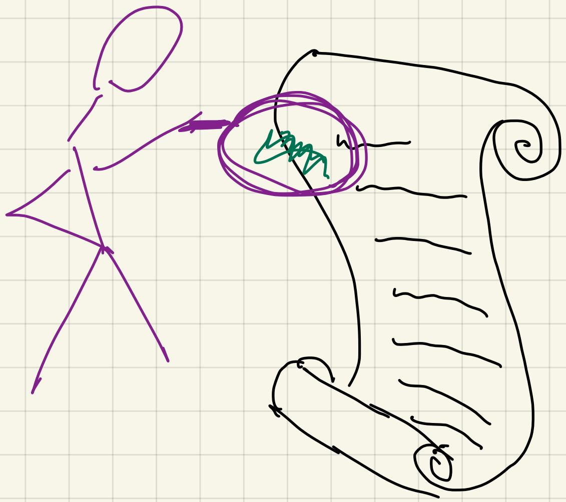
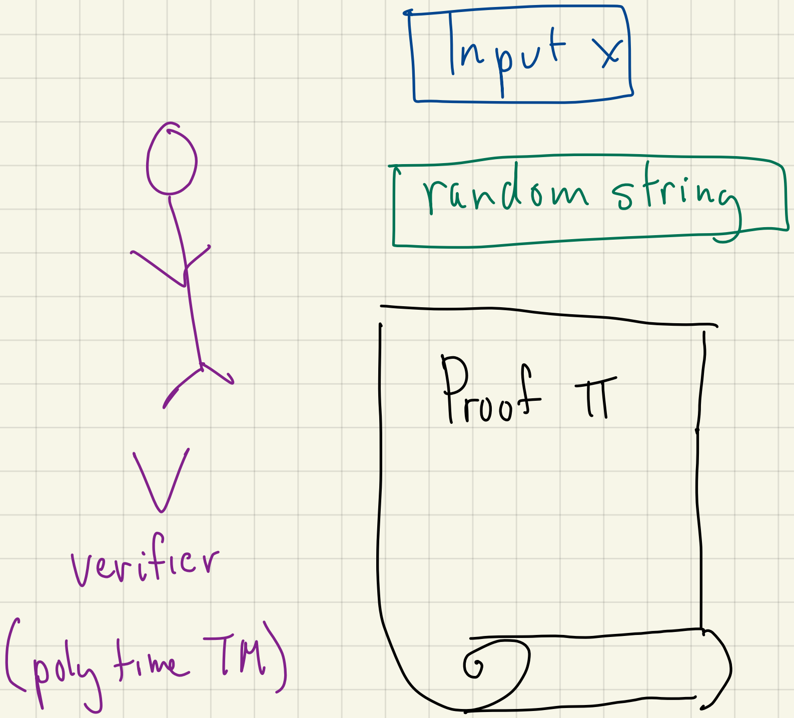
Lecture 22

Probabilistically Checkable Proof Systems



Probabilistically Checkable Proofs



← Theorem to prove

for today:

x is a 3 CNF

Thm x is satisfiable

fixed fctn – ie, what is i^{th} bit?
written down & can't change
created by adversary with unlimited computational power

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

1) $\forall x \in L \;\; \exists \pi$ s.t. $\Pr_{v\text{'s random string}}[v, \pi \text{ accepts}] = 1$

2) $\forall x \notin L \;\; \forall \pi' \;\; \Pr_{v\text{'s random strings}}[v, \pi' \text{ accepts}] \le 1/4$

e.g. $\quad SAT \in PCP(0, n)$

proof = settings of all n vars

V doesn't need to be random!

