Hashing Polynomials and Algebraic Expressions:

(a) Identity Testing of Polynomials
(b) Applications of Polynomial Hashing
(c) Hashing Classes of Algebraic Expressions

Main Goal of Lecture:

Develop techniques for testing equality of Expressions

test $\varepsilon_1 = \varepsilon_2$?

by using test

hash $\varepsilon_1 = \text{hash } \varepsilon_2$?

Goals:

(1) provable bounds on error probability

(2) applicable to largest possible class of expressions

Reading Selection:

Definitions:

polynomial expression: 1 or any variable, or integer, or \( \alpha + \beta, \alpha - \beta, \alpha \cdot \beta, \) or \( \alpha \uparrow \kappa, \) where \( \alpha, \beta \) are polynomial expressions and \( \kappa \) is a positive integer.

Straight Line Program \( \Pi \): Input \( x_1, \ldots, x_n \)

sequence assignments--

\[
\begin{align*}
\text{length (} \theta \text{)} & \quad \begin{cases} 
  x_{n+1} \leftarrow x_{i_1} \theta_1 x_{j_1} \\
  x_{n+2} \leftarrow x_{i_2} \theta_2 x_{j_2} \\
  \vdots 
\end{cases} \\
\text{output } x_L \text{ where } L = \text{length (} \Pi \text{)}. \\
\text{allow operations } \theta_\kappa \in \{+, -, \cdot, \uparrow\} \\
\Pi(x_1, \ldots, x_n) \text{ denotes output value.}
\end{align*}
\]
Notes:

(1) Given a polynomial expression $\alpha$, can construct a straight-line program of size linear in input polynomial $\alpha$.

(2) A straight-line program $\Pi(x_1,\ldots,x_n)$ will yield a polynomial expression $\alpha_{\Pi}$ with integer coefficients where $\deg(\alpha_{\Pi}) \leq 2^\text{length}(\Pi)$.

If $\Pi(x_1,\ldots,x_n)$ is a program over $\mathbb{Q}$, $|\Pi(x_1,\ldots,x_n)| \leq 2^\text{length}(\Pi)$ can be proved by induction on length $\Pi$.

**basis:** true for case $\text{length}(\Pi) = 0$

**induction step:** if true for $\text{length}(\Pi) = k - 1$ and $\Pi(x_1,\ldots,x_k) = \prod_1(x_1\ldots x_k) \theta_k \prod_2(x_1\ldots x_k)$, then $|\Pi(x_1\ldots x_k)| \leq 2^{\text{length}(\Pi)}$.

Q.E.D.
Let Q be an infinite field.
Let $P(x_1, \ldots, x_n)$ be nonzero polynomial degree d.

**Lemma** If $|A| > d$, then there exist at least $(\kappa - d)^n$ elements $a \in A^n$ such that $P(a) \neq 0$.

**Proof**: By induction on $n$

*Basis*: If $n=1$, then $P$ has $\leq d$ roots in $Q$.

*Induction*: Suppose lemma holds for polynomials with less than $n$ variables. Since $P$ nonzero,

- $\exists (a_1, \ldots, a_{n-1}, c)$ s.t.
  - $P(a_1, \ldots, a_{n-1}, c) \neq 0$.

So by induction hypothesis there exist at least $(\kappa - d)^{n-1}$ such $(a_1, \ldots, a_{n-1}) \in A^{n-1}$ s.t.

- $P(a_1, \ldots, a_{n-1}, c) \neq 0$.

But the $P(x_n) = P(a_1, \ldots, a_{n-1}, x_n)$ is nonzero polynomial with at least $\kappa - d$ elements in $A$ s.t. $P(x_n) \neq 0$. *Lemma follows: Q.E.D.*
This is the key Lemma used to justify hashing polynomials!

