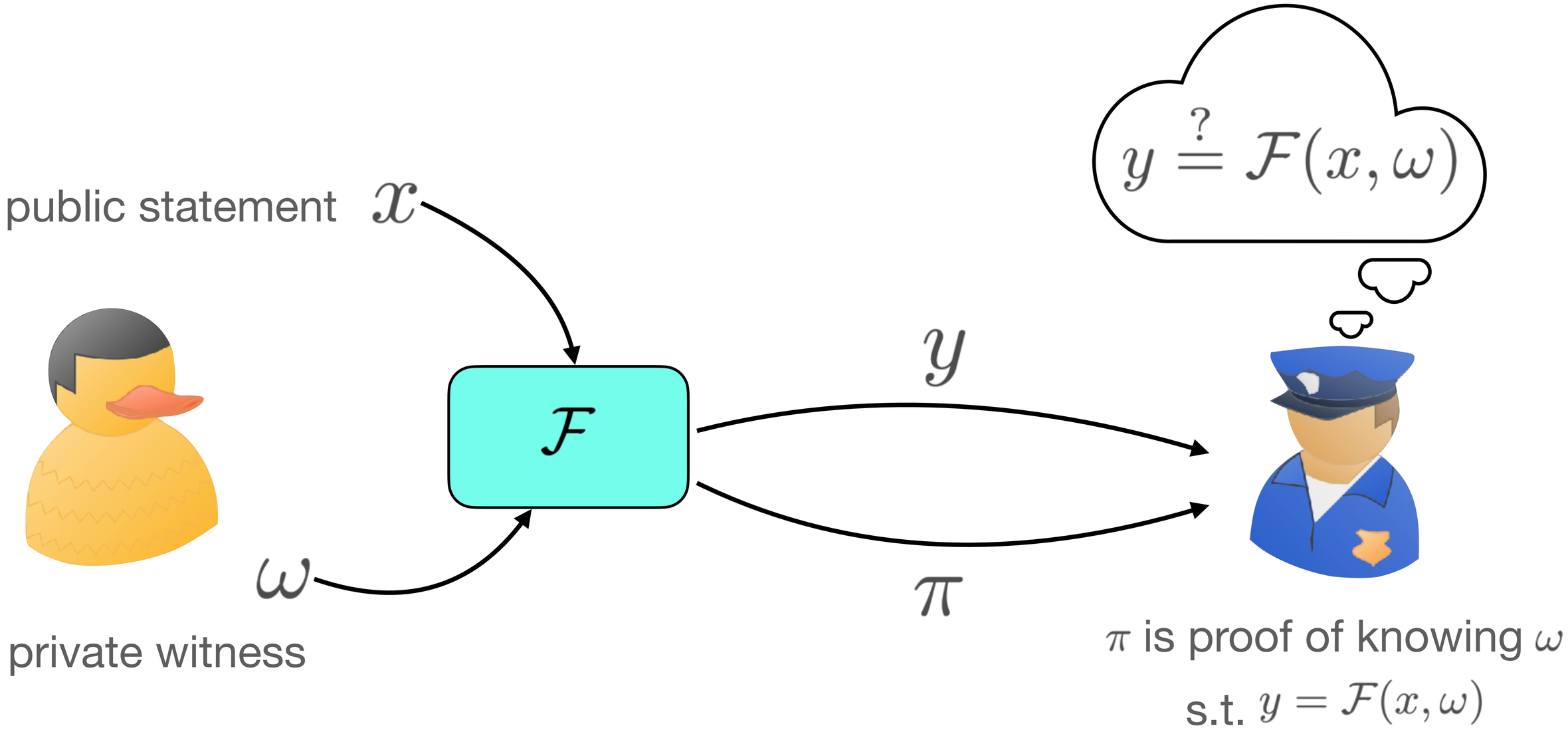
HyperNova

Recursive Arguments for Customizable Constraint Systems

presented by Carlo Brunetta


public statement x
$\mathcal{F}$
y
π
ω
private witness
$y \stackrel{?}{=} \mathcal{F}(x, \omega)$
π is proof of knowing ω
s.t. $y = \mathcal{F}(x, \omega)$

$|\pi|$

Short

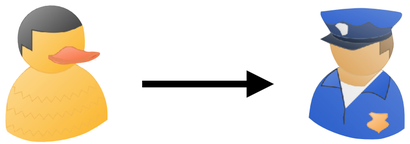
What if $|\mathcal{F}|$ is too big?

