Lecture 23:

Probabilistically Checkable Proof Systems:

check proof in time <u>sublinear</u>
in size of proof

Review:

$x \cdot y = \sum x_i y_i$ "inner product"

$x \circ y = (x_1 y_1 \ldots x_i \cdot y_j \ldots x_n y_n)$ "outer product"

n-bit vectors n^2-bit vector

Fact: if $\bar{a} \neq \bar{b}$ then $\Pr_{\bar{r} \in \{0,1\}^n} [\bar{a} \cdot \bar{r} \neq \bar{b} \cdot \bar{r}] = \frac{1}{2}$ Also true for $\equiv$ mod 2

if $A \cdot B \neq C$ then $\Pr_{\bar{r}} [A \cdot B \cdot \bar{r} \neq C \cdot \bar{r}] \geq 1/2$

proof via pairing argument e.g. lec 20 last page notes

Self-correcting:

if f is $\frac{1}{8}$-close to linear:

define: $g(x)$: Do $O(\log \frac{1}{\beta})$ times

Pick y randomly

$\text{answer}_i \leftarrow f(y) + f(x-y)$

Output most common answer

then: $\forall x$, $\Pr[g(x) = f(x)] \geq 1 - \beta$

Self-testing:

Given f:

Do $O(\frac{1}{\varepsilon})$ times:

Pick x, y randomly

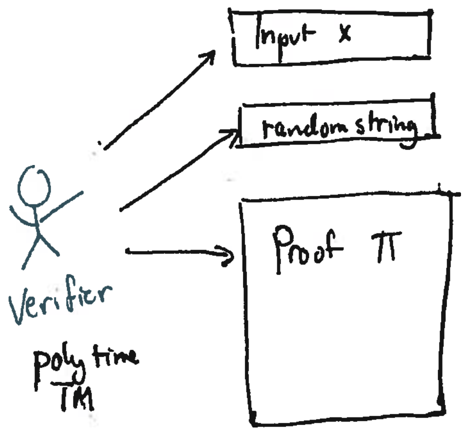
if $f(x) + f(y) \neq f(x+y)$, output fail + halt

Output pass

If f linear, test passes

If f ε-far from linear, $\Pr[\text{test fails}] \geq 3/4$

Probabilistically Checkable Proofs



← fixed fctn – different than prover
Created by adversary with unlimited computational power

def. $L \in PCP(r, q)$ if $\exists V$ (ptime TM) s.t.

1) $\forall x \in L \quad \exists \pi \quad$ s.t. $\quad \Pr_{\text{random strings}}[V, \pi \text{ accepts}] = 1$

2) $\forall x \notin L \quad \forall \pi', \quad \Pr_{\text{random strings}}[V, \pi' \text{ accepts}] < 1/4$

where V uses at most $r(n)$ random bits
+ makes at most $q(n)$ $\underbrace{\text{queries}}_{1 \text{ bit each}}$ to π

e.g. $SAT \in PCP(0, n)$ ← look at all settings of vars

<u>Today</u>: <u>Thm</u> $NP \subseteq PCP(O(n^3), O(1))$

<u>Actually</u>: <u>Thm</u> $NP \subseteq PCP(O(\log n), O(1))$ ⟸ How can it be? Verifier doesn't get to see any significant portion of assignment ??????

3SAT: $F = \bigwedge C_i$ s.t. $C_i = (y_{i_1} \vee y_{i_2} \vee y_{i_3})$ where $y_{i_j} \in \{X_1 \cdots X_n, \bar{X}_1 \cdots \bar{X}_n\}$

is F satisfiable? ← if so, how could you prove this?

A first crack:

π = setting of sat assignment a

⇑ assignment to all n variables

$a_1 = T$
$a_2 = F$
$\vdots$

Protocol for V:

Pick random clause C_i

Check if setting a satisfies C_i

Why good? if a satisfies C then $\Pr[V \text{ succeeds}] = 1$

Why bad? if a doesn't satisfy C, $\exists$ clause i s.t. a doesn't satisfy C_i

So $\Pr[V \text{ finds unsatisfiable clause}] \geq \frac{1}{m}$

since m = # clauses, & could be very big, this isn't so good, need to repeat $O(m)$ times to find unsat clause

