Foundations of Probabilistic Proofs

A course by **Alessandro Chiesa**

Lecture 17

Linear-Length IOP for Machines

Linear-Length IOPs with Sublinear-Time Verification

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

$$\forall \text{ smooth field of size } \Omega(n),\ \mathrm{RICS}(\mathbb{F}) \in \mathrm{IOP}\left[\begin{array}{llll} \varepsilon_c=0 & k=O(\log n) & \Sigma=\mathbb{F} & r=O(\log n\cdot\log|\mathbb{F}|) \quad pt=O(s+n\cdot\log n) \\ \varepsilon_s=1/2 & & \ell=O(n) & q=O(\log n) \quad vt=O(s+n) \end{array}\right]$$

(s: # of non-zero entries in A,B,C)

The verifier running time is optimal, because reading the RICS instance takes time $\Omega(s)$.

Q: Can we achieve linear-length IOPs with sublinear verification?

If we seek sublinear verification then we must target problems where

$$|\text{description of computation}| \ll |\text{described computation}|$$

Today we focus on arithmetic machine computations:

theorem: For every smooth field of size $\Omega(T)$,

$$\mathrm{NTIME}(T,\mathbb{F}) \subseteq \mathrm{IOP}\left[\begin{array}{llll} \varepsilon_c=0 & k=O(\log T) & |\Sigma|=O(T) & r=O(\log T\cdot\log|\mathbb{F}|) \quad pt=O(T\cdot\log T) \\ \varepsilon_s=1/2 & & \ell=O(T) & q=O(\log T) \quad vt=O(n+\log T) \end{array}\right]$$

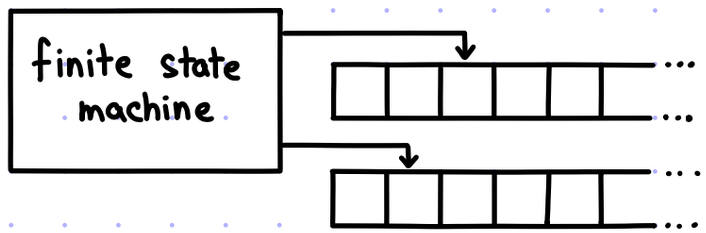
(NTIME: to be defined)

There are IOPs that additionally achieve $pt=O(T)$, using different techniques from today.

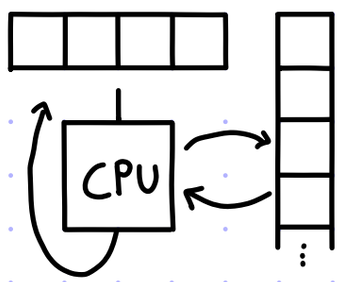
Machine Computations

A machine is an automaton that can read from and write to some type of memory.

If memory = tapes then you get Turing machines:



If memory = RAM then you get register machines:
(≈ a computer)



We will define languages that EFFICIENTLY capture machine computations.

Q: What about IOSAT? We showed a poly(n,logT)-time reduction from NTIME(T) to IOSAT.

A: Yes but the IOSAT instance has $\Omega(T^3)$ constraints.

If we aim for linear-length IOPs then we need reductions to languages that avoid blowups.

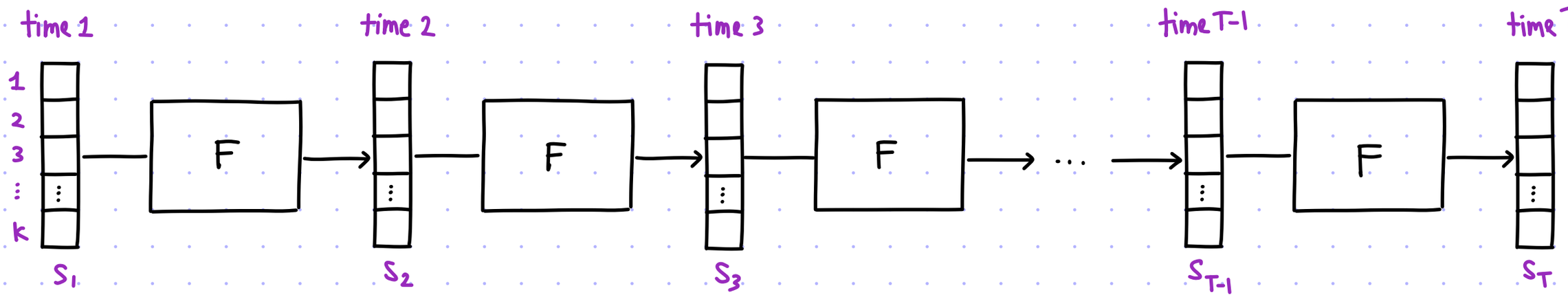
TODAY:
- warm up with IOP for automata computations
- extend to IOP for machine computations

Algebraic Automata [1/2]

We introduce a language that models automata computations over a field $\mathbb{F}$.

We denote by $k \in \mathbb{N}$ the number of registers, so a state is a vector $s \in \mathbb{F}^k$.
The transition function is a circuit $F: \mathbb{F}^k \to \mathbb{F}^k$ that maps a state to the next one.

A T-step computation looks like this:



The computation is specified by the tuple (F, T), which consists of $O(|\mathbb{F}| \cdot \log|\mathbb{F}| + \log T)$ bits.
Performing the computation involves $O(|\mathbb{F}| \cdot T)$ field operations.

Hence for an IOP we would want:
- proof length $\ell = O(|\mathbb{F}| \cdot T)$ field elements
- verifier time $vt = \text{poly}(|\mathbb{F}|, \log T)$ field operations

Algebraic Automata [2/2]

We wish in fact to capture nondeterministic automata computation.

Modifications: • a witness $A_1,\dots,A_k : [T] \to \mathbb{F}$ provides each register's value across the computation
• a circuit $C : \mathbb{F}^{2k} \to \mathbb{F}$ checks the validity of a state transition

We also introduce a separate input $z \in \mathbb{F}^n$, removing the need for hardcoding it in C.

This yields the bounded-halting problem for (algebraic) nondeterministic automata (aka BHA).

