Lecture 4

Polynomial Identity Testing

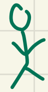
applications to:

- "person on the moon"
- bipartite matching

Polynomial Identity Testing

Is $P(x) = (x+1)^2$ the same as $Q(x) = x^2 + 2x + 1$?

YES!!



What about $P(x) = (x+3)^{38}(x-4)^{83}$

& $Q(x) = (x-4)^{38}(x+3)^{83}$

Obviously not! $P(0) \neq Q(0)$!

Doesn't look like it, but lots of terms to compare!

Problem: given 2 polynomials P, Q

is $P \equiv Q$?

i.e. is $P(x) = Q(x)\ \forall x$?

Problem': given polynomial R

is $R \equiv 0$?

i.e. is $R(x) = 0\ \forall x$?

Let $R(x) = P(x) - Q(x)$ then $R \equiv 0$ iff $P \equiv Q$

Fact: If $R \not\equiv 0$ has degree $\leq d$ then

R has at most d roots* (recall: a "root" is x st. $R(x)=0$)

Algorithm for deciding whether $R \equiv 0$:

pick $d+1$ distinct inputs $x_1 \dots x_{d+1}$
if $\forall i$ $R(x_i)=0$ output "$R \equiv 0$"
else $(\exists i$ st. $R(x_i) \not\equiv 0)$ output "$R \not\equiv 0$"

Runtime: $O(d)$ evaluations of R

* this is true over <u>any</u> field

$\mathbb{Z}$, mod q, ...

prime $> d$

Faster randomized algorithm:

Pick $2d$ distinct inputs $x_1 \cdots x_{2d}$

Do k times:

  Pick $i \in [2d]$, if $R(x_i) \neq 0$ output "$R \not\equiv 0$"

Output "$R \equiv 0$"

Behavior:

if $R \equiv 0$, $\forall x_i$ $R(x_i) = 0$ so always outputs "$R \equiv 0$"

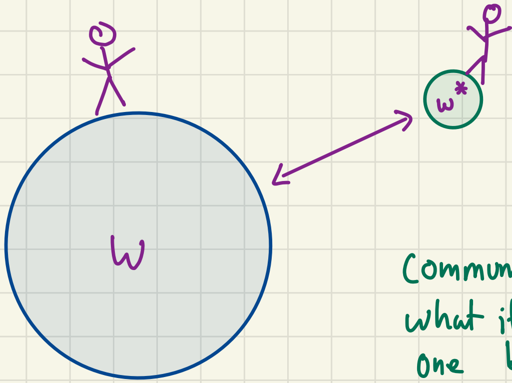
if $R \not\equiv 0$, $\Pr_{i \in [2d]} [\, R(x_i) = 0 \,] \leq \frac{\#\text{ roots}}{2d} \leq 1/2$

$\Pr[\text{err}] = \Pr[\text{ choose root in all } k \text{ iterations}] \leq \frac{1}{2^k}$

$\Rightarrow \Pr[\text{ output "}R \not\equiv 0\text{"}] \geq 1 - \frac{1}{2^k}$

If you are willing to tolerate prob of error $\leq \delta$, pick $k = \log 1/\delta$

Application: "Person-on-the-moon"



Question: Is $w = w^*$?

Communication is expensive!! what if they differ in only one bit?

$w = w_0 \cdots w_n$ (n+1 bit string)

there are lots of primes so q doesn't need to be bigger than $c \cdot n$ — pick q prime $> 2n$

$$\text{Let } P(x) = w_n \cdot x^n + w_{n-1} \cdot x^{n-1} + \ldots + w_1 x + w_0 \mod q$$

$$P^*(x) = w_n^* x^n + w_{n-1}^* x^{n-1} + \ldots + w_1^* x + w_0^* \mod q$$

$$w = w^* \iff P \equiv P^* \quad \text{for } P, P^* \text{ of degree } n$$

$\Theta(n)$ bits of communication

Instead of sending full description of w,

- earthman picks random $r_1 \cdots r_k$ & sends
  $$(r_1, P(r_1)), (r_2, P(r_2)), \ldots (r_k, P(r_k))$$
- man on moon then checks equality

r_i's in $[2n]$ + $p(r_i) \bmod q$ in $[c \cdot n]$ so $O(\log n)$ bits of communication

Multivariate Polynomial Identity Testing

Test if $R(x_1, x_2, \ldots, x_n) \equiv 0$

Total degree: $\max_{s \in \text{terms}} \left(\text{sum. of degrees of } x_i\text{'s in term } s \right)$

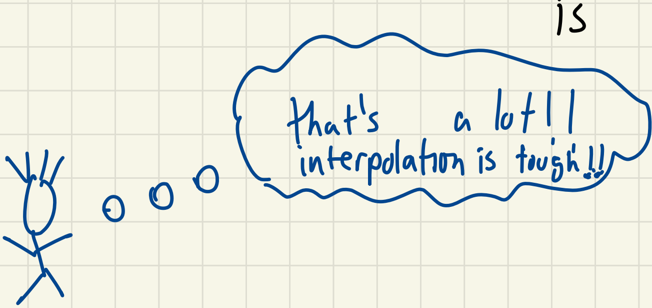
e.g. $\underbrace{2xy}_{\text{deg } 2} + \underbrace{3z^3}_{\text{deg } 3} + \underbrace{4xyz^2}_{\text{deg } 4}$ total deg 4

difficulty 1: $R \not\equiv 0$ can have infinitely many roots

e.g. $R(x,y) = x \cdot y$

$R_2(x,y) = x - y$

difficulty 2: # terms in total degree d poly is $\leq \binom{n}{d}$



Good news!

[Schwartz-Zippel-DeMillo Lipton]

For R of total degree d s.t. $R \not\equiv 0$:

Given S containing $2d$ elements

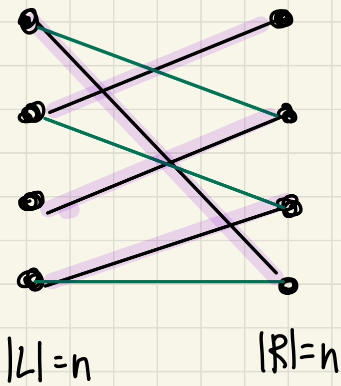
Pick $x_i \in_R S \quad \forall i$ ← Pick $x_1 \ldots x_n$ from "n-dim cube"

Then $\Pr[R(x_1 \ldots x_n) = 0] \leq \frac{d}{|S|}$

Proof induction on d

Application:

Bipartite Perfect Matching



Matching: $M \subseteq E$

no two edges share endpt

Perfect Matching:

$|M| = n$

(all nodes get matched)

Can solve in polytime via flows

Today: another approach!

Note: permutation σ of $[n]$ $\longleftrightarrow$ matching M

$i \rightarrow \sigma(i)$ is edge in matching

main insight:

term drops out if not perfect matching $\rightarrow$ $\prod_{i=1}^{n} a_{i\,\sigma(i)}$ will be 0 if even one of $(i, \sigma(i)) \notin E$

so $\prod_{i=1}^{n} a_{i,\sigma(i)} \neq 0$ iff σ is a matching

so $\underbrace{\mathrm{Det}[A_a] \not\equiv 0}_{\text{Some term remains}}$ iff $\exists$ some σ which is a matching

Det$[A_a]$ is a polynomial!

n^2 vars (1 for each edge)

total degree n

\# terms $n!$ $\leftarrow$ huge!!

Algorithm: Test $\mathrm{Det}[A_a] \not\equiv 0$

also good for parallel algs $\rightarrow$ (need to compute det of integer matrices: $O(n^{\alpha})$ time)