Lecture 13
PCP for NEXP

So far we constructed PCPs for NP:

\[ \text{NP} \leq \text{PCP} \left[ \varepsilon_c = 0, \varepsilon_s = 0.5, \Sigma = \{0, 1\}, l = \exp(n), q = O(1), r = \text{poly}(n) \right] \]
\[ \text{NP} \leq \text{PCP} \left[ \varepsilon_c = 0, \varepsilon_s = 0.5, \Sigma = \{0, 1\}, l = \text{poly}(n), q = \text{poly} \left( \text{log} n \right), r = O(\text{log} n) \right] \]

Today we construct a PCP for NEXP:

**Theorem:** \( \text{NEXP} \leq \text{PCP} \left[ \varepsilon_c = 0, \varepsilon_s = 0.5, \Sigma = \{0, 1\}, l = \exp(n), q = \text{poly}(n), r = \text{poly}(n) \right] \)

**Remarks:**

- \( l = \exp(n) \) 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
  
  [As we see later in the course, one can even achieve \( q = O(1) \) !]
- The PCP verifier runs in \( \text{poly}(n) \) 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 \(|\text{description}| \ll |\text{computation}|\)
2. Design a PCP verifier that only uses description (does not "unroll" the computation)

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

Ex: in \textbf{HSAT} we are given a boolean formula \( \phi: \Sigma_0 \to \Sigma_0 \) and \( \forall \in \mathbb{N} \), and must check

\[ |\{ a \in \Sigma_0^n \mid \phi(a) = 1 \} \cap V | = \nu \]

Ex: in \textbf{TQBF} we are given a boolean formula \( \phi: \Sigma_0 \to \Sigma_0 \) and \( \forall \in \mathbb{N} \), and must check

\[ \forall x_1 \exists x_2 \forall x_3 \ldots \phi(x_1, \ldots, x_n) = 1 \]

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 \textsc{NP}-complete problems where \(|\text{description}| \sim |\text{computation}|:

\[ \text{QESAT}(F) = \{ (p_1, \ldots, p_m) \mid \exists a \in F^n \text{ s.t. } p_1(a) = \ldots = p_m(a) = 0 \} \]

\[ \text{RIGS}(F) = \{ (v, A, B, C) \mid \exists z \in F^n \text{ s.t. } A \cdot B = C \text{ and } z = (v, w) \text{ for some } w \} \]
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 description (does not "unroll" the computation)

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

$P((p_1, \ldots, p_m), a) :=$
1. For every $r \in H_n^s$
   - $p_r = T(p_1, \ldots, p_m; r)$
   - $T_{sc}[r] := \text{eval table for sumcheck to show } p_r = 0$
   - Output $T_{sc}[r]$
2. Output $\alpha : \text{IF}^s \rightarrow \text{IF} [\text{LDE of } \alpha : \text{IN} \rightarrow \text{IF}]$

$V((p_1, \ldots, p_m)) :=$
1. Sample $r \in H_n^s$ and compute
   
   $p_r = \sum_{i,j} r_i \cdot s_j \cdot p_{i,j}$

   2. Run sumcheck to check that
   
   $\sum_{\alpha, \beta \in H_n^s} \hat{c}_r(\alpha, \beta) \hat{a}(\alpha) \hat{a}(\beta) = 0$

   3. Run low-degree test on $T_{\alpha}$

   $V_{\text{LDT}}(\text{IF}, S_n, S_{\text{LVH}})$

\( \text{PCP for } \text{QESAT(IF)} \)}

\( \text{Computing } p_r \text{ and evaluating } \hat{c}_r \text{ takes } \text{poly}(m, n) \text{ time even if } (p_1, \ldots, p_m) \text{ have "structure"} \)
A NEXP-Complete Problem

def: OSAT := \{(m,n,\phi) \mid m,n \in \mathbb{N} \text{ and } \phi: \{0,1\}^{m+3n+3} \rightarrow \{0,1\} \text{ is a boolean formula s.t.} \}

\exists A: \{0,1\}^n \rightarrow \{0,1\}^3 \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 \}

claim: OSAT is NEXP-complete

proof: Suppose L \in 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 \bar{\Phi}_x s.t.

