Lecture 16
Linear-Size IOPs for Arithmetic Computations

We have seen how to trivially adapt the basic PCP for $NTIME(T)$ into an IOP with proof length $T^{1+o(1)}$ and query complexity $(\log T)^{O(k)}$.

Today we see how to achieve linear proof length for computations over large fields.

Recall the following NP-complete language:

$$\text{def: } RICS(\mathbb{F}) = \left\{ (\mathbf{u}, \mathbf{A}, \mathbf{B}, \mathbf{C}) \mid \exists z \in \mathbb{F}^n \text{ s.t. } \mathbf{A} \cdot \mathbf{z} = \mathbf{C} \text{ and } \mathbf{z} = (\mathbf{u}, \mathbf{w}) \text{ for some } \mathbf{w} \right\}.$$

Given $m \times n$ matrices $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{b}_1$, $\mathbf{b}_2$, $\mathbf{c}_1$, $\mathbf{c}_2$, and $\mathbf{w}$:

$$\begin{bmatrix} \mathbf{a}_1 \mathbf{w} & \mathbf{a}_2 \mathbf{w} \\ \mathbf{b}_1 \mathbf{w} & \mathbf{b}_2 \mathbf{w} \\ \mathbf{c}_1 \mathbf{w} & \mathbf{c}_2 \mathbf{w} \end{bmatrix} = \begin{bmatrix} \mathbf{u} \\ \mathbf{w} \end{bmatrix} \text{ i.e. } \left\langle \mathbf{a}_i, \mathbf{z} \right\rangle \left\langle \mathbf{b}_i, \mathbf{z} \right\rangle = \left\langle \mathbf{c}_i, \mathbf{z} \right\rangle, \quad i \in [m].$$

Theorem: For "large smooth" $\mathbb{F}$,

$$RICS(\mathbb{F}) \in \text{IOP}\left[ \varepsilon_c = 0, \varepsilon_s = 0.5, K = O(\log m), \Sigma = \mathbb{F}, l = O(m), q = O(\log m), r = O(\log m) \right]$$

This achieves linear-size IOPs for arithmetic computations!

Note: we cannot conclude that all of NP has linear-size proofs because reductions introduce overheads.

Today we assume for simplicity that $m = n$ (# equations = # variables).
Prior Choices of Encoding

Our recipe to construct PCPs so far has been to set $\Pi = (\Pi a, \Pi_{sat})$ where:

1. $\Pi a$ is (allegedly) the encoding of a candidate assignment $[\text{belongs to } S = \{\text{Enc}(x)\}_{x \in X}]$
2. If $\Pi a$ is close to $\text{Enc}(a)$ for some $a$, $\Pi_{sat}$ facilitates checking that $a$ is satisfying.

1. What encodings did we use for an assignment $a : [n] \rightarrow \mathbb{F}$?

a. For exp-size PCPs we used linear extensions (aka Hadamard code)

$\text{Enc}(a) : \mathbb{F}^n \rightarrow \mathbb{F}$ where $\text{Enc}(a) = (\langle a, c \rangle)_c \in \mathbb{F}^n$

$|\text{Enc}| = \mathcal{O}(1)$

b. For poly-size PCPs we used multivariate low-degree extensions (aka Reed-Muller code)

$\text{Enc}(a) : \mathbb{F}^{\log^* n} \rightarrow \mathbb{F}$ where $\text{Enc}(a) =$ "(IF, \{0,1,3, log n\})-extension of a"

$\text{Enc}(a) : \mathbb{F}^{\frac{\log n}{\log \log n}} \rightarrow \mathbb{F}$ where $\text{Enc}(a) =$ "(IF, H, $\frac{\log n}{\log \log n}$)-extension of a"

Crucially, for a we have linearity test and for b we have (multivariate) low-degree test.

2. How to test satisfiability? For a, random combination. For b, use somewhere for everything.
A New Choice of Encoding

We seek an encoding with:
- \( \| \operatorname{Enc}(a) \| = O(\| a \|) \)
- Constant relative distance: \( a \neq a' \rightarrow \Delta(\operatorname{Enc}(a), \operatorname{Enc}(a')) = O(1) \)

that lets us execute our recipe of \( \Pi = (T\Pi, \Pi_{sat}) \), which in turn means that we need:
- A proximity test: "\( T\Pi \) close to \( \{ \operatorname{Enc}(x) \}_{x} \)" in few queries
- An approach for testing satisfiability (e.g. a replacement for sumcheck protocol)

Satisfying the rate & distance alone is easy (pick any good code over \( \mathbb{F} \)). Additionally satisfying the other requirements is hard.

