Lecture 19

Foundations of Probabilistic Proofs
Fall 2020
Alessandro Chiesa

Linear-Size IOPs with Sublinear-Time Verification

We have proved that arithmetic "circuit-like" computations have linear-size IOPs:

for every field $\mathbb{F}$ of size $\Omega(n)$ that is smooth [smoothness is for the LDT]

$$\mathrm{R1CS}(\mathbb{F}) \subseteq \mathrm{IOP}\begin{bmatrix} \varepsilon_c = 0, \varepsilon_s = 0.5, \Sigma = \mathbb{F}, pt = O(n\log n), vt = O(n) \\ k = O(\log n), r = O(\log n), \ell = O(n), q = O(\log n) \end{bmatrix}$$

The running time of the verifier is optimal, because just reading the statement takes $\Omega(n)$ time.

Similarly to before if we seek sublinear-time verification we need to consider problems whose description is smaller than computation size.

The holy grail would be a statement like the following:

$$\mathrm{NTIME}(T) \subseteq \mathrm{IOP}\begin{bmatrix} \varepsilon_c = 0 & \Sigma = \{0,1\} & pt = O(T) & vt = \mathrm{poly}(n, \log T) \\ \varepsilon_s = 0.5, & k = *, & \ell = O(T), & q = \mathrm{poly}(\log T) \end{bmatrix}$$

This remains a challenging open question.

Instead, we will prove a "large alphabet" relaxation of the theorem:

implies prior theorem with $\ell = O(T\log T)$

<u>theorem:</u> for every field $\mathbb{F}$ of size $\Omega(T)$ that is smooth [smoothness is for the LDT]

$$\mathrm{NTIME}(T, \mathbb{F}) \subseteq \mathrm{IOP}\begin{bmatrix} \varepsilon_c = 0 & \Sigma = \mathbb{F} & pt = O(T\log T) & vt = \mathrm{poly}(n, \log T) \\ \varepsilon_s = 0.5, & k = *, & \ell = O(T), & q = \mathrm{poly}(\log T) \end{bmatrix}$$

Machine Computations

Informally, a machine is an automaton that can read/write to some type of memory.

If memory = tapes then you get Turing machines.

If memory = RAM then you get register machines (very close to how we think of a computer).

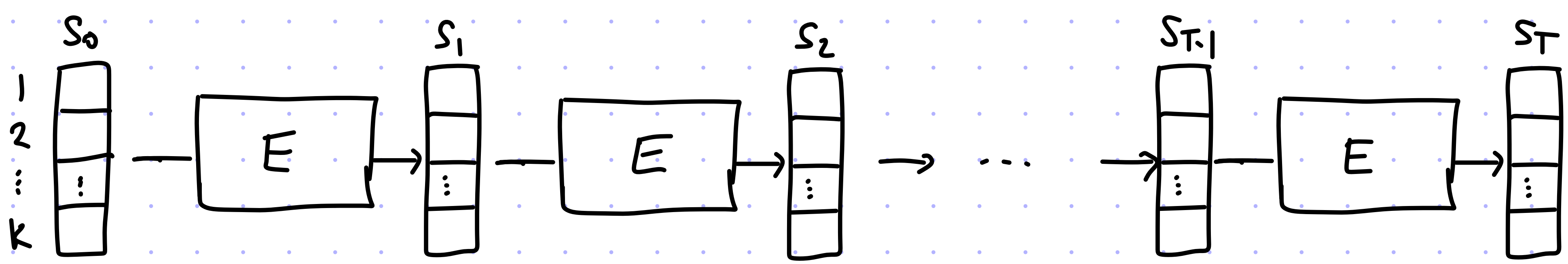
We are going to define languages that model machines that compute over finite fields.

Let's start simple by first doing this for automata (i.e., no memory beyond internal state).

Consider: • $k \in \mathbb{N}$ — number of internal registers, i.e., a state is $s \in \mathbb{F}^k$

• $E : \mathbb{F}^k \to \mathbb{F}^k$ — transition function mapping current state to next state

A T-step computation looks as follows



Specifying the computation requires $O(|E| + \log T)$ bits. our baseline for "linear"

The specified computation involves $O(|E| \cdot T)$ operations, exponentially more in T.

