Polynomial Identity Testing

Question: Given two polynomials $f(x)$ and $g(x)$; we want to verify if $f(x) \stackrel{?}{\equiv} g(x)$.

Representation of polynomials? → Sum of monomials (canonical).

→ Given in other bases

→ Give roots of polynomials → $f = \prod_{i=1}^{d}(x-\alpha_i)$ $g = \prod_{i=1}^{d}(x-\beta_i)$

→ Oracle / Blackbox

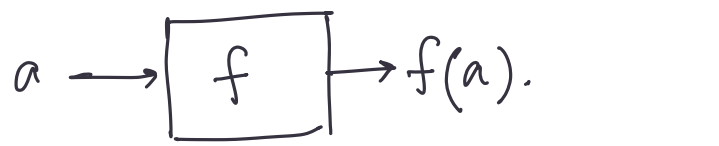
Examples of polynomials:

$$\text{Determinant}(X) = \sum_{\sigma \in S_n} \text{sign}(\sigma) \cdot \prod_{i=1}^{n} x_{i,\sigma(i)}.$$

- $n \times n$ matrix, each of its entries is a distinct variable.
- $\text{sign}(\sigma) = \pm 1$
- S_n is the permutation group on n elements.

Question: Given oracle access, can you do better?

Oracle / Blackbox model



$(a_1, f(a_1)), (a_2, f(a_2)), \ldots, (a_{d+1}, f(a_{d+1}))$

For $g \neq f$,

$\exists\, a_i$ s.t. $f(a_i) \neq g(a_i)$

Fundamental Theorem of Algebra

Any degree d univariate polynomial can have at most d real roots.

→ $\mathbb{C}$: There are exactly d roots.

Interpolating a univariate polynomial:

Given an oracle to $p(x)$, construct its coefficients.

(We are told that $p(x)$ has degree at most d)

Let $p(x) = \sum_{i=0}^{d} c_i \cdot x^i$.

Algorithm:

For $d+1$ non-zero values $a_0, \ldots, a_d$, query for the evaluation of $p(x)$ at $a_0, \ldots, a_d$.

Note that this gives us a full rank linear system.

$$j \in [0,d]: \quad p(a_j) = \sum_{i=0}^{d} c_i \cdot a_j^i$$

$1 \le i,j \le d+1$

$$V_{ij} = a_{i-1}^{j-1}$$

Vandermonde matrix V:

$$\begin{bmatrix} 1 & a_0 & a_0^2 & \cdots & a_0^d \\ 1 & a_1 & a_1^2 & \cdots & a_1^d \\ & & \vdots & & \\ 1 & a_d & a_d^2 & \cdots & a_d^d \end{bmatrix}_{d+1 \times d+1} \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_d \end{bmatrix} = \begin{bmatrix} p(a_0) \\ \vdots \\ p(a_d) \end{bmatrix}_{d+1 \times 1}$$

$$\Rightarrow \bar{C} = V^{-1} \cdot \overline{P(a)}$$

$\rightarrow O(d^3)$ time + query time.

PIT for univariate polynomials

Given $p(x)$ and $q(x)$ as oracles, construct an oracle for the polynomial $r(x) = p(x) - q(x)$.

Degree bound for r is obtained from guarantees for degrees of p and q.

→ Query at $(d+1)$ points.

Check if all evaluations are zero

↳ If YES then assert/return that $f(x) = g(x)$

↳ Else return $f(x) \neq g(x)$.

Question: Can this be extended to multivariate polynomials?

Ex: $x^2 - yz = 0$

Observation: Roots of multivariate polynomials need not be finite.

(Schwartz-Zippel)

Polynomial Zeroes lemma [DeMillo-Lipton-Schwartz Zippel lemma]

$\mathbb{R}$

Let $P(x_1, \ldots, x_n)$ be a non-zero polynomial of degree at most d. Let S be a finite subset of $\mathbb{R}$ s.t $|S| \geq d$. If we assign the values to variables $x_1, \ldots, x_n$ independently and uniformly at random from S, then

$$\Pr_{a_1, \ldots, a_n}\left[P(a_1, \ldots, a_n) = 0\right] \leq \frac{d}{|S|}$$

is independent of n.

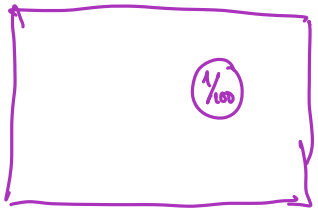
Compare it with fundamental theorem of algebra.

$S = [1, 100d]$

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

Is $f(x) \stackrel{?}{=} g(x)$?

$f(\bar{x}) - g(\bar{x}) \neq 0$



$S \times S \times \cdots \times S$ (n times)

By taking S to be as large gives us a better probabilistic guarantee.

Note that this probability can be brought down by repetitions. (See Lecture 5 notes).

Before seeing the proof, let us see a toy example.

Verifying Matrix Multiplication

Given matrices A, B, C ($n \times n$), we need to verify if $AB = C$.