Verify

PolyLog w.r.t. Evaluation Time

succinct

S



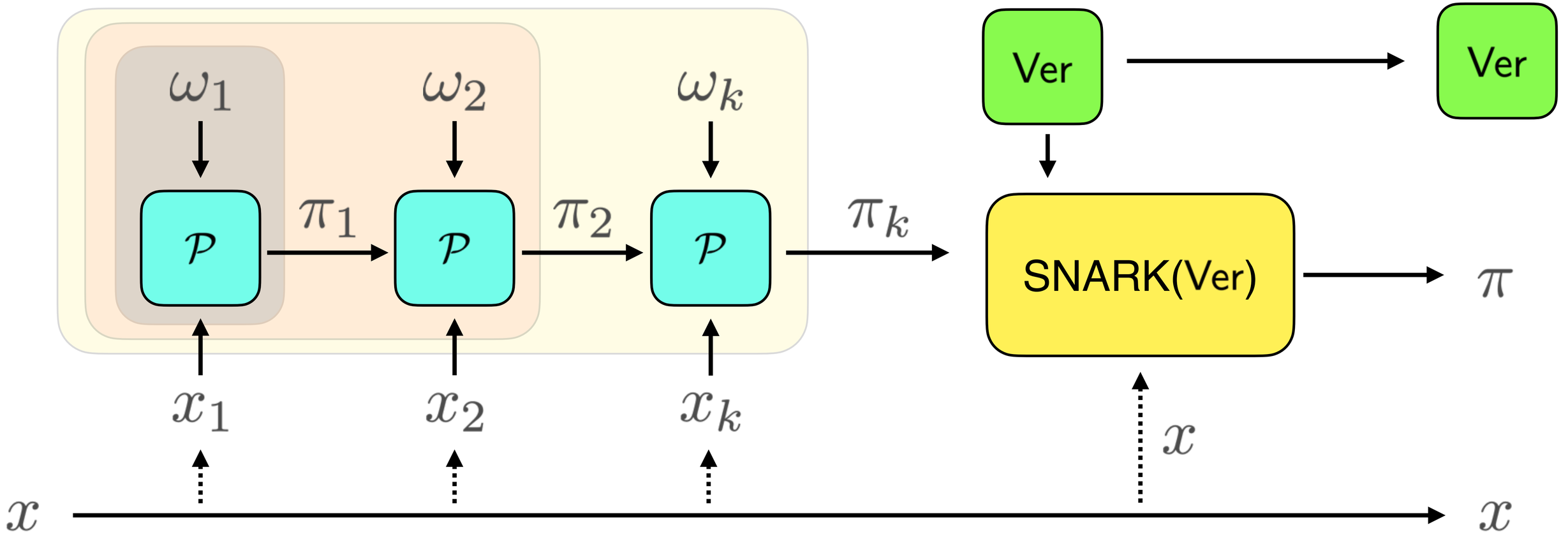
Non interactive

N

π

Proof of “knowing a witness“

ARK



Fast Folding

Longer Proofs

Slower Proving

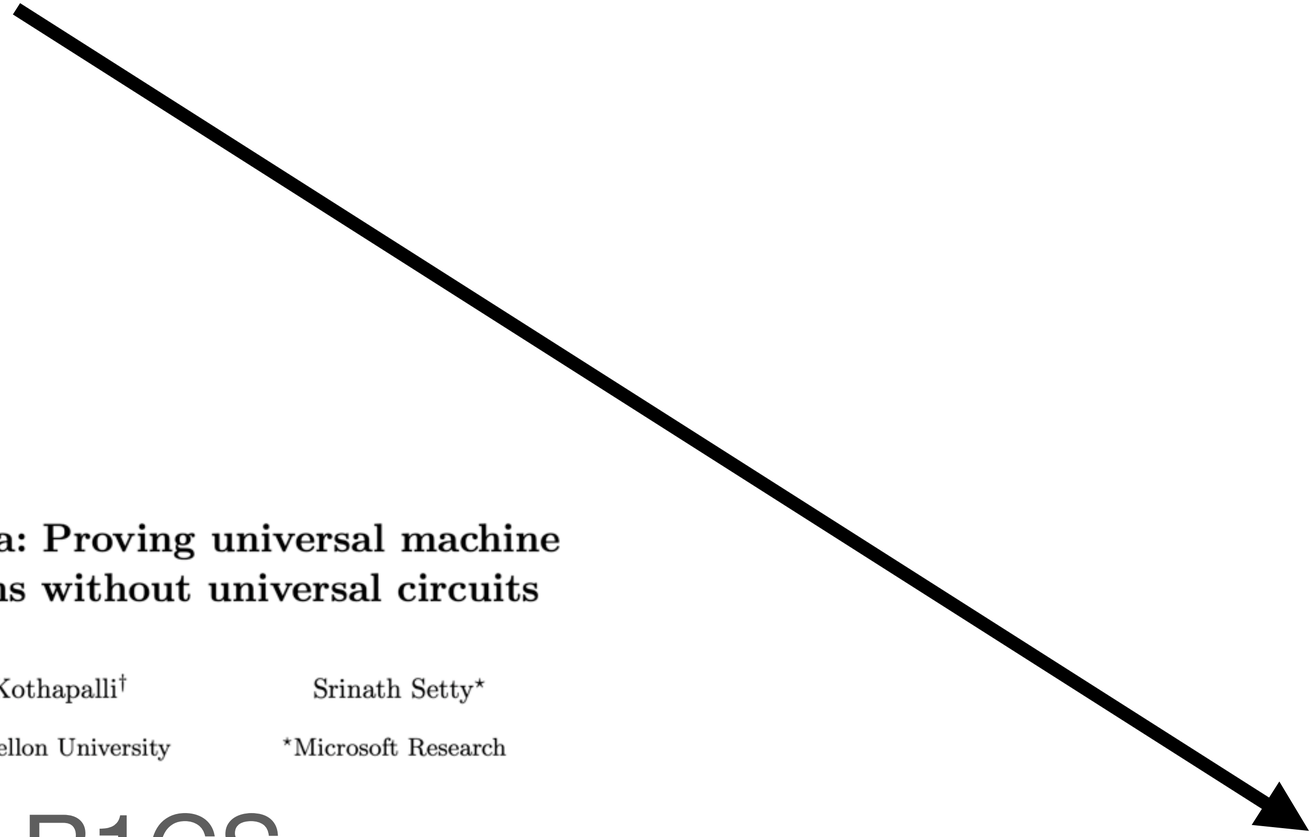
Short Proof

Nova: Recursive Zero-Knowledge Arguments from Folding Schemes

Abhiram Kothapalli[†] Srinath Setty[⋆] Ioanna Tzialla[‡]

