Lecture B.8

PCPs with Sublinear Verification

Summer Graduate School on
Foundations and Frontiers of Probabilistic Proofs
2021.08.04

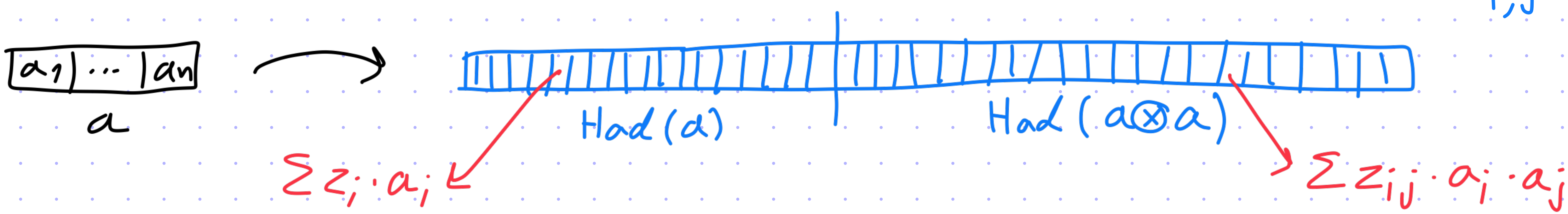
Recap: Hadamard PCP

NP-complete $\Rightarrow$ for all NP

We showed an exp-length, $O(1)$-local PCP for QESAT.

The PCP had 2 parts: (I) Hadamard enc. of sat. assignment $a_1, \dots, a_n$

(II) Hadamard enc. of $a \otimes a = (a_i \cdot a_j)_{i,j}$



We used the linear structure of the Hadamard code to:

1. check all equation at once using random linear comb.
2. check consistency between linear and quadratic terms.
3. locally test the Hadamard enc. using linearity-testing.
4. locally correct the Hadamard enc. using plurality votes.

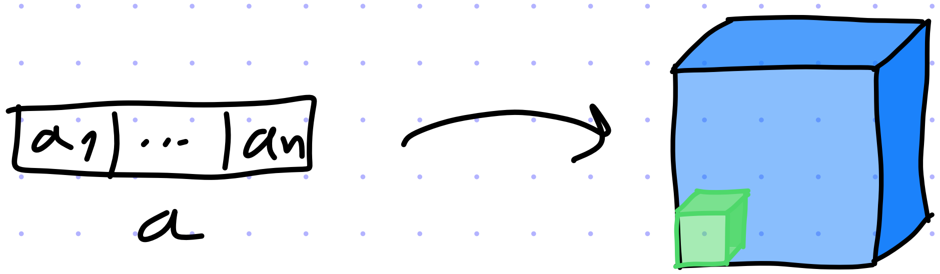
Can we do it without exp encoding?

Recap: Reed—Muller PCP

NP-complete ⇒ for all NP

We showed a poly-length, polylog-local PCP for QESAT.

The PCP was a Reed-Muller enc. of sat. assignment $a_1, \dots, a_n$



$f \in \mathbb{F}[x_1, \dots, x_s], \deg(f) \le d$

"$f(i) = a_i$"

Π_{sc} is eval table of IP prover for sumcheck claim

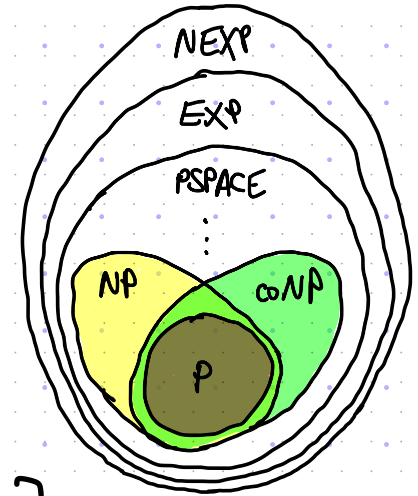
$$\sum_{\alpha, \beta \in H_v^{s_v}} g(\alpha, \beta) = 0$$

We used the structure of the Reed-Muller code to:

1. Reduce m equations to 1 using pseudo-random linear comb.
2. check an equation using the sumcheck protocol.
3. locally test the Reed-Muller enc. using low-deg. testing.

what's next?