The new encoding that we use is: **univariate low-degree extensions**

\( \operatorname{Enc}(a) : \mathbb{F} \rightarrow \mathbb{F} \) where \( \operatorname{Enc}(a) = \) "univariate extension of \( a : \mathbb{F} \rightarrow \mathbb{F} = \) evaluation of \( \sum_{i \in \mathbb{F}} a(i) L_i(X) \) on \( \mathbb{F} \)

Actually we will evaluate on \( L = \oplus \langle 1, H \rangle \) rather than \( \mathbb{F} \) for more flexibility.

This encoding is also known as the Reed-Solomon code:

\[
\text{RS}([1,F, L, d] = \{ f : L \rightarrow F \text{ s.t. } \deg(f) \leq d \}
\]

Today: we temporarily assume that we have a proximity test for univariate extensions, and show how to use this code to construct linear-size IDPs.
Univariate Sumcheck [1/3]

The verifier has oracle access to \( f : \mathbb{F} \rightarrow \mathbb{F} \) s.t. \( \deg(f) \leq d \) and input \((\mathbb{F}, L, d, H, \delta)\), and wants to check the claim \( \sum_{a \in H} \hat{f}(a) = \delta \).

**Attempt 1:** query \( f \) at every \( a \in H \) and add up the answers.

What if \( H \cap L = \emptyset \)?

Deriving \( f(a) \) for a single \( a \in H \) requires \( d + 1 = \Omega(n) \) queries for interpolation.

Even if \( H \subseteq L \), \( |H| = \Theta(n) \) queries is too many.

[And even if \( H \) were small, in the noisy case we would use self-correction, which we don’t have.]

**Attempt 2:** run sumcheck protocol for \( \sum_{a \in H} \hat{f}(a) = \delta \) with \( n = 1 \) (e.g. as IF).

The first (and only) message is the \( d + 1 = \Omega(L) \) coefficients of \( \hat{f} \):

\[
(c_0, c_1, \ldots, c_d)
\]

\( V_f \) : set \( \tilde{f}(x) := \sum_{i=0}^{d} c_i x^i \) and check : \( \sum_{a \in H} \tilde{f}(a) = \delta \) & \( \tilde{f}(s) = f(s) \) for random \( s \in L \)

This is tantamount to reading 1 (huge) symbol from the alphabet \( \Sigma = \mathbb{F}^{d+1} \).

*We need new ideas!*
Univariate Sumcheck [2/3]

The verifier has oracle access to \( f : \mathbb{F} \rightarrow \mathbb{F} \) s.t. \( \deg(f) \leq d \) and input \((\mathbb{F}, L, d, H, \gamma)\), and wants to check the claim \( \sum_{a \in H} \hat{f}(a) = \gamma \).

**Step 1:** reduce the problem to the case \( d < |H| \)

Let \( v_H(x) = \prod_{a \in H} (x-a) \) be the vanishing polynomial of the set \( H \).

Divide \( \hat{f}(x) \) by \( v_H(x) \):

\[
\hat{f}(x) = \hat{h}(x)v_H(x) + \hat{g}(x)
\]

with \( \deg(\hat{g}) < |H| \) \& \( \deg(\hat{h}) = \deg(\hat{f}) - |H| \)

Observe that \( \sum_{a \in H} \hat{f}(a) = \sum_{a \in H} \hat{g}(a) \).

**Step 2:** assume that \( H \) is nice and use algebra

*Lemma:* if \( H \) is a subgroup of \( \mathbb{F}^* \), then \( \sum_{a \in H} \hat{g}(a) = |H| \hat{g}(0) \)

*Proof:* First consider a monomial: \( \sum_{j=0}^{\lfloor |H|/2 \rfloor} (w^i)^j = \sum_{j=0}^{\lfloor |H|/2 \rfloor} (w^i)^j = \frac{1}{2} (1 + (w^i)^{\lfloor |H|/2 \rfloor}) = \begin{cases} 0 & \text{if } i \equiv 0 \mod |H| \\ |H| & \text{if } i \equiv 0 \mod |H| \end{cases} \)

Hence all monomials \( x^i \) for \( i \in |H| \) in \( \hat{g}(x) \) sum to zero, and are left with \( |H| \times g(0) \).

Hence \( \sum_{a \in H} \hat{g}(a) = 0 \) if \( |H| \hat{g}(0) = \gamma \). [Here we saw the case of multiplicative subgroups.]