We are in fact interested in non-deterministic computations, and need an appropriate language.

Algebraic Automata

The bounded-halting problem for automata:

transition function | computation input | time bound

def: BH is the set of instances (E, z, T) where $E: \mathbb{F}^k \to \mathbb{F}^k$, $z \in \mathbb{F}^n$, $T \in \mathbb{N}$ for which
$\exists$ execution trace $A_1, \dots, A_k : [T] \to \mathbb{F}$ s.t.

① each step follows the transition function: $\forall t \in \{0,1,\dots,T-1\}$ $E(A_1(t), \dots, A_k(t)) = A_1(t+1), \dots, A_k(t+1)$

② the first n values of A_1 are z: $A_1|_{[n]} = z$

③ the last value of A_1 is 0: $A_1(T) = 0$

Let's massage this into a more convenient problem:

- Identify $[T]$ with a multiplicative subgroup $H = \langle \omega \rangle \leq \mathbb{F}$ s.t. $|H| = T$.
  Crucially, representing H requires only $O(\log|\mathbb{F}|)$ bits, rather than $O(|H| \log|\mathbb{F}|)$.
- We are interested to check not compute, so we translate the circuit $E: \mathbb{F}^k \to \mathbb{F}^k$ into quadratic equations $p_1, \dots, p_m \in \mathbb{F}[X_1, \dots, X_{2k+\ell}]$ with $m := O(|E|)$ and $\ell := O(|E|)$ auxiliary vars

$(E, z, T) \in BH$ iff $\exists$ augmented execution trace $A_1, \dots, A_k, B_1, \dots, B_\ell : H \to \mathbb{F}$ ↙ size $(k+\ell)T = O(|E| \cdot T)$

- $\forall t \in \{0,1,\dots,T-1\}$: $\{p_j(A_1(\omega^t), \dots, A_k(\omega^t), A_1(\omega \cdot \omega^t), \dots, A_k(\omega \cdot \omega^t), B_1(\omega^t), \dots, B_\ell(\omega^t)) = 0\}_{\forall j \in [m]}$
- $A_1|_{H_{in}} = z$, $A_1(\omega^{T-1}) = 0$

Target-on-Subdomain Testing

Consider the setting where the verifier has oracle access to a function $f: L \to \mathbb{F}$ and wishes to check that $\hat{f}|_H \equiv z$ for a given "target" function $z: H \to \mathbb{F}$. (E.g. z is all 0's.)

We have seen this before: $\hat{f}(x)$ vanishes on H iff $\exists\, \hat{h}(x)$ s.t. $\hat{f}(x) - \hat{z}(x) \equiv \hat{h}(x) v_H(x)$

Hence:

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

Compute $\hat{h}(x) := \dfrac{\hat{f}(x) - \hat{z}(x)}{v_H(x)}$

$f: L \to \mathbb{F}$

$\xrightarrow{h: L \to \mathbb{F}}$

$V((\mathbb{F}, L, H, z))$

- Test that h is δ-close to $RS[\mathbb{F}, L, d - |H|]$
- Sample $\gamma \in L$ and check $f(\gamma) - \hat{z}(\gamma) \stackrel{?}{=} h(\gamma) v_H(\gamma)$

Completeness: if $\hat{f}|_H \equiv z$ then $h := \hat{h}|_L \in RS[\mathbb{F}, L, d - |H|]$ and passes check $\forall \gamma \in L$

Soundness: if $\hat{f}|_H \not\equiv z$ then $\forall h: L \to \mathbb{F}$ we have two cases:

- h is δ-far from $RS[\mathbb{F}, L, d - |H|] \to$ verifier accepts w.p. $\leq \varepsilon_{LDT}(\delta)$
- h is δ-close to $\hat{h}$ of degree $d - |H| \to \hat{f}(x) - \hat{z}(x) \not\equiv \hat{h}(x) v_H(x)$ so verifier accepts w.p. $\frac{d}{|L|} + \delta$

becomes 2δ if f is δ-close to $\hat{f}$

Time complexity of the verifier: [ignore LDT because if using FRI $t_{LDT} = O(\log |L|)$, which is small]

- if $z \neq 0^H$ then: evaluate v_H at γ and evaluate $\hat{z}$ at $\gamma \to \text{poly}(|H|)$
- if $z = 0^H$ then: evaluate v_H at $\gamma \to \text{poly}(|H|)$ in general but $\text{poly}(\log |H|)$ if H is a subgroup!

