Lecture 25
Sublinear Verification for Any Computation

We have seen how to achieve sublinear verification via P0S/IOPs for machine computations. More generally, sublinear verification is achievable iff the description of a computation is shorter than the computation itself (informally, the computation is “structured”). (Indeed, the verifier must at minimum read the description of the computation!)

Q: How can we achieve sublinear verification for any computation?

(Including ones whose shortest description is the computation itself, like a random circuit.)

One approach: Holographic Proofs (a cool-sounding but not very descriptive historical term)

Consider an offline/online model where:

• in the offline phase the description of the computation is “encoded” into an oracle;
• in the online phase the P0S/IOP verifier has oracle access to this oracle, and may check multiple statements wrt different inputs to the computation.

Today we show how to formalize this idea and how to construct a protocol for it.
Indexed Relations and Languages

An indexed relation is a set \( R = \{(\hat{i}, x, w)\} \) where \( \hat{i} \) is the index, \( x \) the instance, and \( w \) the witness. The corresponding indexed language is \( L(R) = \{(\hat{i}, x) \mid \exists w \text{ s.t. } (\hat{i}, x, w) \in R\} \). The valid witnesses for the index-instance pair \( (\hat{i}, x) \) are \( R[(\hat{i}, x)] = \{w \mid (\hat{i}, x, w) \in R\} \).

The index is to be interpreted as the "large" description of a computation. Here are some examples:

- Circuit satisfiability over \( \mathbb{F} \)

  \[
  \text{CSAT}(\mathbb{F}) = \{(\hat{i}, x, w) = (C, x, w) \mid C: \mathbb{F}^n \to \mathbb{F} \text{ is a circuit and } C(x, w) = 0\}
  \]

- Quadratic equations over \( \mathbb{F} \)

  \[
  \text{QESAT}(\mathbb{F}) = \{(\hat{i}, x, w) = (p_1, \ldots, p_m, x, w) \mid p_1, \ldots, p_m \in \mathbb{F}^2[X, \ldots, X_n] \text{ and } p_1(x, w) = \ldots = p_m(x, w) = 0\}
  \]

- Rank-1 constraints over \( \mathbb{F} \)

  \[
  \text{RICS}(\mathbb{F}) = \{(\hat{i}, x, w) = ((A, B, C), x, w) \mid A, B, C \in \mathbb{F}^{m \times n} \text{ and } A \cdot (x) \circ B(x) = C \cdot (x)\}
  \]
Holographic PCPs

A holographic PCP for an indexed language $L$ is a tuple $(I, P, V)$ s.t.

1. **Completeness:** $\forall (i, x) \in L$, for $\Pi_0 := I(i)$ and $\Pi := P(i, x)$, $\mathbb{P}_{\rho} [V^{\Pi_0, \Pi}(x; \rho) = 1] \geq 1 - \epsilon_c$.

2. **Soundness:** $\forall (i, x) \not\in L$, for $\Pi_0 := I(i)$, $\forall \tilde{\Pi} \mathbb{P}_{\rho} [V^{\Pi_0, \tilde{\Pi}}(x; \rho) = 1] \leq \epsilon_s$.

Diagramatically:

- **OFFLINE** (once per index)

  $I(i)$

  $\Pi_0$

- **ONLINE** (any number of times)

  $P(i, x) \rightarrow \Pi \rightarrow V(x) \rightarrow (i, x) \in L?$

  $P(i, x) \rightarrow \Pi' \rightarrow V(x) \rightarrow (i, x') \not\in L$

Efficiency: proof length is $|\Pi_0| + |\Pi|$ and query complexity is $9\rho + 9$.

We can similarly define holographic ITPs and IOFs. (As well as variants for robustness, proximity, ...).
One motivation to study holography is that it naturally leads to preprocessing arguments. These enable sublinear verification for any computation, given a one-time (public) preprocessing step.