**Theorem:** If \( P(x_1 \ldots x_n) \) degree \( d \) in \( Q \),

**Theorem:** If \( \kappa = |A| \geq 2dn \), and \( \alpha \) is a random element of \( A^n \), then

\[
\Pr(P(\alpha) \neq 0) \geq \frac{1}{2}
\]

**Proof:**

\[
\Pr(P(\alpha) \neq 0) = \frac{|\{\alpha: \alpha \in A^n, P(\alpha) \neq 0\}|}{|A^n|}
\]

\[
= \frac{(\kappa - d)^n}{\kappa^n} \quad \text{by Lemma}
\]

\[
= (1 - \frac{d}{\kappa})^n
\]

\[
\geq (1 - \frac{1}{2n})^n \quad \text{since } \kappa \geq 2dn
\]

\[
\geq \left(1 - \frac{1}{2n}\right)^{2n} \geq e^{-1}
\]

\[
\geq e^{-\frac{1}{2}} \quad \text{since } (1 - \frac{1}{2n})^{2n} \geq e^{-1}
\]

\[
\geq \frac{1}{2} \quad \text{since } 2 \geq e^{\frac{1}{2}}
\]

**Q.E.D.**

**Lemma 2:**

If \( \kappa \) is an integer s.t. \( 1 \leq \kappa \leq 2^{2n+2} \),

and \( m \) is randomly chosen from \( \{1, \ldots, 2^n\} \),

then \( \Pr(\kappa \neq 0 \mod m) \geq \frac{1}{4n} \) for \( n \gg 0 \).

**Proof:**

By the prime number theorem, the number of primes less than \( 2^{2n} \)

is at least \( \frac{2^{2n}}{2n} \) for large \( n \).

But \( \kappa \) has at most \( 2n2^n \) prime divisors.

Hence, \( \Pr(\kappa \neq 0 \mod m) \quad \text{(\# primes } \leq 2^{2n} \text{) which don@ divide } \kappa \)

\[
\geq \frac{2^{2n}/2n - 2n2^n}{2^{2n}} \geq \frac{2^{2n}}{4n} \quad \text{Q.E.D.}
\]
Algorithm: Randomized Zero Testing

Input: program \( \pi(x_1, \ldots, x_t) \) length \( r \)

begin
\( n = r + t \)
\[ A = \{1, 2, \ldots, 2t2^r\} \]
for \( i = 1, \ldots, 8n \), do
begin
choose random \( \bar{a} \in A^t \)
choose random \( m \in \{1, \ldots, 2^{2n}\} \)
if \( \pi(\bar{a}) \neq 0 \mod m \),
then return "\( \pi \neq 0 \)"
end
return "\( \pi = 0 \)"
end

Theorem: \( \text{Prob}(\text{correct output}) \geq \frac{1}{2} \)

Proof: If \( \pi \equiv 0 \), then algorithm always correct.

Suppose \( \pi \neq 0 \). By Lemma1,

\( \text{Prob}(\pi(\bar{a}) \neq 0) \geq \frac{1}{2} \). Also, if \( \pi(\bar{a}) \neq 0 \), then

\( \text{Prob}(\pi(\bar{a}) \neq 0 \mod m) \geq \frac{1}{4n} \), so

\( \text{Prob}(\pi(\bar{a}) \neq 0 \mod m) \geq \frac{1}{2} \left( \frac{1}{4n} \right) = \frac{1}{8n} \). Hence,

\( \text{Prob}(\text{correct output}) \geq 1 - \left( 1 - \frac{1}{8n} \right)^{8n} \)

\( \geq 1 - e^{-1} \)

\( \geq \frac{1}{2} \) Q.E.D.
Applications of Polynomial Zero Testing

(1) Given $n \times n$ matrices $A$, $B$, $C$
    problem $A \cdot B = C$?

(2) Given $n$ degree Polynomials $P_1(x)$, $P_2(x)$, $P_3(x)$
    problem $P_1(x) \cdot P_2(x) = P_3(x)$?

(3) Given $n$ bit integers $x_1$, $x_2$, $x_3$
    problem $x_1 \cdot x_2 = x_3$?

(4) Given $n \times n$ Matrix $A$, integer $r$
    problem rank $(A) = r$?

(5) Given graph $G$ of $n$ vertices
    problem does $G$ have perfect matching?

(6) Authentication systems

(7) Testing equality of sets with element addition and deletion operations

Given:
non integer matrices $A,B,C$

Theorem:
Can test $A \cdot B = C$?
in time $O(n^2 \log n)$
with success probability $\geq 1 - \frac{1}{n^c}$, for a constant $c$. 
**Proof:**

Let $K = c \log n$.

