Today:

- Probabilistically Checkable Proof Systems
- Proofs of NP statements can be verified with O(1) queries!
Useful Fact:

Given vectors \( \overline{a} \neq \overline{b} \)
\[
\Pr \left[ \overline{a} \cdot \overline{r} + \overline{b} \cdot \overline{r} \right] \geq \frac{1}{2}
\]

Also true for equality mod 2

Given matrices \( A, B, C \)

if \( A \cdot B + C \) then
\[
\Pr \left[ A \cdot B \cdot \overline{r} + C \cdot \overline{r} \right] \geq \frac{1}{2}
\]

\( O(\ell^4) \) time

Why?

Homework 1 optional problem 2
also: same argument used to
Show Fourier basis is orthogonal
\[
\langle X_s | X_T \rangle = 0 \quad \text{for} \quad s \neq T
\]
Probabilistically Checkable Proofs

---

<table>
<thead>
<tr>
<th>Def</th>
<th>L in PCP(r,g) if ( \exists V ) s.t.</th>
</tr>
</thead>
<tbody>
<tr>
<td>1)</td>
<td>( \forall x \in L \ \exists \Pi ) s.t.</td>
</tr>
<tr>
<td></td>
<td>( \Pr[V, \Pi \text{ accepts}] = 1 )</td>
</tr>
<tr>
<td></td>
<td>( V )'s random strings</td>
</tr>
<tr>
<td>2)</td>
<td>( \forall x \in \mathcal{L}, \forall \Pi' )</td>
</tr>
<tr>
<td></td>
<td>( \Pr[V, \Pi' \text{ accepts}] &lt; \frac{1}{4} )</td>
</tr>
<tr>
<td></td>
<td>( V )'s random strings</td>
</tr>
</tbody>
</table>

---

Theorem 2: For today, we'll focus on 2-CNF. The input is satisfiable.
$\text{SAT} \in PCP(0(n))$

- look at all settings of vars

Today:

\[ \text{Thm} \quad NP \leq PCP(0(n^3), 0(1)) \]

\[ \uparrow \quad \uparrow \]

\[ \emptyset \quad \text{queries} \]

Actually:

\[ \text{Thm} \quad NP \leq PCP(0(\log n), 0(1)) \]

3SAT: \[ F = \bigwedge C_i \text{ st. } C_i = (y_{i1} \lor y_{i2} \lor y_{i3}) \]

where \[ y_{ij} \in \{ x, \bar{x} \} \text{ for } i \leq 3 \]

1. Encode satisfiability of $F$ as a collection of polynomials in variables of assignment

- one for each clause
- low degree
- evaluate to 0 if assignment satisfies clause
- $V$ knows coeffs - depend on structure of clause + vars of clause
Arithmetrization of ZSAT:

boolean formula $F \iff$ arithmetic formula $A(F)$ over $\mathbb{Z}_2$

- $T \iff 1$
- $F \iff 0$
- $X_i \iff X_i$
- $\overline{X_i} \iff 1 - X_i$
- $\alpha \land \beta \iff \alpha \cdot \beta$
- $\alpha \lor \beta \iff 1 - (1 - \alpha)(1 - \beta)$
- $\alpha \lor \beta \lor \gamma \iff 1 - (1 - \alpha)(1 - \beta)(1 - \gamma)$

Example:

$x_1 \lor \overline{x_2} \lor x_3 \iff 1 - (1 - x_1)(1 - (1 - x_2))(1 - x_3)$

$= 1 - (1 - x_1)(x_2)(1 - x_3)$

$F$ satisfied by $\bar{a}$ iff $A(F)(\bar{a}) = 1$
Consider \( C^0(\overline{x}) = (\hat{C}_1(\overline{x}), \hat{C}_2(\overline{x}), \ldots) \)

Note: (1) Complements of arithmeticization of clause \( C_i \)

\( \Rightarrow \) evaluate to 0 if \( x \) satisfies \( C_i \)

(2) each \( \hat{C}_i \) is \( \deg \leq 3 \) poly in \( x \)

(3) \( V \) knows coeffs of each \( \hat{C}_i \)

Need to convince \( V \) that

\( C^0(\overline{a}) = (0, 0, \ldots, 0) \)

who sending \( \overline{a} \)

"weird idea"

Assume \( J \) "little birdie" who tells \( V \)

dot products of \( C^0 \) with random vectors mod 2

\( (V \text{ inputs } \overline{r}) \text{ birdie answers } C(\overline{a}) \cdot \overline{r} \)
Fix \( \vec{a} \)

\[(\hat{C}_1(\vec{a}), \ldots, \hat{C}_m(\vec{a})) \cdot (r_1 \ldots r_m) \]

\[= \sum r_i \hat{C}_i(\vec{a}) \mod 2 \]

\[\Pr \left[ \sum r_i \hat{C}_i(\vec{a}) = 0 \right] = \begin{cases} 1 & \text{if } \forall i \hat{C}_i(\vec{a}) = 0 \\ \frac{1}{2} & \text{otherwise} \\ \text{if } \exists i \text{ s.t. } \hat{C}_i(\vec{a}) \neq 0 \\ \text{if } \hat{C}(\vec{a}) \text{ not satisfied} \end{cases} \]

At this point, one can write a very long proof.

Entry for each choice of \( r_1, \ldots, r_m \) is answered by bi-die.

\[\ldots \]

Birdie can cheat and always answer 0!!

