Foundations of Probabilistic Proofs

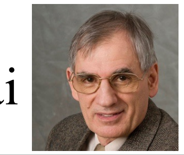
A course by **Alessandro Chiesa**

Lecture 11

PCPs with Sublinear Verification

PCP for NEXP

Non-Deterministic Exponential Time has Two-Prover Interactive Protocols

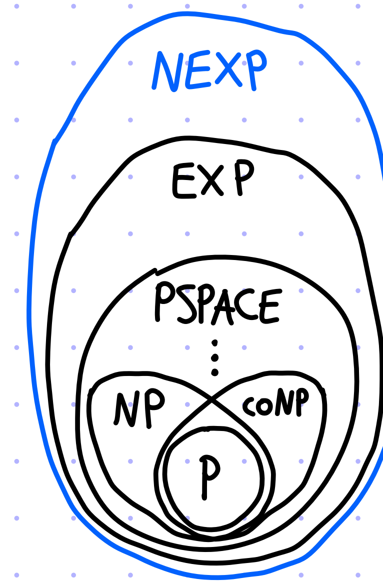
László Babai  Lance Fortnow  Carsten Lund 

So far we constructed PCPs for NP:

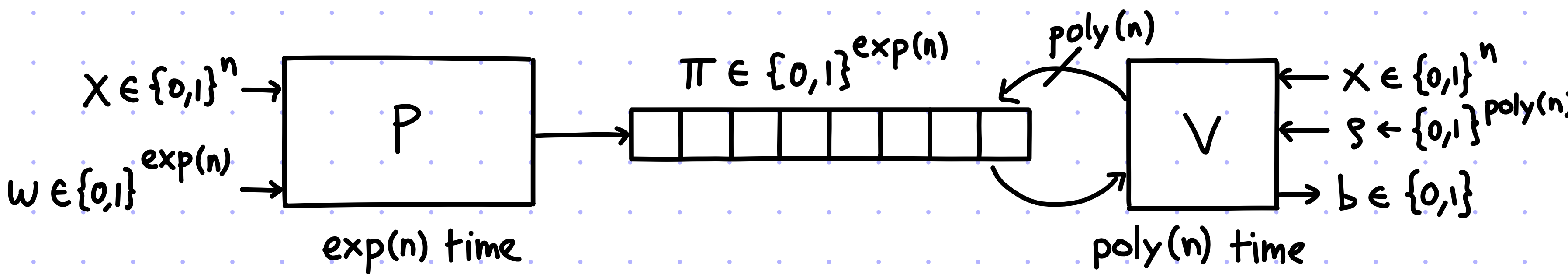
$NP \subseteq PCP[\varepsilon_c = 0, \varepsilon_s = 1/2, \Sigma = \{0,1\}, \ell = \exp(n), q = O(1), r = \text{poly}(n)]$

$NP \subseteq PCP[\varepsilon_c = 0, \varepsilon_s = 1/2, \Sigma = \{0,1\}, \ell = \text{poly}(n), q = \text{poly}(\log n), r = O(\log n)]$

Today we construct a PCP for NEXP:



<u>theorem</u>: $NEXP \subseteq PCP[\varepsilon_c = 0, \varepsilon_s = \frac{1}{2}, \Sigma = \{0,1\}, \ell = \exp(n), q = \text{poly}(n), r = \text{poly}(n)]$



In a prior lecture we proved that $PCP \subseteq NEXP$, so we conclude that $PCP = NEXP$.

- $\ell = \exp(n)$ is the correct regime since the witness and computation have size $\exp(n)$
- $q = \text{poly}(n)$ is exponentially smaller than witness and computation size
- PCP verifier time is $\text{poly}(n)$ (by definition), exponentially smaller than original computation

The first example of "VERIFICATION FASTER THAN COMPUTATION" that we see for PCPs.

Towards Sublinear Verification

To achieve sublinear verification we must:

① consider a problem where $|\text{description}| \ll |\text{computation}|$

② design a PCP verifier that only uses the description (does not "unroll" the computation)

---

① We have seen examples when constructing IPs for "large classes":

Ex: in #SAT we are given a boolean formula $\varphi:\{0,1\}^n \to \{0,1\}$ and $v \in \mathbb{N}$, and must check

$$\left|\{a \in \{0,1\}^n \mid \varphi(a)=1\}\right| \stackrel{?}{=} v$$

Ex: in TQBF we are given a boolean formula $\varphi:\{0,1\}^n \to \{0,1\}$ and must check

$$\forall x_1 \exists x_2 \forall x_3 \cdots \quad \varphi(x_1,\ldots,x_n) \stackrel{?}{=} 1$$

In both cases the description has size $|\varphi|$ but the "computation" has size $2^n \cdot |\varphi|$.

In our lectures on PCPs we have not yet considered such problems.

We have built PCPs for NP-complete problems where $|\text{description}| \sim |\text{computation}|$:

$$\mathrm{QESAT}(\mathbb{F}) = \{(p_1,\ldots,p_m) \mid \exists\, a \in \mathbb{F}^n \text{ s.t. } p_1(a) = \cdots = p_m(a) = 0\}$$

Towards Sublinear Verification [2/2]

To achieve sublinear verification we must:

① consider a problem where |description| ≪ |computation|

② design a PCP verifier that only uses the description (does not "unroll" the computation)

---

② The PCPs that we designed operate on the computation, not its description:

PCP for QESAT($\mathbb{F}$)

$P((p_1,\dots,p_m),a)$

1. For every $\sigma \in \mathbb{F}^{S_e}$:
   - $p_\sigma := T(p_1,\dots,p_m;\sigma)$
   - $\Pi_{sc}[\sigma] :=$ eval table for sumcheck claim "$p_\sigma(a)=0$"
   - output $\Pi_{sc}[\sigma]$
2. Output $\hat{a}:\mathbb{F}^{S_v}\to\mathbb{F}$ as Π_a.