Today: $\quad NP \subseteq PCP\left(O(n^3), O(1)\right)$

Actually: $\quad NP \subseteq PCP\left(O(\log n), O(1)\right)$ $\quad \Leftarrow$ crazy

Verifier can't see almost any of assignment

Let's start with a "warmup":

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

$\underbrace{x \circ y}_{\text{n-bit vector}} = \underbrace{(x_1 y_1, x_1 y_2, x_4 y_3, \ldots, x_i y_j, \ldots, x_n y_n)}_{n^2\text{-bit vector}}$ "outer product"

Fact: if $\underbrace{\bar{a} \neq \bar{b}}_{\text{n-bitvector}}$ then $\Pr_{\bar{r} \in_R \{0,1\}^n}[\bar{a} \cdot \bar{r} \neq \bar{b} \cdot \bar{r}] \geq \frac{1}{2}$

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

also true for "= mod 2"

takes $O(n^2)$ time to compute: $\underbrace{A \cdot (\underbrace{B \cdot \bar{r}})}$

Proof of fact: if $a_i \neq b_i$ pair n-bit strings that agree on all but i^{th} location

so $\bar{r} = (r_1 \ldots r_i \ldots r_n)$ then either $\bar{a} \cdot \bar{r} \neq \bar{b} \cdot \bar{r}$ why?

$\bar{s} = (r_1 \ldots \bar{r}_i \ldots r_n)$ or $\bar{a} \cdot \bar{s} \neq \bar{b} \cdot \bar{s}$

$\bar{a} \cdot \bar{s} = (\bar{a} \cdot \bar{r}) \pm a_i$ different

$\bar{b} \cdot \bar{s} = (\bar{b} \cdot \bar{r}) \pm b_i$

$\Rightarrow$ in each pair at least one is $\neq$

note that this proof works "mod 2"

Fact: if $\bar{a} \neq \bar{b}$ then $\Pr_{\bar{r} \in_R \{0,1\}^n} [\bar{a} \cdot \bar{r} \neq \bar{b} \cdot \bar{r}] \geq \frac{1}{2}$

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

Example "application": setting: given vector $\bar{a} = (a_1, a_2, \ldots a_n)$

in one step: • can query a_i

• $\underbrace{\text{can specify } \bar{y} \text{ \& query } \bar{a} \cdot \bar{y}}_{}$

a bit strange but what if all these answers were written down for you?

to test if $\bar{a} \equiv (0, 0, \ldots, 0)$:

Do several times:

pick $\bar{r} \in_R \{0,1\}^n$

if $\bar{a} \cdot \bar{r} \neq 0$ output "Fail"

Output PASS

behavior: if $\bar{a} = (0, 0, \ldots 0)$ will always PASS

if $\bar{a} \neq (0, \ldots 0)$ then FACT $\Rightarrow \Pr_{\bar{r}} [\bar{a} \cdot \bar{r} \neq 0] \geq 1/2$

$\Rightarrow O(1)$ query 0-testing for vector in strange model

setting: given vector $\bar{a} = (a_1, a_2 \ldots a_n)$
in **one** step: • can query a_i
• can specify $\bar{y}$ & query $\bar{a} \cdot \bar{y}$

Making the model "less strange":

first idea:

"Proof" = write out all answers to $\bar{a} \cdot \bar{y}$

answer vector

$\bar{a} \cdot (0,0, \ldots 0)$ 0
$\bar{a} \cdot (0,0, \ldots 1)$ 0
$\bar{a} \cdot (0,0, \ldots 1,0)$ 0
$\bar{a} \cdot (0,0, \ldots 1,1)$ ⋮

to test if $\bar{a} = (0, 0, \ldots, 0)$:

Do several times:

pick $\bar{r} \in_R \{0,1\}^n$

ask proof for value of $\bar{a} \cdot \bar{r}$

if $\bar{a} \cdot \bar{r} \neq 0$ output "Fail"

Output PASS

Problem: proof can cheat

write all 0's in answer vector

How can we check that proof doesn't cheat?

test on $\bar{r}$'s that we know answer to? is this easier than just looking at every entry of $\bar{a}$?

WILL COME BACK TO THIS

3SAT:

$F = \bigwedge C_i$ s.t. $C_i = (y_{i_1} \vee y_{i_2} \vee y_{i_3})$

where $y_{i_j} \in \{ X_1 \cdots X_n, \bar{X}_1 \cdots \bar{X}_n \}$

here use $\bar{X}$ notation for complement

First crack:

π = setting of sat assignment a

$a_1 = T \quad a_2 = F \quad a_3 = T \ldots$

1 0 1 ...

