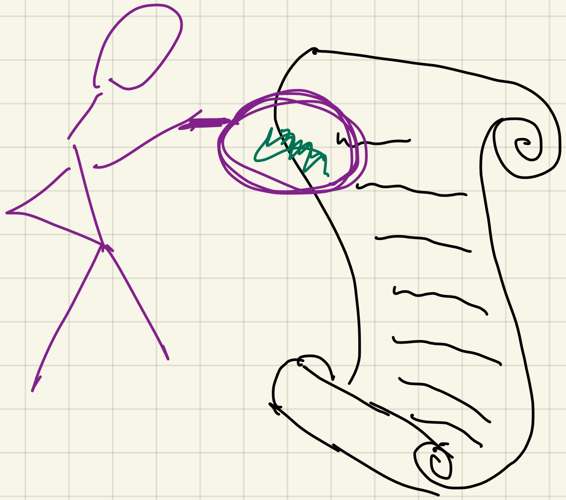
Lecture 22

Probabilistically Checkable Proof Systems



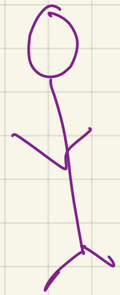
Probabilistically Checkable Proofs

$\boxed{\text{Input } x}$ ← Theorem you want to prove

for today: x is 3CNF

Thm x is satisfiable

$\boxed{\text{random string}}$



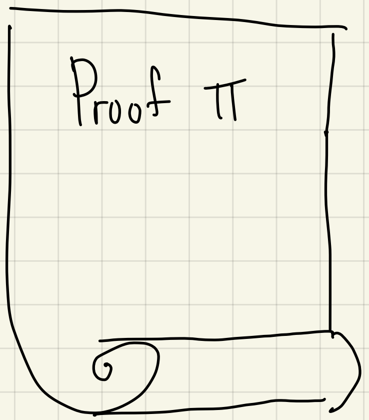
V

verifier

(poly time TM)

using r random bits

+ q queries to proof



fixed fctn

Verifier can query: what is ith bit?

Charged per query

proof doesn't change based on past questions of verifier

Created by adversary who knows verifier's algorithm + has unlimited computational power

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

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

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

e.g. $SAT \in PCP(0, n)$ ← proof settings of all n vars
V doesn't need any randomness

Today: $NP \subseteq PCP(O(n^3), O(1))$ ← crazy?

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

Let's start with a "warmup":

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

$x \circ y = (x_1 y_1, x_1 y_2, x_1 y_3, \ldots, x_i y_j, \ldots, x_n y_n)$ "outer product"

$\underbrace{x \circ y}_{\text{n-bit vectors}}$ ... $\underbrace{(x_1y_1, \ldots, x_ny_n)}_{n^2 \text{ bit vector}}$

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 $\Pr_{\bar{r}}[A \cdot \underbrace{B \cdot \bar{r}}_{A\cdot(B\cdot\bar{r})} \neq \underbrace{C \cdot \bar{r}}_{\text{take}}] \geq 1/2$

$A\cdot(B\cdot\bar{r})$ take $O(n^2)$ to compute

also true for "= mod 2"

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

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

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

$(2^{n-1}$ pairs$)$

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

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

so $\bar{a}\cdot\bar{s} \neq \bar{b}\cdot\bar{s}$

note 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

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

what if these answers were written for you?

why should you believe they are correct?

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 $\bar{a} = (0,\ldots 0)$ 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 algorithm for n-bit vector in strange model setting: given vector $\bar{a} = (a_1, a_2, \dots a_n)$

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 $\bar{a}\cdot\bar{y}$

$\bar{r}$	answer vector
$\bar{a}\cdot(0,0,\dots 0)$	0
$\bar{a}\cdot(0,0,\dots 1)$	0
$\bar{a}\cdot(0,0,\dots 1,0)$	0
$\bar{a}\cdot(0,0,\dots 1,1)$	0
	0

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

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 in answer vector ~ write all 0's

How can we check proof doesn't cheat?

test proof? on $\bar{r}$'s 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

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

$\boxed{1\ 0\ 1\ \ldots}$

V's protocol given formula & a:

Pick random clause C_i & check if a satisfies

$F = (x_1 \vee \bar{x}_2 \vee x_3)(x_2 \vee \bar{x}_3 \vee x_4)$

$a = (x_1 = T, x_2 = F, x_3 = F, x_4 = F, \ldots)$

pick clause 1

good? if a satisfies z, always passes

if a doesn't satisfy z, at least one clause not satisfied

$\Rightarrow \Pr[\text{pick unsatisfied clause}] \geq \frac{1}{\#\text{clauses}}$ ☹

Arithmetization of 3SAT:

Boolean formula F $\Longleftrightarrow$ arithmetic formula $A(F)$ over $\underbrace{\mathbb{Z}_2}_{\text{mod } 2}$

$$T \Longleftrightarrow 1$$

$$F \Longleftrightarrow 0$$

$$x_i \Longleftrightarrow x_i$$

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

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

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

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

example: $x_1 \vee \bar{x}_2 \vee x_3 \Longleftrightarrow 1 - (1-x_1)(\overbrace{x_2}^{1-(1-x_2)})(1-x_3)$

Key point: F satisfied by assignment 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 $y_{i_j} \in \{ x_1 \dots x_n, \bar{x}_1 \dots \bar{x}_n \}$

$T \iff 1$

$F \iff 0$

$x_i \iff x_i$

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

$\alpha \wedge \beta \iff \alpha \cdot \beta$

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

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

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

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

$\Rightarrow$ evaluates to 0 if x satisfies C_i

$\Rightarrow C^{\sigma}(x) = (0, \dots 0)$ if x satifies F

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

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

Need to convince V that $C^{\sigma}(a) = (\hat{C}_1(a), \hat{C}_2(a) \dots) = (0, \dots 0)$ 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 $\mathcal{C}(a)\cdot r \quad \forall \ r \in \{0,1\}^n$
- if $\forall \ i, \ \hat{C}_i(a) = 0, \quad \Pr_r[\mathcal{C}(a)\cdot r = 0] = 1$

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

  $\qquad \Pr[\mathcal{C}(a)\cdot r = 1]$

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

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

$\mathcal{C}(a) = (\hat{C}_1(a), \hat{C}_2(a), \ldots) \overset{?}{=} (0, 0, \ldots 0)$

complement

mod 2 arithmetic

$T \Leftrightarrow 1$

$F \Leftrightarrow 0$

$x_i \Leftrightarrow x_i$

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

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

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

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

why believe proof? can write all 0's even if $\mathcal{C}(a)\cdot r \neq 0$

$\Rightarrow$ will need to do more

V doesn't know

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

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

V does know: depend on r_i's + coeffs in $\hat{C}_i$

example

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

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

$$\Rightarrow \hat{C}_1(a) = 1 - a_1 - a_2 + a_1 a_2$$

since complement of

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

$$\Rightarrow \hat{C}_2(a) = a_1 - a_1 a_2$$

$$\sum_{i=1}^{2} r_i \cdot \hat{C}_i(a) = r_1(1 - a_1 - a_2 + a_1 a_2) + r_2(a_1 - a_1 a_2)$$

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

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

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

$\hat{C}(a) = (\hat{C}_1(a), \hat{C}_2(a), \dots) \overset{?}{=} (0, 0, \dots 0)$

complement

$T \Leftrightarrow 1$

$F \Leftrightarrow 0$

$x_i \Leftrightarrow x_i$

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

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

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

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

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

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 all 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 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 g

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
$\forall x, \ \Pr[\text{output} = g(x)] \geq 1-\beta$

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