   (The LDE of $a:[n]\to\mathbb{F}$.)

$\left\{ \boxed{\Pi_{sc}[\sigma]} \right\}_{\sigma\in\mathbb{F}^{S_e}}$

$\boxed{\Pi_a}$

$V((p_1,\dots,p_m))$

1. Sample $\sigma \in \mathbb{F}^{S_e}$.
2. Compute $p_\sigma := \sum_{0\le j_1,\dots,j_{S_e}<|H_e|} \sigma_1^{j_1}\cdots\sigma_{S_e}^{j_{S_e}}\cdot p_{j_1\dots j_{S_e}}$.
3. Run sumcheck to check that $p_\sigma(a)=0$:

   $$\sum_{\alpha,\beta\in H_v^{S_v}} \hat{C}_\sigma(\alpha,\beta)\cdot\hat{a}(\alpha)\cdot\hat{a}(\beta)=0$$

   $\boxed{V_{sc}(\mathbb{F}, H_v, \underbrace{2S_v}_{\text{vars}}, \underbrace{0}_{\text{sum}}, \underbrace{2\cdot(|H_v|-1)}_{\text{degree}})}$

4. Run (individual) low-degree test on Π_a:

   $\boxed{V_{LDT}(\mathbb{F}, \underbrace{S_v}_{\text{vars}}, \underbrace{|H_v|-1}_{\text{ind degree}})}$

computing p_σ and evaluating $\hat{C}_\sigma$ takes time poly(m,n) even if $(p_1,\dots,p_m)$ has "structure"

Interlude: Cook-Levin Theorem

We review the ideas that underlie the Cook-Levin theorem:

theorem: SAT is NP-hard   ($\forall L \in NP$, L is polynomial-time reducible to SAT)

Fix $L \in NP$ and let M_L be a nondeterministic Turing machine that decides L.

For an instance x, $M_L(x)$ accepts if and only if $x \in L$.

The proof expresses a valid (and accepting) execution of $M_L(x)$ as a SAT instance φ.

So let us review nondeterministic Turing machines.

A nondeterministic Turing machine M is specified by:

Q states

I input alphabet

Γ tape alphabet, $I \subseteq \Gamma$, $\sqcup \in \Gamma \setminus I$

$f : Q \times \Gamma \to 2^{Q \times \Gamma \times \{L,R\}}$ nondeterministic transition function; $2^{(\cdot)}$: power set (set of all subsets)

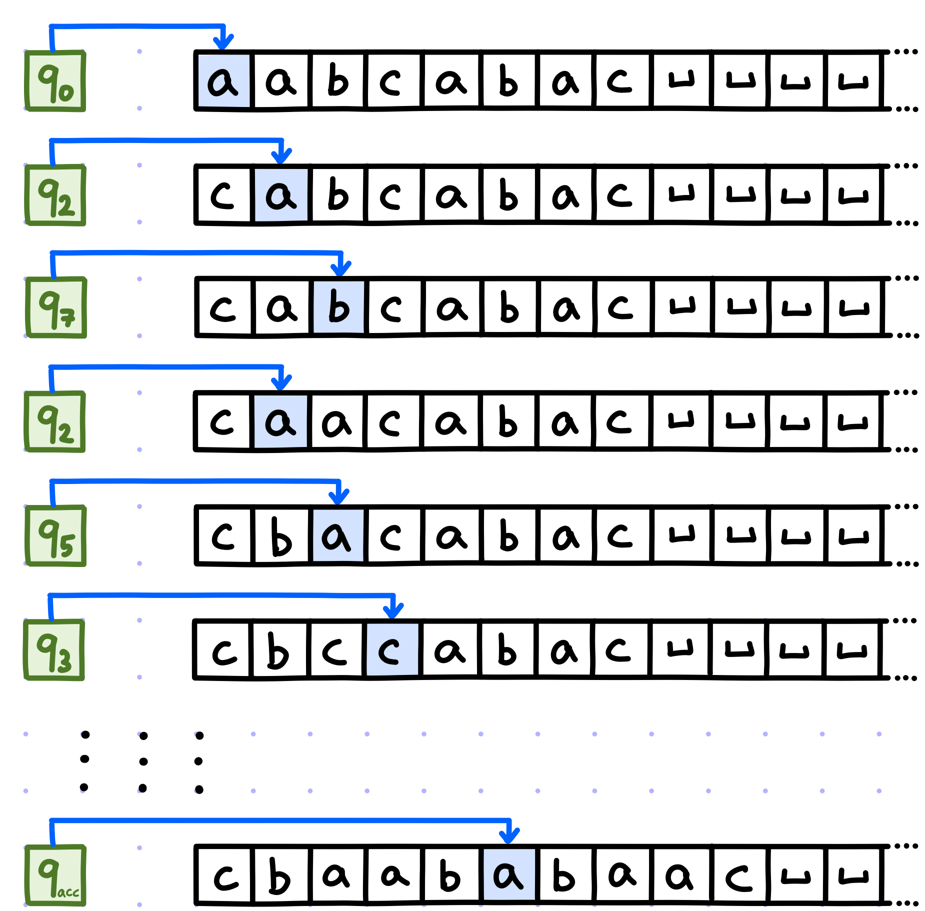
q_0 start state

q_{accept} accept state

A configuration of M is $(\underset{\text{state}}{q}, \underset{\text{head position}}{j}, \underset{\text{tape contents}}{\sigma})$.