[†]Carnegie Mellon University [⋆]Microsoft Research [‡]New York University

$|\mathcal{F}|$ R1CS



SuperNova: Proving universal machine executions without universal circuits

Abhiram Kothapalli[†] Srinath Setty[⋆]

[†]Carnegie Mellon University [⋆]Microsoft Research

$|\mathcal{F}_i|$ R1CS

"Pay-per-use"

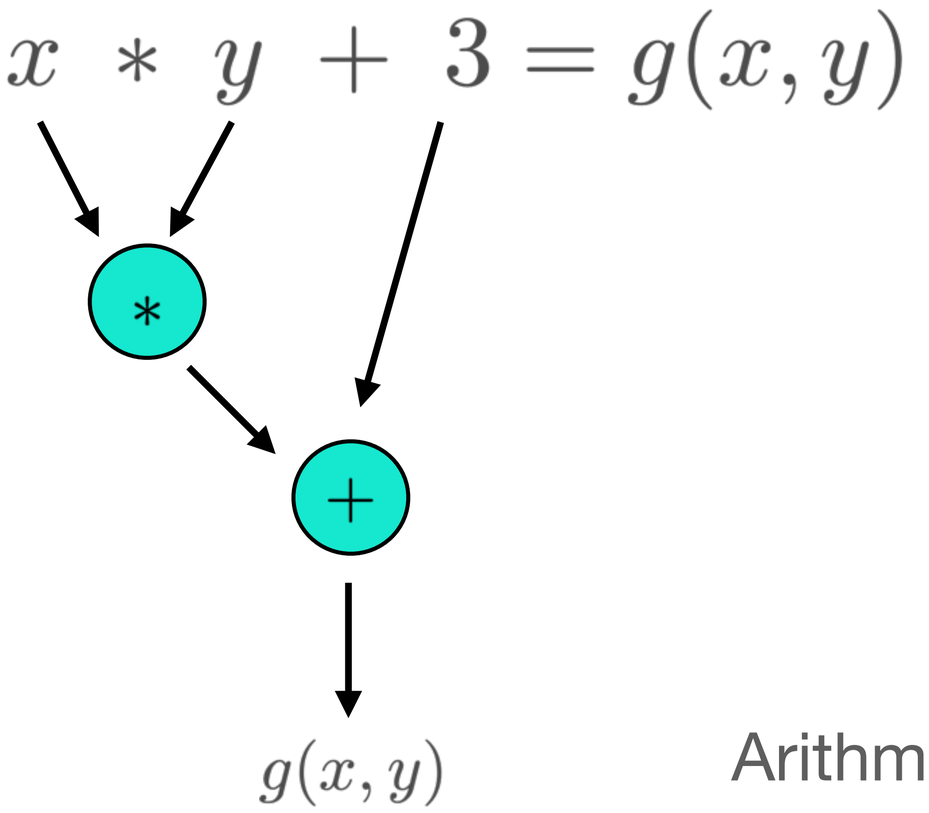
HyperNova: Recursive arguments for customizable constraint systems

CCS

Abhiram Kothapalli[†] Srinath Setty[⋆]

[†]Carnegie Mellon University [⋆]Microsoft Research

1 Multi Scalar Multiplication , 1 Sum-Check



Representation

Rank-1 Constraint Systems (**R1CS**)

Permutations over Lagrange-bases for Oecumenical Noninteractive arguments of Knowledge (**PLONK**)

Arithmetic Intermediate Representation (**AIR**)

Customizable Constraint Systems (**CCS**)

Relations and Circuits

Definition 12 (CCS [STW23]). *We define the customizable constraint system (CCS) relation $\mathcal{R}_{\mathsf{CCS}}$ as follows. Let the public parameter consists of size bounds $m, n, N, \ell, t, q, d \in \mathbb{N}$ where $n > \ell$.*

An $\mathcal{R}_{\mathsf{CCS}}$ structure s consists of:

- *a sequence of matrices $M_1, \ldots, M_t \in \mathbb{F}^{m \times n}$ with at most $N = \Omega(\max(m, n))$ non-zero entries in total;*
- *a sequence of q multisets $[S_1, \ldots, S_q]$, where an element in each multiset is from the domain $\{1, \ldots, t\}$ and the cardinality of each multiset is at most d.*
- *a sequence of q constants $[c_1, \ldots, c_q]$, where each constant is from $\mathbb{F}$.*

An $\mathcal{R}_{\mathsf{CCS}}$ instance consists of public input $\mathsf{x} \in \mathbb{F}^{\ell}$. An $\mathcal{R}_{\mathsf{CCS}}$ witness consists of a vector $w \in \mathbb{F}^{n-\ell-1}$. An $\mathcal{R}_{\mathsf{CCS}}$ structure-instance tuple (s, x) is satisfied by an $\mathcal{R}_{\mathsf{CCS}}$ witness w if

$$\sum_{i=1}^{q} c_i \cdot \bigcirc_{j \in S_i} M_j \cdot z = \mathbf{0},$$

where $z = (w, 1, \mathsf{x}) \in \mathbb{F}^n$, $M_j \cdot z$ denotes matrix-vector multiplication, $\bigcirc$ denotes the Hadamard product between vectors, and $\mathbf{0}$ is an m-sized vector with entries equal to the the additive identity in $\mathbb{F}$.