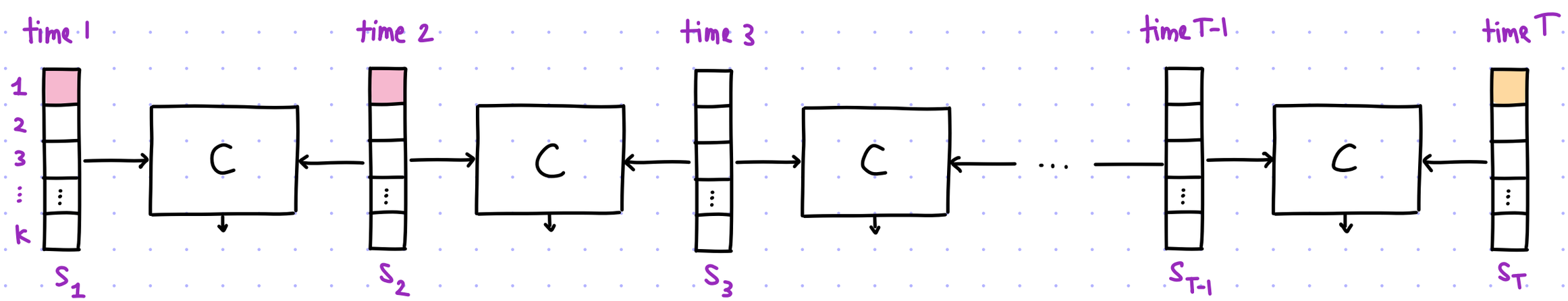
def: $\mathrm{BHA}(\mathbb{F})$ is the set of instances (C, z, T) where $C : \mathbb{F}^{2k} \to \mathbb{F}$ (transition checker), $z \in \mathbb{F}^n$ (computation input), $T \in \mathbb{N}$ (time bound) for which

$\exists$ execution trace $A_1,\dots,A_k : [T] \to \mathbb{F}$ s.t.

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

② the first n values of A_1 are z: $A_1([n]) = z$

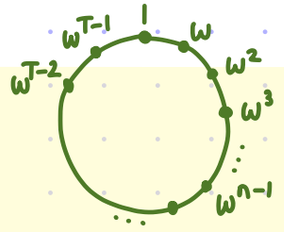
③ the last value of A_1 is 0: $A_1(T) = 0$



Arithmetization of BHA

We arithmetize BHA via univariate polynomials:

representing H takes $O(\log|\mathbb{F}|)$ bits
(an arbitrary set would take $O(|H|\cdot\log|\mathbb{F}|)$ bits)



lemma: Let $H=\langle\omega\rangle$ be a multiplicative subgroup of $\mathbb{F}$ with $|H|=T$.

There is a polynomial-time transformation R s.t.

① $R(C:\mathbb{F}^{2k}\to\mathbb{F})$ outputs quadratic equations $p_1,\dots,p_m\in\mathbb{F}[X_1,\dots,X_{2k+\ell}]$ with $m,\ell=O(|C|)$

② $(C,z,T)\in BHA(\mathbb{F}) \iff \exists$ polynomials $\hat{A}_1,\dots,\hat{A}_k,\hat{B}_1,\dots,\hat{B}_\ell\in\mathbb{F}[X]$ of degree $<T$ s.t.

- $\forall a\in H\setminus\{\omega^{T-1}\}$, $\{p_j(\hat{A}_1(a),\dots,\hat{A}_k(a),\hat{A}_1(\omega\cdot a),\dots,\hat{A}_k(\omega\cdot a),\hat{B}_1(a),\dots,\hat{B}_\ell(a))=0\}_{j\in[m]}$
- $\hat{A}_1(\{1,\omega,\dots,\omega^{n-1}\})=z$
- $\hat{A}_1(\omega^{T-1})=0$

proof: Translate the arithmetic circuit $C:\mathbb{F}^{2k}\to\mathbb{F}$ into quadratic polynomials $p_1,\dots,p_m\in\mathbb{F}[X_1,\dots,X_{2k+\ell}]$ (with $m,\ell=O(|C|)$) that capture C's computation:

$C(\alpha)=\beta$ iff $\exists\gamma\in\mathbb{F}^\ell$ s.t. $\forall j\in[m]$ $p_j(\alpha,\beta,\gamma)=0$.

Set functions $B_1,\dots,B_\ell:[T]\to\mathbb{F}$ to record each step's auxiliary variables arising from the translation.

Identify $[T]$ with $H=\{1,\omega,\dots,\omega^{T-1}\}$ (and $[n]$ with $\{1,\omega,\dots,\omega^{n-1}\}$).

Take univariate LDEs. Completeness and soundness follow. ■

Zero-on-Subset Test

if $d<|H|$ then $\hat{f}|_H \equiv 0 \to \hat{f} \equiv 0$

Given oracle access to $f: L \to \mathbb{F}$ that is δ-close to $\hat{f}$ of degree at most d (with $d \geq |H|$)

check that $\hat{f}|_H \equiv 0$.

We saw this before: $\hat{f}(x)$ is 0 on $H \leftrightarrow \exists\, \hat{h}(x)$ s.t. $\hat{f}(x) \equiv \hat{h}(x) \cdot V_H(x)$ where $V_H(x) = \prod_{a \in H}(X-a)$

Hence:

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

Compute $\hat{h}(x) := \frac{\hat{f}(x)}{V_H(x)}$.

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

$h: L \to \mathbb{F}$ ⟶

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

- Test that h is close to degree $d-|H|$.
- Sample $\gamma \leftarrow L$ and check $f(\gamma) \stackrel{?}{=} h(\gamma) V_H(\gamma)$.

Completeness: if $\hat{f}|_H \equiv 0$ then $h := \hat{h}(L)$ has degree $d-|H|$ and passes the consistency check $\forall \gamma \in L$

Soundness: if $\hat{f}|_H \not\equiv 0$ then $\forall\, h: L \to \mathbb{F}$ there are two cases:

- h is δ-far from degree $d-|H|$ → verifier accepts w.p. $\leq \varepsilon_{LDT}(\delta)$
- h is δ-close to $\hat{h}(x)$ of degree $d-|H|$ → $\hat{f}(x) \not\equiv \hat{h}(x) V_H(x)$ so verifier accepts w.p. $\leq \frac{d}{|L|} + 2\delta$

Verifier time: time(V_{LDT}) (for FRI it's $O(\log|L|)$) plus time to evaluate V_H at $\gamma \in L$.

Evaluating $V_H(x) = \prod_{a \in H}(X-a)$ at $\gamma \in \mathbb{F}$ takes poly$(|H|)$ field operations in general.

But if H is a multiplicative subgroup then $V_H(X) = X^{|H|}-1$, so only $O(\log|H|)$ in this special case.

CRUCIAL FOR US TODAY

[Similarly, if H is an additive subgroup then $V_H(x)$ has $\leq \dim(H)$ non-zero coefficients.]