We encode the configuration as a string $\#\sigma_1 q \sigma_2 \#$ (delimiter: $\#$)

where j is the first symbol of σ_2 in $\sigma = \sigma_1\sigma_2$.



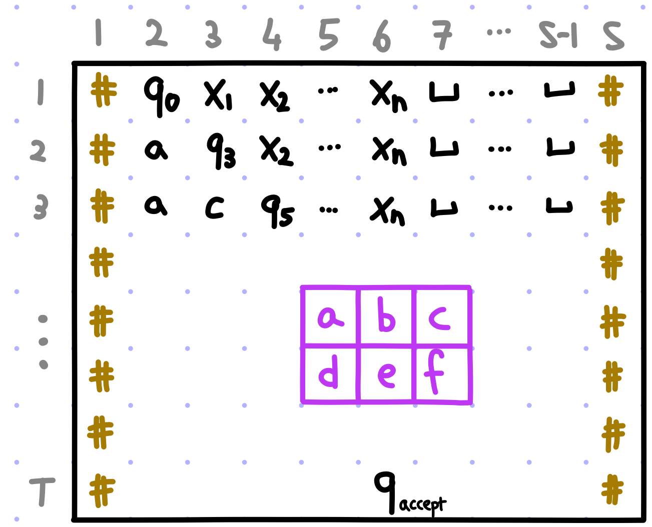
Interlude: Cook-Levin Theorem [2/2]

A computation trace of $M(x)$ is a valid list of configurations starting from $(q_0, 1, x\sqcup\sqcup\cdots)$.

We design φ s.t. $\varphi(z)=1 \leftrightarrow z$ is a (boolean encoding of) computation trace of $M(x)$ that accepts.

Suppose $M(x)$ runs in time T (# of configurations) and space S (longest encoded configuration).

The variables are $\{z_{i,j,\sigma}\}_{i\in[T],\, j\in[S],\, \sigma\in\Sigma}$ where $\Sigma := Q \cup \Gamma \cup \{\#\}$.



The SAT formula φ has 4 parts: $\varphi = \varphi_{cell} \wedge \varphi_{start} \wedge \varphi_{move} \wedge \varphi_{accept}$.

- each cell contains exactly one symbol:

$$\varphi_{cell} := \bigwedge_{i\in[T]} \bigwedge_{j\in[S]} \left[\left(\bigvee_{\sigma\in\Sigma} z_{i,j,\sigma} \right) \wedge \left(\bigwedge_{\substack{\sigma,\tau\in\Sigma \\ \sigma\neq\tau}} \overline{z_{i,j,\sigma} \wedge z_{i,j,\tau}} \right) \right]$$

- starting configuration is $(q_0, 1, x\sqcup\sqcup\cdots)$:

$$\varphi_{start} := z_{1,1,\#} \wedge z_{1,2,q_0} \wedge \left(\bigwedge_{j=1}^{n} z_{1,2+j,x_j} \right) \wedge \left(\bigwedge_{j=n+3}^{S-1} z_{1,j,\sqcup} \right) \wedge z_{1,S,\#}$$

- ending configuration contains q_{accept}:

$$\varphi_{accept} := \bigvee_{j\in[S]} z_{T,j,q_{accept}}$$

- each 2x3 window is legal:

$$\left\{ \begin{smallmatrix} a & b & c \\ a & b & c \end{smallmatrix}, \begin{smallmatrix} a & b & c \\ a & b & q \end{smallmatrix}, \begin{smallmatrix} a & b & c \\ q & b & c \end{smallmatrix}, \begin{smallmatrix} a & q & c \\ a & d & q' \end{smallmatrix}, \cdots \right\} \in W$$

$$\left\{ \begin{smallmatrix} a & b & c \\ a & d & c \end{smallmatrix}, \begin{smallmatrix} a & b & c \\ a & q & c \end{smallmatrix}, \begin{smallmatrix} a & q & c \\ a & b & c \end{smallmatrix}, \begin{smallmatrix} a & q & c \\ q' & d & q'' \end{smallmatrix}, \cdots \right\} \notin W$$

$$\varphi_{move} := \bigwedge_{1\leq i<T} \bigwedge_{1<j<S} \bigvee_{\begin{smallmatrix} a & b & c \\ d & e & f \end{smallmatrix} \in W} \left(\begin{matrix} z_{i,j-1,a} \wedge z_{i,j,b} \wedge z_{i,j+1,c} \\ \wedge z_{i+1,j-1,d} \wedge z_{i+1,j,e} \wedge z_{i+1,j+1,f} \end{matrix} \right)$$

where W is the set of legal windows.

A NEXP-Complete Problem

def: OSAT := $\left\{ (m,n,\varphi) \;\middle|\; \begin{array}{l} m,n \in \mathbb{N},\ \varphi : \{0,1\}^{m+3n+3} \to \{0,1\} \text{ boolean formula} \\ \exists A : \{0,1\}^n \to \{0,1\}\ \forall w \in \{0,1\}^m\ \forall v_1,v_2,v_3 \in \{0,1\}^n\ \varphi(w,v_1,v_2,v_3,A(v_1),A(v_2),A(v_3)) = 0 \end{array} \right\}$.

claim: OSAT is NEXP-complete

proof: Suppose that $L \in$ NEXP and let M be a NEXP machine deciding L.
Let x be an input to M.

