algorithms for finding matching [Karp-Upfal-Wigderson, Mulmuley-Vazirani-Vazirani]

Polynomial Identity Testing

- ✓ Definition of the problem
- Connection to lower bounds (hardness)
  - Kabanets-Implagliazzo
  - Dvir-S-Yehudayoff
  - Heintz-Schnorr, Agrawal
  - Agrawal-Vinay
- Survey of positive results
- Some proofs:
  - Sparse polynomials
  - Partial derivatives technique
  - Depth-3 circuits
  - Read-Once formulas
- Connection to polynomial factorization

Hardness: PIT $\equiv$ lower bounds

[Kabanets-Impagliazzo]:

- $2^{\Omega(n)}$ lower bound for Permanent $\Rightarrow$ PIT in $n^{\text{polylog}(n)}$ time
- PIT $\in$ P $\Rightarrow$ super-polynomial lower bounds: Boolean for **NEXP**, or arithmetic for Permanent

[Dvir-S-Yehudayoff]: (almost) same as K-I for bounded depth circuits

[Heintz-Schnorr,Agrawal]: Polynomial time Black-Box PIT $\Rightarrow$ Exponential lower bounds for arithmetic circuits

Lesson: derandomizing PIT essentially equivalent to proving lower bounds for arithmetic circuit

Non Black-Box P.I.T. $\Leftrightarrow$ Lower Bounds [K-I]

[Valiant,Toda,Impagliazzo-Kabanets-Wigderson]:

$\mathbf{NEXP} \subseteq \mathbf{P}/\text{Poly} \Rightarrow$ Perm is $\mathbf{NEXP}$-complete

K-I: Perm has poly size arith. circuit $\Rightarrow$ Perm in $\mathbf{NP}^{\text{PIT}}$

Idea: guess circuit for Perm. verify correctness using self reducibility and PIT.

$\Rightarrow$: If $\mathbf{NEXP} \subseteq \mathbf{P}/\text{Poly}$ and Perm has poly size circuits and PIT in $\mathbf{P}$ then $\mathbf{NEXP}$ in $\mathbf{NP}$ $\Rightarrow\Leftarrow$

Other direction follows by using arithmetic version of N-W generator and Kaltofen's factorization theorem.

Black-Box P.I.T. $\Rightarrow$ Lower Bounds

[Heintz-Schnorr,Agrawal]: Black-Box P.I.T for size s circuits in time poly(s) (i.e. poly(s) size hitting set) implies exponential lower bounds for arithmetic circuits:

Given $H=\{p_i\}$, find non-zero $\log(|H|)+1$-variate polynomial f such that $f(p_i)=0$ for all i.

$\Rightarrow$ f does not have size s circuits

Gives lower bounds for f in **PSPACE**

Conjecture [Agrawal]:
$H=\{(y_1,\ldots, y_n) : y_i=y^{k^i \bmod r}, k,r < s^{20}\}$ is a hitting set for size s circuits

Importance of $\Sigma\Pi\Sigma\Pi$ circuits

[Agrawal-Vinay,Raz]: Exponential lower bounds for $\Sigma\Pi\Sigma\Pi$ circuits imply exponential lower bounds for general circuits.

Proof: 1. Depth reduction a-la $\mathbb{P}=\mathbb{NC}^2$ [Valiant-Skyum-Berkowitz-Rackoff] 2. Break the circuit in the middle and interpolate each part using $\Sigma\Pi$ circuits.

Cor [Agrawal-Vinay]: Polynomial time PIT of $\Sigma\Pi\Sigma\Pi$ circuits gives quasi-polynomial time PIT for general circuits.

Proof: By [Heintz-Schnorr,Agrawal] polynomial time PIT $\Rightarrow$ exponential lower bounds for $\Sigma\Pi\Sigma\Pi$ circuits. [Agrawal-Vinay] $\Rightarrow$ exponential lower bounds for general circuits. Now use [K-I].