$$g(x, y) = x \cdot y + 3$$

$$M_1 = \begin{pmatrix} 1 & 0 & 0 & 0 \end{pmatrix}$$

$$M_2 = \begin{pmatrix} 0 & 1 & 0 & 0 \end{pmatrix}$$

$$S_1 = [1, 2]$$

$$M_3 = \begin{pmatrix} 0 & 0 & 0 & 1 \end{pmatrix}$$

$$S_2 = [3]$$

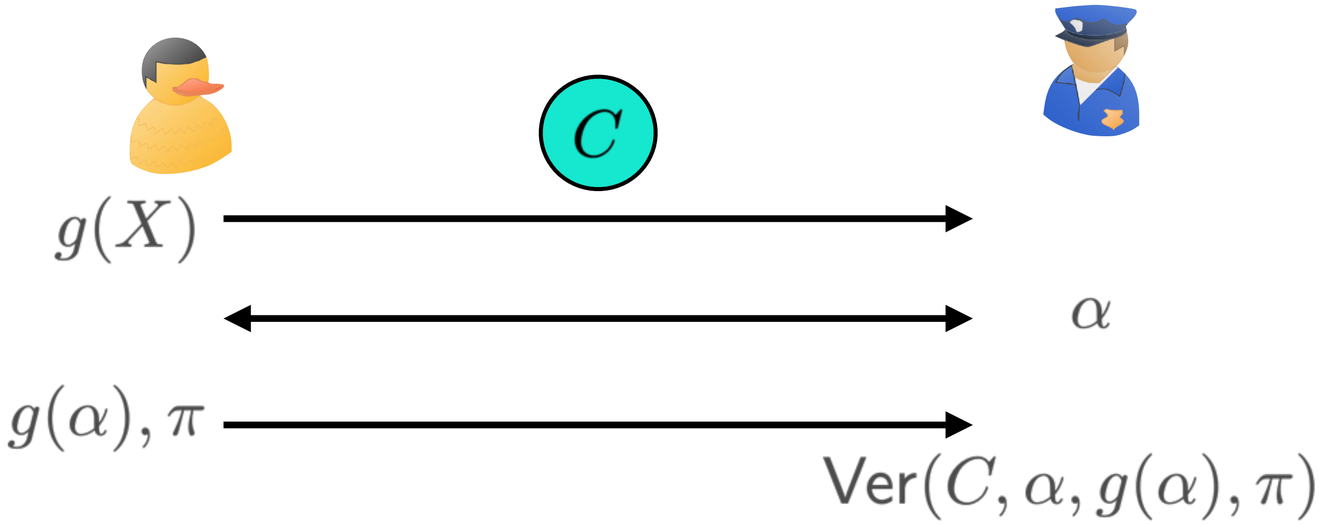
$$c_1 = c_2 = 1$$

$$z = (\omega, 1, \mathbf{x}) = (\underbrace{x, y}_{\omega}, 1, \underbrace{3}_{\mathbf{x}})$$

$$c_1(M_1 \cdot z \circ M_2 \cdot z) + c_2 M_3 \cdot z = x \cdot y + 3$$

Definition 6. *An extractable polynomial commitment scheme for multilinear polynomials over finite field $\mathbb{F}$ is a tuple of four protocols $\mathsf{PC} = (\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval})$:*

- $\mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda, \ell)$*: takes as input ℓ (the number of variables in a multivariate polynomial); produces public parameters* pp.
- $\mathcal{C} \leftarrow \mathsf{Commit}(\mathsf{pp}, g)$*: takes as input a ℓ-variate multilinear polynomial $g \in \mathbb{F}[X_1, \ldots, X_\ell]$; produces a commitment $\mathcal{C}$.*
- $b \leftarrow \mathsf{Open}(\mathsf{pp}, \mathcal{C}, g)$*: verifies the opening of commitment $\mathcal{C}$ to the ℓ-variate multilinear polynomial $g \in \mathbb{F}[X_1, \ldots, X_\ell]$; outputs $b \in \{0, 1\}$.*
- $b \leftarrow \mathsf{Eval}(\mathsf{pp}, \mathcal{C}, r, v, \ell, g)$ *is a protocol between a PPT prover $\mathcal{P}$ and verifier $\mathcal{V}$. Both $\mathcal{V}$ and $\mathcal{P}$ hold a commitment $\mathcal{C}$, the number of variables ℓ, a scalar $v \in \mathbb{F}$, and $r \in \mathbb{F}^\ell$. $\mathcal{P}$ additionally knows an ℓ-variate multilinear polynomial $g \in \mathbb{F}[\ell]$. $\mathcal{P}$ attempts to convince $\mathcal{V}$ that $g(r) = v$. At the end of the protocol, $\mathcal{V}$ outputs $b \in \{0, 1\}$.*



Completeness: if correct, must accept

Binding: once g committed, cannot change

Knowledge Soundness
evaluating the polynomial is a succinct argument of knowledge

Additively Homomorphic

$$\mathsf{Commit}(\mathsf{pp}, g_1) + \mathsf{Commit}(\mathsf{pp}, g_2) = \mathsf{Commit}(\mathsf{pp}, g_1 + g_2).$$

$$g : \{0,1\}^l \to \mathbb{F} \longrightarrow \widetilde{g} : \mathbb{F}^l \to \mathbb{F}$$

$$\forall\, x \in \{0,1\}^l \qquad \widetilde{g}(x) = g(x)$$

Definition 1 (Multilinear polynomial). *A multivariate polynomial is called a multilinear polynomial if the degree of the polynomial in each variable is at most one.*