1. \bar{\Phi}_x is satisfiable iff M accepts x
2. \bar{\Phi}_x has \(N_v = 2^{\text{poly}(\log n)}\) variables and \(N_c = 2^{\text{poly}(\log n)}\) clauses — set \(n := \log N\)
3. there is a poly(\(\log n\))-size circuit \(D_x: \{0,1\}^{3n+3} \rightarrow \{0,1\}^3\) that specifies \(\bar{\Phi}_x\)'s clauses:

\[ D_x(v_1, v_2, v_3, c_1, c_2, c_3) = 1 \text{ iff } \bar{\Phi}_x \text{ contains clause } \bigwedge_{i=1}^{3} (x_{v_i} \oplus c_i) \]

Therefore, \(x \in L\) iff \(\exists A: \{0,1\}^n \rightarrow \{0,1\}^3\)

\[ \forall v_1, v_2, v_3 \in \{0,1\}^n \forall c_1, c_2, c_3 \in \{0,1\}^3 \quad D_x(v_1, v_2, v_3, c_1, c_2, c_3) \land \left( \bigwedge_{i=1}^{3} A(v_i) \oplus c_i \right) = 0 \]
A NEXP-Complete Problem

**Def:** OSAT := \{ (m,n,\phi) \mid 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\}^3 \text{ for which}

\forall w \in \{0,1\}^m \forall v_1,v_2,v_3 \in \{0,1\}^3 \quad \phi(w,v_1,v_2,v_3,A(v_1),A(v_2),A(v_3)) = 0 \}

**Claim:** OSAT is NEXP-complete

**Proof:** [continued]

Therefore, \( x \in L \) iff \( \exists A : \{0,1\}^n \to \{0,1\}^3 \)

\[ \forall v_1,v_2,v_3 \in \{0,1\}^n \forall c_1,c_2,c_3 \in \{0,1\}^3 \quad D_x(v_1,v_2,v_3,c_1,c_2,c_3) \land \left( \bigvee_{i=1}^{3} A(v_i) \oplus c_i \right) = 0 \]

Finally, to make \( D_x \) a formula, apply the Cook-Levin Theorem to \( D_x \) to get a boolean formula \( \psi : \{0,1\}^{m+3n+3} \to \{0,1\}^3 \) of size \( \text{poly}(1Dx1)=\text{poly}(1x1) \) such that

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

Now define

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

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. $T(F, (m, n, \phi))$ outputs a circuit $\hat{\phi} : F^{m+3n+3} \rightarrow F$ of total degree $|\phi|$
2. $(m, n, \phi) \in \text{OSAT}$ if and only if there exists a multilinear $\hat{A} : F^n \rightarrow F$ s.t. $\hat{A}$ is boolean on $\mathbb{F}_2^{m+n}$ and

$$\forall w \in \{0, 1\}^m \forall v_1, v_2, v_3 \in \{0, 1\}^n \quad \hat{\phi}(w, v_1, v_2, v_3, \hat{A}(v_1), \hat{A}(v_2), \hat{A}(v_3)) = 0$$

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

[Recall: $x \land y \mapsto x \cdot y$, $x \lor y \mapsto 1 - (1-x)(1-y)$, $\overline{x} \mapsto 1-x$.]

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

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

Soundness: if $(m, n, \phi) \in \text{OSAT}$ then $\exists$ multilinear $\hat{A} : F^n \rightarrow F$ either $\hat{A}$ is not boolean on $\{0, 1\}^n$ or $\exists w \in \{0, 1\}^m \forall v_1, v_2, v_3 \in \{0, 1\}^n \quad \hat{\phi}(w, v_1, v_2, v_3, \hat{A}(v_1), \hat{A}(v_2), \hat{A}(v_3)) = \phi(w, v_1, v_2, v_3, A(v_1), A(v_2), A(v_3)) \neq 0$
Part 2: Zero-on-Subcube Test

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

Idea: reduce to sumcheck

Let $\text{int}: \mathbb{H} \to \{0, 1, \ldots, |\mathbb{H}|-1\}$ be an efficiently computable bijection. Consider the polynomial $g(x_1, \ldots, x_n) = \sum_{a_1, \ldots, a_n \in \mathbb{H}} f(a_1, \ldots, a_n) x_1^{\text{int}(a_1)} \cdots x_n^{\text{int}(a_n)}$.