Rational Constraints

We rewrite the conditions arising from the arithmetization of BHA as polynomial identities:

- accepting condition:

$$\hat{A}_1(\omega^{T-1}) = 0 \leftrightarrow X-\omega^{T-1} \mid \hat{A}_1(X) \leftrightarrow \exists\ \hat{h}_0 \text{ of degree } < |H|-1 \text{ s.t. } \hat{A}_1(X) \equiv \hat{h}_0(X)\cdot(X-\omega^{T-1})$$

- input consistency condition:

$$\hat{A}_1(\{1,\omega,\dots,\omega^{n-1}\}) = z \leftrightarrow \Pi_{t\in[n]}(X-\omega^{t-1}) \mid \hat{A}_1(X)-\hat{z}(X)$$
$$\leftrightarrow \exists\ \hat{h}_z \text{ of degree } < |H|-n \text{ s.t. } \hat{A}_1(X)-\hat{z}(X) \equiv \hat{h}_z(X)\cdot\Pi_{t\in[n]}(X-\omega^{t-1})$$

- state transition condition:

$$\forall j\in[m] \quad \{ p_j(\hat{A}_1(a),\dots,\hat{A}_k(a),\hat{A}_1(\omega\cdot a),\dots,\hat{A}_k(\omega\cdot a),\hat{B}_1(a),\dots,\hat{B}_\ell(a)) = 0 \}_{a\in H\setminus\{\omega^{T-1}\}}$$

$$\leftrightarrow \frac{V_H(X)}{X-\omega^{T-1}} \mid p_j(\hat{A}_1(x),\dots,\hat{A}_k(x),\hat{A}_1(\omega\cdot x),\dots,\hat{A}_k(\omega\cdot x),\hat{B}_1(x),\dots,\hat{B}_\ell(x))$$

$$\leftrightarrow \exists\ \hat{h}_j \text{ of degree } < |H| \text{ s.t. } p_j(\hat{A}_1(x),\dots,\hat{A}_k(x),\hat{A}_1(\omega\cdot x),\dots,\hat{A}_k(\omega\cdot x),\hat{B}_1(x),\dots,\hat{B}_\ell(x)) \equiv \hat{h}_j(x)\cdot\frac{V_H(X)}{X-\omega^{T-1}}$$

Each of these can be viewed as a RATIONAL CONSTRAINT:

$$\frac{N(x)}{D(x)} \rightarrow \forall a\in\mathbb{F} \text{ s.t. } D(a)=0 \text{ it must be that } N(a)=0$$

IOP for Algebraic Automata

[L is an evaluation domain disjoint from H]

theorem: For every smooth field of size $\Omega(T)$,

$$\mathrm{BHA}(\mathbb{F}) \subseteq \mathrm{IOP}\left[\begin{array}{lllll} \varepsilon_c = 0 & k = O(\log T) & |\Sigma| = O(T) & r = O(\log T \cdot \log|\mathbb{F}|) & pt = O(|C| \cdot T \cdot \log T) \\ \varepsilon_s = 1/2 & & \ell = O(|C| \cdot T) & q = O(|C| \cdot \log T) & vt = O(|C| \cdot \log T) + \mathrm{poly}(|z|) \end{array}\right]$$

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

1. $\forall\, i \in [k]\quad f_i := \hat{A}_i(L) \in RS[\mathbb{F}, L, |H|]$
2. Derive auxiliary trace $B_1, \ldots, B_\ell : H \to \mathbb{F}$ from the execution trace $A_1, \ldots, A_k : H \to \mathbb{F}$
3. $\forall\, i \in [\ell]\quad g_i := \hat{B}_i(L) \in RS[\mathbb{F}, L, |H|]$
4. $\forall\, j \in [m]\quad h_j := \hat{h}_j(L) \in RS[\mathbb{F}, L, |H|]$

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

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

$$\hat{h}_z(x) := \frac{\hat{A}_1(x) - \hat{z}(x)}{\prod_{t \in [n]} (X - \omega^{t-1})}$$

6. $h_0 := \hat{h}_0(L) \in RS[\mathbb{F}, L, |H| - 1]$

$$\hat{h}_0(x) := \frac{\hat{A}_1(x) - 0}{X - \omega^{T-1}}$$

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

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

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

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

$h_0 : L \to \mathbb{F}$

$V((C, z, T))$

1. Sample $\gamma \in L$ and check that:
   - $\forall\, j \in [m]$

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

   - $f_1(\gamma) - \hat{z}(\gamma) \stackrel{?}{=} h_z(\gamma) \cdot \prod_{t \in [n]} (\gamma - \omega^{t-1})$
   - $f_1(\gamma) - 0 \stackrel{?}{=} h_0(\gamma) \cdot (\gamma - \omega^{T-1})$
2. Test for low degree:
   - $\forall\, i \in [k]\quad V_{LDT}^{f_i}(\mathbb{F}, L, |H|) \stackrel{?}{=} 1$
   - $\forall\, i \in [\ell]\quad V_{LDT}^{g_i}(\mathbb{F}, L, |H|) \stackrel{?}{=} 1$
   - $\forall\, j \in [m]\quad V_{LDT}^{h_j}(\mathbb{F}, L, |H|) \stackrel{?}{=} 1$

$V_{LDT}^{h_z}(\mathbb{F}, L, |H| - n) \stackrel{?}{=} 1 \quad V_{LDT}^{h_0}(\mathbb{F}, L, |H| - 1) \stackrel{?}{=} 1$

Completeness

Suppose that $A_1,\dots,A_k : H \to \mathbb{F}$ is a witness for $(C,z,T) \in BHA(\mathbb{F})$.

- For each $j \in [m]$:

  $\forall a \in H \setminus \{\omega^{T-1}\}$

  $$P_j\left(\begin{matrix}\hat{A}_1(a),\dots,\hat{A}_k(a)\\ \hat{A}_1(\omega\cdot a),\dots,\hat{A}_k(\omega\cdot a)\end{matrix}, \hat{B}_1(a),\dots,\hat{B}_\ell(a)\right) = 0$$

  $\to V_H(x)/(x-\omega^{T-1})$ divides

  $$P_j\left(\begin{matrix}\hat{A}_1(x),\dots,\hat{A}_k(x)\\ \hat{A}_1(\omega\cdot x),\dots,\hat{A}_k(\omega\cdot x)\end{matrix}, \hat{B}_1(x),\dots,\hat{B}_\ell(x)\right)$$

  $\to \hat{h}_j(x)$ is defined

- $\hat{A}_1(\{1,\omega,\dots,\omega^{n-1}\}) \equiv z \to \prod_{t\in[n]}(x-\omega^{t-1})$ divides $\hat{A}_1(x)-\hat{z}(x) \to \hat{h}_z(x)$ is defined
- $\hat{A}_1(\omega^{T-1}) = 0 \to x-\omega^{T-1}$ divides $\hat{A}_1(x)-0 \to \hat{h}_0(x)$ is defined

---

Moreover:

- proof length (in field elements): $O((k+\ell+m)\cdot|L|) = O((k+\ell+m)\cdot|H|) = O(|C|\cdot T)$
- query complexity: $O((k+m+\ell)\cdot\log|L|) = O(|C|\cdot\log T)$
- prover time (in field operations): $O((k+m+\ell)\cdot|L|\cdot\log|L|) = O(|C|\cdot T\cdot\log T)$
- verifier time (in field operations): $O((k+m+\ell)\cdot\log|L|) + \text{poly}(|z|) = O(|C|\cdot\log T) + \text{poly}(n)$

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

1. $\forall\, i\in[k]\quad f_i := \hat{A}_i(L) \in RS[\mathbb{F},L,|H|]$
2. Derive auxiliary trace $B_1,\dots,B_\ell : H\to\mathbb{F}$ from the execution trace $A_1,\dots,A_k : H\to\mathbb{F}$
3. $\forall\, i\in[\ell]\quad g_i := \hat{B}_i(L) \in RS[\mathbb{F},L,|H|]$
4. $\forall\, j\in[m]\quad h_j := \hat{h}_j(L) \in RS[\mathbb{F},L,|H|]$

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

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

   $$\hat{h}_z(x) := \frac{\hat{A}_1(x)-\hat{z}(x)}{\prod_{t\in[n]}(x-\omega^{t-1})}$$

6. $h_0 := \hat{h}_0(L) \in RS[\mathbb{F},L,|H|-1]$

   $$\hat{h}_0(x) := \frac{\hat{A}_1(x)-0}{x-\omega^{T-1}}$$

$\{f_i : L\to\mathbb{F}\}_{i\in[k]}$
$\{g_i : L\to\mathbb{F}\}_{i\in[\ell]}$
$\{h_j : L\to\mathbb{F}\}_{j\in[m]}$
$h_z : L\to\mathbb{F}$
$h_0 : L\to\mathbb{F}$
$\longrightarrow$

$V((C,z,T))$

1. Sample $\gamma\in L$ and check that:
   - $\forall\, j\in[m]$

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

   - $f_1(\gamma)-\hat{z}(\gamma) \overset{?}{=} h_z(\gamma)\cdot\prod_{t\in[n]}(\gamma-\omega^{t-1})$
   - $f_1(\gamma)-0 \overset{?}{=} h_0(\gamma)\cdot(\gamma-\omega^{T-1})$
2. Test for low degree:
   - $\forall\, i\in[k]\quad V_{LDT}^{f_i}(\mathbb{F},L,|H|) \overset{?}{=} 1$
   - $\forall\, i\in[\ell]\quad V_{LDT}^{g_i}(\mathbb{F},L,|H|) \overset{?}{=} 1$
   - $\forall\, j\in[m]\quad V_{LDT}^{h_j}(\mathbb{F},L,|H|) \overset{?}{=} 1$

   $V_{LDT}^{h_z}(\mathbb{F},L,|H|-n) \overset{?}{=} 1$ $V_{LDT}^{h_0}(\mathbb{F},L,|H|-1) \overset{?}{=} 1$

Soundness [1/2]

Suppose that $(C,z,T) \notin BHA(\mathbb{F})$.

① A function is far from RS.

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

OR

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

OR

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

OR

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

OR

- h_0 is δ-far from $RS[\mathbb{F},L,|H|-1]$

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

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

1. $\forall\, i \in [k]$ $f_i := \hat{A}_i(L) \in RS[\mathbb{F},L,|H|]$
2. Derive auxiliary trace $B_1,\dots,B_\ell : H \to \mathbb{F}$ from the execution trace $A_1,\dots,A_k : H \to \mathbb{F}$
3. $\forall\, i \in [\ell]$ $g_i := \hat{B}_i(L) \in RS[\mathbb{F},L,|H|]$
4. $\forall\, j \in [m]$ $h_j := \hat{h}_j(L) \in RS[\mathbb{F},L,|H|]$

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

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

$$\hat{h}_z(x) := \frac{\hat{A}_1(x)-\hat{z}(x)}{\prod_{t\in[n]}(X-\omega^{t-1})}$$

6. $h_0 := \hat{h}_0(L) \in RS[\mathbb{F},L,|H|-1]$

$$\hat{h}_0(x) := \frac{\hat{A}_1(x)-0}{X-\omega^{T-1}}$$

$\{f_i : L \to \mathbb{F}\}_{i\in[k]}$
$\{g_i : L \to \mathbb{F}\}_{i\in[\ell]}$
$\{h_j : L \to \mathbb{F}\}_{j\in[m]}$
$h_z : L \to \mathbb{F}$
$h_0 : L \to \mathbb{F}$

$V((C,z,T))$

1. Sample $\gamma \in L$ and check that:
   - $\forall j \in [m]$
   $$P_j\left(\begin{matrix}f_1(\gamma),\dots,f_k(\gamma)\\ f_1(\omega\cdot\gamma),\dots,f_k(\omega\cdot\gamma)\end{matrix}, g_1(\gamma),\dots,g_\ell(\gamma)\right) \overset{?}{=} h_j(\gamma)\cdot\frac{V_H(\gamma)}{\gamma-\omega^{T-1}}$$
   - $f_1(\gamma)-\hat{z}(\gamma) \overset{?}{=} h_z(\gamma)\cdot\prod_{t\in[n]}(\gamma-\omega^{t-1})$
   - $f_1(\gamma)-0 \overset{?}{=} h_0(\gamma)\cdot(\gamma-\omega^{T-1})$
2. Test for low degree:
   - $\forall i \in [k]$ $V_{LDT}^{f_i}(\mathbb{F},L,|H|) \overset{?}{=} 1$
   - $\forall i \in [\ell]$ $V_{LDT}^{g_i}(\mathbb{F},L,|H|) \overset{?}{=} 1$
   - $\forall j \in [m]$ $V_{LDT}^{h_j}(\mathbb{F},L,|H|) \overset{?}{=} 1$

   $V_{LDT}^{h_z}(\mathbb{F},L,|H|-n) \overset{?}{=} 1$ $V_{LDT}^{h_0}(\mathbb{F},L,|H|-1) \overset{?}{=} 1$

② All functions are close to (unique) polynomials $(\hat{f}_i)_{i\in[k]}, (\hat{g}_i)_{i\in[\ell]}, (\hat{h}_j)_{j\in[m]}, \hat{h}_z, \hat{h}_0$ of the appropriate degree. ($\delta \leq \frac{1}{2}(1-\frac{|H|}{|L|})$)