By the Cook-Levin Theorem (& SAT→3SAT reduction), can reduce (M,x) to a 3CNF Φ_x s.t.

- Φ_x has $N_v = 2^{\mathrm{poly}(|x|)}$ variables (and $N_c = 2^{\mathrm{poly}(|x|)}$ clauses),
- $M(x)=1 \leftrightarrow \exists A : [N_v] \to \{0,1\}\ \ \Phi_x(A) = 1$.

Set $n = \log N_v = \mathrm{poly}(|x|)$, and relabel $[N_v]$ as $\{0,1\}^n$.

Moreover, $\exists$ poly(|x|)-size circuit $D_x : \{0,1\}^{3n+3} \to \{0,1\}$ that specifies Φ_x's clauses:

$D_x(v_1,v_2,v_3,c_1,c_2,c_3) = 1 \leftrightarrow \Phi_x$ contains clause $\bigvee_{i=1}^{3} (x_{v_i} \oplus c_i)$

Therefore $M(x)=1 \leftrightarrow \exists A : \{0,1\}^n \to \{0,1\}$ s.t.

$$\forall v_1,v_2,v_3 \in \{0,1\}^n\ \ \forall c_1,c_2,c_3 \in \{0,1\}\ \ D_x(v_1,v_2,v_3,c_1,c_2,c_3) \wedge \overline{\left(\bigvee_{i=1}^{3} A(v_i) \oplus c_i \right)} = 0 .$$

A NEXP-Complete Problem

<u>def:</u> $\text{OSAT} := \left\{ (m,n,\varphi) \;\middle|\; \begin{array}{l} m,n \in \mathbb{N},\ \varphi:\{0,1\}^{m+3n+3} \to \{0,1\} \text{ boolean formula} \\ \exists A:\{0,1\}^n \to \{0,1\}\ \forall w \in \{0,1\}^m\ \forall v_1,v_2,v_3 \in \{0,1\}^n\ \varphi(w,v_1,v_2,v_3,A(v_1),A(v_2),A(v_3)) = 0 \end{array} \right\}.$

<u>claim:</u> OSAT is NEXP-complete

<u>proof:</u> [continued]

Therefore $M(x)=1 \leftrightarrow \exists A:\{0,1\}^n \to \{0,1\}$ s.t.

$$\forall v_1,v_2,v_3 \in \{0,1\}^n\ \forall c_1,c_2,c_3 \in \{0,1\}\ D_x(v_1,v_2,v_3,c_1,c_2,c_3) \wedge \overline{\left(\bigvee_{i=1}^{3} A(v_i) \oplus c_i\right)} = 0 .$$

Reduce the boolean circuit D_x to a boolean formula $\Psi_x:\{0,1\}^{m'+3n+3} \to \{0,1\}$ with $m' = O(|D_x|) = \text{poly}(|x|)$ and $|\Psi_x| = O(|D_x|) = \text{poly}(|x|)$ s.t.

$$\forall v_1,v_2,v_3 \in \{0,1\}^n\ \forall c_1,c_2,c_3 \in \{0,1\}\ D_x(v_1,v_2,v_3,c_1,c_2,c_3) = 1 \leftrightarrow \exists w' \in \{0,1\}^{m'}\ \Psi_x(v_1,v_2,v_3,c_1,c_2,c_3,w') = 1 .$$

Define $\varphi(w,v_1,v_2,v_3,a_1,a_2,a_3) := \Psi_x(v_1,v_2,v_3,c_1,c_2,c_3,w') \wedge \overline{\left(\bigvee_{i=1}^{3} a_i \oplus c_i\right)}$

where $w = (c_1,c_2,c_3,w') \in \{0,1\}^m$ and $m = 3 + m'$.

In sum, $M(x) = 1 \leftrightarrow \exists A:\{0,1\}^n \to \{0,1\}$ s.t.

$$\forall w \in \{0,1\}^m\ \forall v_1,v_2,v_3 \in \{0,1\}^n\ \varphi(w,v_1,v_2,v_3,A(v_1),A(v_2),A(v_3)) = 0 . \qquad \blacksquare$$

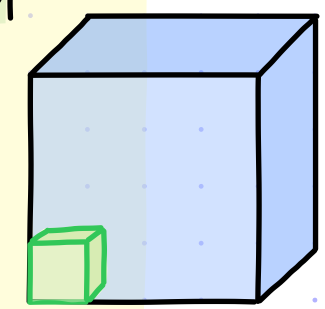
Part 1: Arithmetization of OSAT

<u>claim:</u> There is a polynomial-time transformation T s.t.

① $T(\mathbb{F},(m,n,\varphi))$ outputs a circuit $\hat{\varphi}: \mathbb{F}^{m+3n+3} \to \mathbb{F}$ s.t. $|\hat{\varphi}| \leq |\varphi|$ and $\deg_{tot}(\hat{\varphi}) \leq |\varphi|$

② $(m,n,\varphi) \in OSAT \leftrightarrow \exists$ multilinear $\hat{A}: \mathbb{F}^n \to \mathbb{F}$ s.t.

- booleanity: $\hat{A}$ is boolean on $\{0,1\}^n$
- zero on subcube: $\forall w \in \{0,1\}^m \ \forall v_1,v_2,v_3 \in \{0,1\}^n \ \hat{\varphi}(w,v_1,v_2,v_3,\hat{A}(v_1),\hat{A}(v_2),\hat{A}(v_3)) = 0$



<u>proof:</u>

The transformation T outputs $\hat{\varphi} :=$ arithmetize$(\mathbb{F},\varphi)$.