A similar statement holds for additive subgroups.
Univariate Sumcheck [3/3]

The verifier has oracle access to \( f : L \rightarrow \mathbb{F} \) s.t. \( \deg(f) \leq d \) and input \((\mathbb{F}, L, d, H, \gamma)\), and wants to check the claim \( \sum_{a \in H} \hat{f}(a) = \gamma \).

\[ P((\mathbb{F}, L, d, H, \gamma), f) \]

Compute \( \hat{h}(x) \) with \( \deg(\hat{h}) = \deg(f) - |H| \)
and \( \hat{p}(x) \) with \( \deg(\hat{p}) < |H| - 1 \) s.t.
\[
\hat{f}(x) = \hat{h}(x) \cdot v_H(x) + (x \hat{p}(x) + \gamma/|H|) \]

\[ Vf : L \rightarrow \mathbb{F}((\mathbb{F}, L, d, H, \gamma)) \]

- test that \( h \) is \( \delta \)-close to degree \( d - |H| \)
and that \( p \) is \( \delta \)-close to degree \( |H| - 1 \)
- sample \( s \in L \) and check that
\[
f(s) = h(s) \cdot v_H(s) + (s \cdot p(s) + \gamma/|H|) \]

Analysis: If \( \sum_{a \in H} \hat{f}(a) = \gamma \) then verifier accepts w.p. 1. If \( \sum_{a \in H} \hat{f}(a) \neq \gamma \) then distinguish between:
1. \( \hat{h} \) or \( \hat{p} \) is \( \delta \)-far from (respective) low-degree sets \( \rightarrow \) low-degree test accepts w.p. \( \leq \epsilon \text{Lot}(s) \)
2. \( \hat{h} \) and \( \hat{p} \) both \( \delta \)-close to (unique) \( h \) and \( p \)

\[
\hat{f}(x) \neq \hat{h}(x) \cdot v_H(x) + (x \hat{p}(x) + \gamma/|H|) \text{ so identity test accept w.p.} \leq \frac{d}{|H|} + 2\delta.
\]

[or else \( \hat{f} \) would sum to \( \gamma \).]
Checking Linear Equations

The verifier has oracle access to \( f, g : L \rightarrow F \) of degree \( \leq d \) and input \((F, L, d, H, M)\), and wants to check the claim

\[
\hat{g}|_H \equiv M \cdot \hat{f}|_H.
\]

Idea: reduce to a univariate sumcheck claim

\[
\left\{ \hat{g}(a) = \sum_{b \in H} M[a, b] \cdot \hat{f}(b) \right\}_{a \in H} \text{ if } \sum_{a \in H} \left( \hat{g}(a) - \sum_{b \in H} M[a, b] \hat{f}(b) \right) \times \text{int}(a) \equiv 0
\]

For any \( r \in H \), the evaluation at \( r \) can be written as:

\[
\sum_{a \in H} \text{int}(a) \hat{g}(a) - \left( \sum_{b \in H} M[b, a] \text{int}(b) \right) \hat{f}(a), \text{ equivalently } \sum_{a \in H} \hat{F}(a) \hat{g}(a) - \hat{F}(a) \hat{f}(a)
\]

\[
P((F, L, d, H, M), f, g)
\]

\[
\forall f, g : L \rightarrow F \quad ((F, L, d, H, M))
\]

The soundness error is

\[
\frac{|H| - 1}{|F|} + \varepsilon_{\text{sc}}
\]

univariate sumcheck for \( \sum_{a \in H} \hat{f}(a) \hat{g}(a) - \hat{f}(a) \hat{f}(a) = 0 \)

\[
\text{query f, g at s} \quad \text{eval } \hat{f}, \hat{g} \text{ at s}
\]

\{ can be done in \( O(|H| + |M|) \) ops
IOP for R1CS: Construction

Set \( z := (u, w) \in \mathbb{F}^n \).

Shift \( w \) as follows:
\[
\forall a \in \text{Haux} \quad w'(a) = w(a) - \hat{w}(a) / v_{\text{Hin}}(a)
\]

Compute \( f_w := w'_1 \).

For each \( M \in \{A, B, C\} \):
\[
\text{compute } f_M := M^2 \bigg| L
\]

Compute \( \hat{h}(x) := \frac{\hat{A}_2(x) \hat{B}_2(x) - \hat{C}_2(x)}{v_H(x)} \).