Notation: $x = (x_1 \cdots x_n)$
$y = (y_1 \cdots y_n)$

3SAT:

$F = \bigwedge C_i$

↖ i^{th} clause

$C_i = (y_{i_1} \vee y_{i_2} \vee y_{i_3})$ where $y_{i_j} \in \{x_1 \cdots x_n \; \bar{x}_1 \cdots \bar{x}_n\}$

"Arithmetization" of 3SAT:

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

Same as $GF(2)$
domain $= \{0,1\}$
$+ \equiv +$ mod 2

T	$\Leftrightarrow$	1
F	$\Leftrightarrow$	0
x_i	$\longleftrightarrow$	x_i
$\bar{x}_i$	$\Leftrightarrow$	$1 - x_i$
$\alpha \wedge \beta$	$\Leftrightarrow$	$\alpha \cdot \beta$
$\alpha \vee \beta$	$\Leftrightarrow$	$1 - (1-\alpha)(1-\beta)$
$\alpha \vee \beta \vee \gamma$	$\Leftrightarrow$	$1 - (1-\alpha)(1-\beta)(1-\gamma)$

examples

$$(x_1 \vee x_2) \wedge \bar{x}_3 \Leftrightarrow (1 - (1-x_1)(1-x_2)) \cdot (1-x_3)$$

$$x_1 \vee \bar{x}_2 \vee x_3 \Leftrightarrow 1 - (1-x_1)(1-(1-x_2))(1-x_3)$$
$$= 1 - (1-x_1)(x_2)(1-x_3)$$

For $a = (a_1 \cdots a_n)$:
- F satisfied by a iff $A(a) = 1$
- F satisfiable iff $A(F) \neq 1$

Consider $C(x) = (\hat{c}_1(x), \hat{c}_2(x), \cdots)$

$\boxed{\text{NOTE}}$ complements of each clause C_i evaluate to 0 iff x satisfies the clause

$\boxed{\text{Note}}$. each $\hat{C}_i$ is degree ≤ 3 poly in x
and verifier knows its coefficients!!

- Won't arithmetize the whole formula, just each clause separately $\Rightarrow$ low degree
- (oh by the way, take the complement)

High level idea: special encoding of assignment

- proof "writes out" <u>all</u> linear fctns of assignment
  deg 2
  deg 3

  - possible "confusion": "symmetric" for linear case

$$f_x(a) = x \cdot a = A_a(x)$$

inner product

idea of switching role of x & a is important here!

  - for deg 2, deg 3

$$B_a(y) = (a \circ a)^T \cdot y$$

$$C_a(z) = (a \circ a \circ a)^T \cdot z$$

A_a, B_a & C_a are all linear

$\Rightarrow$ can test & self-correct

- are A_a, B_a & C_a consistent? (e.g. from same a?)

- is "a" a sat assignment?
  (this is where we "win" over obvious encoding)

example

$G = (x_1 \vee x_2) \wedge (\bar{x}_1 \vee x_2)$

$\bar{A}(C_1) = x_1 + x_2 - x_1 x_2 \quad \Rightarrow \quad A(C_1)(a) = 1 - a_1 - a_2 + a_1 a_2$

$\bar{A}(C_2) = 1 - x_1 + x_1 x_2 \quad \Rightarrow \quad A(C_2)(a) = \cdot a_1 - a_1 a_2$

evaluate at $x = a$:

$\sum r_i C_i(a) = r_1(1 - a_1 - a_2 + a_1 a_2) + r_2(\quad a_1 - a_1 a_2)$

$= (r_1 \cdot \quad) \cdot 1 + (-r_1 + r_2) \cdot a_1 + (-r_1 \quad) \cdot a_2 + (r_1 - r_2) a_1 a_2$

r_1	r_2	$\sum r_i C_i(a)$	sat case $a^+ = (0,1)$	unsat case $a^- = (0,0)$
0	0	0	0	0
0	1	$a_1 - a_1 a_2$	0	0
1	0	$1 - a_1 + -a_2 + a_1 a_2$	$1-0-1+0=0$	$1-0-0+0=1$
1	1	$1 - a_2$	$1-1=0$	$1-0=1$

Need to convince Verifier that $C(\bar{a}) = (0, 0, \dots, 0)$ w/o sending a

How do you test if a vector is all 0?