Definition 2 (Low-degree polynomial). *A multivariate polynomial g over a finite field $\mathbb{F}$ is called low-degree polynomial if the degree of g in each variable is exponentially smaller than $|\mathbb{F}|$.*

$$\widetilde{g}(x_1, \ldots, x_l) = \sum_{e \in \{0,1\}^l} g(e) \cdot \prod_{i=1}^{l} (x_i \cdot e_i + (1 - x_i) \cdot (1 - e_i))$$

$$= \langle \left(g(0), \cdots, g(2^l - 1)\right), \left(\widetilde{eq}(x, 0), \cdots, \widetilde{eq}(x, 2^l - 1)\right) \rangle$$

$$M \in \mathbb{F}^{m \times n} \qquad M(x, y) : [m] \times [n] \to \mathbb{F}$$

$$M(x, y) : \{0,1\}^{\log(m)} \times \{0,1\}^{\log(n)} \to \mathbb{F} \longrightarrow M \longrightarrow \widetilde{M}$$

Definition 13 (Committed CCS). *Let* $PC = (\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval})$ *denote an additively-homomorphic polynomial commitment scheme for multilinear polynomials over a finite field* $\mathbb{F}$.

We define the committed customizable constraint system (CCCS) relation $\mathcal{R}_{\mathsf{CCCS}}$ *as follows. Let the public parameter consists of size bounds* $m, n, N, \ell, t, q, d \in \mathbb{N}$ *where* $n = 2 \cdot (\ell + 1)$ *and* $\mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda, s' - 1)$. *Let* $s = \log m$ *and* $s' = \log n$.

An $\mathcal{R}_{\mathsf{CCCS}}$ *structure* s *consists of:*

- *a sequence of sparse multilinear polynomials in* $s + s'$ *variables* $\widetilde{M}_1, \ldots, \widetilde{M}_t$ *such that they evaluate to a non-zero value in at most* $N = \Omega(m)$ *locations over the Boolean hypercube* $\{0,1\}^s \times \{0,1\}^{s'}$;
- *a sequence of* q *multisets* $[S_1, \ldots, S_q]$, *where an element in each multiset is from the domain* $\{1, \ldots, t\}$ *and the cardinality of each multiset is at most* d.
- *a sequence of* q *constants* $[c_1, \ldots, c_q]$, *where each constant is from* $\mathbb{F}$.

An $\mathcal{R}_{\mathsf{CCCS}}$ *instance is* (C, x), *where* C *is a commitment to a multilinear polynomial in* $s' - 1$ *variables and* $\mathsf{x} \in \mathbb{F}^\ell$. *An* $\mathcal{R}_{\mathsf{CCCS}}$ *witness consists of a multilinear polynomial* $\widetilde{w}$ *in* $s' - 1$ *variables. An* $\mathcal{R}_{\mathsf{CCCS}}$ *structure-instance tuple is satisfied by an* $\mathcal{R}_{\mathsf{CCCS}}$ *witness if* $\mathsf{Commit}(\mathsf{pp}, \widetilde{w}) = C$ *and if for all* $x \in \{0,1\}^s$,

$$\sum_{i=1}^{q} c_i \cdot \left(\Pi_{j \in S_i} \left(\sum_{y \in \{0,1\}^{\log m}} \widetilde{M}_j(x, y) \cdot \widetilde{z}(y) \right) \right) = 0,$$

where $\widetilde{z}$ *is an* s'*-variate multilinear polynomial such that* $\widetilde{z}(x) = \widetilde{(w, 1, \mathsf{x})}(x)$ *for all* $x \in \{0,1\}^{s'}$.

Similar to CCS but in multilinear polynomial representation

Commits the extended multilinear polynomial of the witness

$$\mathsf{Commit}(\mathsf{pp}, \widetilde{\omega}) = C$$

Definition 14 (Linearized committed CCS). *Let* $PC = (\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval})$ *denote an additively-homomorphic polynomial commitment scheme for multilinear polynomials over a finite field* $\mathbb{F}$.

We define the linearized committed customizable constraint system (LCCS) relation $\mathcal{R}_{\mathsf{LCCCS}}$ *as follows. Let the public parameter consists of size bounds* $m, n, N, \ell, t, q, d \in \mathbb{N}$ *where* $n = 2 \cdot (\ell + 1)$ *and* $\mathsf{pp} \leftarrow \mathsf{Gen}(1^\lambda, s' - 1)$. *Let* $s = \log m$ *and* $s' = \log n$.

An $\mathcal{R}_{\mathsf{LCCCS}}$ *structure* s *consists of:*

- *a sequence of sparse multilinear polynomials in* $s + s'$ *variables* $\widetilde{M}_1, \ldots, \widetilde{M}_t$ *such that they evaluate to a non-zero value in at most* $N = \Omega(m)$ *locations over the Boolean hypercube* $\{0,1\}^s \times \{0,1\}^{s'}$;
- *a sequence of* q *multisets* $[S_1, \ldots, S_q]$, *where an element in each multiset is from the domain* $\{1, \ldots, t\}$ *and the cardinality of each multiset is at most* d.
- *a sequence of* q *constants* $[c_1, \ldots, c_q]$, *where each constant is from* $\mathbb{F}$.

An $\mathcal{R}_{\mathsf{LCCCS}}$ *instance is* $(C, u, \mathsf{x}, r, v_1, \ldots, v_t)$, *where* $u \in \mathbb{F}$, $\mathsf{x} \in \mathbb{F}^\ell$, $r \in \mathbb{F}^s$, $v_i \in \mathbb{F}$ *for all* $i \in [t]$, *and* C *is a commitment to a multilinear polynomial in* $s' - 1$ *variables. An* $\mathcal{R}_{\mathsf{LCCCS}}$ *witness is a multilinear polynomial* $\widetilde{w}$ *in* $s' - 1$ *variables.*

An $\mathcal{R}_{\mathsf{LCCCS}}$ *structure-instance tuple is satisfied by an* $\mathcal{R}_{\mathsf{LCCCS}}$ *witness if* $\mathsf{Commit}(\mathsf{pp}, \widetilde{w}) = C$ *and if for all* $i \in [t]$, $v_i = \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_i(r, y) \cdot \widetilde{z}(y)$, *where* $\widetilde{z}$ *is an* s'*-variate multilinear polynomial such that* $z(x) = (w, u, \mathsf{x})(x)$ *for all* $x \in \{0,1\}^{s'}$.