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

If $f |_{\mathbb{H}^n} \neq 0$ then $g \neq 0$, and in particular $\Pr_{r_1, \ldots, r_n \in \mathbb{F}} [g(r_1, \ldots, r_n) = 0] \leq \frac{n \cdot (|\mathbb{H}|-1)}{|\mathbb{F}|}$.

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

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

In sum it suffices to run sumcheck on this claim:

$$\sum_{a_1, \ldots, a_n \in \mathbb{H}} f(a_1, \ldots, a_n) \hat{r}_1(a_1) \cdots \hat{r}_n(a_n)$$

for random $r_1, \ldots, r_n \in \mathbb{F}$. 
Part 2: Zero-on-Subcube Test

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

For every \( r_1, \ldots, r_n \in \mathbb{F} \),
output eval table \( T_{sc}[r_1, \ldots, r_n] \)
of IP prov for sumcheck claim
\[ \sum f(a_1, \ldots, a_n) \prod_{i \in [n]} \hat{f}_i(a_i) = 0 \quad a_1, \ldots, a_n \in H \]

**Proof Length:**
\[ |T_{sc}| = |\mathbb{F}|^n \cdot O(|\mathbb{F}|^n \cdot (1H + d)) = |\mathbb{F}|^{O(n)} \cdot (1H + d) \]

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

**Soundness:** if \( f|_{\mathbb{F}^n} \not\equiv 0 \) then, except w.p. \( \leq \frac{n \cdot (1H + d)}{|\mathbb{F}|} \) over \( r_1, \ldots, r_n \in \mathbb{F} \), \( \sum_{a_1, \ldots, a_n \in H} f(a_1, \ldots, a_n) \prod_{i \in [n]} \hat{f}_i(a_i) \not\equiv 0 \) so \( V_{sc} \) accepts w.p. \( \leq \frac{n \cdot (1H + d)}{|\mathbb{F}|} \).

\[ \forall f: \mathbb{F}^n \rightarrow \mathbb{F} \quad (\mathbb{F}, H, n) \]
Sample \( r_1, \ldots, r_n \in \mathbb{F} \).
Run sumcheck for the claim
\[ \sum_{a_1, \ldots, a_n \in H} f(a_1, \ldots, a_n) \prod_{i \in [n]} \hat{f}_i(a_i) = 0 \]

- **Query Complexity:**
  - \( O(n \cdot (|H| + d)) \) els from \( T_{sc} \)
  - 1 elt from \( f \)
- **Running Time:**
  - \( \text{poly}(n, |H|, d) \) from \( V_{sc} \)
  - \( \text{poly}(n, |H|) \) from \( \mathbb{F} \)


1. Query \( f \) at \( (s_1, \ldots, s_n) \)
2. For \( i = 1, \ldots, n \): Evaluate \( \hat{f}_i(x) \) at \( s_i \).
Putting the Two Parts Together

1. Compute $\hat{\phi} := T(\mathbb{F}, (m, n, \emptyset))$ for $|\mathbb{F}| = \text{poly}(1/\emptyset)$.

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

3. For every $\vec{r} \in \mathbb{F}$:
   - output sumcheck proof $\Pi_{sc}^{(1)}[\vec{r}]$ for $\sum_{a \in \{0, 1\}^n} \Pi_A(a)(1 - \Pi_A(a)) \prod_{i \in [n]} \hat{f}_i(a_i) = 0$.