Polynomial Identity Testing

- ✓ Definition of the problem
- ✓ Connection to lower bounds (hardness)
  - ✓ Kabanets-Implagliazzo
  - ✓ Dvir-S-Yehudayoff
  - ✓ Heintz-Schnorr, Agrawal
  - ✓ Agrawal-Vinay
- Survey of positive results
- Some proofs:
  - Sparse polynomials
  - Partial derivatives technique
  - Depth-3 circuits
  - Read-Once formulas
- Connection to polynomial factorization

Randomized algorithms for PIT

Schwartz,Zippel,DeMillo-Lipton:

Evaluate C at a random input

Gives error-randomness tradeoff

Chen-Kao: trade time for error over $\mathbb{R}$:

$\pi_i = \pm p_{i,1}^{\frac{1}{2}} \dots \pm p_{i,r}^{\frac{1}{2}}$, for different primes, random signs

Then $C \equiv 0$ iff $C(\pi_1,\pi_2,\dots,\pi_n) = 0$

Truncating after t digits gives error $O(1/t)$

Intuition: random conjugate won't vanish mod 2^{-t}

For multilinear polynomials, C-K use n random bits for $1/\text{poly}$ error, S-Z-DM-L use $n\log(n)$ bits for error $\frac{1}{2}$.

Lewin-Vadhan: generalized C-K to finite fields:

irreducible polynomials $\leftrightarrow$ primes,

power series $\leftrightarrow$ square roots. Truncation mod x^t.

Randomized algorithms for PIT

Agrawal-Biswas:

Observe: $C \equiv 0$ iff $C(y, y^D, y^{D^2}, \ldots, y^{D^n}) \equiv 0$

Problem: degree too large

A-B give a "small" set of polynomials $\{f_i(y)\}$ s.t.
$C \equiv 0$ iff $\forall i\ C(y, y^D, y^{D^2}, \ldots, y^{D^n}) \equiv 0 \bmod f_i(y)$

- Similar idea used in primality test of A-K-S
- Uses less random bits than S-Z-DM-L
- Non black-box

Agrawal's conjecture:

$\{(y_1, \ldots, y_n) : y_i = y^{k^i \bmod r}, k, r < s^{20}\}$ is a hitting set for size s circuits

Deterministic algorithms for PIT

- $\Sigma\Pi$ circuits (a.k.a., sparse polys) [BenOr-Tiwari, Grigoriev-Karpinski, Klivans-Spielman,…]
  - Black-Box in polynomial time
- Non-commutative formulas [Raz-S]
  - Non-Black-Box in polynomial time
- $\Sigma\Pi\Sigma(k)$ circuits [Dvir-S,Kayal-Saxena,Arvind-Mukhopadhyay,Karnin-S,Saxena-Seshadri,Kayal-Saraf]
  - Black-Box in quasi-polynomial time*
  - Non-Black-Box in time $n^{O(k)}$
- Sum of k Read-once formulas [S-Volkovich]
  - Black-Box in $n^{O(\log(n)\ +k)}$
  - Non-Black-Box in time $n^{O(k)}$
- Multilinear $\Sigma\Pi\Sigma\Pi(k)$: [Karnin-Mukhopadhyay-S-Volkovich] NEW!
  - Black-Box in quasi-polynomial time

Why study restricted models

- [Agrawal-Vinay] PIT for $\Sigma\Pi\Sigma\Pi$ circuits implies PIT for general depth.
- Gaining insight to more general questions:

  Intuitively: lower bounds imply PIT

  Multilinear formulas: super polynomial bounds [Raz] but no PIT algorithms

  Not even for Depth-3 multilinear formulas!

  Sum of ROFs, depth-3,4 multilinear formulas relaxations of the more general problem
- Interesting results: Structural theorems for $\Sigma\Pi\Sigma(k)$ and $\Sigma\Pi\Sigma\Pi(k)$ circuits.

Polynomial Identity Testing

- ✓ Definition of the problem
- ✓ Connection to lower bounds (hardness)
  - ✓ Kabanets-Implagliazzo
  - ✓ Agrawal
  - ✓ Dvir-S-Yehudayoff
  - ✓ Agrawal-Vinay
- ✓ Survey of positive results
- Some proofs:
  - Sparse polynomials
  - Partial derivatives technique
  - Depth-3 circuits
  - Read-Once formulas
- Connection to polynomial factorization

Proofs – tailored for the model

Proofs usually use `weakness' inherent in model

- Depth 2: few monomials. Substituting y^{a_i} to x_i we can control `collapses' of different monomials.
- Non Commutative formulas: Polynomial has few linearly independent partial derivatives [Nisan]. Keep track of a basis for derivatives to do PIT.
- $\Sigma\Pi\Sigma(k)$: setting a linear function to zero reduces top fan-in. If $k=2$ then multiplication gates must be the same. Calls for induction.
- Multilinear $\Sigma\Pi\Sigma\Pi(k)$: in some sense `combination' of sparse polynomials and multilnear $\Sigma\Pi\Sigma(k)$.
- Read-Once-Formulas: sub formulas of root contain ½ of variables.