PCP for NEXP



So far we constructed PCPs for NP:

$$NP \subseteq PCP[\varepsilon_c = 0, \varepsilon_s = 0.5, \Sigma = \{0,1\}, \ell = \exp(n), q = O(1), r = poly(n)]$$

$$NP \subseteq PCP[\varepsilon_c = 0, \varepsilon_s = 0.5, \Sigma = \{0,1\}, \ell = poly(n), q = poly(\log n), r = O(\log n)]$$

Today we construct a PCP for NEXP:

<u>theorem</u>: $NEXP \subseteq PCP[\varepsilon_c = 0, \varepsilon_s = 0.5, \Sigma = \{0,1\}, \ell = \exp(n), q = poly(n), r = poly(n)]$

<u>Remarks</u>:

- $\ell = \exp(n)$ is the correct regime since the witness and computation have size $\exp(n)$
- $q = poly(n)$ is exponentially smaller than witness and computation size
- the PCP verifier runs in poly(n) time, exponentially smaller than original computation!

this is the first instance of "verification faster than computation" that we see for PCPs!

Towards Sublinear Verification

To achieve sublinear verification we must

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

② design a PCP verifier that only uses 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 $\phi:\{0,1\}^n \to \{0,1\}$ and $v \in \mathbb{N}$, and must check

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

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

$$\forall x_1 \exists x_2 \forall x_3 \cdots \phi(x_1,\dots,x_n) \stackrel{?}{=} 1$$

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

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

We have built PCPs for NP-complete problems where |description| ~ |computation|:

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

Towards Sublinear Verification

To achieve sublinear verification we must

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

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

---

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

PCP for QESAT(F)

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

1. For every $r \in H_e^{S_e}$:
   - $p_r = T(p_1,\dots,p_m;r)$
   - $\Pi_{sc}[r] :=$ eval table for sumcheck to show $p_r(a)=0$
   - output $\Pi_{sc}[r]$
2. output $\hat{a}: \mathbb{F}^{S_v} \to \mathbb{F}$ [LDE of $a: [n] \to \mathbb{F}$]

$\Pi_{sc}[r]$

$r \in H_e^{S_e}$

$\hat{a}$

Π_a

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

1. sample $r \in H_e^{S_e}$ and compute $p_r := \sum_{0 \le j_1,\dots,j_{S_e} < |H_e|} r_1^{j_1} \cdots r_{S_e}^{j_{S_e}} \cdot p_{j_1 \dots j_{S_e}}$
2. run sumcheck to check that $\sum_{\alpha,\beta \in H_v^{S_v}} \hat{C}_r(\alpha,\beta)\,\hat{a}(\alpha)\,\hat{a}(\beta) = 0$

$V_{sc}(\mathbb{F}, H_v, 2S_v, 0)$

3. run low-degree test on Π_a

$V_{LDT}(\mathbb{F}, S_v, S_v|H_v|)$

computing p_r and evaluating $\hat{C}_r$ takes poly(m,n) time even if $(p_1,\dots,p_m)$ have "structure"

A NEXP-Complete Problem

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

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

<u>proof:</u> Suppose $L \in \text{NEXP}$ and let M be a NEXP machine deciding L.
Let x be an input to M. By the Cook–Levin Theorem, there is a 3CNF Φ_x s.t.

① Φ_x is satisfiable iff M accepts x

② Φ_x has $N_v = 2^{\text{poly}(|x|)}$ variables and $N_c = 2^{\text{poly}(|x|)}$ clauses — set $n := \log N_v$

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

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

Therefore, $x \in L$ iff $\exists A : \{0,1\}^n \to \{0,1\}$

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

A NEXP-Complete Problem

__def:__ $\text{OSAT} := \left\{ (m,n,\phi) \;\middle|\; \begin{array}{c} m,n \in \mathbb{N} \text{ and } \phi : \{0,1\}^{m+3n+3} \to \{0,1\} \text{ is a boolean formula s.t.} \\ \exists A : \{0,1\}^n \to \{0,1\} \text{ for which} \\ \forall w \in \{0,1\}^m \; \forall v_1,v_2,v_3 \in \{0,1\}^n \;\; \phi(w,v_1,v_2,v_3,A(v_1),A(v_2),A(v_3)) = 0 \end{array} \right\}$.