[Recall: $x \wedge y \mapsto x \cdot y$, $x \vee y \mapsto 1-(1-x)\cdot(1-y)$, $\bar{x} \mapsto 1-x$.]

This ensures that $|\hat{\varphi}| \leq |\varphi|$ and $\deg_{tot}(\hat{\varphi}) \leq |\varphi|$ and $\hat{\varphi} \equiv \varphi$ on every boolean input.

<u>COMPLETENESS:</u> if $A:\{0,1\}^n \to \{0,1\}$ is a witness for $(m,n,\varphi) \in OSAT$ then $\hat{A}$ = "multilinear extension of A" satisfies the booleanity and the zero-on-subcube conditions.

$$\hat{A}(x) = \sum_{u \in \{0,1\}^n} A(u) \cdot \prod_{i=1}^{n} (u_i x_i + (1-u_i)(1-x_i))$$

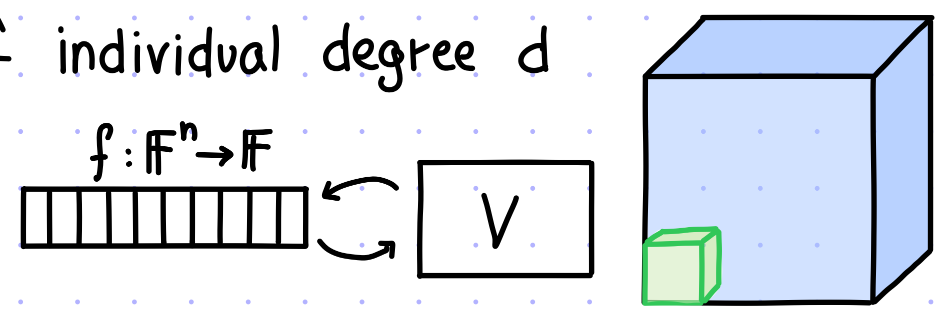
<u>SOUNDNESS:</u> if $(m,n,\varphi) \notin OSAT$ then $\forall$ multilinear $\hat{A}: \mathbb{F}^n \to \mathbb{F}$

either $\hat{A}$ is not boolean on $\{0,1\}^n$

or $\exists w \in \{0,1\}^m \ \exists v_1,v_2,v_3 \in \{0,1\}^n \ \hat{\varphi}(w,v_1,v_2,v_3,\hat{A}(v_1),\hat{A}(v_2),\hat{A}(v_3)) = \varphi(w,v_1,v_2,v_3,\hat{A}(v_1),\hat{A}(v_2),\hat{A}(v_3)) \neq 0$. ■

Part 2: Zero-on-Subcube Test

Given oracle access to $f: \mathbb{F}^n \to \mathbb{F}$ that is δ-close to $\hat{f}$ of individual degree d check that $\hat{f}|_{H^n} \equiv 0$.



Idea #1: query f at every point in H^n and check if 0

Problem: if even 1 corruption is in H^n then test may accept w.p. 1 even if $\hat{f}|_{H^n} \not\equiv 0$ $\to$ test is not sound

Idea #2: locally correct the value of f at every point in H^n (and check if 0)

Problem: $|H|^n$ is too many queries

Idea #3: run sumcheck protocol on sum of squares $\sum_{a \in H^n} \hat{f}(a)^2 \stackrel{?}{=} 0$

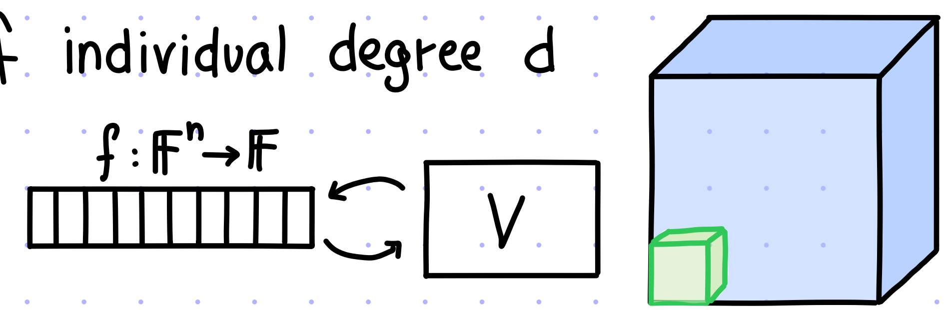
Problem: for every finite field $\mathbb{F}$, $\sum_i c_i^2 = 0 \not\Rightarrow \forall i \; c_i = 0$ over $\mathbb{F}$

If $\mathrm{char}(\mathbb{F}) > 0$ (e.g. $\mathbb{F}$ is finite) then the implication does not hold over $\mathbb{F}$.

The implication holds for some fields with $\mathrm{char}(\mathbb{F}) = 0$ (e.g. $\mathbb{R}$ and $\mathbb{Q}$ but not $\mathbb{C}$).

Part 2: Zero-on-Subcube Test

Given oracle access to $f: \mathbb{F}^n \to \mathbb{F}$ that is δ-close to $\hat{f}$ of individual degree d check that $\hat{f}|_{H^n} \equiv 0$.



Final Idea: randomized reduction to sumcheck

Let $\text{int}: H \to \{0,1,\dots,|H|-1\}$ be an efficiently computable bijection.