"Weird idea:" assume $\exists$ little birdie who tells V dot products of C with random vectors (mod 2)

Fix a

$$(\hat{c}_1(a), \dots, \hat{c}_m(a)) \cdot (r_1 \cdots r_m) \equiv \sum r_i \hat{c}_i(a) \mod 2$$

$$\Pr\left[\sum r_i \hat{c}_i(a) = 0\right] = \begin{cases} 1 & \text{if } \forall_i \ \hat{c}_i(a) = 0 \quad \leftarrow C(a) \text{ satisfied} \\ 1/2 & \text{o.w. } (\exists i \text{ s.t. } \hat{c}_i(a) \neq 0) \leftarrow C(a) \text{ not satisfied} \end{cases}$$

$\boxed{\text{* see also p 6. 3a}}$

$\Rightarrow$ different behavior when C(a) is satisfied

why? remember the pairing argument we did in lecture 20? (see last pg of notes)

But: Why believe the birdie? (e.g. birdie can just answer "0" all the time?)

does it help? 1) We know r_i's

2) we know $\boxed{\text{coeffs}}$ of polys of $\hat{c}_i$'s

3) $\hat{c}_i$'s have deg ≤ 3 in a_i's

V doesn't know these

$$\text{so} \quad \sum r_i \hat{c}_i(a) = \Gamma + \underbrace{\sum_i a_i \alpha_i}_{\mathrm{I}} + \underbrace{\sum_{i,j} a_i a_j \beta_{ij}}_{\mathrm{II}} + \underbrace{\sum_{i,j,k} a_i a_j a_k \gamma_{ijk}}_{\mathrm{III}} \pmod 2$$

V does know these

- depend on r_i's + coeffs of polys
- do not depend on a_i's
- computed by V
- since working mod 2, all values are $\in \{0,1\}$

from here on:

$\alpha_i \to x_i$
$\beta_{ij} \to y_{ij}$
$\gamma_{ijk} \to z_{ijk}$

} no relation to variables of 3SAT

(hopefully all of same a)

these are functions, and we really only care about their value at input that corresponds to what V computes from coeffs of polys & $\hat{C}_i$'s

def

A = all linear fctns evaluated at assignment a

$A : \mathbb{F}_2^n \to \mathbb{F}_2$ $\quad A(\bar{x}) = \sum_{i=1}^{n} a_i x_i = a^T \cdot x$

V knows this

B = all deg 2 fctns evaluated at assignment a

$B : \mathbb{F}_2^{n^2} \to \mathbb{F}_2$ $\quad B(\bar{y}) = \sum_{i,j} a_i a_j y_{ij} = (a \circ a)^T \cdot y$

outer product: if $z = b \circ c$, $z_{ij} = b_i \cdot c_j$ (n^2 vector, n vectors)

C = all deg 3 fctns evaluated at a

$C : \mathbb{F}_2^{n^3} \to \mathbb{F}_2$ $\quad C(\bar{z}) = \sum_{ijk} a_i a_j a_k z_{ijk} = (a \circ a \circ a)^T \cdot z$

Huge!! Proof π contains:

supposed to be A, B, C but we need to check this

~~Complete~~ description of truth tables of $\tilde{A}$, $\tilde{B}$, $\tilde{C}$. for $\boxed{\text{all}}$ inputs x, y, z

- we only need the values at one input !! ← namely $x = \alpha$, $y = \beta$, $z = \gamma$
- but this makes the checks a lot easier to do

What does verifier need to check in π?

(1) $\tilde{A}$, $\tilde{B}$, $\tilde{C}$ are of right form

- all are linear fctns ⟸ • can only test that they are close to linear • however, can self-correct!
- Correspond to same assignment a

  i.e. $\tilde{A}(x) = a^T \cdot x \Rightarrow \tilde{B}(y) = (a \circ a)^T \cdot y \Rightarrow \tilde{C}(z) = (a \circ a \circ a)^T \cdot z$

  • test that self-corrections are consistent

(2) $\bar{a}$ is a sat assignment

- all $\hat{C}_i$'s evaluate to 0 on a

How to do (2):

- Test $\tilde{A}, \tilde{B}, \tilde{C}$ are all $\frac{1}{8}$-close to linear fcts (ie. Pass if linear, Fail if $\geq \frac{1}{8}$-far from linear) in $O(1)$ queries