__claim:__ OSAT is NEXP-complete

__proof:__ [continued]

Therefore, $x \in L$ iff $\exists A : \{0,1\}^n \to \{0,1\}$

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

Finally, to make D_x a formula, apply the Cook-Levin Theorem to D_x to get a boolean formula $\psi : \{0,1\}^{m'+3n+3} \to \{0,1\}$ of size $\text{poly}(|D_x|) = \text{poly}(|x|)$ such that

$$D_x(v_1,v_2,v_3,c_1,c_2,c_3) = 1 \quad \text{iff} \quad \exists w' \in \{0,1\}^{m'} \; \psi(w',v_1,v_2,v_3,c_1,c_2,c_3) = 1$$

Now define

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

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

Part 1: Arithmetization of OSAT

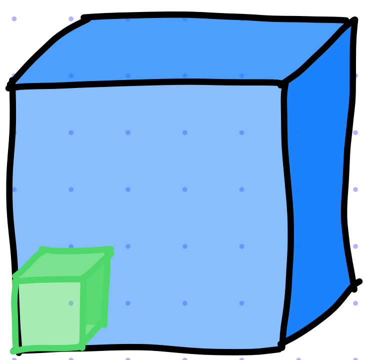
claim: there is a polynomial-time transformation T s.t.

① $T(\mathbb{F}, (m,n,\phi))$ outputs a circuit $\hat{\phi}: \mathbb{F}^{m+3n+3} \to \mathbb{F}$ of total degree $|\phi|$

② $(m,n,\phi) \in$ OSAT iff $\exists$ multilinear $\hat{A}: \mathbb{F}^n \to \mathbb{F}$ s.t. $\hat{A}$ is boolean on $\{0,1\}^n$ and

$$\underbrace{\forall w \in \{0,1\}^m \, \forall v_1, v_2, v_3 \in \{0,1\}^n \;\; \hat{\phi}(w, v_1, v_2, v_3, \hat{A}(v_1), \hat{A}(v_2), \hat{A}(v_3)) = 0}_{\text{"zero on subcube" testing}}$$

proof:



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

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

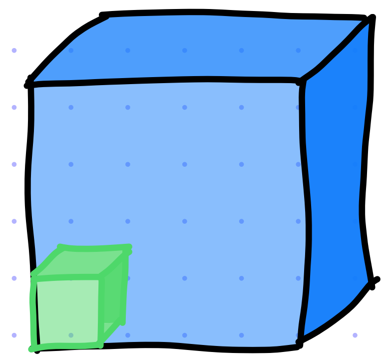
This ensures that the total degree of $\hat{\phi}$ is $\leq |\phi|$ and $\hat{\phi} \equiv \phi$ on every boolean input.

Completeness: if $A: \{0,1\}^n \to \{0,1\}$ is a witness for $(m,n,\phi) \in$ OSAT then $\hat{A} =$ "multilinear extension of A" satisfies the booleanity condition and the vanishing condition

Soundness: if $(m,n,\phi) \notin$ OSAT then $\forall$ multilinear $\hat{A}: \mathbb{F}^n \to \mathbb{F}$ either $\hat{A}$ is not boolean on $\{0,1\}^n$ or $\exists w \in \{0,1\}^m \; \exists v_1, v_2, v_3 \in \{0,1\}^n$ $\hat{\phi}(w, v_1, v_2, v_3, \hat{A}(v_1), \hat{A}(v_2), \hat{A}(v_3)) = \phi(w, v_1, v_2, v_3, \hat{A}(v_1), \hat{A}(v_2), \hat{A}(v_3)) \neq 0$

Part 2: Zero-on-Subcube Test [1/2]

Given oracle access to a low-degree $f: \mathbb{F}^n \to \mathbb{F}$, check that $f|_{H^n} \equiv 0$.



Idea: reduce to sumcheck

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

Consider the polynomial $g(X_1,\ldots,X_n) := \sum_{a_1,\ldots,a_n \in H} f(a_1,\ldots,a_n) X_1^{\text{int}(a_1)} \cdots X_n^{\text{int}(a_n)}$.

