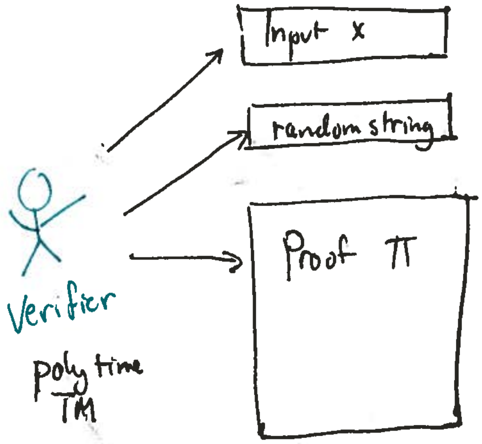
Probabilistically Checkable Proofs



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

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

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

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

where V uses at most $r(n)$ random bits

+ makes at most $q(n)$ queries to π
1 bit each

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

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

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

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

3SAT:

$F = \bigwedge C_i$

↖ ith 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 $\mathbb{Z}_2$

$T \Leftrightarrow 1$

$F \Leftrightarrow 0$

$x_i \longleftrightarrow x_i$

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

$\alpha \wedge \beta \longleftrightarrow \alpha \cdot \beta$

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

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

examples

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

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

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

- F satisfied by $\bar{a}$ iff $A(\bar{a})=1$
- F satisfiable iff $A(F) \neq 1$

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

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

$\boxed{\text{NOTE}}$ 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!!

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

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 $\bar{a}$

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

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

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

* see also p 6.3a

why? remember the pairing argument we did when proving that Fourier coeffs are orthogonal? same argument works here.

But: Why believe the birdie?

does it help? 1) We know r_i's

2) we know 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

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

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

Useful Fact

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

Freivald's test $\Rightarrow$ if matrices $A \cdot B \neq C$ then $\Pr_{\bar{r}}[A \cdot B \cdot \bar{r} \neq C \cdot \bar{r}] \geq 1/2$

also true for equality mod 2

Pf. pairing vectors that differ in coordinate where $a_i \neq b_i$ or $A \cdot B_{ij} \neq C_{ij}$

(as in proof of orthogonality of parity basis) ∎

(hopefully all of same $\bar{a}$)

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

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

V knows this

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

outer product: if $z = b \circ c$ (n^2 vector; n vectors)

$z_{ij} = b_i \cdot c_j$

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

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 $\bar{x}, \bar{y}, \bar{z}$

- we only need the values at one input !!
- 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 $\bar{a}$

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

  - test that self-corrections are consistent

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

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

How to do (2):

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

$\boxed{\#\text{random bits} = O(n^3) \quad \#\text{queries} = O(1) \quad \text{runtime} = O(n^3)}$

- From now on, use self-corrector to get $sc\text{-}\tilde{A}, sc\text{-}\tilde{B}, sc\text{-}\tilde{C}$ for all inputs ($sc\text{-}\tilde{A} \Leftrightarrow a$, $sc\text{-}\tilde{B} \Leftrightarrow b$, $sc\text{-}\tilde{C} \Leftrightarrow c$)
  - 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)

$\boxed{\text{Per query to s-c: } \#\text{random bits} = O(1) \quad \#\text{queries} = O(1) \quad \text{runtime} = O(n^3)}$

- Consistency test:

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

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

Tester:

Pick random $\bar{x}_1, \bar{x}_2, \bar{x}, \bar{y}$

Test that $sc\text{-}\tilde{A}(\bar{x}_1) \cdot sc\tilde{A}(\bar{x}_2) \left(= \sum a_i x_{1i} \cdot \sum a_j x_{2j} = \sum_{i,j} a_i a_j x_{1i} x_{2j}\right)$

$= sc\tilde{B}(\bar{x}_1 \circ \bar{x}_2)$

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

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

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

Note: these are not uniformly distributed vectors

$\boxed{\#\text{random bits} = O(n^3) \quad \#\text{queries} = O(1) \quad \text{runtime} = O(n^3)}$