- $\exists j \in [m]$ $P_j\left(\begin{matrix}\hat{f}_1(x),\dots,\hat{f}_k(x)\\ \hat{f}_1(\omega\cdot x),\dots,\hat{f}_k(\omega\cdot x)\end{matrix}, \hat{g}_1(x),\dots,\hat{g}_k(x)\right) \neq \hat{h}_j(x)\cdot\frac{V_H(x)}{X-\omega^{T-1}}$ → verifier accepts w.p. $\leq \frac{2|H|-2}{|L|} + (2k+\ell+m)\cdot\delta$

  (γ random in L → $\omega\cdot\gamma$ random in L)
- $\hat{f}_1(x)-\hat{z}(x) \neq \hat{h}_z(x)\cdot\prod_{t\in[n]}(X-\omega^{t-1})$ → verifier accepts w.p. $\leq \frac{|H|-1}{|L|} + 2\delta$
- $\hat{f}_1(x)-0 \neq \hat{h}_0(x)\cdot(X-\omega^{T-1})$ → verifier accepts w.p. $\leq \frac{|H|-1}{|L|} + 2\delta$

Note: Must ensure that $\max\{\varepsilon_{LDT}(\delta), \frac{2|H|-2}{|L|} + (2k+\ell+m)\cdot\delta\} \leq \frac{1}{2}$.

How ?

With t consistency tests the errors reduce to $\left(\frac{2|H|-2}{|L|} + (2k+\ell+m)\cdot\delta\right)^t, \left(\frac{|H|-1}{|L|}+2\delta\right)^t, \left(\frac{|H|-1}{|L|}+2\delta\right)^t$.

In this case, must ensure that $\max\{\varepsilon_{LDT}(\delta), \left(\frac{2|H|-2}{|L|} + (2k+\ell+m)\cdot\delta\right)^t\} \leq \frac{1}{2}$.

Soundness [2/2]

We need that

$$\max\left\{\varepsilon_{LDT}(\delta), \frac{2|H|-2}{|L|} + (2k+\ell+m)\cdot\delta\right\} \leq \frac{1}{2}.$$

This requires considering

$$\delta = O\left(\frac{1}{2k+\ell+m}\right) = O\left(\frac{1}{|C|}\right).$$

In FRI $\varepsilon_{LDT}(\delta) \leq \frac{1}{2}$ requires $O(\frac{1}{\delta}\log|L|)$ queries by $O(\frac{1}{\delta})$ repetitions of the query phase.

This yields a total of $O(\frac{|C|}{\delta}\cdot\log|L|) = O(|C|^2\cdot\log|L|)$ queries. This is ok but can do better!

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

1. $\forall\, i\in[k]\quad f_i := \hat{A}_i(L) \in RS[\mathbb{F},L,|H|]$
2. Derive auxiliary trace $B_1,\ldots,B_\ell : H\to\mathbb{F}$ from the execution trace $A_1,\ldots,A_k : H\to\mathbb{F}$
3. $\forall\, i\in[\ell]\quad g_i := \hat{B}_i(L) \in RS[\mathbb{F},L,|H|]$
4. $\forall\, j\in[m]\quad h_j := \hat{h}_j(L) \in RS[\mathbb{F},L,|H|]$

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

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

$$\hat{h}_z(x) := \frac{\hat{A}_1(x)-\hat{z}(x)}{\prod_{t\in[n]}(X-\omega^{t-1})}$$

6. $h_0 := \hat{h}_0(L) \in RS[\mathbb{F},L,|H|-1]$

$$\hat{h}_0(x) := \frac{\hat{A}_1(x)-0}{X-\omega^{T-1}}$$

$\{f_i : L\to\mathbb{F}\}_{i\in[k]}$
$\{g_i : L\to\mathbb{F}\}_{i\in[\ell]}$
$\{h_j : L\to\mathbb{F}\}_{j\in[m]}$
$h_z : L\to\mathbb{F}$
$h_0 : L\to\mathbb{F}$

$V((C,z,T))$

1. Sample $\gamma\in L$ and check that:
   - $\forall\, j\in[m]$
     $$P_j\left(\begin{matrix}f_1(\gamma),\ldots,f_k(\gamma)\\ f_1(\omega\cdot\gamma),\ldots,f_k(\omega\cdot\gamma)\end{matrix}, g_1(\gamma),\ldots,g_\ell(\gamma)\right) \overset{?}{=} h_j(\gamma)\cdot\frac{V_H(\gamma)}{\gamma-\omega^{T-1}}$$
   - $f_1(\gamma) - \hat{z}(\gamma) \overset{?}{=} h_z(\gamma)\cdot\prod_{t\in[n]}(\gamma-\omega^{t-1})$
   - $f_1(\gamma) - 0 \overset{?}{=} h_0(\gamma)\cdot(\gamma-\omega^{T-1})$
2. Test for low degree:
   - $\forall\, i\in[k]\quad V_{LDT}^{f_i}(\mathbb{F},L,|H|) \overset{?}{=} 1$
   - $\forall\, i\in[\ell]\quad V_{LDT}^{g_i}(\mathbb{F},L,|H|) \overset{?}{=} 1$
   - $\forall\, j\in[m]\quad V_{LDT}^{h_j}(\mathbb{F},L,|H|) \overset{?}{=} 1$

   $V_{LDT}^{h_z}(\mathbb{F},L,|H|-n) \overset{?}{=} 1$ $V_{LDT}^{h_0}(\mathbb{F},L,|H|-1) \overset{?}{=} 1$

Option 1: vectorized LDT, to get error $\max\left\{\varepsilon_{LDT}(d), \frac{2|H|-2}{|L|} + 2\cdot\delta\right\}$.

Run the $k+\ell+m$ instances of FRI with the same randomness to get correlated agreement across functions.

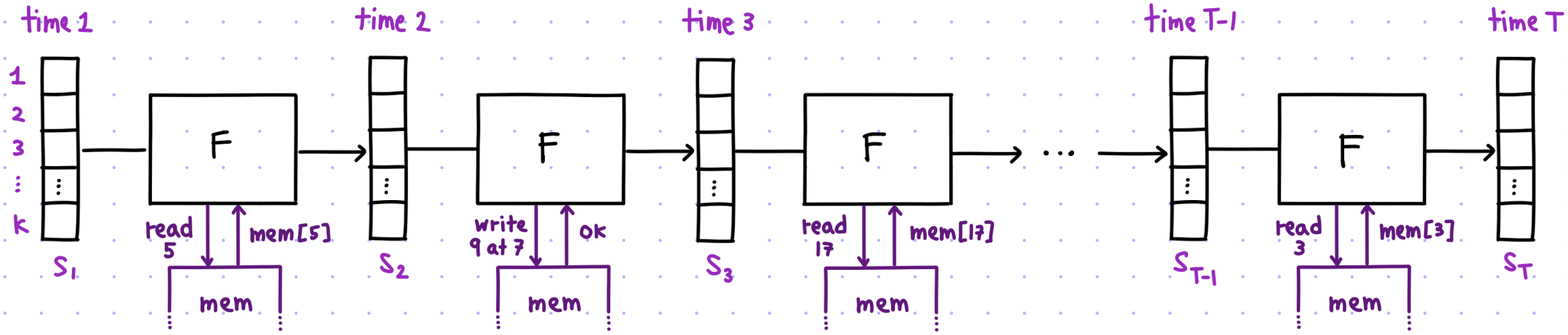
This yields $O(|C|\cdot\log|L|)$ queries (as e.g. $\delta \leq \frac{1}{4}$ suffices).