  E.g. if H is a multiplicative subgroup then $v_H(x) = x^{|H|} - 1$. Crucial for us today.

IOP for Algrebraic Automata

[L is an evalwation domain disjoint from H]

$P((E, z, T), A)$

- Run computation on trace $A_1, \dots, A_k$ augment it with $B_1, \dots, B_\ell$
- For each $i \in [k]$:
  compute $f_i := \hat{A}_i|_L \in RS[\mathbb{F}, L, |H|-1]$
- For each $i \in [\ell]$
  compute $g_i := \hat{B}_i|_L \in RS[\mathbb{F}, L, |H|-1]$
- For each $j \in [m]$:
  compute $h_j := \hat{h}_j(x)|_L \in RS[\mathbb{F}, L, |H|-1]$

$$\hat{h}_j(x) := \frac{P_j\left(\begin{matrix}\hat{A}_1(x), \dots, \hat{A}_k(x) \\ \hat{A}_1(\omega \cdot x), \dots, \hat{A}_k(\omega \cdot x)\end{matrix}, \hat{B}_1(x), \dots, \hat{B}_\ell(x)\right)}{V_H(x) / (X - \omega^{T-1})}$$

- $h_z := \hat{h}_z(x)|_L \in RS[\mathbb{F}, L, |H|-1-n]$

$$\hat{h}_z(x) := \frac{\hat{A}_1(x) - \hat{z}(x)}{V_{H_{in}}(x)}$$

- $h_o := \hat{h}_o(x)|_L \in RS[\mathbb{F}, L, |H|-1-1]$

$$\hat{h}_o(x) := \frac{\hat{A}_1(x)}{(X - \omega^{T-1})}$$

$\{f_i : L \to \mathbb{F}\}_{i \in [k]}$

$\{g_i : L \to \mathbb{F}\}_{i \in [\ell]}$

$\{h_j : L \to \mathbb{F}\}_{j \in [m]}$

$h_z, h_o : L \to \mathbb{F}$

$\longrightarrow$

$V((E, z, T))$

- Test each of the received function for the appropriate degree

  [we will come back to this]

- Sample $\gamma \in L$ and check that:

- $\forall j \in [m]$

$$h_j(\gamma) \frac{V_H(\gamma)}{\gamma - \omega^{T-1}} \stackrel{?}{=} P_j\left(\begin{matrix}f_1(\gamma), \dots, f_k(\gamma) \\ f_1(\omega \cdot \gamma), \dots, f_k(\omega \cdot \gamma)\end{matrix}, g_1(\gamma), \dots, g_\ell(\gamma)\right)$$

- $h_z(\gamma) V_{H_{in}}(\gamma) \stackrel{?}{=} f_1(\gamma) - \hat{z}(\gamma)$
- $h_o(\gamma)(\gamma - \omega^{T-1}) \stackrel{?}{=} f_1(\gamma)$

Completeness

Suppose that $A_1,\dots,A_k:[T]\to\mathbb{F}$ is a witness for $(E,z,T)\in BH$.

- The prover can evaluate E at each time step to augment the trace with $B_1,\dots,B_\ell:[T]\to\mathbb{F}$ that satisfy all m quadratic equations $p_1,\dots,p_m$ derived from E. So the prover can find $\hat{h}_1(x),\dots,\hat{h}_m(x)$.
- A_1 agrees with z on first n entries, and is 0 on last entry so $\hat{h}_z,\hat{h}_o$ too can be found.