A preprocessing argument system for an indexed language $L$ looks as follows:

\begin{itemize}
  \item **OFFLINE** (once)
  \begin{itemize}
    \item $PK$
    \item $IA$
    \item $VK$
  \end{itemize}

  \begin{itemize}
    \item $I$
  \end{itemize}

  \begin{itemize}
    \item $\text{VK is a short commitment to } I$
  \end{itemize}

  \begin{itemize}
    \item (e.g. to the circuit being proved)
  \end{itemize}

\end{itemize}

\begin{itemize}
  \item **ONLINE** (multiple)
  \begin{itemize}
    \item $x$
    \item $w$
  \end{itemize}

  \begin{itemize}
    \item $PA$
  \end{itemize}

  \begin{itemize}
    \item $VA$
  \end{itemize}

  \begin{itemize}
    \item $x$
  \end{itemize}

\end{itemize}

In the past, we have seen how:

\begin{itemize}
  \item PCP (or IOP) + CRH $\rightarrow$ succinct argument
\end{itemize}

Now we see how:

\begin{itemize}
  \item holographic PCP (or IOP) + CRH $\rightarrow$ succinct argument
  \item preprocessing PCP (or IOP) + CRH $\rightarrow$ succinct argument
\end{itemize}
From Holography to Preprocessing

**Setup:** Everyone has access to a collision-resistant function \( h \) (sampled from a family \( H \)).

**Offline:** Anyone can compute the key pair for an index \( \bar{i} \) (reusable any number of times):

1. Compute the encoded index: \( \bar{T}_0 := I(\bar{i}) \).
2. Commit to encoded index: \( \bar{T}_0 := N \bar{T}_0(\bar{T}_0) \).
3. Output key pair: \( (pk, vk) := (\bar{i}, \bar{T}_0, r_{T0}) \).

**Online:** Anyone can use the key pair to prove/verify statements \( (\bar{i}, x) \in \mathcal{L} \):

\[
\begin{align*}
\text{time}(P_a) &= \text{time}(P) + O_\lambda(n) \\
\text{time}(V_a) &= \text{time}(V) + O_\lambda(n \cdot \log n)
\end{align*}
\]
We have proved that NP has PCPs with polynomial proof length and polylogarithmic query complexity. The PCP verifier did not (and could not) run in sublinear time because it had to read the description of the NP statement being proved, in that case the list of quadratic equations.

We show how to achieve sublinear verification time with the help of an indexer:

**Theorem:** $\text{QESAT}(F) \in \text{HPCP} \left[ \begin{array}{c} E_c = 0, \quad \Sigma = F \\ \Sigma_e = \frac{1}{2}, \quad \ell = \text{iFF} \left( \frac{\log n}{\log \log n} \right) \\ v_t = \text{poly}(1x, \log n) \end{array} \right]$

Here we mean the indexed language $\{ (\hat{w}, x) = ((p_1, \ldots, p_m), x) \mid \exists w \text{ s.t. } p_i(x, w) = \ldots = p_m(x, w) \}$.

This implies, via the holography $\rightarrow$ preprocessing connection, a preprocessing succinct argument for $\text{QESAT}(F)$ where $\text{time}(I_d) = \text{poly}_x(n)$, $\text{time}(P_a) = \text{poly}_x(n)$, and $\text{time}(V_a) = \text{poly}_x(1x, \log n)$.

The ability to verify any (not necessarily structured) computation in sublinear time is convenient to "program" and is useful for cryptographic applications (e.g., recursive proofs).

We now prove the theorem by modifying the construction that we have already.
Fix $H_v, H_e \in \mathbb{F}$ of sizes $O(\log n), O(\log m)$ and set $s_v = \log n / \log |H_v|$ and $s_e = \log m / \log |H_e|$. 