V's protocol given formula + a:

Pick random clause C_i + check if $\bar{a}$ satisfies

good? $\bar{a}$ satisfies $\bar{c}$ ✓

$\bar{a}$ doesn't satisfy $\bar{c} \Rightarrow \exists i$ s.t. $C_i(\bar{a}) \neq T$

Pick i with prob $= \frac{1}{\#\text{clauses}}$:(

$F = (X_1 \vee \bar{X}_2 \vee X_3)(X_2 \vee \bar{X}_3 \vee X_4)$

$\bar{a} = (X_1 = T, X_2 = F, X_3 = F, X_4 = F, \ldots)$

random clause $(X_2 \vee \bar{X}_3 \vee X_4)$ ✓

F T F

Arithmetization of 3SAT:

Boolean formula F $\Longleftrightarrow$ arithmetic formula $A(F)$ over $\mathbb{Z}_2$

$\underbrace{}_{\text{mod } 2}$

$$T \Longleftrightarrow 1$$

$$F \Longleftrightarrow 0$$

$$x_i \Longleftrightarrow x_i$$

$$\bar{x}_i \Longleftrightarrow 1-x_i$$

$$\alpha \wedge \beta \Longleftrightarrow \alpha \cdot \beta$$

$$\alpha \vee \beta \Longleftrightarrow 1-(1-\alpha)(1-\beta)$$

$$\alpha \vee \beta \vee \gamma \Longleftrightarrow 1-(1-\alpha)(1-\beta)(1-\gamma)$$

<u>example</u>: $x_1 \vee \bar{x}_2 \vee x_3 \Longleftrightarrow 1-(1-x_1)(1-(1-x_2))(1-x_3)$

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

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

$F = \bigwedge C_i$ s.t. $C_i = (y_{i_1} \vee y_{i_2} \vee y_{i_3})$

where $y_{i_j} \in \{ x_1 \dots x_n, \bar{x}_1 \dots \bar{x}_n \}$

$T \Leftrightarrow 1$

$F \Leftrightarrow 0$

$x_i \Leftrightarrow x_i$

$\bar{x}_i \Leftrightarrow 1 - x_i$

$\alpha \wedge \beta \Leftrightarrow \alpha \cdot \beta$

$\alpha \vee \beta \Leftrightarrow 1 - (1-\alpha)(1-\beta)$

$\alpha \vee \beta \vee \gamma \Leftrightarrow 1 - (1-\alpha)(1-\beta)(1-\gamma)$

Consider $\mathcal{C}(x) = (\hat{C}_1(x), \hat{C}_2(x), \dots)$

s.t. $\hat{C}_i(x)$ = complement of arithmetization of clause C_i

$\Rightarrow$ evaluates to 0 if x satisfies C_i

$\Rightarrow \mathcal{C}(x) = (0, 0, \dots, 0)$ if x satisfies F

Observe (1) each $\hat{C}_i$ is deg ≤ 3 poly in x

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

Note we are only concerned about # bits V sees in proof, NOT V's runtime which is going to be superlinear

Need to convince V that

$\mathcal{C}(a) = (\hat{C}_1(a), \hat{C}_2(a), \dots) = (0, 0, \dots 0)$ WITHOUT SENDING assignment a

High level idea: special encoding of assignment

Encode satisfiability of F as a collection of polys in vars of assignment

- one for each clause
- eval to 0 if assignment satisfies clause
- low degree
- V knows coeffs - depend on structure of clause & vars of clause.