Depth 2 ($\Sigma\Pi$) circuits

$f(x_1,\ldots,x_n) = M_1+\ldots+M_m$ sum of m degree d monomials

Idea: replace x_i by y^{a_i} so that all monomials map uniquely, interpolate resulting polynomial.

Problem: a_i–s need to grow fast (gives high degree)

[Klivans-Speilman]: for large prime p, $k \leq p$ set $a_i = k^{i-1} \bmod p$. Evaluate at $np+1$ different y-s.

$x_1^{e_1} \cdot x_2^{e_2} \cdot \ldots \cdot x_n^{e_n}$ mapped to $y\wedge(e_1+e_2k+\ldots+e_nk^{n-1}) = y^{E(k)}$

m monomials define m polynomials $E_1(k), \ldots, E_m(k)$. They are mapped 1-1 if k is not root of any E_i-E_j. Holds for a large fraction of the k's.

Better constructions are known

Non commutative formulas

Special case: set-multilinear depth-3 circuits

$X = X_1 \sqcup X_2 \sqcup \ldots \sqcup X_d$, $X_i = \{x_{i,1}, \ldots, x_{i,n}\}$

Multiplication gate: $M_i = L_{i,1}(X_1) \cdot \ldots \cdot L_{i,d}(X_d)$

$C = M_1 + \ldots + M_s$

Main observation: dimension of partial derivatives of C according to $X_1, \ldots, X_k$ (any k) is at most s (spanned by $L_{i,k+1}(X_{k+1}) \cdot \ldots \cdot L_{i,d}(X_d)$ $i=1 \ldots s$)

Algorithm [Raz-S]: compute a basis for all derivatives according to $X_1, \ldots, X_k$ starting from $k=1$ to $k=d$. $C \equiv 0$ if at the end all basis elements are 0

Same idea also in the general case

Depth-3 circuits ($\Sigma\Pi\Sigma(k)$ circuits)

$L = \Sigma_{t=1\ldots n} a_t \cdot x_t + a_0$, $M_i = \Pi_{j=1\ldots d_i} L_{i,j}$, $C = \Sigma_{i=1\ldots k} M_i$

Definition:

- C simple if no linear function appears in all the M_i-s
- C minimal if no subset of mult. gates sums to zero

Main tool [Dvir S]: If $C \equiv 0$ simple and minimal then $\dim(\mathrm{span}(L_{i,j})) \leq \mathrm{Rank}(k,d) = (\log(d))^{k}$*

Lesson: If $C \equiv 0$ then it is very structured

Non Black-Box Algorithm: find partition to sub-circuits of low dimension (after removal of g.c.d.) and brute force verify that they vanish.

Improved $\mathbf{n^{O(k)}}$ algorithm by [Kayal-Saxena].

Black-Box PIT for $\Sigma\Pi\Sigma(k)$

Black-Box algorithm [Karnin-S]: restrict C to a low dim subspace such that the dimensions of any sub-circuit is not reduced by too much.

Idea: such map preserves structure of C

Claim: $C|_V \equiv 0$ iff $C \equiv 0$

Can find poly set of V-s of dimension Rank(k,d)

Gives: $\mathrm{poly}(n) \cdot d^{O(\mathrm{Rank}(k,d))}$ time algorithm

[Saxena-Seshadri]: finite $\mathbb{F}$, $\mathrm{Rank}(k,d) < k^3 \log(d)$

[Kayal-Saraf]: over $\mathbb{Q},\mathbb{R}$ $\mathrm{Rank}(k,d) < k^k$

Improve [Dvir-S] and [Karnin-S] (plug and play)

To see the proofs come to the PIT session!

Black-Box PIT for multilinear $\Sigma\Pi\Sigma\Pi$ (k) [Karnin-Mukhopadhyay-S-Volkovich]

$C = \Sigma_{i=1\ldots k} M_i$ s.t. $M_i = P_{i1}\cdot\ldots\cdot P_{id}$, P_{ij} is size s multilinear $\Sigma\Pi$ circuit. $P_{i1},\ldots,P_{id}$ variable disjoint

Observe: in each M_i, at most polylog(n) P_{ij}-s have more than n/polylog(n) variables.

$\Rightarrow M_i = A_i \cdot B_i$, A_i = quasi-poly sparse and B_i = product of sparse P_{ij} on n/polylog(n) vars

Claim: $\exists$ polylog(n) vars that after deriavating or substituting zeroes to them, $C' = \Sigma_{i=1\ldots k} B'_i \neq 0$

Proof: each operation reduces by half the number of monomials of some A_i

Black-Box PIT for multilinear $\Sigma\Pi\Sigma\Pi$ (k)

$C' = \Sigma_{i=1...k} B'_i \neq 0$ s.t. B_i = product of sparse, variable disjoint, P_{ij}–s on n/polylog(n) vars

Claim: C' contains non-zero multilinear $\Sigma\Pi\Sigma(k)$ circuit

Proof: randomly fix all vars appearing with x_1…

Can derandomize using PIT for sparse polys.