Option 2: single LDT on random linear combination, to get error $\max\left\{\varepsilon_{LDT}(d), O(\frac{|L|}{|\mathbb{F}|}) + \frac{2|H|-2}{|L|} + 2\cdot\delta\right\}$. (error for correlated agreement)

Verifier sends random $(\alpha_i)_{i\in[k]}, (\beta_i)_{i\in[\ell]}, (\gamma_j)_{j\in[m]}$ and the prover and verifier run 1 instance of FRI on the "virtual function" $\sum_{i\in[k]}\alpha_i f_i + \sum_{i\in[\ell]}\beta_i g_i + \sum_{j\in[m]}\gamma_j h_j$. This yields $O(|C|+\log|L|)$ queries.

From Automata to Machines [1/2]

We add (random-access) memory:



BUT: reducing such computations to BHA is expensive (incurs polynomial blowups)

We would have to augment the state with all of the memory:

$s_i' := (s_i, mem_i)$.

The new automaton has state size $K' := K + |mem| = \Omega(T)$,

and a state function F' that additionally maintains memory.

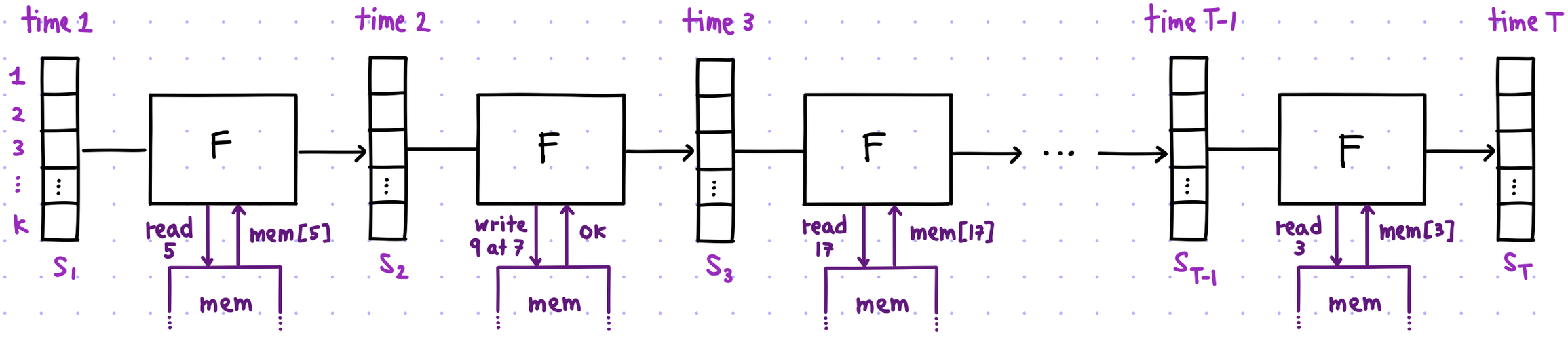
The BHA computation then has size $|F'| \cdot T \geq K' \cdot T = \Omega(T^2)$.

HOPE: each computation step interacts with memory at a single location,

so we may be able to avoid maintaining all of memory at every step.

From Automata to Machines [2/2]

We add (random-access) memory:



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 step.

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

The memory trace is valid iff for every two adjacent pairs

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

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 BH-type language that EFFICIENTLY captures machine computations.

The BHM Language [1/2]

The notion of a memory trace leads to a language convenient for machines.

Similarly to BHA (the language for automata) we consider:

- (non-determinism) a witness $A_1,\dots,A_k : [T] \to \mathbb{F}$,
- (checker instead of function) a circuit $C : \mathbb{F}^{3k} \to \mathbb{F}$,
- (no hardcoded input in C) separate input $z \in \mathbb{F}^n$.

Moreover, to efficiently capture memory, we introduce an additional witness:

a permutation $\pi : [T] \to [T]$ and additionally provide to C the state at time $\pi(t)$.

This yields the bounded-halting problem for (algebraic) nondeterministic machines (aka BHM).

def: $\mathrm{BHM}(\mathbb{F})$ is the set of instances (C, z, T) where $C : \mathbb{F}^{3k} \to \mathbb{F}$, $z \in \mathbb{F}^n$, $T \in \mathbb{N}$ for which

$\exists$ execution trace $A_1,\dots,A_k : [T] \to \mathbb{F}$ and permutation $\pi : [T] \to [T]$ s.t.

① the transition checker validates each step: $\forall t \in \{1,\dots,T-1\}$ $C\left(A_1(t),\dots,A_k(t), A_1(t+1),\dots,A_k(t+1), A_1(\pi(t)),\dots,A_k(\pi(t))\right) = 0$