Similar to CCCS

$$\sum_{i=1}^{q} c_i \cdot \left(\Pi_{j \in S_i} \left(\sum_{y \in \{0,1\}^{\log m}} \widetilde{M}_j(x, y) \cdot \widetilde{z}(y) \right) \right) = 0,$$

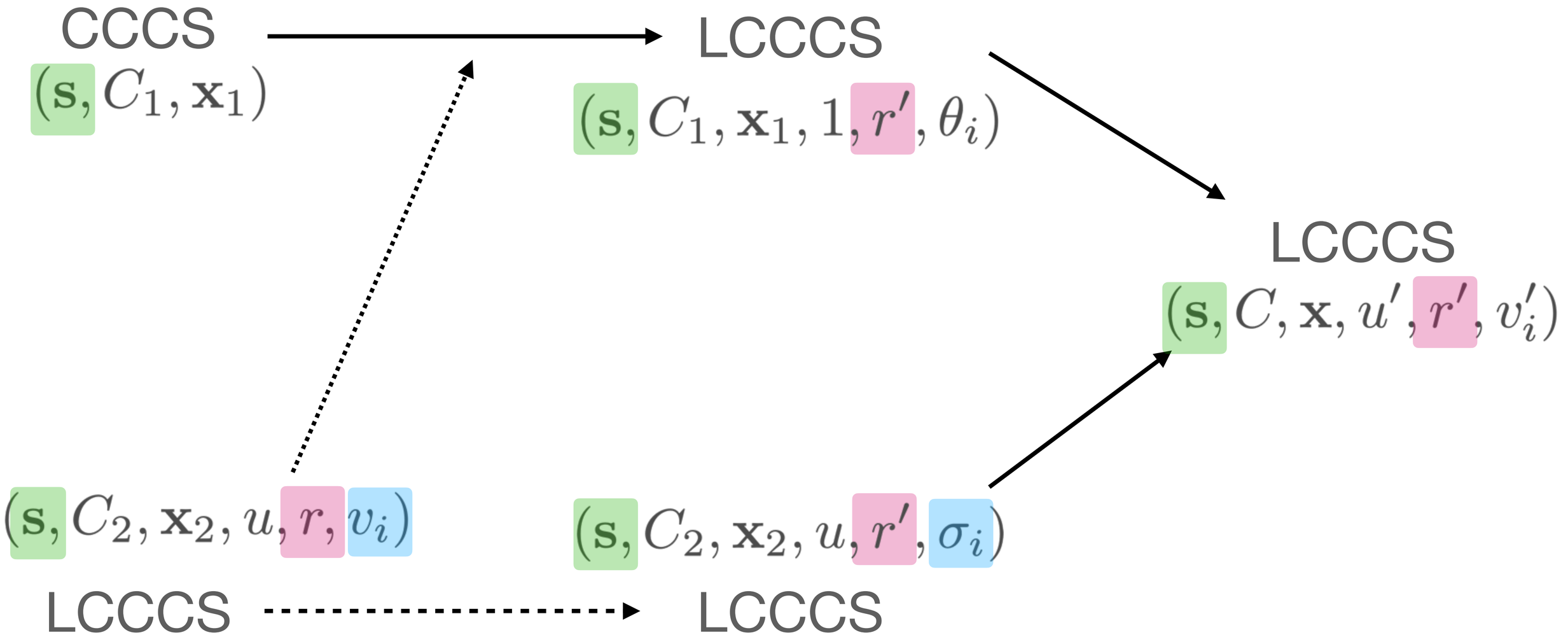
LCCCS specific evaluation in r for each S_i

$$v_i = \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_i(r, y) \cdot \widetilde{z}(y)$$

New element u in z

$$z(x) = (\widetilde{w, u, \mathsf{x}})(x)$$

[Non] Interactive Folding Scheme


CCCS
$(\mathbf{s}, C_1, \mathbf{x}_1)$
LCCCS
$(\mathbf{s}, C_1, \mathbf{x}_1, 1, r', \theta_i)$
LCCCS
$(\mathbf{s}, C, \mathbf{x}, u', r', v_i')$
$(\mathbf{s}, C_2, \mathbf{x}_2, u, r, v_i)$
$(\mathbf{s}, C_2, \mathbf{x}_2, u, r', \sigma_i)$
LCCCS
LCCCS

2.2 The sum-check protocol

Suppose there is an ℓ-variate low-degree polynomial, g, where the degree of each variable in g is at most d. Suppose that a verifier $\mathcal{V}$ is interested in checking a claim of the following form by an untrusted prover $\mathcal{P}$:

$$T = \sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_\ell \in \{0,1\}} g(x_1, x_2, \ldots, x_\ell)$$

For any ℓ-variate polynomial g with degree at most d in each variable, the sum-check protocol satisfies the following properties.