Consider the polynomial
$$g(x_1,\dots,x_n) := \sum_{a_1,\dots,a_n \in H} \hat{f}(a_1,\dots,a_n)\, x_1^{\text{int}(a_1)} \cdots x_n^{\text{int}(a_n)} .$$

If $\hat{f}|_{H^n} \equiv 0$ then $g \equiv 0$.

If $\hat{f}|_{H^n} \not\equiv 0$ then $g \not\equiv 0$, so $\Pr_{\sigma_1,\dots,\sigma_n \in \mathbb{F}^n}[g(\sigma_1,\dots,\sigma_n)=0] \leq \frac{n\cdot(|H|-1)}{|\mathbb{F}|}$.

Hence it suffices to check that $\sum_{a_1,\dots,a_n \in H} \hat{f}(a_1,\dots,a_n)\, \sigma_1^{\text{int}(a_1)} \cdots \sigma_n^{\text{int}(a_n)}$ for random $\sigma_1,\dots,\sigma_n \in \mathbb{F}$.

To make the addend a polynomial: $\forall \sigma \in \mathbb{F}$ define $\hat{\sigma}(x) := \sum_{a \in H} \sigma^{\text{int}(a)} \cdot L_{a,H}(x)$.

a-th Lagrange polynomial over H

In sum it suffices to run sumcheck on this claim:

$$\sum_{a_1,\dots,a_n \in H} \hat{f}(a_1,\dots,a_n)\, \hat{\sigma}_1(a_1) \cdots \hat{\sigma}_n(a_n) \quad \text{for random } \sigma_1,\dots,\sigma_n \in \mathbb{F}.$$

Part 2: Zero-on-Subcube Test

$\hat{f}|_{H^n} \stackrel{?}{\equiv} 0$

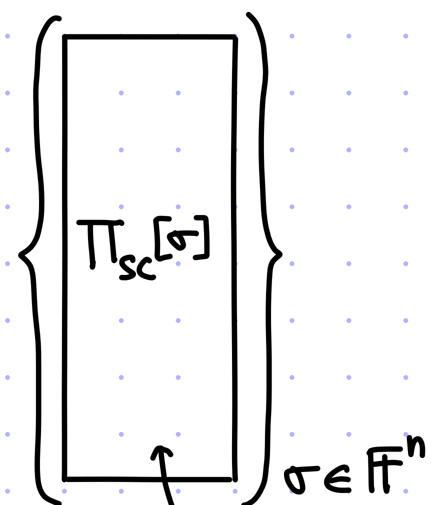
$P(\mathbb{F}, H, n, f)$

For every $\sigma_1, \dots, \sigma_n \in \mathbb{F}$:

output eval table $\Pi_{sc}[\sigma_1, \dots, \sigma_n]$ of

IP prover for sumcheck claim

$$\sum_{a_1,\dots,a_n \in H} \hat{f}(a_1,\dots,a_n) \cdot \prod_{i\in[n]} \hat{\sigma}_i(a_i) = 0$$



δ-close to $\hat{f} \in \mathbb{F}^{\leq d}[X_1,\dots,X_n]$

$V^{f:\mathbb{F}^n \to \mathbb{F}}(\mathbb{F}, H, n, d)$

Sample $\sigma_1, \dots, \sigma_n \in \mathbb{F}$.

Run sumcheck for the claim:

$$\sum_{a_1,\dots,a_n \in H} \hat{f}(a_1,\dots,a_n) \prod_{i\in[n]} \hat{\sigma}_i(a_i) = 0.$$

$V_{sc}(\mathbb{F}, H, n, 0, d)$

field domain vars sum degree

$(\rho_1,\dots,\rho_n) \Rightarrow \hat{f}(\rho_1,\dots,\rho_n) \cdot \prod_{i\in[n]} \hat{\sigma}_i(\rho_i)$

1. Query f at $(\rho_1,\dots,\rho_n)$.
2. For every $i \in [n]$: evaluate $\hat{\sigma}_i$ at ρ_i.

proof length: $|\Pi_{sc}| = |\mathbb{F}|^n \cdot O(|\mathbb{F}|^n \cdot (|H|+d)) = |\mathbb{F}|^{O(n)} \cdot (|H|+d)$

query complexity:

- n queries to Π_{sc} (each retrieving $|H|+d$ elts)
- 1 random query to f

verifier time: $\mathrm{poly}(n, |H|, d)$ for V_{sc} + $n \cdot \mathrm{poly}(|H|)$ to evaluate $\{\hat{\sigma}_i\}_{i\in[n]}$

COMPLETENESS: if $f \equiv \hat{f} \wedge \hat{f}|_{H^n} \equiv 0$ then $\forall \sigma_1,\dots,\sigma_n \in \mathbb{F}$ $\sum_{a_1,\dots,a_n \in H} \hat{f}(a_1,\dots,a_n) \prod_{i\in[n]} \hat{\sigma}_i(a_i) = 0$ so V_{sc} accepts w.p. 1

SOUNDNESS: if $\Delta(f,\hat{f}) < \delta \wedge \hat{f}|_{H^n} \not\equiv 0$ then, except w.p. $\leq \frac{n \cdot (|H|-1)}{|\mathbb{F}|}$ over $\sigma_1,\dots,\sigma_n \in \mathbb{F}$, $\sum_{a_1,\dots,a_n \in H} \hat{f}(a_1,\dots,a_n) \prod_{i\in[n]} \hat{\sigma}_i(a_i) \neq 0$,

so V_{sc} accepts w.p. $\leq \frac{n \cdot (|H|-1+d)}{|\mathbb{F}|} + \delta$ (regardless of PCP string $\tilde{\pi}$).