Choose $k$ random vectors $\bar{x}_1, \ldots, \bar{x}_k$
each of size $n$, from elements in $\{-1, 1\}$

If $\exists i \in \{1, \ldots, k\}$ s.t. $A(B\bar{x}_i) \neq (C\bar{x}_i)$
then output "$A \cdot B \neq C$"
else output "$A \cdot B = C$"

**Note:** if $A \cdot B = C$, then no errors ever!

**Else:** if $A \cdot B \neq C$, $\forall i \in \{1, \ldots, k\}$

$\text{Prob}(A \cdot (B \cdot \bar{x}) \neq C \bar{x})$
$= \text{Prob}(D\bar{x}_i \neq 0) \text{ where } D = A \cdot B - C \neq 0$
$\geq \frac{1}{2}$ since at most $2^{n-1}$ out of $2^n$
vectors $\bar{x}$ have $D \cdot \bar{x} = 0$ if $D \neq 0$.

So, $\text{Prob}(A \cdot (B \cdot \bar{x}_i) \neq C \bar{x}_i \text{ for } i \in \{1, \ldots, k\})$
$\geq 1 - 2^{-k} = 1 - n^{-\epsilon}$.

**Given Polynomials:** $P_1(x) \cdot P_2(x), P_3(x)$ degree $n$.

**Theorem:** Can test $P_1(x) \cdot P_2(x) = P_3(x)$? in expected $O(n)$ arithmetic steps.

**Proof:** Fix error prob. $\epsilon \in \left(0, \frac{1}{2}\right)$.

Let
$k = \left\lceil \frac{1}{\epsilon} \right\rceil$,

$w = 2^{\lceil \log(kn) \rceil}$

Choose random $x_0 \in \{-w+1, -w+2, \ldots, 0, \ldots, w-1, w\}$

if $P_1(x_0) \cdot P_2(x_0) - P_3(x_0) \neq 0$
then return "$P_1(x) \cdot P_2(x) \neq P_3(x)$"
else "$P_1(x) \cdot P_2(x) = P_3(x)$"

**Note:** If $P_1 \cdot P_2 = P_3$, then never any error!
If $P_1 \cdot P_2 \neq P_3$, then, since the polynomial
$Q = P_1 \cdot P_2 - P_3$ has degree $\leq 2n$,

$\Rightarrow$ error probability $\leq \frac{2n}{2w} = \frac{n}{w} \leq \epsilon$ Q.E.D.
Application to Perfect Matching

Let \( G = (V, E) \) be an undirected graph with vertex set \( V = \{1, \ldots, n\} \).

A perfect matching of \( G \) is a set of \( n \) edges on \( E \) with no common endpoints.

Define \( n \times m \) matrix \( M \) such

\[
M_{ij} = \begin{cases} x_{ij} & \text{if } (i, j) \in E \\ 1 & \text{if } i = j \\ 0 & \text{otherwise} \end{cases}
\]

Let \( x_{ij} = -x_{ji} \) be indeterminate variables.

Lemma (Edmonds): \( G \) has perfect matching iff determinate \((M) \neq 0\).

⇒ Randomized Algorithm for matching test:

[1] Choose each \( x_{ij} \) to be a random integer in \( \{1, \ldots, n^c\} \)

[2] If determinate \((M) = 0\)

then return, "no perfect matching",

else, return, "a perfect matching exists".

Can set \( c > \alpha 3 \) to get error \( < \frac{1}{n^\alpha} \).
Strongly Universal Hash Functions
(Wegman and Carter)

Let $H$ be a set of hash fns $A \rightarrow B$

def: $H$ is strongly universal$_n$ if
\[
\forall a_1 \ldots a_n \in A \quad \forall b_1 \ldots b_n \in B
\]
then \( \frac{|H|}{|B|^n} \) functions in $H$ take $a_i \rightarrow b_i$
for $i = 1, \ldots, n$.

Example: Let $A, B$ be sets in some finite field

Let $H$= class of polynomials degree $n$ of one variable.

Claim: $H$ is strongly universal$_n$.

Proof: Given $a_1, \ldots, a_n$, $b_1, \ldots, b_n$
\( \exists \) exactly one polynomial degree $n$
that interpolates through distinguished pairs $a_i \rightarrow b_i$ for all $i = 1, \ldots, n$.