Can be done in $O(n^\omega)$ time where ω is the matrix mult. exponent. [Alman–Virginia Williams, LeGall]

Question: Can randomness speed this up?

Let S be a finite subset of $\mathbb{R}$. Let $\bar{x} = (x_1, \ldots, x_n)$ be constructed randomly such that each entry is chosen independently and uniformly at random.

Test if $ABx = Cx$. If "YES" Return "$AB = C$". Else return $AB \neq C$.

Runtime? $O(n^2)$

→ First multiply Bx and multiply w/ A: $A(Bx)$

Probability of success:

Note that ABx is a vector whose entries are linear forms in x. Similarly Cx is a vector of linear forms in x.

Let $ABx = (L_1(x), L_2(x), \ldots, L_n(x))^T$ and let Cx be $(L_1'(\bar{x}), L_2'(\bar{x}), \ldots, L_n'(\bar{x}))^T$.

If $AB = C$ then $\forall\, x$, $AB\bar{x} = C\bar{x}$.

$\Rightarrow$ for all $1 \leq i \leq n$, $L_i(\bar{x}) - L_i'(\bar{x}) = 0$.

Else $AB \neq C$ then $\exists x, i$ s.t $L_i(\bar{x}) - L_i'(\bar{x}) \neq 0$.

Claim: If $AB \neq C$ then $\Pr[ABx = Cx] \leq \frac{1}{|S|}$.

Proof: For us a bad event is when we have

$L_i(\bar{x}) - L_i'(\bar{x}) = 0 \quad \forall\, i \in [n]$.

Rephrasing this we just need to show that

$\Pr[\overset{\forall i}{L_i(x) - L_i'(x) = 0}] \leq \frac{1}{|S|}$.

Thus bad event prob

$$\Pr\left[\bigwedge_{i=1}^{n} (L_i(\bar{x}) - L_i'(\bar{x}) = 0)\right] \leq \max_i \left\{\Pr[L_i(\bar{x}) - L_i'(\bar{x}) = 0]\right\}$$

Claim: For any linear polynomial $L(x)$:

$$\Pr_{\bar{a}}[L(\bar{a}) = 0] \leq \frac{1}{|S|} \cdot \frac{|S|^{n-1}}{|S|^n}$$

} Principle of deferred decision.

$L(x) = \sum_{i=1}^{n} b_i x_i$; is $L(x) = 0$ or not.

$b_n x_n = -\sum_{i=1}^{n-1} b_i x_i$

$\frac{1}{|S|}$ fraction of inputs $L(x) = 0$.

Fix $x_1, \ldots, x_{n-1} = a_1, \ldots, a_{n-1}$ $\quad x_n$

$|S|^n$, $|S|^{n-1}$ points that are zeroes of $L(\bar{x})$.

Proof of Polynomial Zeroes lemma:

Proof by induction on n.

Edmond's criterion for Perfect matching.

$G = ([n], [n], E)$

Question: $\exists$ a perfect matching?

Perfect matching (M)

- $\rightarrow M \subseteq E$
- $\rightarrow$ every vertex $\in$ only one edge in M
- $\rightarrow$ edges in M cover all vertices (no isolated vertex).

A PM bipartite graphs is a permutation / bijection

Tutte's matrix (M) $\downarrow$ $n \times n$ matrix

$\{x_{11}, \ldots, x_{nn}\}$.

$1 \le i, j \le n$: $M_{ij} = \begin{cases} x_{ij} & \text{if } \exists \text{ an edge between nodes } i \text{ in } L \text{ and } j \text{ in } R \\ 0 \end{cases}$

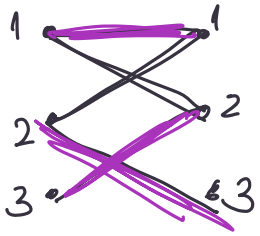
[

A bipartite graph G has a Perfect matching iff $\text{Det}(M) \neq 0$

$$\text{Det}(X) = \sum_{\sigma \in S_n} \text{sign}(\sigma) \prod_{i \in [n]} x_{i\sigma(i)}$$

$e_1, \ldots, e_m$

$G \rightarrow G \setminus \{e\}$ $\quad \bar{e} \in M$



$$M = \begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} x_{11} & x_{12} & 0 \\ x_{21} & 0 & x_{23} \\ 0 & x_{32} & 0 \end{bmatrix} \end{array}$$

$x_{32}\, x_{11}\, x_{23}$

$\text{Det}(M) = -1 \cdot x_{32}(x_{11}\, x_{23} - 0)$ ✓

$$\text{Perm}(M) = \sum_{\sigma \in S_n} \prod_{i \in [n]} x_{i\sigma(i)}$$ } Polynomial

Perm(M) counts the no. of perfect matchings.

$$M = \begin{bmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$ → Perm(M) gives us no. of PMs in G.

$$\underline{HC(G)} = \sum_{\substack{\sigma \in S_n \\ \sigma \text{ is 1-cycle}}} \prod_{i=1}^{n} x_{i\sigma(i)}.$$ }

2