Putting the Two Parts Together

$P((m,n,\varphi),A)$

1. Compute $\hat{\varphi} := T(\mathbb{F},(m,n,\varphi))$.
2. For every $\sigma \in \mathbb{F}^n$:

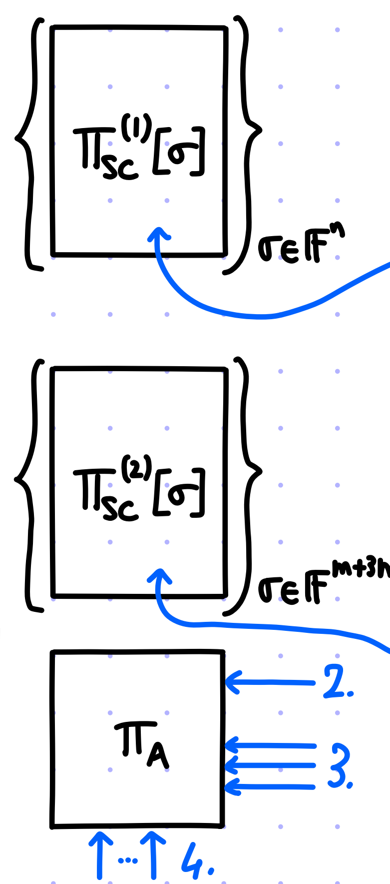
   output sumcheck proof $\pi_{SC}^{(1)}[\sigma]$ for

   $$\sum_{a\in\{0,1\}^n} \hat{A}(a)\cdot(\hat{A}(a)-1)\cdot\prod_{i\in[n]}\hat{\sigma}_i(a_i)=0 .$$
3. For every $\sigma \in \mathbb{F}^{m+3n}$:

   output sumcheck proof $\pi_{SC}^{(2)}[\sigma]$ for

   $$\sum_{\substack{a=(w,v_1,v_2,v_3)\\ \in\{0,1\}^{m+3n}}} \hat{\varphi}(w,v_1,v_2,v_3,\hat{A}(v_1),\hat{A}(v_2),\hat{A}(v_3))\cdot\prod_{i\in[m+3n]}\hat{\sigma}_i(a_i)=0$$
4. Output $\hat{A}:\mathbb{F}^n\to\mathbb{F}$ as π_A.

   (The multilinear extension of $A:\{0,1\}^n\to\{0,1\}$.)



$V((m,n,\varphi))$

1. Compute $\hat{\varphi} := T(\mathbb{F},(m,n,\varphi))$.
2. Sample $\sigma\in\mathbb{F}^n$ and run sumcheck for

   $$\sum_{a\in\{0,1\}^n} \hat{A}(a)\cdot(\hat{A}(a)-1)\cdot\prod_{i\in[n]}\hat{\sigma}_i(a_i)=0 .$$

   $\boxed{V_{SC}(\underset{\text{field}}{\mathbb{F}},\underset{\text{domain}}{\{0,1\}},\underset{\text{vars}}{n},\underset{\text{sum}}{0},\underset{\text{degree}}{3})}$

   $(\rho_1,\dots,\rho_n) \Rightarrow$
   - query π_A at $(\rho_1,\dots,\rho_n)$
   - for every $i\in[n]$: eval $\hat{\sigma}_i(x)$ at ρ_i
3. Sample $\sigma\in\mathbb{F}^{m+3n}$ and run sumcheck for

   $$\sum_{\substack{a=(w,v_1,v_2,v_3)\\ \in\{0,1\}^{m+3n}}} \hat{\varphi}(w,v_1,v_2,v_3,\hat{A}(v_1),\hat{A}(v_2),\hat{A}(v_3))\cdot\prod_{i\in[m+3n]}\hat{\sigma}_i(a_i)=0$$

   $\boxed{V_{SC}(\underset{\text{field}}{\mathbb{F}},\underset{\text{domain}}{\{0,1\}},\underset{\text{vars}}{m+3n},\underset{\text{sum}}{0},\underset{\text{degree}}{|\varphi|+1})}$

   $(\rho_1,\dots,\rho_{m+3n}) \Rightarrow$
   - query π_A at $(\rho_{m+1},\dots,\rho_{m+n})$
     $(\rho_{m+n+1},\dots,\rho_{m+2n})$
     $(\rho_{m+2n+1},\dots,\rho_{m+3n})$
   - for every $i\in[m+3n]$: eval $\hat{\sigma}_i$ at ρ_i
   - eval $\hat{\varphi}$ at $(\rho_1,\dots,\rho_{m+3n},\text{ans}_1,\text{ans}_2,\text{ans}_3)$
4. $V_{LDT}^{\pi_A}(\mathbb{F},\underset{\text{vars}}{n},\underset{\text{degree}}{\text{ind}\le 1})$

Analysis

$P((m,n,\varphi),A)$

1. Compute $\hat{\varphi} := T(\mathbb{F},(m,n,\varphi))$.
2. For every $\sigma \in \mathbb{F}^n$:

   output sumcheck proof $\Pi_{SC}^{(1)}[\sigma]$ for

$$\sum_{a\in\{0,1\}^n} \hat{A}(a)\cdot(\hat{A}(a)-1)\cdot\prod_{i\in[n]}\hat{\sigma}_i(a_i)=0.$$