$\boxed{\begin{array}{l}\#\text{random bits} = O(n^3) \\ \#\text{queries} = O(1) \\ \text{runtime} = O(n^3)\end{array}}$

- From now on, use self-corrector to get $sc\text{-}\tilde{A}$, $sc\text{-}\tilde{B}$, $sc\text{-}\tilde{C}$ for all inputs
  - use error parameter that is small enough to do union bound over all queries to $\tilde{A}\,\tilde{B}\,\tilde{C}$ (but will only be constant)

$\boxed{\begin{array}{l}\text{Per query to S-C:} \\ \#\text{random bits} = O(1) \\ \#\text{queries} = O(1) \\ \text{runtime} = O(n^3)\end{array}}$

- Consistency test:

  Goal: Pass iff $sc\text{-}\tilde{B} = sc\text{-}\tilde{A} \circ sc\text{-}\tilde{A}$

  $sc\text{-}\tilde{C} = sc\text{-}\tilde{A} \circ sc\text{-}\tilde{B}$

Outer Product Tester:

Pick random x_1, x_2, x, y

Test that $sc\text{-}\tilde{A}(x_1) \cdot sc\text{-}\tilde{A}(x_2) \left(= \sum a_i x_{1i} \cdot \sum a_j x_{2j} = \sum_{i,j} a_i a_j x_{1i} x_{2j} \right) = sc\text{-}\tilde{B}(x_1 \circ x_2)$

$sc\text{-}\tilde{A}(x) \cdot sc\text{-}\tilde{A}(y) = \left(\sum_i a_i x_i \cdot \sum_{jk} a_j a_k y_{jk} \right) = \sum_{ijk} a_i a_j a_k x_i \ldots$

$= sc\text{-}\tilde{C}(x \circ y)$

assuming $\tilde{A}+\tilde{B}+\tilde{C}$ correspond to a_i's

Note: these are not uniformly distributed vecto[rs]

$\boxed{\begin{array}{l}\#\text{random bits} = O(n^3) \\ \#\text{queries} = O(1) \\ \text{runtime} = O(n^3)\end{array}}$

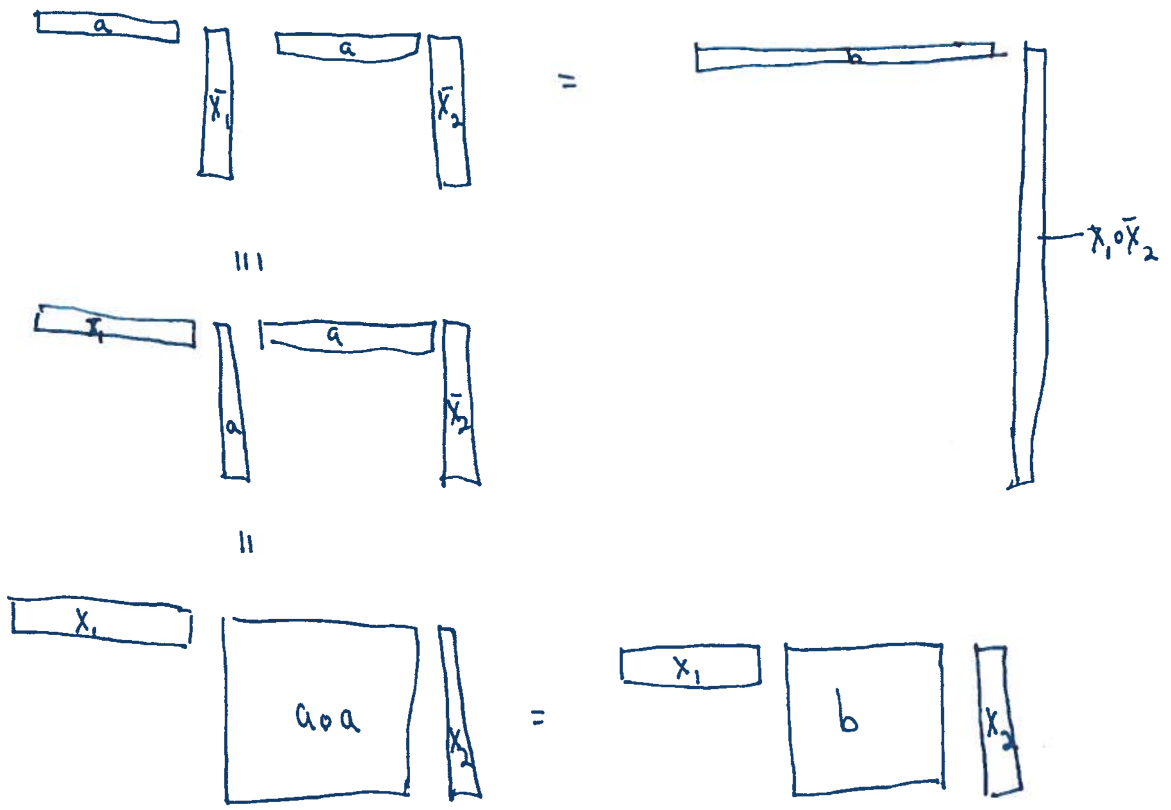
Does it work? Given, sc-$\tilde{A}$ $= A$, sc-$\tilde{B}$ $= B$ + sc $\tilde{C}$ $= C$ are linear fctns