1. $\hat{p}(X_1, \ldots, X_{s_e}) := \sum_{o \in H_v} X_1^{o_1} \cdots X_{s_e}^{o_{s_e}} \cdot P_{o_1 \cdots o_{s_e}}$ 

2. $\hat{c}(X_1, \ldots, X_{s_e}, Y_v, Z, \ldots, Z_{s_v}) := \sum_{\alpha, \beta \in H_v} \hat{p}(X_1, \ldots, X_{s_e})[\alpha, \beta] \cdot I(\alpha, Y) \cdot I(\beta, Z)$ 

3. Output $\Pi_0 : \mathbb{F}^{s_e + 2s_v} \rightarrow \mathbb{F}$ where $\Pi_0 := \hat{c}|_{\mathbb{F}^{s_e + 2s_v}}$

\[ P(\mathbf{x} = (p_1, \ldots, p_m), x, \mathbf{u}) \]

1. Output $\Pi_a : \mathbb{F}^{s_v} \rightarrow \mathbb{F}$ the $(\mathbb{F}, H_v, s_v)$-extension of $a : \mathbb{F} \rightarrow \mathbb{F}$ 

2. For every $r_1, \ldots, r_{s_e} \in \mathbb{F}$:
   - $P_r := \sum_{o \in H_v} r_1^{o_1} \cdots r_{s_e}^{o_{s_e}} \cdot P_{o_1 \cdots o_{s_e}}$
   - output $\Pi_{sc}[r] :=$ eval table for sumcheck to show $P_r(a) = 0$

3. Output $\Pi_{sc}$ that proves that $\Pi_a$ is consistent with $x$.

\[ V(\mathbf{u} = (p_1, \ldots, p_m), x) \]

1. Run low-degree test on $\Pi_a$

2. Sample $r_1, \ldots, r_{s_e} \in \mathbb{F}$ and compute $P_r := \sum_{o \in H_v} r_1^{o_1} \cdots r_{s_e}^{o_{s_e}} \cdot P_{o_1 \cdots o_{s_e}}$

3. Run sumcheck protocol to check that $\sum_{\alpha, \beta \in H_v} P_r[\alpha, \beta] \cdot \hat{a}(\alpha) \cdot \hat{a}(\beta) = 0$

4. Check input consistency proof $\Pi_{sc}$ (query $\Pi_a$ and evaluate $x$ at a point).
Holographic IOP for NP

We have obtained holographic PCEs for NP with polynomial size and polylogarithmic query complexity. Can we improve the proof length by using IOPs instead of PCEs?

This requires some new ideas but can be done:

**Theorem:** For “large smooth” $\mathcal{F}$, $\text{RICS}(\mathcal{F}) \in \text{HIOp} \left[ \begin{array}{l} E_c = 0, k = O(\log s), \Sigma = \mathcal{F} \\ E_s = \frac{1}{2} \\ l = O(s), q = O(\log s) \end{array} \right]$

$s = \#\text{ of non-zero entries in } A, B, C$

Here we mean the indexed language $\{ (i, x) = (A, B, C, x) \mid A, B, C \in \mathcal{F}^{\text{max}} \text{ and } \exists w \text{ s.t. } A(\tilde{x}) \circ B(\tilde{x}) = C(\tilde{x}) \}$

This theorem builds on the non-holographic counterpart that we have already seen:

$\text{RICS}(\mathcal{F}) \in \text{IOp} \left[ \begin{array}{l} E_c = 0, k = O(\log m), \Sigma = \mathcal{F} \\ E_s = \frac{1}{2} \\ l = O(m), q = O(\log m) \end{array} \right]$

It requires one new idea: a holographic subroutine for linear equations.

$\hat{\delta}|_H \equiv N \cdot \hat{f}|_H$

The goal is easy if we allow $l = O(m \cdot n)$.

To achieve $l = O(s)$ (and $vt = O(\log s)$) we need to use algebraic tricks related to Lagrange polynomials.
Interactive Proofs
arithmetization, sumcheck, low-degree extensions, GKR, IP=PSPACE, limitations, ZK

Interactive Oracle Proofs
linear-size proofs, univariate sumcheck, FRI protocol

Probabilistically Checkable Proofs
exponential-size PCP, polynomial-size PCPs
linearity testing, low-degree testing, zero testing

Proof Composition
robust proofs, proximity proofs, composition, PCP Theorem

And more!
parallel repetition, sliding scale conjecture, PCP/IP limitations, holography