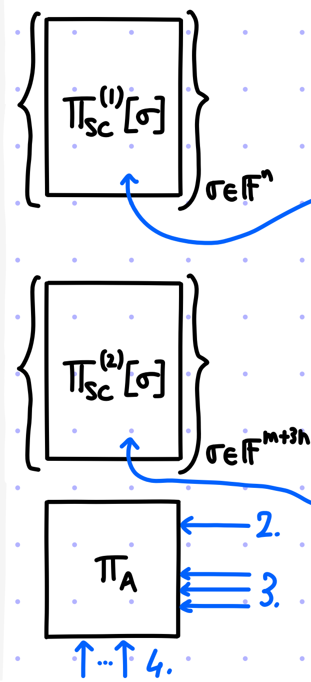
3. For every $\sigma \in \mathbb{F}^{m+3n}$:

   output sumcheck proof $\Pi_{SC}^{(2)}[\sigma]$ for

$$\sum_{\substack{a=(w,v_1,v_2,v_3)\\ \in\{0,1\}^{m+3n}}} \hat{\varphi}(w,v_1,v_2,v_3,\hat{A}(v_1),\hat{A}(v_2),\hat{A}(v_3))\cdot\prod_{i\in[m+3n]}\hat{\sigma}_i(a_i)=0$$

4. Output $\hat{A}:\mathbb{F}^n\to\mathbb{F}$ as Π_A.

   (The multilinear extension of $A:\{0,1\}^n\to\{0,1\}$.)



$V((m,n,\varphi))$

1. Compute $\hat{\varphi} := T(\mathbb{F},(m,n,\varphi))$.
2. Sample $\sigma \in \mathbb{F}^n$ and run sumcheck for

$$\sum_{a\in\{0,1\}^n} \hat{A}(a)\cdot(\hat{A}(a)-1)\cdot\prod_{i\in[n]}\hat{\sigma}_i(a_i)=0.$$

$$V_{SC}(\underset{\text{field}}{\mathbb{F}},\underset{\text{domain}}{\{0,1\}},\underset{\text{vars}}{n},\underset{\text{sum}}{0},\underset{\text{degree}}{3})$$

$(\rho_1,\dots,\rho_n) \Rightarrow$
- query Π_A at $(\rho_1,\dots,\rho_n)$
- for every $i\in[n]$: eval $\hat{\sigma}_i(x)$ at ρ_i

3. Sample $\sigma \in \mathbb{F}^{m+3n}$ and run sumcheck for

$$\sum_{\substack{a=(w,v_1,v_2,v_3)\\ \in\{0,1\}^{m+3n}}} \hat{\varphi}(w,v_1,v_2,v_3,\hat{A}(v_1),\hat{A}(v_2),\hat{A}(v_3))\cdot\prod_{i\in[m+3n]}\hat{\sigma}_i(a_i)=0$$

$$V_{SC}(\underset{\text{field}}{\mathbb{F}},\underset{\text{domain}}{\{0,1\}},\underset{\text{vars}}{m+3n},\underset{\text{sum}}{0},\underset{\text{degree}}{|\varphi|+1})$$

$(\rho_1,\dots,\rho_{m+3n}) \Rightarrow$
- query Π_A at $(\rho_{m+1},\dots,\rho_{m+n})$, $(\rho_{m+n+1},\dots,\rho_{m+2n})$, $(\rho_{m+2n+1},\dots,\rho_{m+3n})$
- for every $i\in[m+3n]$: eval $\hat{\sigma}_i$ at ρ_i
- eval $\hat{\varphi}$ at $(\rho_1,\dots,\rho_{m+3n},ans_1,ans_2,ans_3)$

4. $V_{LDT}^{\Pi_A}(\mathbb{F},\underset{\text{vars}}{n},\underset{\text{degree}}{\text{ind}\le 1})$

- soundness error

$$\max\left\{\varepsilon_{LDT}(\delta),\ 4\delta+O\left(\frac{n\cdot 3}{|\mathbb{F}|}\right)+O\left(\frac{(m+3n)\cdot(|\varphi|+1)}{|\mathbb{F}|}\right)\right\} = O(1)$$

- proof length (in field elements)

$$|\Pi_A|+|\Pi_{SC}^{(1)}|+|\Pi_{SC}^{(2)}| = |\mathbb{F}|^n+|\mathbb{F}|^n\cdot O(|\mathbb{F}|^n\cdot 1)+|\mathbb{F}|^{m+3n}\cdot O(|\mathbb{F}|^{m+3n}\cdot|\varphi|) = |\mathbb{F}|^{poly(m,n)} = 2^{poly(m,n,\log|\varphi|)}$$

setting $|\mathbb{F}|=poly(|\varphi|)$

- query complexity

$$(1+3+q_{LDT})+n\cdot O(1)+(m+3n)\cdot|\varphi| = poly(n)+poly(|\varphi|) = poly(|\varphi|)$$

- verifier time (in field operations)

$$poly(|\varphi|)+poly(n)+poly(|\varphi|)+t_{LDT} = poly(|\varphi|)$$

Bibliography

PCP for NEXP

In MIP model, uses arithmetization and zero check over $\mathbb{Q}$

- [BFL 1991]: Non-deterministic exponential time has two-prover interactive protocols, by László Babai, Lance Fortnow, Carsten Lund.
- [BFLS 1991]: Checking computations in polylogarithmic time, by László Babai, Lance Fortnow, Leonid Levin, Mario Szegedy.
- [BS 2008]: Short PCPs with polylog query complexity, by Eli Ben-Sasson, Madhu Sudan.
- [HS 2000]: Small PCPs with low query complexity, by Prahladh Harsha and Madhu Sudan.
- (Cook–Levin Theorem) by Michael Sipser.

Alternative zerochecks