$P((E,z,T),A)$

- Run computation on trace $A_1,\dots,A_k$ augment it with $B_1,\dots,B_\ell$
- For each $i\in[k]$:
  compute $f_i:=\hat{A}_i|_L\in RS[\mathbb{F},L,|H|-1]$
- For each $i\in[\ell]$
  compute $g_i:=\hat{B}_i|_L\in RS[\mathbb{F},L,|H|-1]$
- For each $j\in[m]$:
  compute $h_j:=\hat{h}_j(x)|_L\in RS[\mathbb{F},L,|H|-1]$

$$\hat{h}_j(x):=\frac{p_j\left(\begin{matrix}\hat{A}_1(x),\dots,\hat{A}_k(x)\\ \hat{A}_1(\omega\cdot x),\dots,\hat{A}_k(\omega\cdot x)\end{matrix},\hat{B}_1(x),\dots,\hat{B}_\ell(x)\right)}{V_H(x)/(x-\omega^{T-1})}$$

- $h_z:=\hat{h}_z(x)|_L\in RS[\mathbb{F},L,|H|-1-n]$
  $\hat{h}_z(x):=\dfrac{\hat{A}_1(x)-\hat{z}(x)}{V_{H_{in}}(x)}$
- $h_o:=\hat{h}_o(x)|_L\in RS[\mathbb{F},L,|H|-1-1]$
  $\hat{h}_o(x):=\dfrac{\hat{A}_1(x)}{(x-\omega^{T-1})}$

$$\{f_i:L\to\mathbb{F}\}_{i\in[k]},\ \{g_i:L\to\mathbb{F}\}_{i\in[\ell]},\ \{h_j:L\to\mathbb{F}\}_{j\in[m]},\ h_z,h_o:L\to\mathbb{F}\ \longrightarrow$$

$V((E,z,T))$

- Test each of the received function for the appropriate degree
  [we will come back to this]
- Sample $\gamma\in L$ and check that:
  - $\forall j\in[m]$
    $h_j(\gamma)\dfrac{V_H(\gamma)}{\gamma-\omega^{T-1}}\stackrel{?}{=}p_j\left(\begin{matrix}f_1(\gamma),\dots,f_k(\gamma)\\ f_1(\omega\cdot\gamma),\dots,f_k(\omega\cdot\gamma)\end{matrix},g_1(\gamma),\dots,g_\ell(\gamma)\right)$
  - $h_z(\gamma)V_{H_{in}}(\gamma)\stackrel{?}{=}f_1(\gamma)-\hat{z}(\gamma)$
  - $h_o(\gamma)(\gamma-\omega^{T-1})\stackrel{?}{=}f_1(\gamma)$

Moreover:
- proof length: $O((k+\ell+m)|L|)=O((k+\ell+m)|H|)=O(|E|\cdot T)$ elts
- query complexity: $O((k+\ell+m)\log|L|)=O(|E|\log T)$
- prover time: $O((k+\ell+m)|L|\log|L|)=O(|E|T\log T)$
- verifier time: $O((k+\ell+m)\log|L|)+poly(n)=O(|E|\log T)+poly(n)$

Soundness

Suppose that $(E,z,T) \notin BH$.

There are two cases:

① One of the functions is far from RS.

- $\exists i \in [k]$ f_i is δ-far from $RS[\mathbb{F}, L, |H|-1]$

or

- $\exists i \in [\ell]$ g_i is δ-far from $RS[\mathbb{F}, L, |H|-1]$

or

- $\exists j \in [m]$ h_j is δ-far from $RS[\mathbb{F}, L, |H|-1]$

or

- h_z is δ-far from $RS[\mathbb{F}, L, |H|-1-n]$

or

- h_o is δ-far from $RS[\mathbb{F}, L, |H|-1-1]$

$\Rightarrow$ verifier accepts w.p. $\leq \varepsilon_{LDT}(\delta)$

$P((E,z,T),A)$

- Run computation on trace $A_1, \dots, A_k$ augment it with $B_1, \dots, B_\ell$
- For each $i \in [k]$:
  compute $f_i := \hat{A}_i|_L \in RS[\mathbb{F}, L, |H|-1]$
- For each $i \in [\ell]$
  compute $g_i := \hat{B}_i|_L \in RS[\mathbb{F}, L, |H|-1]$
- For each $j \in [m]$:
  compute $h_j := \hat{h}_j(x)|_L \in RS[\mathbb{F}, L, |H|-1]$

$$\hat{h}_j(x) := \frac{p_j\left(\frac{\hat{A}_1(x), \dots, \hat{A}_k(x)}{\hat{A}_1(\omega \cdot x), \dots, \hat{A}_k(\omega \cdot x)}, \hat{B}_1(x), \dots, \hat{B}_\ell(x)\right)}{V_H(x)/(x-\omega^{T-1})}$$