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

If $f|_{H^n} \not\equiv 0$ then $g \not\equiv 0$, and in particular $\Pr_{r_1,\ldots,r_n \in \mathbb{F}}[g(r_1,\ldots,r_n)=0] \leq \frac{n \cdot (|H|-1)}{|\mathbb{F}|}$.

Hence it suffices to check that $\sum_{a_1,\ldots,a_n \in H} f(a_1,\ldots,a_n) r_1^{\text{int}(a_1)} \cdots r_n^{\text{int}(a_n)}$ for random $r_1,\ldots,r_n \in \mathbb{F}$.

To make the addend a polynomial: $\forall r \in H$ define $\hat{r}(X) := \sum_{a \in H} r^{\text{int}(a)} L_{a,H}(X)$.

In sum it suffices to run sumcheck on this claim:

$$\sum_{a_1,\ldots,a_n \in H} f(a_1,\ldots,a_n)\, \hat{r}_1(a_1) \cdots \hat{r}_n(a_n) \quad \text{for random } r_1,\ldots,r_n \in \mathbb{F}.$$

Part 2: Zero-on-Subcube Test [2/2]

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

individual degree $\leq d$

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

For every $r_1,\dots,r_n \in \mathbb{F}$:

output eval table $\pi_{SC}[r_1,\dots,r_n]$ of IP prover for sumcheck claim

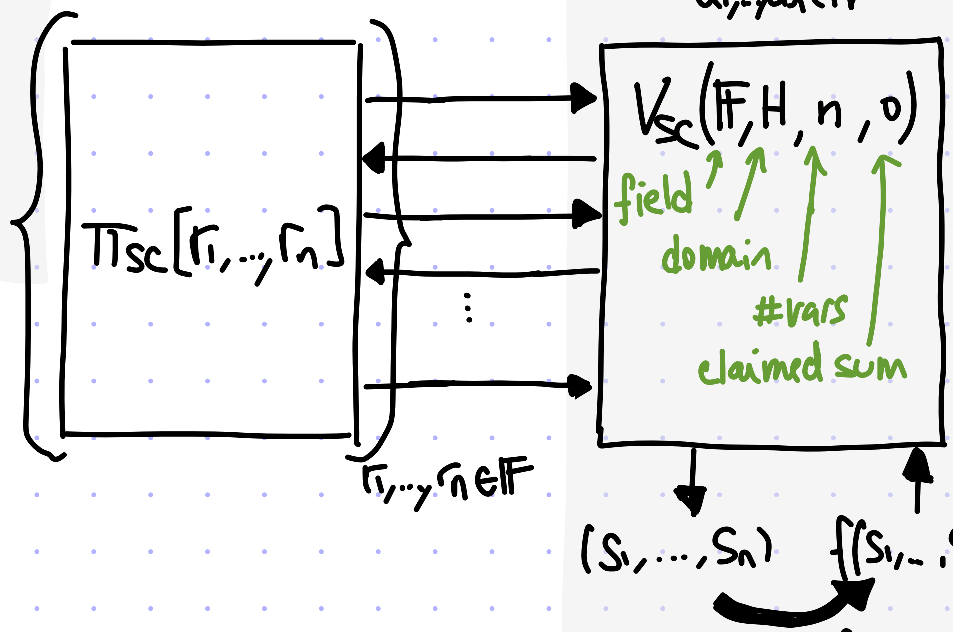
$$\sum_{a_1,\dots,a_n \in H} f(a_1,\dots,a_n) \prod_{i\in[n]} \hat{r}_i(a_i) = 0$$

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

Sample $r_1,\dots,r_n \in \mathbb{F}$.

Run sumcheck for the claim

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



1. query f at $(s_1,\dots,s_n)$
2. for $i=1,\dots,n$: evaluate $\hat{r}_i(x)$ at s_i

query complexity:
- $O(n\cdot(|H|+d))$ elts from π_{SC}
- 1 elt from f

running time:
- $\mathrm{poly}(n,|H|,d)$ from V_{SC}
- $\mathrm{poly}(n,|H|)$ from

proof length:

$|\pi_{SC}| = |\mathbb{F}|^n \cdot O(|\mathbb{F}|^n \cdot (|H|+d))$

$= |\mathbb{F}|^{O(n)} \cdot (|H|+d)$

Completeness: if $f|_{H^n} \equiv 0$ then $\forall r_1,\dots,r_n \in \mathbb{F}$ $\sum_{a_1,\dots,a_n \in H} f(a_1,\dots,a_n)\prod_{i\in[n]}\hat{r}_i(a_i) = 0$ so V_{SC} accepts w.p. 1