② the first n values of A_1 are z: $A_1([n]) = z$

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

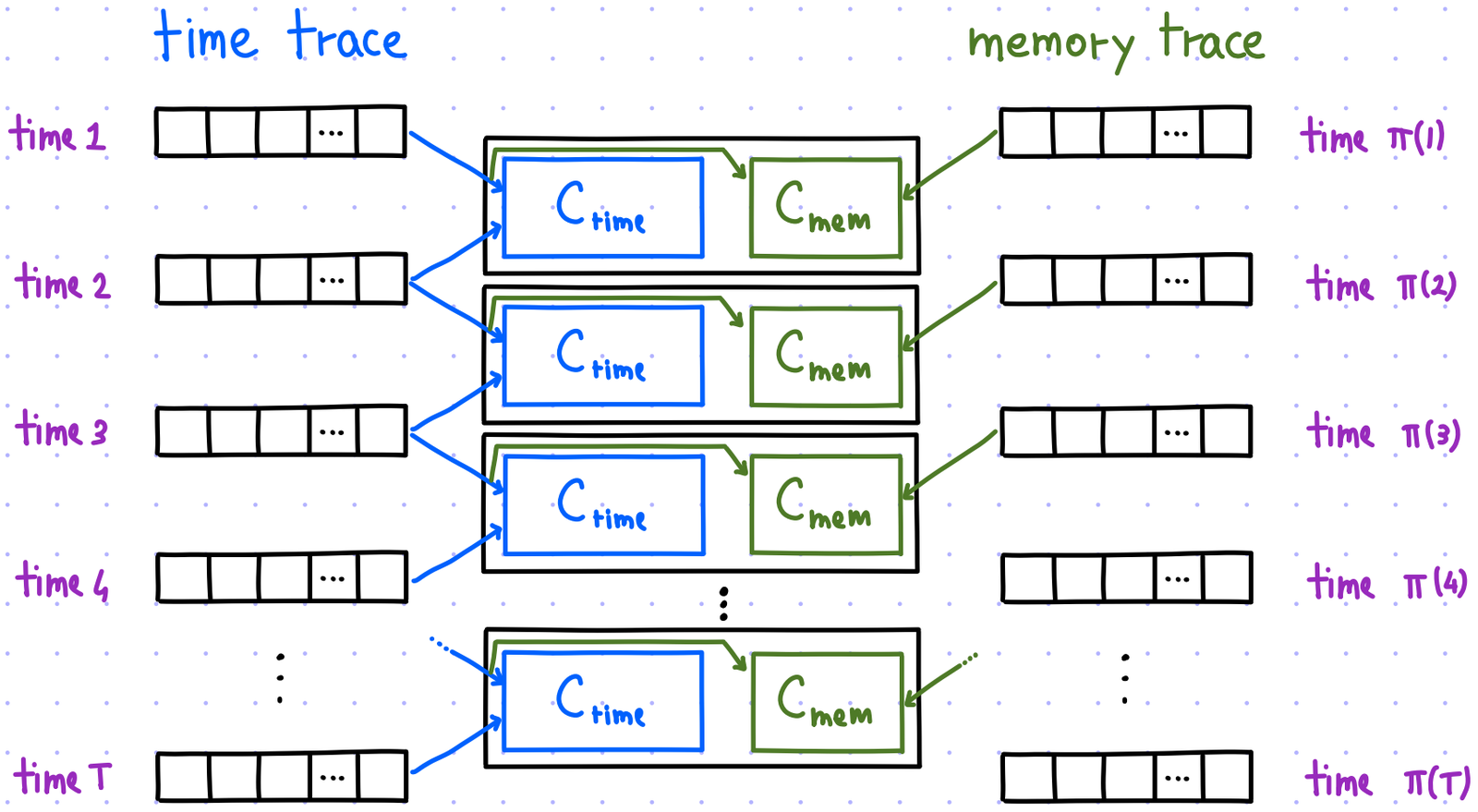
How does this odd language efficiently capture machine computations?

The BHM Language [2/2]

The circuit C is tasked to check:

① time transitions: $\forall\, t \in \{1,\dots,T-1\}$, the transition from state t to state $t+1$
(assuming the correctness of memory operations)

② memory transitions: $\forall\, t \in \{1,\dots,T-1\}$, the memory invariant from state t to state $\pi(t)$



Completeness:

set π to sort the time trace by address and then by time

Soundness:

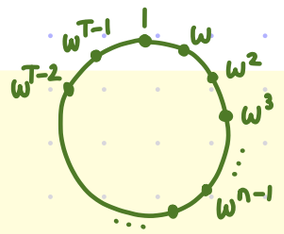
for any permutation π, if all memory transitions hold then memory behaves correctly

The circuit C is constructed as:

$$C(a_1,\dots,a_k,b_1,\dots,b_k,c_1,\dots,c_k) := C_{time}(a_1,\dots,a_k,b_1,\dots,b_k) \vee C_{mem}(a_1,\dots,a_k,c_1,\dots,c_k)$$

Arithmetization of BHM

The arithmetization of BHM is similar to that of BHA.



lemma: Let $H=\langle\omega\rangle$ be a multiplicative subgroup of $\mathbb{F}$ with $|H|=T$.

There is a polynomial-time transformation R s.t.

① $R(C:\mathbb{F}^{3k}\to\mathbb{F})$ outputs quadratic equations $p_1,\dots,p_m\in\mathbb{F}[X_1,\dots,X_{2k+\ell}]$ with $m,\ell=O(|C|)$

② $(C,z,T)\in BHM(\mathbb{F}) \leftrightarrow \exists$ polynomials $\hat{A}_1,\dots,\hat{A}_k,\hat{B}_1,\dots,\hat{B}_\ell\in\mathbb{F}[X]$ of degree $<T$ & permutation $\pi:[T]\to[T]$ s.t.

- $\forall a\in H\setminus\{\omega^{T-1}\}$, $\left\{P_j\begin{pmatrix}\hat{A}_1(a),\dots,\hat{A}_k(a) & ,\hat{B}_1(a),\dots,\hat{B}_\ell(a)\\ \hat{A}_1(\omega\cdot a),\dots,\hat{A}_k(\omega\cdot a) & ,\hat{A}_1(\pi(a)),\dots,\hat{A}_k(\pi(a))\end{pmatrix}=0\right\}_{j\in[m]}$
- $\hat{A}_1(\{1,\omega,\dots,\omega^{n-1}\})=z$
- $\hat{A}_1(\omega^{T-1})=0$

The corresponding rational constraints follow directly:

$$\left\{\frac{P_j\begin{pmatrix}\hat{A}_1(x),\dots,\hat{A}_k(x) & ,\hat{B}_1(x),\dots,\hat{B}_\ell(x)\\ \hat{A}_1(\omega\cdot x),\dots,\hat{A}_k(\omega\cdot x) & ,\hat{u}_1(x),\dots,\hat{u}_k(x)\end{pmatrix}}{V_H(x)/(x-\omega^{T-1})}\right\}_{j\in[m]},\quad \frac{\hat{A}_1(x)-\hat{z}(x)}{\prod_{t\in[n]}(x-\omega^{t-1})},\quad \frac{\hat{A}_1(x)-0}{x-\omega^{T-1}}.$$

QUESTION: how do we check that $\hat{u}_1,\dots,\hat{u}_k$ are obtained via a permutation from $\hat{A}_1,\dots,\hat{A}_k$?

IOP for Algebraic Machines

The arithmetization of BHM directly leads to an IOP for BHM with similar parameters as the IOP for BHA: linear proof length, logarithmic verification,...