- $h_z := \hat{h}_z(x)|_L \in RS[\mathbb{F}, L, |H|-1-n]$

$$\hat{h}_z(x) := \frac{\hat{A}_1(x) - \hat{z}(x)}{V_{H_{in}}(x)}$$

- $h_o := \hat{h}_o(x)|_L \in RS[\mathbb{F}, L, |H|-1-1]$

$$\hat{h}_o(x) := \frac{\hat{A}_1(x)}{(x-\omega^{T-1})}$$

$\{f_i : L \to \mathbb{F}\}_{i \in [k]}$

$\{g_i : L \to \mathbb{F}\}_{i \in [\ell]}$

$\{h_j : L \to \mathbb{F}\}_{j \in [m]}$

$h_z, h_o : L \to \mathbb{F}$

$\longrightarrow$

$V((E,z,T))$

- Test each of the received function for the appropriate degree
  [we will come back to this]
- Sample $\gamma \in L$ and check that:
  - $\forall j \in [m]$
    $h_j(\gamma) \frac{V_H(\gamma)}{\gamma - \omega^{T-1}} \stackrel{?}{=} p_j\left(\frac{f_1(\gamma), \dots, f_k(\gamma)}{f_1(\omega \cdot \gamma), \dots, f_k(\omega \cdot \gamma)}, g_1(\gamma), \dots, g_\ell(\gamma)\right)$
  - $h_z(\gamma) V_{H_{in}}(\gamma) \stackrel{?}{=} f_1(\gamma) - \hat{z}(\gamma)$
  - $h_o(\gamma)(\gamma - \omega^{T-1}) \stackrel{?}{=} f_1(\gamma)$

② all functions are close to (unique) polynomials $\{\hat{f}_i\}_{i \in [k]}, \{\hat{g}_i\}_{i \in [\ell]}, \{\hat{h}_j\}_{j \in [m]}, \hat{h}_z, \hat{h}_o$ of the appropriate degree.

ⓘ $\exists j \in [m]$ $\hat{h}_j(x) \frac{V_H(x)}{x-\omega^{T-1}} \neq p_j\left(\frac{\hat{f}_1(x), \dots, \hat{f}_k(x)}{\hat{f}_1(\omega \cdot x), \dots, \hat{f}_k(\omega \cdot x)}, \hat{g}_1(x), \dots, \hat{g}_\ell(x)\right) \rightarrow$ consistency test passes w.p. $\leq \frac{2|H|-2}{|L|} + (2k+\ell)\delta$

or

ⓘⓘ $\hat{h}_z(x) V_{H_{in}}(x) \neq \hat{f}_1(x) - \hat{z}(x) \rightarrow$ consistency check accepts w.p. $\leq \frac{|H|-1}{|L|} + 2\delta$