Conclusion: need a black-box way of \`isolating' polylog(n) variables while applying a sparse-PIT for the remaining vars.

[S-Volkovich]: generator for read-once-formulas having this property.

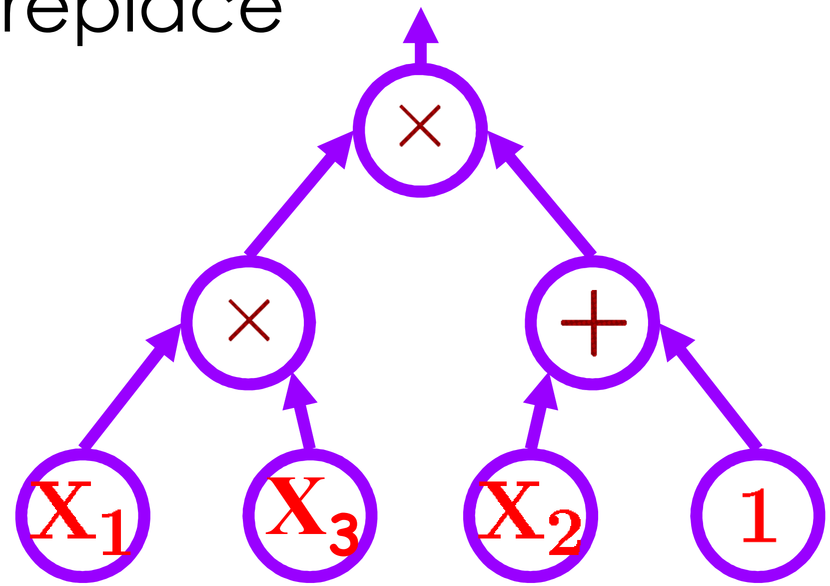
Read-Once formulas (ROFs)

A formula where every variable labels at most one leaf.

Preprocessed ROF: can replace each x_i with $T_i(x_i)$

Sum of ROFs:
$F = F_1 + F_2 + \ldots + F_k$
each F_i is a (P-)ROF

Result: Black-Box PIT for sum of k (P)ROFs in time $n^{O(\log(n) + k)}$



Generator for ROFs

$A = \{a_1, a_2, a_3, \ldots, a_n\} \subseteq \mathbb{F}$

Let $u_i(x)$ be such that $u_i(a_j) = \delta(i,j)$

Def: For every $i \in [n]$ and $k \geq 1$:

$$G^i_k(\mathbf{y},\mathbf{z}) \triangleq u_i(y_1)\cdot z_1 + u_i(y_2)\cdot z_2 + \ldots + u_i(y_k)\cdot z_k$$

$$G_k(\mathbf{y},\mathbf{z}) \triangleq (G^1_k, G^2_k, \ldots, G^n_k)$$

Crucial Property: $G_k|_{(y_k = a_m)} = G_{k-1} + z_k\cdot\bar{e}_m$

$G_k(\mathbf{y},\mathbf{z})$ enables isolation of any $k' \leq k$ variables

In addition, $G_k(\mathbf{y},\mathbf{z})$ is generator for 2^k sparse polynomials (needed for $\Sigma\Pi\Sigma\Pi$ PIT)

PIT for ROFs

Theorem: Let P be a non-zero ROP then $P(G_{\log(n)+1}) \neq 0$. Moreover, if P is a non-constant polynomial then so is $P(G_{\log(n)+1})$

Proof idea: induction on structure of formula. If the top gate is $\times$ then by induction we are ok. If top gate is $+$, then one son has few variables. Can keep a variable that belongs to small son 'alive'.

Sum of ROFs

$F = F_1 + F_2 + \ldots + F_k$

Idea: PIT for ROFs gives a justifying set for any k ROFs of size $n^{O(\log n)}$

Justifying set: contains at least one input $(a_1,\ldots,a_n)$ such that if F_i depends on x_m then $F_i(a_1,\ldots,a_{m-1},x_m,a_{m+1},\ldots,a_n)$ depends on x_m .

By changing $x_i \leftarrow x_i+a_i$ assume that all the F_i-s are **0**-justifyied.