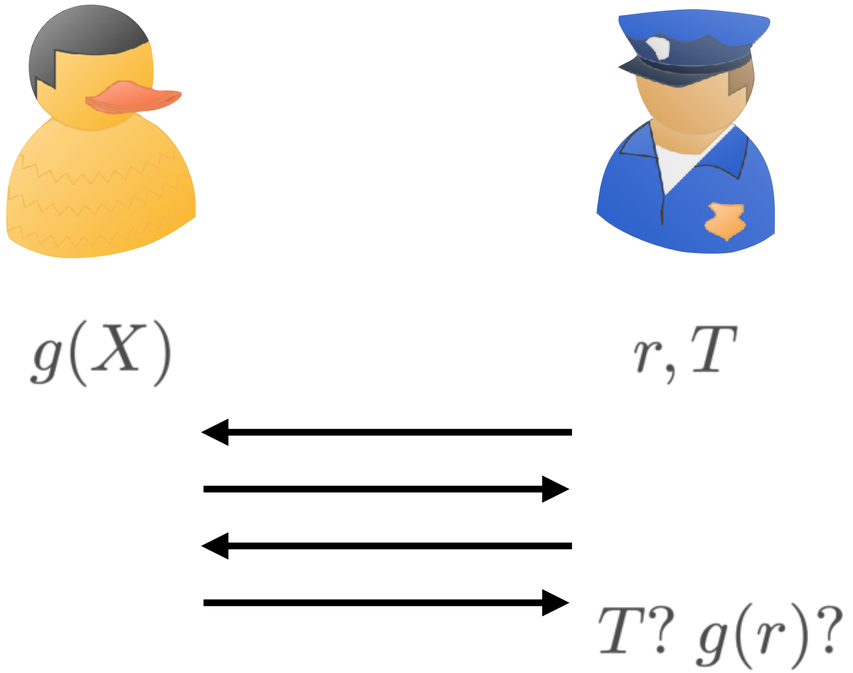
1. Completeness: If $T = \sum_{x \in \{0,1\}^\ell} g(x)$, then for all $r \in \mathbb{F}^\ell$,

$$\Pr_r[\langle \mathcal{P}, \mathcal{V}(r) \rangle (g, \ell, d, T) = g(r)] = 1.$$

2. Soundness: If $T \neq \sum_{x \in \{0,1\}^\ell} g(x)$, then for any $\mathcal{P}^\star$ and for all $r \in \mathbb{F}^\ell$,

$$\Pr_r[\langle \mathcal{P}^\star, \mathcal{V}(r) \rangle (g, \ell, d, T) = c \wedge g(r) = c] \leq \ell \cdot d / |\mathbb{F}|.$$

3. Succinctness: The communication cost is $O(\ell \cdot d)$ elements of $\mathbb{F}$.



$$T = \sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} \cdots \sum_{x_\ell \in \{0,1\}} g(x_1, x_2, \ldots, x_\ell)$$

Construction 2 (A multi-folding scheme for CCS). We construct a multi-folding scheme for $(\mathcal{R}_{\mathsf{LCCCS}}, \mathcal{R}_{\mathsf{CCCS}}, \mathsf{compat}, \mu = 1, \nu = 1)$, where compat is defined as follows.

$\underline{\mathsf{compat}(\mathsf{s}_1, \mathsf{s}_2) \to \{\mathsf{true}, \mathsf{false}\}}$

1. If $\mathsf{s}_1 = \mathsf{s}_2$, then return true, otherwise return false

Let $\mathsf{s}_1 = \mathsf{s}_2 = ([\widetilde{M}_1, \ldots, \widetilde{M}_t], [S_1, \ldots, S_q], [c_1, \ldots, c_q])$. Let $\mathsf{PC} = (\mathsf{Gen}, \mathsf{Commit}, \mathsf{Open}, \mathsf{Eval})$ denote an additively-homomorphic polynomial commitment scheme for multilinear polynomials.

$\underline{\mathcal{G}(1^\lambda) \to \mathsf{pp}}$:

1. Sample size bounds $m, n, N, \ell, t, q, d \in \mathbb{N}$ with $n = 2 \cdot (\ell + 1)$
2. $\mathsf{pp}_{\mathsf{PC}} \leftarrow \mathsf{Gen}(1^\lambda, \log n - 1)$
3. Output $(m, n, N, \ell, t, q, d, \mathsf{pp}_{\mathsf{PC}})$

$\underline{\mathcal{K}(\mathsf{pp}, (\mathsf{s}_1, \mathsf{s}_2)) \to (\mathsf{pk}, \mathsf{vk})}$:

1. Let $\mathsf{pk} \leftarrow (\mathsf{pp}, \mathsf{s}_1)$ and $\mathsf{vk} \leftarrow \mathsf{pp}$
2. Output $(\mathsf{pk}, \mathsf{vk})$

The verifier $\mathcal{V}$ takes a linearized committed CCS instance $(C_1, u, \mathsf{x}_1, r_x, v_1, \ldots, v_t)$. and a committed CCS instance (C_2, x_2) and the prover $\mathcal{P}$, in addition to the two instances, takes witnesses to both instances, $\widetilde{w_1}$ and $\widetilde{w_2}$.

Let $s = \log m$ and $s' = \log n$. Let $\widetilde{z_1} = \widetilde{(w_1, u, \mathsf{x}_1)}$ and $\widetilde{z_2} = \widetilde{(w_2, 1, \mathsf{x}_2)}$.

Sum-check proves evaluation

The prover and the verifier proceed as follows.