Note: We are only concerned that V is poly time, here will not be sublinear

note that solving SAT in poly time would be impressive ☺

However, want # queries to proof to be constant

Idea for proof:

- proof contains $\mathscr{C}(a)\cdot r \quad \forall \quad r \in \{0,1\}^n$
- if $\forall\ i,\ \hat{C}_i(a)=0,\ \Pr_r[\mathscr{C}(a)\cdot r = 0]=1$

  if $\exists\, i$ st. $\hat{C}_i(a)\neq 0,\ \Pr[\mathscr{C}(a)\cdot r=0]=\frac{1}{2}$

$F = \bigwedge C_i$ st. $C_i = (y_{i_1} \vee y_{i_2} \vee y_{i_3})$

where $y_{i_j} \in \{x_1 \dots x_n, \bar{x}_1 \dots \bar{x}_n\}$

$\mathscr{C}(a) = (\underbrace{\hat{C}_1}_{\text{complement}}(a), \hat{C}_2(a), \dots) \overset{?}{=} (0,0,\dots 0)$

$T \Leftrightarrow 1$

$F \Leftrightarrow 0$

$x_i \Leftrightarrow x_i$

$\bar{x}_i \Leftrightarrow 1-x_i$

$\alpha \wedge \beta \Leftrightarrow \alpha\cdot\beta$

$\alpha \vee \beta \Leftrightarrow 1-(1-\alpha)(1-\beta)$

$\alpha \vee \beta \vee \gamma \Leftrightarrow 1-(1-\alpha)(1-\beta)(1-\gamma)$

$\Leftrightarrow$ mod 2 arithmetic

why believe the proof?

can write all 0's, even if $\mathscr{C}(a)\cdot r \neq 0$

$\Rightarrow$ will need to do more in order to believe the proof. e.g. more proof
more verification of proof

What does $\mathscr{C}(a)\cdot r$ look like?

V doesn't know these

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

V does know these (& so does proof)
depend on r_i's, coeffs of polys in $\hat{C}_i$ but not on a_i's

example

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

$$A(C_1) = 1 - (1-x_1)(1-x_2) = x_1 + x_2 - x_1 x_2$$

$$\Rightarrow \quad C_1(a) = 1 - a_1 - a_2 + a_1 a_2$$

$$A(C_2) = 1 - (x_1)(1-x_2) = 1 - x_1 + x_1 x_2$$

$$\Rightarrow \quad C_2(a) = a_1 - a_1 a_2$$

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

$$= \underbrace{r_1 \cdot 1 + r_2 \cdot 0}_{\text{deg } 0} + \underbrace{(-r_1 + r_2) \cdot a_1 + (-r_1) \cdot a_2}_{\text{deg } 1} + \underbrace{(r_1 - r_2) \cdot a_1 a_2}_{\text{deg } 2}$$

$F = \bigwedge C_i$ s.t. $C_i = (y_{i_1} \vee y_{i_2} \vee y_{i_3})$

where $y_{i_j} \in \{x_1 \cdots x_n, \bar{x}_1 \cdots \bar{x}_n\}$

$$\hat{C}(a) = (\underbrace{\hat{C}_1(a)}_{\text{complement}}, \hat{C}_2(a), \ldots) \overset{?}{=} (0, 0, \ldots 0)$$

$T \Leftrightarrow 1$

$F \Leftrightarrow 0$

$x_i \Leftrightarrow x_i$

$\bar{x}_i \Leftrightarrow 1 - x_i$

$\alpha \wedge \beta \Leftrightarrow \alpha \cdot \beta$

$\alpha \vee \beta \Leftrightarrow 1 - (1-\alpha)(1-\beta)$

$\alpha \vee \beta \vee \gamma \Leftrightarrow 1 - (1-\alpha)(1-\beta)(1-\gamma)$

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

r_1	r_2	$\sum r_i \hat{C}_i(a)$	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	$1 - a_2$	$1-1=0$	$1-0=1$

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

- for deg 2, 3: $B_a(y) = (a \circ a)^T \cdot y$

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

A_a, B_a, C_a are all linear fctns $\Rightarrow$ can test linearity + self-correct

Proof can cheat!
- what if A_a, B_a, C_a actually come from different assignments
- is a satisfying?

linear fctn : $\forall x, y \quad f(x) + f(y) = f(x+y)$

Self-correcting :

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

define $g(x)$ as follows:

Do $O(\log \frac{1}{\beta})$ times

Pick y randomly

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

Output most common answer_i

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 $f(x) + f(y) \neq f(x+y)$ Fail

Pass

if f linear passes

if f ε-far from linear, fails