$A(x) = a^T x$
$B(y) = b^T y \overset{?}{=} (a \circ a)^T y$
$C(z) = c^T z \overset{?}{=} (a \circ a \circ a)^T z$

if $b = a \circ a$
+ $c = a \circ a \circ a \doteq a \circ b$ ✓ (by green argument on previous page)

else, if $b \neq a \circ a$

$$A(\bar{x}_1) \cdot A(\bar{x}_2) = B(\bar{x}_1 \circ \bar{x}_2)$$



Fact [Freivald's test] if vectors $a \neq b$ then $\Pr[a \cdot r \neq b \cdot r] \geq 1/2$
if matrixes $A \cdot B \neq C$ then $\Pr_{\text{random vector } r}[A \cdot B \cdot r \neq C r] \geq 1/2$

Same "proof" as for "weird idea"

note: $\bar{x}$'s are playing role of $\bar{r}$'s here

$\Rightarrow \Pr[(a \circ a) \cdot x_2 \neq b \cdot x_2] \geq 1/2 \Rightarrow \Pr[x_1 \cdot [(a \circ a) \cdot x_2] \neq x_1 \cdot [b \cdot x_2]] \geq 1/4$

So test fails with prob $\geq 1/4$!!!

How to do (2):

- recall, we are making calls to self corrector, so we are recovering linear fctns a, $a \otimes a$, $a \circ a \circ a$
- we don't actually know a, but it represents the assignment
- is a satisfying? ie. are all $\hat{C}_i(a) = 0$?

Satisfiability Test:

Pick $r \in_R \mathbb{Z}_2^n$

Compute Γ, α_i's, β_{ij}'s, γ_{ijk}'s ← fctns of r + coeffs of polys from constraints

$\downarrow$ x_i's $\downarrow$ y_{ij}'s $\downarrow$ z_{ijk}'s

$\boxed{\begin{array}{l}\#\text{random bits} = O(n) \\ \#\text{queries} = O(1)\end{array}}$

query proof to get
$$SC\text{-}\tilde{A}(\alpha_1 \ldots \alpha_n) = w_0$$
$$SC\text{-}\tilde{B}(\beta_{11} \ldots \beta_{nn}) = w_1$$
$$SC\text{-}\tilde{C}(\gamma_{111} \ldots \gamma_{nnn}) = w_2$$

Verify $0 = \Gamma + w_0 + w_1 + w_2 \pmod 2$

hopefully means $\sum r_i \hat{C}_i(a) = 0$

Why does it work?

if $\forall i$, $\hat{C}_i(a) \equiv 0$ then pass with prob 1 ✓

if $\exists i$ s.t. $\hat{C}_i(a) \not\equiv 0$ then $(0\,0 \ldots 0) \neq (\hat{C}_1(a) \ldots \hat{C}_m(\bar{a}))$

so $\Pr\left[\sum r_i \hat{C}_i(a) = 0 \bmod 2 = \sum 0 \cdot r_i\right] = \frac{1}{2}$

after k times, pass all K times with prob $= \frac{1}{2^k}$ ✓