For each \( M \in \{A, B, C\} \):
\[
\text{compute } \hat{g}_M(x) \text{ and } \hat{h}_M(x) \text{ s.t.}
\]
\[
\hat{f}(x) \hat{M}_2(x) - \hat{f}_M(x) \hat{z}(x) = \hat{h}_M(x) v_H(x) + x \hat{r}_M(x)
\]

[actually the three sumchecks can be merged into one via random orfs]

View \( H \) in 2 parts:
\[
\begin{array}{l|l}
\text{Hin} & \text{Haux} \\
\hline
u & w
\end{array}
\]

\[
V((u, A, B, C))
\]

\[
f_w, f_A, f_B, f_C, h : L \to \mathbb{F}
\]

\[
f : L \to \mathbb{F} \text{ is defined as}
\]
\[
f(a) := f_w(a) v_{\text{Hin}}(a) + \hat{w}(a)
\]

For each \( M \in \{A, B, C\} \):
\[
\text{univariate sumcheck for}
\]
\[
\sum_{a \in H} \hat{f}_M(a) f(a) - \hat{f}_M(a) f(a) = 0
\]
\[
p_h, h_M : L \to \mathbb{F}
\]

- Sample \( s \in L \) at random.

- \( f_A(s) f_B(s) - f_C(s) = h(s) v_H(s) \)

- For each \( M \in \{A, B, C\} \):
\[
\hat{f}(s) f_M(s) - \hat{f}_M(s) f(s) = h_M(s) v_H(s) - s p_h(s)
\]

- Test that:
  - \( f_A, f_B, f_C \) are \( \delta \)-close to degree \( 1H1-1 \)
  - \( h \) is \( \delta \)-close to degree \( 1H1-2 \)
  - \( h_A, h_B, h_C \) are \( \delta \)-close to degree \( 1H1-2 \)
  - \( g_A, g_B, g_C \) are \( \delta \)-close to degree \( 1H1-2 \)
IOP for R1CS: Soundness

Suppose that \((u, A, B, C) \in \text{R1CS}\).
If any of the sent functions is \(\delta\)-far from what we are done. So suppose all are \(\delta\)-close.
Let \(\hat{f}_u, \hat{f}_A, \hat{f}_B, \hat{f}_C, \hat{h}, \hat{p}_A, \hat{p}_B, \hat{p}_C, \hat{h}_C\) be the unique closest low-degree polynomials.

One of the following must be true.

1. the Hadamard product condition is violated: \(\hat{f}_A \cdot \hat{f}_B \neq \hat{f}_C\).
2. one of the linear conditions is violated: \(\exists \{A, B, C\} s.t. \hat{f}_M \neq M \cdot \hat{f}_H\).

In case 1: \(\hat{f}_A(x) \cdot \hat{f}_B(x) - \hat{f}_C(x) \neq \hat{h}(x) V_H(x)\) so the verifier accepts w.p. \(\leq \frac{2\|H\| - 2}{11} + 4\delta\).

In case 2: except w.p. \(\frac{11H - 1}{111}\) over \(\text{def}\), \(\hat{f}(x) \hat{f}_H(x) - \hat{f}(x) \hat{f}(x) \neq \hat{h}_M(x) V_H(x) + X \hat{P}_H(x)\) in which case the verifier accepts w.p. \(\leq \frac{2\|H\| - 2}{111} + 4\delta\).
IOP for R1CS: Efficiency

- **proof complexity (in field elts):**
  \( O(1|L| + L_{LDT}) = O(n + L_{LDT}) = O(n) \)

- **query complexity:**
  \( O(1) + q_{LDT} = O(\log n) \)

- **round complexity:**
  \( O(1) + k_{LDT} = O(\log n) \)

- **randomness complexity:**
  \( O(1) + r_{LDT} = O(\log n) \)

- **prover time:** \([\ast]\)
  \( O(1|L| \log |L|) + pt_{LDT} = O(n \log n) \)

- **verifier time:** \([\ast]\)
  \( O(1|L|) + vt_{LDT} = O(n) \)

We have constructed IOPs of linear size for R1CS:

**Theorem:** For every field \( \mathbb{F} \) of size \( \omega(n) \) that is smooth

\[
\text{R1CS}(\mathbb{F}) \leq \text{IOP} \left[ \varepsilon_e = 0, \varepsilon_s = 0.5, \Sigma = \mathbb{F}, pt = O(n \log n), vt = O(n) \right]
\]

\( k = O(1|L| \log n), r = O(1|L| \log n), l = O(n), q = O(\log n) \)

We are left to construct a univariate LDT with logarithmically-many queries.