Lecture 23

Probabilistically Checkable Proof Systems

- from earlier lectures & homeworks:
    - Freivald's test
    - self-testing correcting linear fctns
- model
- $NP \subseteq PCP(n^3, 1)$
    - arithmetization

Recall some useful facts

Freivald's test

if vectors $a \neq b$ then $\Pr_{r \in \{0,1\}^n}[a \cdot r \neq b \cdot r] \geq \frac{1}{2}$

if matrices $A \cdot B \neq C$ then $\Pr_{r \in \{0,1\}^n}[A \cdot B \cdot r \neq C \cdot r] \geq \frac{1}{2}$

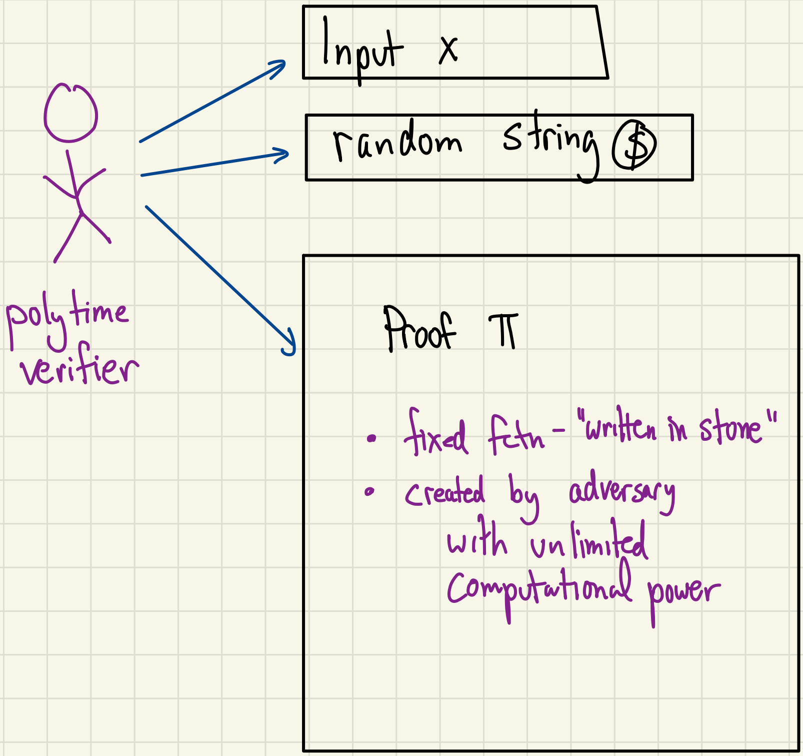
<u>Pf</u>. pair vectors that differ in coordinate i s.t. $a_i \neq b_i$

or $A \cdot B_{ij} = C_{ij}$

(as in proof of orthogonality of Fourier basis) $\blacksquare$

<u>Comment</u> also true for equality mod 2

The Model



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

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

arbitrary

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

V uses $\leq r(n)$ random bits & makes $\leq q(n)$ queries to π 1 bit each

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

Today $NP \subseteq PCP(O(n^3), O(1))$

Actually $NP \subseteq PCP(O(\log n), O(1))$

} verifier can't see 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 $y_{i_j} \in \{x_1 .. x_n \bar{x}_1 \cdots \bar{x}_n\}$

Is F satisfiable?

if so, how would you prove it?

First Crack:

π = settings of sat assignment a

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

Protocol for V:

pick random clause C_i

Check if setting $\bar{a}$ satisfies C_i

Why good?

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

Why bad?

if $\bar{a}$ doesn't satisfy C,

$\exists$ clause i s.t. $\bar{a}$ doesn't satisfy C_i

So $\Pr[V \text{ finds unsat } C_i] \geq \frac{1}{m}$

not so great since m can be big & need to repeat $O(m)$ times to find one

Arithmetization of SAT

boolean formula F	arithmetic formula $A(F)$ over $\mathbb{Z}_2$
T	1
F	0
x_i	x_i
$\overline{x}_i$	$1-x_i$
$\alpha \wedge \beta$	$\alpha \cdot \beta$
$\alpha \vee \beta$	$1-(1-\alpha)(1-\beta)$
$\alpha \vee \beta \vee \gamma$	$1-(1-\alpha)(1-\beta)(1-\gamma)$

examples:

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

$x_1 \vee \overline{x}_2 \vee x_3$ $\qquad$ $1-(1-x_1)(1-(1-x_2))(1-x_3)$

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

degree ≤ 3

F satisfied by a iff $A(a)=1$.

