Polynomial Identity Testing

Assume that \( P \) is given as a black box oracle

\[
x \rightarrow P \rightarrow P(x)
\]

Given \( P, Q \), check if \( P(c) = Q(c) \)

Why \( b.o.b. \) ?

\[
P(x) = (x+3)^{38}(x-4)^{83}
\]

[Can expand BUT exponentially many terms]

Define \( R(x) \equiv P(x) - Q(x) \)

Check if \( R(c) = 0 \) ?

Strategy: Try some values of \( x \)
Assume: \( R(\circ) \) has degree \( d \)

\[ R(x) = (x-a_1)(x-a_2)\ldots(x-a_d) \]

**IF** \( R(\circ) \neq 0 \)

then only \( d \) inputs can make it 0

**SO,** try \((d+1)\) inputs 🥰

**Randomized:**

For any set \( S \subseteq F \)

\( S \) contains \( \leq d \) roots

Pick \( x \in S \): 🔄

\[ P[R(x) = 0] \leq \frac{d}{|S|} \]

(Choose \( |S| = 2d \))

Assuming \( R(\circ) \neq 0 \)
Application

Minimize communication

$W \leftrightarrow W^*$

Check $w = w^*$?

$w = w_1 \; w_2 \; w_3 \; \ldots \ldots \; w_n$

$P(n) = \sum_{i=1}^{n} w_i \cdot x^i$

$P^*(x) = \sum_{i=1}^{n} w_i^* \cdot x^i$

Best deterministic algorithm requires $\Omega(n)$ bits (we get $\log n$)

Choose $x \in \mathbb{R} \; [12 \ldots \; 2n] = S$

Repeat $\log(n)$ times

Check if $P(x) = P^*(x)$?
Multi-variate Case

Check if \( R(x, y) = 0 \) \( \forall x, y \)?

Infinitely many roots 😞

Total degree:

\[
\begin{align*}
2x + 3xy^2 & < 1x + 2 = 3 \\
& \text{Take the max: 3}
\end{align*}
\]

\[
\begin{align*}
xyz^3 + x^4 & < 1 + 1 + 3 = 5 \\
& \text{Take max: 5}
\end{align*}
\]

[Schwartz – Zippel] Lemma

\( R(x_1, x_2, \ldots, x_n) \) has t. degree \( d \) over \( F \)

1. Pick \( S \subseteq F \)
2. Choose \( r_1, r_2, r_3, \ldots, r_n \in_R S \)
3. \( \Pr \left[ R(r_1, \ldots, r_n) = 0 \right] \leq \frac{d}{1 + 1} \)

Assuming \( R(0) \neq 0 \)
Proof by induction on $d$

Behavior:

- $R(0) = 0$ Always detected
- $R(0) \neq 0$ Detected w.p. $1 - \frac{d}{1.51}$
Bipartite Matching

\[ |L| = n \quad |R| = n \]

Perfect Matching

\[ M \subseteq E \]

\[ \text{No two edges share endpoint} \]

Each vertex is represented.

\[ \rightarrow \text{Only possible if } |L| = |R| \]

Q: Is there a perfect matching?

A: Can be solved using flows

Q: What if we want to solve in parallel?

(full discussion relegated to pset 2)
**Tutte Matrix:** \( A_a = \{ a_{u,v} \}_{u \in L \setminus V R} \)

\[ a_{u,v} = \begin{cases} 
X_{u,v} & \text{if } (u,v) \in E \\
0 & \text{o.w.}
\end{cases} \]

**Claim:** \( G \) has p.m. iff. \( \det[A_a] \neq 0 \)

**Proof:**

\[ \det[A_a] = \sum \text{sign}(\sigma) \prod_{i=1}^{n} a_{i, \sigma(i)} \]

\( \sigma \) is a permutation of \([n]\)

**Notice:** permutation \( \sigma \) (of \([n]\)) \( \leftrightarrow \) Matching \( M \)

\( i - \sigma(i) \) is an edge in the matching

**If** \( \sigma \) is not a matching, then this is \( \emptyset \)

\( \Rightarrow \) Product is \( \emptyset \)
Also Note: No cancellations!
(All monomials are distinct)

Why do this?
Det. polynomial has \((n!)\) terms
(too many)

- Instead, use [Schwartz–Zippel]

- Det can be computed FAST!
  - \(O(n^w)\) - sequential \((w \approx 2.38\ldots)\)
  - \(O(\log^2 n)\) - parallel w/ poly \((n)\) processors
DNF Sampling (OR of ANDs)

\[ \Phi = (x_1 \land x_2) \lor (\neg x_2 \land x_3) \lor \ldots \]

\[ \Phi = C_1 \lor C_2 \lor \ldots \]

How to Satisfy?

- Pick one clause & satisfy it!

**DNF-SAT \rightarrow easy**

Q: Find random uniformly satisfying assignment

**Strategy:**

- Pick random arbitrary clause \( C_i \)
- Set vars in \( \Phi C_i \) to make it true
- Set other vars randomly.

Define \[ A_i = \{ \bar{x} \in \{0,1\}^n | \bar{x} \text{ satisfies } C_i \} \]
Strategy samples uniformly from $A_i$.

Note: Can compute $|A_i| \rightarrow \frac{1}{2}$ (#vars not in $C_i$)

Idea: 1. Choose $i$ w.o.p. $\frac{|A_i|}{\sum_{j=1}^{m} |A_j|}$

2. Return uniform sample from $A_i$

Problem: (not uniform)

Solution: For a proposed $\overrightarrow{x}$

Let $C_{\overrightarrow{x}} = \{ A_i \ s.t. \ \overrightarrow{x} \in A_i \}$

# of ways $\overrightarrow{x}$ can be returned

3. Return $\overrightarrow{x}$ w.o.p. $\frac{1}{C_{\overrightarrow{x}}}$

Makes prob. of any $\overrightarrow{x}$ the same

Otherwise, try again (w.o.p. $1 - \frac{1}{C_{\overrightarrow{x}}}$)