Soundness: if $f|_{H^n} \not\equiv 0$ then, except w.p. $\leq \frac{n\cdot(|H|-1)}{|\mathbb{F}|}$ over $r_1,\dots,r_n \in \mathbb{F}$, $\sum_{a_1,\dots,a_n \in H} f(a_1,\dots,a_n)\prod_{i\in[n]}\hat{r}_i(a_i) \neq 0$

so V_{SC} accepts w.p. $\leq \frac{n\cdot(|H|-1+d)}{|\mathbb{F}|}$.

Putting the Two Parts Together

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

0. Compute $\hat{\phi} := T(\mathbb{F},(m,n,\phi))$ for $|\mathbb{F}| = \mathrm{poly}(|\phi|)$.

1. Output $\Pi_A : \mathbb{F}^n \to \mathbb{F}$ that equals the multilinear extension of $A : \{0,1\}^n \to \{0,1\}$

2. For every $r_1,\dots,r_n \in \mathbb{F}$:

   output sumcheck proof $\Pi_{SC}^{(1)}[r_1,\dots,r_n]$

   for $\sum_{a \in \{0,1\}^n} \Pi_A(a)(1-\Pi_A(a)) \prod_{i\in[n]} \hat{r}_i(a_i) = 0$

3. For every $r_1,\dots,r_{m+3n} \in \mathbb{F}$:

   output sumcheck proof $\Pi_{SC}^{(2)}[r_1,\dots,r_{m+3n}]$

   for $\sum_{\substack{a=(w,v_1,v_2,v_3)\\ \in\{0,1\}^{m+3n}}} \hat{\phi}(w,v_1,v_2,v_3,\Pi_A(v_1)\Pi_A(v_2),\Pi_A(v_3)) \prod_{i\in[m+3n]} \hat{r}_i(a_i) = 0$

$\boxed{\Pi_A}$ 2. 3. 4.

$\left\{\boxed{\Pi_{SC}^{(1)}[r_1,\dots,r_n]}\right\}_{r_1,\dots,r_n\in\mathbb{F}}$

$\left\{\boxed{\Pi_{SC}^{(2)}[r_1,\dots,r_{m+3n}]}\right\}_{r_1,\dots,r_{m+3n}\in\mathbb{F}}$

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

1. Compute $\hat{\phi} := T(\mathbb{F},(m,n,\phi))$ for $\mathbb{F}$ of size $\mathrm{poly}(|\phi|)$

2. Sample $r_1,\dots,r_n \in \mathbb{F}$ & run sumcheck for claim

   $$\sum_{a\in\{0,1\}^n} \Pi_A(a)(1-\Pi_A(a)) \prod_{i\in[n]} \hat{r}_i(a_i) = 0$$

   $\boxed{V_{SC}(\mathbb{F},\{0,1\},n,0)}$

   $(s_1,\dots,s_n)$

   - query Π_A at $(s_1,\dots,s_n)$
   - for $i=1,\dots,n$: eval $\hat{r}_i(x)$ at s_i

3. Sample $r_1,\dots,r_{m+3n} \in \mathbb{F}$ & run sumcheck for claim:

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

   $\boxed{V_{SC}(\mathbb{F},\{0,1\},m+3n,0)}$

   $(s_1,\dots,s_{m+3n})$

   - query Π_A at $(s_{m+1},\dots,s_{m+n}),(s_{m+n+1},\dots,s_{m+2n}),(s_{m+2n+1},\dots,s_{m+3n})$
   - for $i=1,\dots,m+3n$: eval $\hat{r}_i(x)$ at s_i
   - evaluate $\hat{\phi}$ at (s,ans_1,ans_2,ans_3)

4. low-degree test Π_A for total degree n [poly(n) queries]