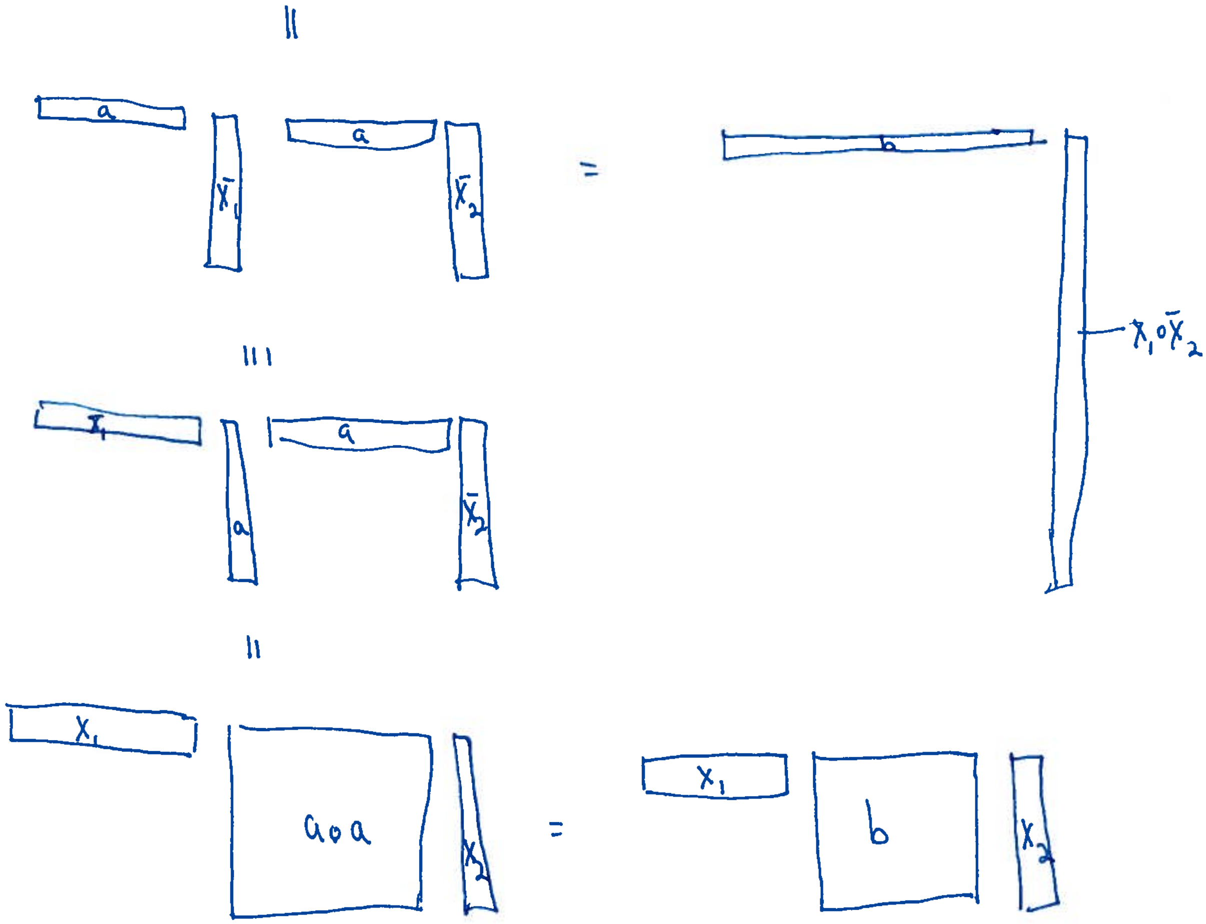
Does it work? Given, $\text{sc-}\tilde{A}$, $\text{sc-}\tilde{B}$ + $\text{sc-}\tilde{C}$ are linear fctns
($\tilde{A} = A$, $\tilde{B} = B$, $\tilde{C} = C$)

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

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 matrices $A \cdot B \neq C$ 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, $a\circ a$, $a\circ a\circ a$
- 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

| # random bits $= O(n)$
| # queries $= O(1)$

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 $(0 0 \ldots 0) \not\equiv (\hat{C}_1(a) \ldots \hat{C}_m(\bar{a}))$

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 $= \frac{1}{2^k}$ ✓