PROVIDED we have a suitable IOP for this subproblem:

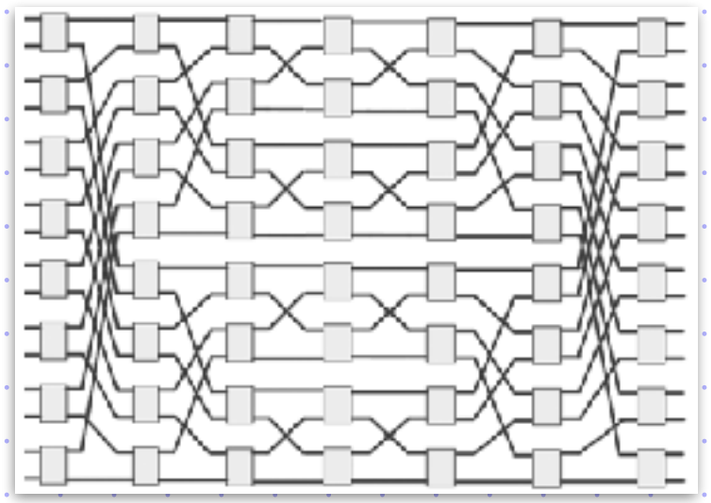
> **VECTOR PERMUTATION CHECK**
>
> Given oracle access to $f_1,\dots,f_k, g_1,\dots,g_k : L \to \mathbb{F}$ that are δ-close to degree d, check that $\exists\, \pi : H \to H$ s.t. $\forall i \in [k]$ $\forall a \in H$ $\hat{g}_i(a) = \hat{f}_i(\pi(a))$.

In the IOP we would apply this to $A_1,\dots,A_k, u_1,\dots,u_k$.

Approach #1: routing networks

Design an IOP to ensure the valid operation of every switch in a T-packet routing network.

PROBLEM: the network has size $O(T \cdot \log T)$ (more than linear)



Approach #2: roots of univariate polynomials

We see how to do this for $k=1$. (The case $k>1$ can be randomly reduced to $k=1$.)

Permutation Check

The verifier has oracle access to $f,g: L \to \mathbb{F}$ and wishes to check:

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

IDEA: the condition is equivalent to "$\{\hat{g}(a)\}_{a\in H}$ and $\{\hat{f}(a)\}_{a\in H}$ equal as multi-sets", in turn equivalent to $\prod_{a\in H}(X-\hat{g}(a)) \equiv \prod_{a\in H}(X-\hat{f}(a))$.

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

$P((L,H),(f,g))$

1. Compute partial products:
   - $f_A: L \to \mathbb{F}$ s.t. $\hat{f}_A(\omega^i) := \prod_{j<i}(r-\hat{f}(\omega^j))$
   - $g_A: L \to \mathbb{F}$ s.t. $\hat{g}_A(\omega^i) := \prod_{j<i}(r-\hat{g}(\omega^j))$
2. Compute $h_1,h_2,h_3,h_4,h_5: L \to \mathbb{F}$ s.t.

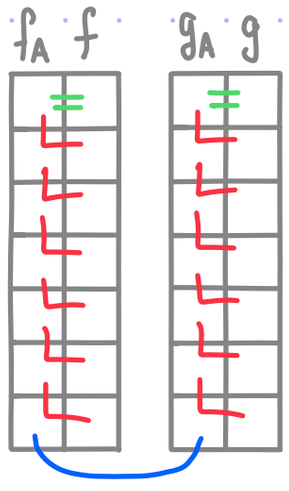
$$\hat{h}_1(X) \coloneqq \frac{\hat{f}_A(X)-(r-\hat{f}(X))\cdot\hat{f}_A(\omega^{-1}X)}{V_H(X)/(X-1)} \qquad \hat{h}_3(X) \coloneqq \frac{\hat{g}_A(X)-(r-\hat{g}(X))\cdot\hat{g}_A(\omega^{-1}X)}{V_H(X)/(X-1)}$$

$$\hat{h}_2(X) \coloneqq \frac{\hat{f}_A(X)-(r-\hat{f}(X))}{X-1} \qquad \hat{h}_4(X) \coloneqq \frac{\hat{g}_A(X)-(r-\hat{g}(X))}{X-1}$$

$$\hat{h}_5(X) \coloneqq \frac{\hat{f}_A(X)-\hat{g}_A(X)}{X-\omega^{T-1}}$$

$r \in \mathbb{F}$ ←

$f_A, g_A, h_1, \ldots, h_5: L \to \mathbb{F}$ →



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

1. Sample $r \leftarrow \mathbb{F}$.
2. Sample $\gamma \in L$ and check:

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

$$f_A(\gamma)-(r-f(\gamma)) \stackrel{?}{=} h_2(\gamma)\cdot(\gamma-1)$$

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

$$g_A(\gamma)-(r-g(\gamma)) \stackrel{?}{=} h_4(\gamma)\cdot(\gamma-1)$$

$$f_A(\gamma)-g_A(\gamma) \stackrel{?}{=} h_5(\gamma)\cdot(\gamma-\omega^{T-1})$$

3. Test functions for low degree.

Bibliography

Linear-length IOPs for machines

Early work on PCPs for machine computation

- [BBCGGHPRSTV 2016]: Computational integrity with a public random string from quasi-linear PCPs, by Eli Ben-Sasson, Iddo Ben-Tov, Alessandro Chiesa, Ariel Gabizon, Daniel Genkin, Matan Hamilis, Evgenya Pergament, Michael Riabzev, Mark Silberstein, Eran Tromer, Madars Virza.
- [BBHR 2018]: Scalable, transparent, and post-quantum secure computational integrity, by Eli Ben-Sasson, Iddo Bentov, Yinon Horesh, Michael Riabzev. (▸Video).
- [BCGGRS 2019]: Linear-size constant-query IOPs for delegating computation, by Eli Ben-Sasson, Alessandro Chiesa, Lior Goldberg, Tom Gur, Michael Riabzev, Nick Spooner.

Frontends

- [Robson 1991]: An O(T log T) reduction from RAM computations to satisfiability, by John Robson.
- [BEGKN 1992]: Checking the correctness of memories, by Manuel Blum, Will Evans, Peter Gemmel, Sampath Kannan, Moni Naor.
- [BCTV 2013]: Succinct non-interactive zero knowledge for a von Neumann architecture, by Eli Ben-Sasson, Alessandro Chiesa, Eran Tromer, Madars Virza.
- [BCGTV 2013]: SNARKs for C: verifying program executions succinctly and in zero knowledge, by Eli Ben-Sasson, Alessandro Chiesa, Daniel Genkin, Eran Tromer, Madars Virza.
- [SAL 2020]: Verifiable state machines: Proofs that untrusted services operate correctly, by Srinath Setty, Sebastian Angel, Jonathan Lee.
- [STW 2024]: Unlocking the lookup singularity with Lasso, by Srinath Setty, Justin Thaler, Riad Wahby. (▸Lasso + Jolt), (▸Lookup landscape).
- Verifiable virtual machines: Cairo, Jolt, Nexus, RISC0, Succinct. (▸Intro to zkVMs).