Q.E.D.
Applications of Polynomial Hashing to Authentication System:

Let $M =$ possible message set
$T =$ authentication tags

1. public knows set functions $H$ from $M \rightarrow T$
2. sender / receiver share secret random $f \in H$
3. sender sends message $m$ in $M$ with authentication tag $f(m)$

   case: $H =$ strongly universal$_2$ set fns $M \rightarrow T$
   = polynomials degree $< |M|$

Claim: unbreakable with prob $\geq 1 - \frac{1}{|T|}$

Proof: If $f$ random fn in $H$ forger must pick correct $fn f$ from $H' = \{ h \in G | f(m) = h(m) \}$ and substitute $m'$ for $m$ s.t. $f(m') = f(m)$, but, by definition of strongly universal$_2$ fns, only $\frac{1}{|T|}$ of fns in $H'$ map $m'$ to $f(m)$. Q.E.D.

Application to Testing Set Equality

Given: set elements $A = \{a_1, \ldots, a_n\}$ and sets $S_1, \ldots, S_m$ initially empty

Operations:
1. add element $a_i$ to set $S_j$
2. delete element $a_i$ from set $S_j$
3. test equality $S_{j_1} = S_{j_2}$?

Implementation:
Use set hash fn $H$, which is strongly universal$_n$ for each $n$.
Each $f \in H$ maps from $A$ to $B$.
assume: $B$ is group with operation $\oplus$ and inverse

Example: Analyze following implementation
(Use variables $V_1, \ldots, V_m$ initially all fixed $b_0 \in B$.)

Operational:

<table>
<thead>
<tr>
<th>Operation</th>
<th>Implementation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S_j \leftarrow S_j \cup {a_i}$</td>
<td>$V_j \leftarrow V_j \oplus f(a_i)$</td>
</tr>
<tr>
<td>$S_j \leftarrow S_j \setminus {a_i}$</td>
<td>$V_j \leftarrow V_j \oplus f(a_i)^{-1}$</td>
</tr>
<tr>
<td>test $S_{j_1} = S_{j_2}$?</td>
<td>test $V_{i_1} = V_{i_2}$?</td>
</tr>
</tbody>
</table>
Hashing Algebraic Expressions

(Gonnet, "Determining Equilibrium of Expressions in Random Polynomial Time", 1984 STOC)

Generalizations:

(1) complex arithmetic expressions

Partial Results:

(2) expressions with roots & rational components
(3) expressions with exponents
(4) expressions with trigonometric fns

Hashing Complex Expressions

Assume $p$ prime $> 2$

Lemma: $\exists i$ s.t. $i^2 = -1 \mod p$, iff $p = 4k + 1$ for some $k$.

Proof: Since any prime $p > 2$ is odd so $(p-1)/2$ is integer.

Let $\alpha$ be generator of mult. group of $Z_p$.
Then $\alpha^{p-1} \equiv 1 \mod p$ and $\alpha^{(p-1)/2} \equiv -1 \mod p$.
Thus $i^2 \equiv \alpha^{(p-1)/2} \equiv -1 \mod p$ if $i = \alpha^k$ where $k = (p-1)/4$. Q.E.D.

Example: For $p = 13$, $i^2 = -1 \mod p$ for $i = 5$.

Then: Can do equivalence testing of complex expressions in random polynomial time.
**Hashing Expressions with Constant Exponents in Finite Fields**

*Expressions:*

\[ E^E \] allow \( E \) to have \(+, -, x, +\) operations.

(Compute \( E \mod p \).)

requires \( E' \) only to have \(+, -\) operations.

(Compute \( E' \mod p - 1 \).)

Since multiplication group in \( \mathbb{Z}_p \) is a cyclic group with one less element than entire group \( \mathbb{Z}_p \).

---

**Hashing Expressions with Square Roots**

*Proposition:*

If \( p = 4nj + 1 \) is prime \( > 2 \),

then \( \sqrt{j} \mod p \) is defined.

---

**Hashing Expressions with Trigonometric Functions**

(no provable method)

*Extensions:*

(Morton)

Can extend construction to find \( e, \pi \) s.t. \( e^{i\pi} = -1 \) for certain primes \( p \).

*Open Problem:*

\( \Rightarrow \) get a provable method for identity testing of trigonometric functions \( \sin(x), \cos(x) \), etc.

*Idea:*

Use equivalences

\[ \sin(x) = \frac{e^{ix} - e^{-ix}}{2i} \]

\[ \cos(x) = \frac{e^{ix} + e^{-ix}}{2} \]