I.e. assigning zeros to all variables but x_i keeps dependence on x_i

Hardness of representation

Hardness of representation: no sum of $k < n/3$ 0-justified ROFs can compute $x_1 \cdot x_2 \cdot \ldots \cdot x_n$

Proof Idea: By induction on k. By taking partial derivatives and making substitutions, can remove some of the ROFs but preserve the structures of F and $x_1 \cdot x_2 \cdot \ldots \cdot x_n$.

Theorem: Let F be a sum of k 0-justified ROFs. Let $\mathcal{A}$ be a set of all vectors in $\{0,1\}^n$ of Hamming weight $\leq k$. Then $F \equiv 0 \Leftrightarrow F|_{\mathcal{A}} \equiv 0$.

Idea: For $n \leq k$ clear. For large n, set $x_i = 0$. Induction implies $x_i \mid F$. Hence $x_1 \cdot x_2 \cdot \ldots \cdot x_n \mid F$ $\Rightarrow\Leftarrow$

Polynomial Identity Testing

- ✓ Definition of the problem
- ✓ Connection to lower bounds (hardness)
  - ✓ Kabanets-Implagliazzo
  - ✓ Dvir-S-Yehudayoff
  - ✓ Heintz-Schnorr, Agrawal
  - ✓ Agrawal-Vinay
- ✓ Survey of positive results
- ✓ Some proofs:
  - ✓ Sparse polynomials
  - ✓ Partial derivatives technique
  - ✓ Depth-3 circuits
  - ✓ Read-Once formulas
- Connection to polynomial factorization

PIT and Factoring

f is *composed* if $f(X) = g(X|_S)\cdot h(X|_T)$ where S and T are disjoint

[S-Volkovich]: PIT is equivalent to factoring to *decomposable* factors.

$\Leftarrow$: $f \equiv 0$ iff $f+y\cdot z$ has two decomposable factors.

$\Rightarrow$: Claim: If we have a (BB or NBB) PIT for all circuits of the form $C_1 + C_2\cdot C_3$, where $C_i \in \mathcal{M}$ then given (BB or NBB) $C \in \mathcal{M}$ we can deterministically output (BB or NBB) all decomposable factors of C.

Decomposable factoring using PIT

Claim: If we have a (BB or NBB) PIT for all circuits of the form $C_1 + C_2 \cdot C_3$, where $C_i \in \mathcal{M}$ then given (BB or NBB) $C \in \mathcal{M}$ we can deterministically output (BB or NBB) all decomposable factors of C.

Idea: Using PIT find a justifying assignment $\mathbf{a}$ for C. Set $x_n = a_n$ and factor (recursively).

Assume $S_1, \ldots, S_k$ is the partition of $[n-1]$.

For every S_i check weather

$$C(\mathbf{a}) \cdot C \equiv C(X_{Si} \leftarrow a_{Si}) \cdot C(X_{[n]\setminus Si} \leftarrow a_{[n]\setminus Si})$$

If yes, add S_i to the partition. At the end put all the remaining vars in a new set.

PIT and factoring

- Deterministic decomposable factoring is equivalent to lower bounds:
  - Deterministic factoring implies $\mathbf{NEXP}$ does not have small arithmetic circuits
  - Lower bounds imply Deterministic decomposable factoring
- PIT $\equiv$ factoring for multilinear polynomials
- Deterministic decomposable factoring for depth-2, $\Sigma\Pi\Sigma(k)$, sum of read-once…
- Open problem: is PIT equivalent to general factorization?

Summary of talk

- ✓ Definition of the problem
- ✓ Connection to lower bounds (hardness)
  - ✓ Kabanets-Implagliazzo
  - ✓ Dvir-S-Yehudayoff
  - ✓ Heintz-Schnorr, Agrawal
  - ✓ Agrawal-Vinay
- ✓ Survey of positive results
- ✓ Some proofs:
  - ✓ Sparse polynomials
  - ✓ Partial derivatives technique
  - ✓ Depth-3 circuits
  - ✓ Read-Once formulas
- ✓ Connection to polynomial factorization

Some `accessible' open problems

1. Give a Black-Box PIT algorithm for non-commutative formulas
2. Solve PIT for depth-3 circuits
3. Solve PIT for multilinear depth-3 circuits
4. Black-Box PIT for set-multilinear depth-3 circuits
5. Polynomial time PIT for (sum of) ROFs
6. P.I.T. for depth-4 with restricted fan-in
7. P.I.T. for read-k formulas (can do it for k=2)
8. Is PIT equivalent to general factorization?

Thank You!