or

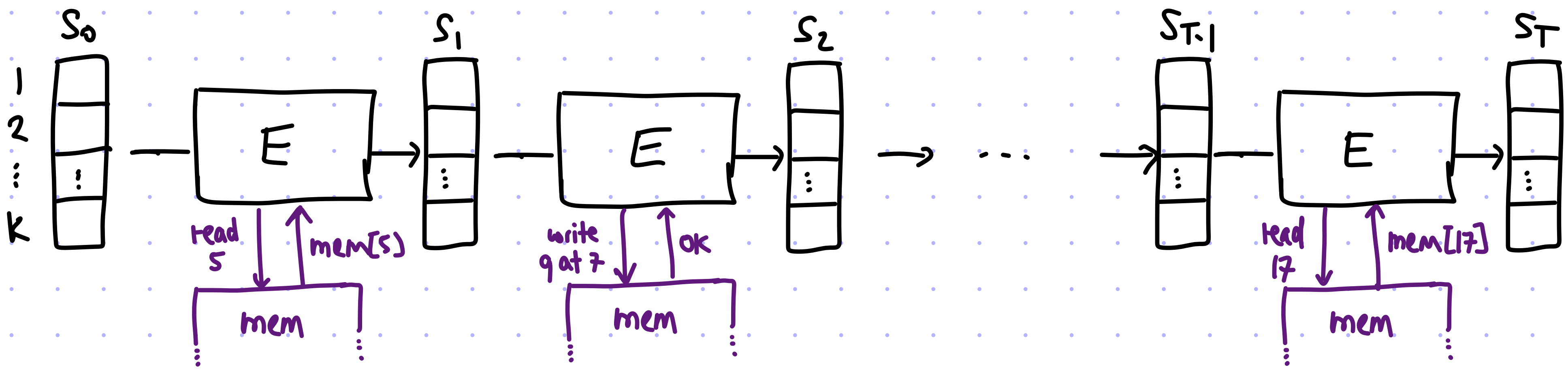
ⓘⓘⓘ $\hat{h}_o(x)(x-\omega^{T-1}) \neq \hat{f}_1(x) \rightarrow$ consistency check accepts w.p. $\leq \frac{|H|-1}{|L|} + 2\delta$

several options to make this < 1:

- set proximity parameter $\delta = O\left(\frac{1}{2k+\ell}\right) = O\left(\frac{1}{|E|}\right)$
  $\rightarrow$ this requires setting repetition parameter t in FRI to $t = O(|E|)$ to ensure that $\varepsilon_{LDT}(\delta) = O(1)$
- repeat consistency test $t = O(\log |E|)$ times, as the term becomes $\left(\frac{2|H|-2}{|L|} + (2k+\ell)\delta\right)^t$
- send random coefficients to prover & test $\sum_i \alpha_i f_i + \sum_i \beta_i g_i$ instead of individually
  $\rightarrow$ distortion statements imply the error becomes $\frac{2|H|-2}{|L|} + 2\delta$ (due to column distance)

From Automata to Machines

We now add memory:



If we extend the state with all of memory, we end up with T^2 variables — well beyond linear.

Observation: it suffices to check correctness of memory operations, "what you wrote is what you read".

Consider the memory trace ordered first by address and then by time stamp:

op	addr	time	val (read or written)
r	2	7	13
r	2	19	13
w	2	22	0
r	2	31	0
r	5	1	3
w	5	6	2
w	7	2	9
⋮			

The trace is correct iff for every two adjacent pairs

$$(op, addr, time, val), (op', addr', time', val')$$

the following holds

- if $addr = addr'$ then $time < time'$ and $(op' = r \rightarrow val' = val)$
- if $addr \neq addr'$ then $addr < addr'$

This leads to a language that represents machine computations...

Memory from a Permuted Trace

<u>lemma:</u> There is a polynomial-time reduction R s.t.

- $R(E,z,T)$ outputs quadratic equations $p_1,\ldots,p_m \in \mathbb{F}[X_1,\ldots,X_{3k+\ell}]$ with $m,\ell = O(|E|)$
- $(E,z,T) \in BH$ iff $\exists$ augmented execution trace $A_1,\ldots,A_k,B_1,\ldots,B_\ell : H \to \mathbb{F}$

  & permutation $\pi : [T] \to [T]$ such that

  - $\forall t \in \{0,1,\ldots,T-1\}$: $\left\{ p_j\left(\begin{matrix} A_1(\omega^t),\ldots,A_k(\omega^t),A_1(\omega\cdot\omega^t),\ldots,A_k(\omega\cdot\omega^t),B_1(\omega^t),\ldots,B_\ell(\omega^t) \\ A_1(\omega^{\pi(t)}),\ldots,A_k(\omega^{\pi(t)}) \end{matrix} \right) = 0 \right\}_{\forall j \in [m]}$
  - $A_1|_{H_{in}} = z$, $A_1(\omega^{T-1}) = 0$

<u>proof:</u> Set $p_1,\ldots,p_m$ to be the quadratic equations obtained by translating the transition function & also the logic for "what you wrote is what you read".