Strange Arithmetization:

arithmetize complement of each clause separately

$$\mathcal{C}(x) = \left(\hat{C}_1(x), \hat{C}_2(x), \ldots \right)$$

$x = (x_1 \cdots x_n)$

Complements of each clause C_i

evaluate to 0 if x satisfies C_i

each $\hat{C}_i(x)$ is degree ≤ 3 poly in x

& verifier knows the coefficients

Need to convince verifier that

$\mathcal{C}(a) = (0, 0, \ldots 0)$ w/o sending a

how to test vector is all 0?

weird idea: try to use "Freivald's test"??

how? assume $\exists$ little birdie who tells V dot products of $\mathcal{C}(a)$ with random vectors (mod 2)

Freivald's test on $C(a)$:

Fix a:

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

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

$C(a)$ satisfied

$$= \begin{cases} 1 & \text{if } \forall i \ \hat{C}_i(a) = 0 \\ \frac{1}{2} & \text{o.w.} \end{cases}$$

$C(a)$ not satisfied

Problem why believe the birdie?

Believing the birdie

1) we choose r_i's

2) we know coeffs of polys in $\hat{C}_i$'s

3) polys of $\hat{C}_i$'s are degree ≤ 3 in a_i's

V doesn't know these

so:

$$\sum r_i \hat{C}_i(a) = \Gamma + \sum_i a_i \alpha_i + \sum_{i,j} a_i a_j \beta_{ij} + \sum_{i,j,k} a_i a_j a_k \gamma_{ijk} \pmod 2$$

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 vars of 3SAT

example

$$(x_1 \vee \bar{x}_2 \vee x_3)(\bar{x}_1 \vee x_2) \Longrightarrow \left(\overline{(1 - x_2 + x_1 x_2 + x_2 x_3 - x_1 x_2 x_3)}, \overline{(1 - x_1 + x_1 x_2)}\right)$$

$$\Longrightarrow \left((x_2 - x_1 x_2 - x_2 x_3 + x_1 x_2 x_3), (x_1 - x_1 x_2)\right)$$

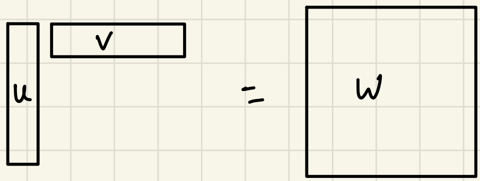
$$r_1 \cdot (x_2 - x_1 x_2 - x_2 x_3 + x_1 x_2 x_3) + r_2 \cdot (x_1 - x_1 x_2)$$

$$= 0 \cdot 1 + r_2 \cdot x_1 + r_1 \cdot x_2 - (r_1 + r_2) x_1 x_2 - r_1 \cdot x_2 x_3$$

$$+ 0 \cdot x_1 x_3 + r_1 \cdot x_1 x_2 x_3$$

Functions for the "birdy"

def [outer product] $w = u \circ v$ if $w_{ij} = u_i \cdot v_j$



V knows these

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

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

$C : \mathbb{F}_2^{n^3} \to \mathbb{F}_2$ $\quad C(y) = \sum_{ijk} a_i a_j a_k z_{ijk}$

$$= (a \circ a \circ a)^T \cdot \underline{z}$$

Proof $\Uparrow$:

Complete description of truth tables $\underbrace{\tilde{A}, \tilde{B}, \tilde{C}}_{}$

hopefully A, B, C
but need to check

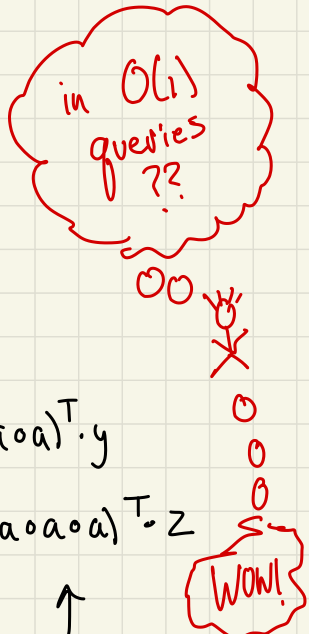
V really only needs to know A, B, C at input x, y + z (which it knows)

other entries help in checking!!

- check that tables of correct forms (linear fctns)
- self-correct to get values of linear fctn at x, y, z

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 close-to-linear
  but can self-correct

- correspond to same assigment 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 according to

(2) a is SAT assignment

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

How to do (1):

#randombits $O(n^3)$
#queries $O(1)$
runtime $O(n^3)$

- Test $\tilde{A}, \tilde{B}, \tilde{C}$ each $\frac{1}{8}$-close to linear via linearity test

  (Pass if linear, fail if $\frac{1}{8}$-far)

  $O(1)$ queries

- from now on, access $\tilde{A}, \tilde{B}, \tilde{C}$ via self-corrector on all inputs.

  $$sc\text{-}\tilde{A}, \; sc\text{-}\tilde{B}, \; sc\text{-}\tilde{C}$$
  $$\updownarrow \qquad \updownarrow \qquad \updownarrow$$
  $$a \qquad b \qquad c$$

  use confidence parameter that is small enough to do union bnd over all queries to $sc\text{-}\tilde{A}$, $sc\text{-}\tilde{B}$, $sc\text{-}\tilde{C}$ s.t. can assume always get right answer with high (constant) probability

- test consistency of $sc\text{-}\tilde{A}, sc\text{-}\tilde{B}, sc\text{-}\tilde{C}$

  i.e. $b = a \circ a$ & $c = a \circ b$

Consistency test:

Pick random x_1, x_2, x, y

assuming $\tilde{A}, \tilde{B}, \tilde{C}$ actually correspond to a's

test that (1) $sc\text{-}\tilde{A}(x_1) \cdot sc\text{-}\tilde{A}(x_2)$

$$= \sum_i a_i x_{1i} \cdot \sum_j a_j x_{2j} = \sum_{i,j} a_i a_j x_{1i} x_{2j}$$

$$= sc\text{-}\tilde{B}(x_1 \circ x_2)$$

#random bits $O(n^2)$

#queries $O(1)$

runtime $O(n^3)$

(2) $sc\text{-}\tilde{A}(x) \cdot sc\text{-}\tilde{B}(y)$

$$= \sum_i a_i x_i \cdot \sum_{j,k} a_j a_k y_{jk} = \sum_{ijk} a_i a_j a_k x_i y_{jk}$$

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

note $x_1 \circ x_2$ + $x \circ y$ are not unif dist vectors. (that's why we call $sc\text{-}\tilde{A}$, $sc\text{-}\tilde{B}$, $sc\text{-}\tilde{C}$ instead of $\tilde{A}, \tilde{B}, \tilde{C}$ directly)

proof of consistency test: let a, b, c be linear fctns corresponding to $sc\text{-}\tilde{A}$, $sc\text{-}\tilde{B}$, $sc\text{-}\tilde{C}$

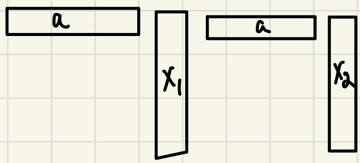
if $b = a \circ a$ + $c = a \circ a \circ a$ then test passes ✓

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

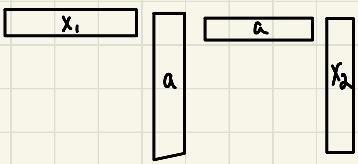
$$sc\text{-}\tilde{A}(x_1)\cdot sc\text{-}\tilde{A}(x_2) = A(x_1)\cdot A(x_2) \overset{?}{=} B(x_1 \circ x_2) = sc\text{-}\tilde{B}(x_1 \circ x_2)$$

$$A(x_1)\cdot A(x_2) = a\cdot x_1 \cdot a\cdot x_2 \qquad\qquad B(x_1 \circ x_2) = b\cdot(x_1 \circ x_2)$$

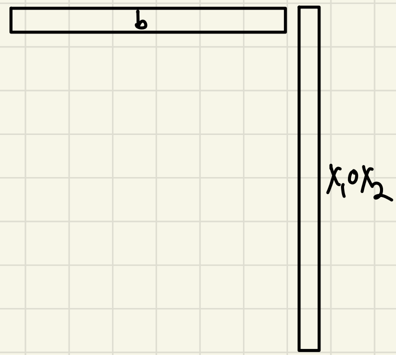
$=$



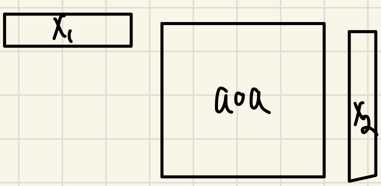
$=$



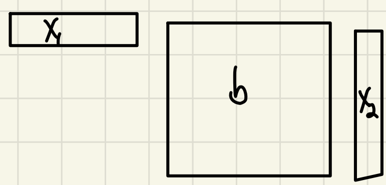
$=$



$=$



$\overset{?}{=}$



if $b \neq a \circ a$: $\Pr_{x_1, x_2}\left[x_1 \cdot (a \circ a)\cdot x_2 \neq x_1 \cdot (b \cdot x_2)\right]$

$\geq \frac{1}{2} \cdot \Pr\left[(a \circ a) x_2 \neq b x_2\right]$

$\geq 1/4$

(note, x's are playing role of "r"'s here)

similar argument for $C \neq a \circ a \circ a$

How to do (2):

recall:

- we call self-corrector, so recovering consistent linear fctns a, $a \circ a$, $a \circ a \circ a$
- we don't actually know a, but it represents the assignment
- does it satisfy? i.e. are all $\hat{C}_i(a) = 0$?

Satisfiability Test:

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

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

$\alpha \to x$, $\beta \to y$, $\gamma \to z$

query proof to get
$$SC\text{-}\hat{A}(\alpha) = w_0$$
$$SC\text{-}\tilde{B}(\beta) = w_1$$
$$SC\text{-}\tilde{C}(\gamma) = w_2$$

Verify $0 = \Gamma + w_0 + w_1 + w_2$ ← hopefully means $\sum r_i \hat{C}_i(a) = 0$

does it work?

if $\forall i$, $\hat{C}_i(a) = 0 \Rightarrow$ always pass

if $\exists i$ s.t. $\hat{C}_i(a) \neq 0 \Rightarrow$

$(0 \dots 0) \neq (\hat{C}_1(a) \dots \hat{C}_n(a))$

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

$\Rightarrow \Pr[\text{passes all } k \text{ times}] \leq 1/2^k$

$\blacksquare$