So far, no check for consistency with \( \hat{C}_m(\vec{a}) \)

So, why believe the birdie?
Recall:
we know $r_i$'s
we know coeff of polys of $c_i$'s
$c_i$'s have deg $\leq 3$ in $a_i$'s
we do not know $a_i$'s

\[ \sum_i r_i c_i(a) = \Gamma + \sum_i a_i x_i + \sum_{i,j} a_i a_j \beta_{ij} + \sum_{i,j,k} a_i a_j a_k \delta_{ijk} \]

From here on:
\[ a_i \rightarrow x_i \]
\[ \beta_{ij} \rightarrow y_{ij} \] no reln to vars of $\mathbb{F}$
\[ \delta_{ijk} \rightarrow z_{ijk} \] as $z_{ijk}$

V knows these (as does prof),
depend on $r_i$'s coeff of polys
do not depend on $a_i$'s
since worked mod 2, all
values $\in \{0,1\}$

Idea: make birdie write all answers for
all choices of $r_i$'s
V check consistency
(and later check satisfying the assignment)
We will do something stronger & easier to check
better idea

make birdie write out answers to all

3 separate parts of proof

linear fn's of $\tilde{a}$

deg 2 "  "  "
deg 3 "  "  "

we only care about 1 lin fn of $\tilde{a}$

deg 2
deg 3

will use to check that birdie wrote down a proper encoding of $\tilde{a}$

def $A : \mathbb{F}_2^n \to \mathbb{F}_2$ $A(x) = \sum a_i x_i = a^T \cdot \tilde{x}$

def $B : \mathbb{F}_2^n \to \mathbb{F}_2$ $B(y) = \sum a_i a_j y_{i,j} = (a \cdot a)^T \cdot \tilde{y}$

out product

if $Z = b \cdot c$
$Z_{i,j} = b_i \cdot c_j$

def $C : \mathbb{F}_2^3 \to \mathbb{F}_3$ $C(z) = \sum a_i a_j a_k z_{i,j,k} = (a \cdot a \cdot a)^T \cdot \tilde{z}$
Proof contains:

Complete description of truth tables of $\bar{A}, \bar{B}, \bar{C}$ for all inputs $\bar{x}, \bar{y}, \bar{z}$

Supposed to be $A, B, C$

but $V$ needs to check

What to check:

1. $\bar{A}, \bar{B}, \bar{C}$ are of right form
   - all are linear fits $\Rightarrow sc-\bar{A}$ will always answer according to closest value
   - linearity test + self-correct
   - correspond to same assignment $\bar{a}$
   - test all self-corrections consistent

2. $\bar{a}$ is a sat assignment
   - all $\bar{C}_n$'s evaluate to 0 on $\bar{a}$
How to do (N):

. Test \( A, \hat{B}, \hat{C} \) are all \( \frac{1}{8} \) close to linear functions
  
  \#random bits O(n^3)
  \#queries O(i)
  runtime O(n^3)
  . Pass if linear
  . Fail if \( \geq \frac{1}{9} \) far from linear
    in O(i) queries

. From now on use self-correcter per query to get

  \#random bits O(n^3)
  \#queries O(i)
  runtime O(n^3)
  sc-\( \hat{A} \), sc-\( \hat{B} \), sc-\( \hat{C} \) lin fcn

  can query on all inputs

  use really small error bound on S-C
  so, if union bound over all calls to sc-\( \hat{A} \), sc-\( \hat{B} \), sc-\( \hat{C} \)
  will never see error
Consistency Test:
are $sc-\tilde{A}$, $sc-\tilde{B}$, and $sc-\tilde{C}$ from some assignment $\bar{a}$?

Tester:
pick random $\bar{X}_1 \bar{X}_2 \bar{X}_3 \bar{Y}$

Test that $sc-\tilde{A}(\bar{X}_1) \cdot sc-\tilde{A}(\bar{X}_2)$

$$= \sum a_i x_i \cdot \sum a_j x_2j$$

$$= \sum \omega a_\omega x_1 \omega x_2j$$

$$= sc-\tilde{B}(\bar{X}_1, \bar{X}_2)$$

Assume $A \equiv B \equiv C$ correspond to some $\bar{a}$

# random bits $O(n^2)$

# queries $O(1)$

Random $O(n^3)$

Test that $sc-\tilde{A}(\bar{X}) \cdot sc-\tilde{B}(\bar{g})$:

$$= \sum a_\omega x_\omega \cdot \sum_{jk} a_j a_k y_{jk}$$

$$= \sum a_i a_j a_k x_i y_{jk}$$

$$= sc-\tilde{C}(\bar{X}, \bar{g})$$

Role:
not unit dist queries
but $sc-C$ helps here
Is it a good test?

given $\begin{align*}
sc - a \\
sc - b \\
sc - c
\end{align*}$

$\exists$ all lin forms $A(x) = a^T x$

$B(y) = b^T y$

$C(z) = c^T z$

hopefully $b^T = (a_0 a)^T$

$c^T = (a_0 b)^T$

$= (a_0 a_0 a)^T$

If $b = a_0 a$ then test pass

via green argument $✓$

else

if $b \neq a_0 a$

$$A(x_1) \cdot A(x_2)$$

$= B(x_1 \circ x_2)$

by def of

$$x_1 \circ x_2$$

with what prob $?$
if $a \cdot a \neq b$ then
$\Pr[(a \cdot a) \cdot x_2 \neq b \cdot x_2] \geq \frac{1}{2}$

$\Pr[x_1 \cdot (a \cdot a) \cdot x_2 \neq x_1 \cdot b \cdot x_2] \geq \frac{1}{a} \cdot \frac{1}{2} \geq \frac{1}{4}$

so test fails with $\Pr[b] \geq \frac{1}{4}$