Analysis

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

1. Output $\Pi_A:\mathbb{F}^n\to\mathbb{F}$ that equals the multilinear extension of $A:\{0,1\}^n\to\{0,1\}$

2. For every $r_1,\dots,r_n\in\mathbb{F}$:
   output sumcheck proof $\pi_{SC}^{(1)}[r_1,\dots,r_n]$
   for $\sum_{a_1,\dots,a_n\in\{0,1\}} \Pi_A(a)(1-\Pi_A(a))\prod_{i\in[n]}\hat{r}_i(a_i)=0$

3. For every $r_1,\dots,r_{m+3n}\in\mathbb{F}$:
   output sumcheck proof $\pi_{SC}^{(2)}[r_1,\dots,r_{m+3n}]$
   for $\sum_{\substack{a=(w,v_1,v_2,v_3)\\ \in\{0,1\}^{m+3n}}} \hat{\phi}(w,v_1,v_2,v_3,\Pi_A(v_1)\Pi_A(v_2),\Pi_A(v_3))\prod_{i\in[m+3n]}\hat{r}_i(a_i)=0$

Π_A — 2. 3. 4.

$\pi_{SC}^{(1)}[r_1,\dots,r_n]$ — $r_1,\dots,r_n\in\mathbb{F}$

$\pi_{SC}^{(2)}[r_1,\dots,r_{m+3n}]$ — $r_1,\dots,r_{m+3n}\in\mathbb{F}$

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

1. Compute $\hat{\phi}:=T(\mathbb{F},(m,n,\phi))$ for $\mathbb{F}$ of size $poly(|\phi|)$
2. Sample $r_1,\dots,r_n\in\mathbb{F}$ & run sumcheck for claim
   $$\sum_{a_1,\dots,a_n\in\{0,1\}^n}\Pi_A(a)(1-\Pi_A(a))\prod_{i\in[n]}\hat{r}_i(a_i)=0$$
   $V_{SC}(\mathbb{F},\{0,1\},n,0)$ → $(s_1,\dots,s_n)$
   - query Π_A at $(s_1,\dots,s_n)$
   - for $i=1,\dots,n$: eval $\hat{r}_i(x)$ at s_i
3. Sample $r_1,\dots,r_{m+3n}\in\mathbb{F}$ & run sumcheck for claim:
   $$\sum_{\substack{a=(w,v_1,v_2,v_3)\\ \in\{0,1\}^{m+3n}}}\hat{\phi}(w,v_1,v_2,v_3,\Pi_A(v_1)\Pi_A(v_2),\Pi_A(v_3))\prod_{i\in[m+3n]}\hat{r}_i(a_i)=0$$
   $V_{SC}(\mathbb{F},\{0,1\},m+3n,0)$ → $(s_1,\dots,s_{m+3n})$
   - query Π_A at $(s_{m+1},\dots,s_{m+n}),(s_{m+n+1},\dots,s_{m+2n}),(s_{m+2n+1},\dots,s_{m+3n})$
   - for $i=1,\dots,m+3n$: eval $\hat{r}_i(x)$ at s_i
   - evaluate $\hat{\phi}$ at (s,ans_1,ans_2,ans_3)
4. low-degree test Π_A for total degree n [poly(n) queries]

- **soundness error:**
  $$\varepsilon_{LDT}+O(1)+O\left(\frac{n\cdot n}{|\mathbb{F}|}\right)+O\left(\frac{(m+3n)\cdot(|\phi|\cdot n)}{|\mathbb{F}|}\right)\Rightarrow |\mathbb{F}|=poly(|\phi|) \text{ suffices}$$

- **proof length:**
  $$|\Pi_A|+|\pi_{SC}^{(1)}|+|\pi_{SC}^{(2)}|$$
  $$=|\mathbb{F}|^n+|\mathbb{F}|^n\cdot O(|\mathbb{F}|^n\cdot 1)+|\mathbb{F}|^{m+3n}\cdot O(|\mathbb{F}|^{m+3n}\cdot|\phi|)$$
  $$=O(|\mathbb{F}|^{poly(m,n)})=2^{poly(m,n,\log|\phi|)}$$

- **query complexity:**
  $$(1+3+q_{LDT})+n\cdot O(1)+(m+3n)\cdot|\phi|$$
  $$=poly(n)+poly(|\phi|)$$
  $$=poly(|\phi|)$$

- **verifier time**
  $$poly(|\phi|)+poly(n)+poly(|\phi|)+t_{LDT}$$
  $$=poly(|\phi|)$$