4. For every $\vec{r} \in \mathbb{F}$:
   - output sumcheck proof $\Pi_{sc}^{(2)}[\vec{r}]$ for $\sum_{a = (w, v_1, v_2, v_3, \Pi_A(v_1) \Pi_A(v_2), \Pi_A(v_3))} \prod_{i \in [m+3n]} \hat{f}_i(a_i) = 0$.

5. Sample $\vec{r}_1, \ldots, \vec{r}_{m+3n} \in \mathbb{F}$ and run sumcheck for claim:
   - $\sum_{a = (w, v_1, v_2, v_3, \Pi_A(v_1) \Pi_A(v_2), \Pi_A(v_3))} \prod_{i \in [m+3n]} \hat{f}_i(a_i) = 0$.

6. Sample $\vec{r}_1, \ldots, \vec{r}_{m+3n} \in \mathbb{F}$ and run sumcheck for claim:
   - $\sum_{a = (w, v_1, v_2, v_3, \Pi_A(v_1) \Pi_A(v_2), \Pi_A(v_3))} \prod_{i \in [m+3n]} \hat{f}_i(a_i) = 0$.

7. For every $i = 1, \ldots, n$:
   - eval $\hat{f}_i(x)$ at $S_i$.

8. Low-degree test $\Pi_A$ for total degree $n$ [poly(n) queries].
Analysis

\[ P((m,n,\phi), A) \]

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

2. For every \( r_1, \ldots, r_n \in \mathbb{F} \):
   - output sumcheck proof \( \Pi_{\text{sc}}^{(1)}[r_1, \ldots, r_n] \)
   - for \( \sum \Pi_{\Pi_A}(a)(1-\Pi_A(a)) \prod_{i \in [n]} \hat{\phi}_i(a_i) = 0 \) for all \( a_{r_1, \ldots, r_n} \in \mathbb{F}^n [r_1, \ldots, r_n] \)

3. For every \( r_1, \ldots, r_{m+3n} \in \mathbb{F} \):
   - output sumcheck proof \( \Pi_{\text{sc}}^{(2)}[r_1, \ldots, r_{m+3n}] \)
   - for \( \sum \hat{\phi}(w,v,u,v_2,\Pi_A(v_1),\Pi_A(v_2),\Pi_A(v_3)) \prod_{i \in [m+3n]} \hat{\phi}_i(a_i) = 0 \) for all \( a = (w,v,u,v_2,\Pi_A(v_1),\Pi_A(v_2),\Pi_A(v_3)) \) \in \mathbb{F}^{m+3n} \)

- **Soundness error:**
  \[ \varepsilon_{\text{sc}} = O(1) + O\left(\frac{n \cdot n}{|\mathbb{F}|} \right) + O\left(\frac{(m+3n) \cdot |\mathbb{F}|}{|\mathbb{F}|} \right) \Rightarrow |\mathbb{F}| = \text{poly}(1/\varepsilon) \] suffices

- **Proof length:**
  \[ |\Pi_A| + |\Pi_{\text{sc}}^{(1)}| + |\Pi_{\text{sc}}^{(2)}| \]
  \[ = |\mathbb{F}|^n + |\mathbb{F}|^n \cdot O\left(\frac{n \cdot n}{|\mathbb{F}|} \right) + |\mathbb{F}|^{m+3n} \cdot O\left(\frac{(m+3n) \cdot |\mathbb{F}|}{|\mathbb{F}|} \right) = O\left(\frac{|\mathbb{F}|}{\text{poly}(m,n)} \right) = 2 \text{poly}(m,n,\log |\mathbb{F}|) \]

- **Query complexity:**
  \[ (1 + q_{\text{sc}}) + n \cdot O(1) + (m+3n) \cdot 1 |\mathbb{F}| = \text{poly}(n) + \text{poly}(1/\varepsilon) = \text{poly}(1/\varepsilon) \]

- **Verifier time:**
  \[ \text{poly}(1/\varepsilon) + \text{poly}(n) + \text{poly}(1/\varepsilon) + t_{\text{sc}} = \text{poly}(1/\varepsilon) \]