<u>Completeness:</u> choose π to be the permutation that reorders the trace by address then time, so that the memory checks pass

<u>Soundness:</u> for any choice of permutation π, either some memory check fails, or the read/write operations are all correct so the transition function is fed the correct values.

Permutation Check

Consider the setting where the verifier has oracle access to $f,g : L \to \mathbb{F}$ and wishes to check the claim

$$\text{"}\exists \pi : H \to H \text{ s.t. } \forall a \in H \;\; \hat{g}(a) = \hat{f}(\pi(a))\text{"}.$$

Idea: the condition is equivalent to asking if $\{\hat{g}(a)\}_{a\in H}$ and $\{\hat{f}(a)\}_{a\in H}$ equal as multisets, which in turn is true iff $\prod_{a\in H}(X-\hat{g}(a)) \equiv \prod_{a\in H}(X-\hat{f}(a))$.

This directly leads to a protocol when $H = \langle \omega \rangle$:

$P((L,H),(f,g))$	$f,g : L \to \mathbb{F}$	$V((L,H))$
	$\xleftarrow{\;r \in \mathbb{F}\;}$	Sample $r \leftarrow \mathbb{F}$

P: Compute partial products:

- $f_\pi : L \to \mathbb{F}$ s.t. $\hat{f}_\pi(\omega^i) := \prod_{j\le i}(r - \hat{f}(\omega^j))$
- $g_\pi : L \to \mathbb{F}$ s.t. $\hat{g}_\pi(\omega^i) := \prod_{j\le i}(r - \hat{g}(\omega^j))$

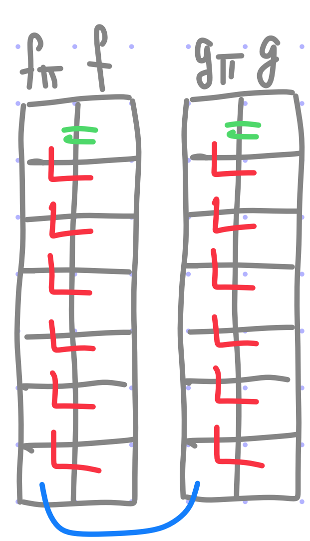
Compute $h_1, h_2, h_3, h_4, h_5 : L \to \mathbb{F}$ s.t.

$$\hat{h}_1(x) \doteq \frac{f_\pi(x) - (r - f(x))f_\pi(\omega^{-1}x)}{V_H(x)/(x-1)} \qquad \hat{h}_3(x) \doteq \frac{g_\pi(x) - (r - g(x))g_\pi(\omega^{-1}x)}{V_H(x)/(x-1)}$$

$$\hat{h}_2(x) \doteq \frac{f_\pi(x) - (r - f(x))}{(x-1)} \qquad \hat{h}_4(x) \doteq \frac{g_\pi(x) - (r - g(x))}{(x-1)}$$

$$\hat{h}_5(x) \doteq \frac{f_\pi(x) - g_\pi(x)}{(x - \omega^{T-1})}$$

$$\xrightarrow{\;f_\pi, g_\pi, h_1, \dots, h_5 : L \to \mathbb{F}\;}$$



V:

- Test that all received functions are LD.
- Sample $\gamma \in L$ and check:

$$h_1(\gamma)\frac{V_H(\gamma)}{\gamma-1} \stackrel{?}{=} f_\pi(\gamma) - (r - f(\gamma))f_\pi(\omega^{-1}\gamma)$$

$$h_2(\gamma)(\gamma-1) \stackrel{?}{=} f_\pi(\gamma) - (r - f(\gamma))$$

$$h_3(\gamma)\frac{V_H(\gamma)}{\gamma} \stackrel{?}{=} g_\pi(\gamma) - (r - g(\gamma))g_\pi(\omega^{-1}\gamma)$$

$$h_4(\gamma)(\gamma-1) \stackrel{?}{=} g_\pi(\gamma) - (r - g(\gamma))$$

$$h_5(\gamma)(\gamma - \omega^{T-1}) = f_\pi(\gamma) - g_\pi(\gamma)$$