1. $\mathcal{V} \to \mathcal{P}$: $\mathcal{V}$ samples $\gamma \xleftarrow{\$} \mathbb{F}, \beta \xleftarrow{\$} \mathbb{F}^s$, and sends them to $\mathcal{P}$.
2. $\mathcal{V}$: Sample $r'_x \xleftarrow{\$} \mathbb{F}^s$.
3. $\mathcal{V} \leftrightarrow \mathcal{P}$: Run the sum-check protocol $c \leftarrow \langle \mathcal{P}, \mathcal{V}(r'_x) \rangle(g, s, d+1, \sum_{j \in [t]} \gamma^j \cdot v_j)$, where:

$$g(x) \coloneqq \left(\sum_{j \in [t]} \gamma^j \cdot L_j(x) \right) + \gamma^{t+1} \cdot Q(x)$$

$$L_j(x) \coloneqq \widetilde{eq}(r_x, x) \cdot \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x, y) \cdot \widetilde{z}_1(y) \right)$$

$$Q(x) \coloneqq \widetilde{eq}(\beta, x) \cdot \left(\sum_{i=1}^{q} c_i \cdot \prod_{j \in S_i} \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x, y) \cdot \widetilde{z}_2(y) \right) \right)$$

$g(r'_x)$ is the output

$$\begin{aligned} v_j &= \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(r_x, y) \cdot \widetilde{z}_1(y) \\ &= \sum_{x \in \{0,1\}^s} \widetilde{eq}(r_x, x) \cdot \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x, y) \cdot \widetilde{z}_1(y) \right) \\ &= \sum_{x \in \{0,1\}^s} L_j(x) \end{aligned}$$

$$\begin{aligned} 0 &= \sum_{i=1}^{q} c_i \cdot \prod_{j \in S_i} \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(\beta, y) \cdot \widetilde{z}_2(y) \right) \\ &= \sum_{x \in \{0,1\}^s} \widetilde{eq}(\beta, x) \cdot \sum_{i=1}^{q} c_i \cdot \prod_{j \in S_i} \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x, y) \cdot \widetilde{z}_2(y) \right) \\ &= \sum_{x \in \{0,1\}^s} Q(x) \end{aligned}$$

$$\begin{aligned} \sum_{j \in [t]} \gamma^j \cdot v_j &= \sum_{x \in \{0,1\}^s} \left(\left(\sum_{j \in [t]} \gamma^j \cdot L_j(x) \right) + \gamma^{t+1} \cdot Q(x) \right) \\ &= \sum_{x \in \{0,1\}^s} g(x) \end{aligned}$$

$$g(x) \coloneqq \left(\sum_{j\in[t]} \gamma^j \cdot L_j(x)\right) + \gamma^{t+1} \cdot Q(x)$$

$$L_j(x) \coloneqq \widetilde{eq}(r_x, x) \cdot \left(\sum_{y\in\{0,1\}^{s'}} \widetilde{M}_j(x,y) \cdot \widetilde{z}_1(y)\right)$$

$$Q(x) \coloneqq \widetilde{eq}(\beta, x) \cdot \left(\sum_{i=1}^{q} c_i \cdot \prod_{j\in S_i} \left(\sum_{y\in\{0,1\}^{s'}} \widetilde{M}_j(x,y) \cdot \widetilde{z}_2(y)\right)\right)$$

4. $\mathcal{P} \to \mathcal{V}$: $((\sigma_1, \ldots, \sigma_t), (\theta_1, \ldots, \theta_t))$, where for all $i \in [t]$:

$$\sigma_i = \sum_{y\in\{0,1\}^{s'}} \widetilde{M}_i(r'_x, y) \cdot \widetilde{z}_1(y)$$
$$\theta_i = \sum_{y\in\{0,1\}^{s'}} \widetilde{M}_i(r'_x, y) \cdot \widetilde{z}_2(y)$$

5. $\mathcal{V}$: Compute $e_1 \leftarrow \widetilde{eq}(r_x, r'_x)$ and $e_2 \leftarrow \widetilde{eq}(\beta, r'_x)$, and abort if:

$$c \neq \left(\sum_{j\in[t]} \gamma^j \cdot e_1 \cdot \sigma_j + \gamma^{t+1} \cdot e_2 \cdot \left(\sum_{i=1}^{q} c_i \cdot \prod_{j\in S_i} \theta_j\right)\right)$$

Compute new partial evaluations

$$\begin{aligned} c &= g(r'_x) \\ &= \left(\left(\sum_{j\in[t]} \gamma^j \cdot L_j(r'_x)\right) + \gamma^{t+1} \cdot Q(r'_x)\right) \\ &= \left(\left(\sum_{j\in[t]} \gamma^j \cdot e_1 \cdot \sigma_j\right) + \gamma^{t+1} \cdot e_2 \sum_{i\in[q]} c_i \cdot \prod_{j\in S_i} \theta_j\right). \end{aligned}$$

This implies that the verifier will not abort.

and verifying they are correct

6. $\mathcal{V} \to \mathcal{P}$: $\mathcal{V}$ samples $\rho \xleftarrow{\$} \mathbb{F}$ and sends it to $\mathcal{P}$.

7. $\mathcal{V}, \mathcal{P}$: Output the folded linearized committed CCS instance $(C', u', \mathsf{x}', r'_x, v'_1, \ldots, v'_t)$, where for all $i \in [t]$:

$$
\begin{aligned}
C' &\leftarrow C_1 + \rho \cdot C_2 \\
u' &\leftarrow u + \rho \cdot 1 \\
\mathsf{x}' &\leftarrow \mathsf{x}_1 + \rho \cdot \mathsf{x}_2 \\
v'_i &\leftarrow \sigma_i + \rho \cdot \theta_i
\end{aligned}
$$

Homomorphic Additive Commitment Scheme

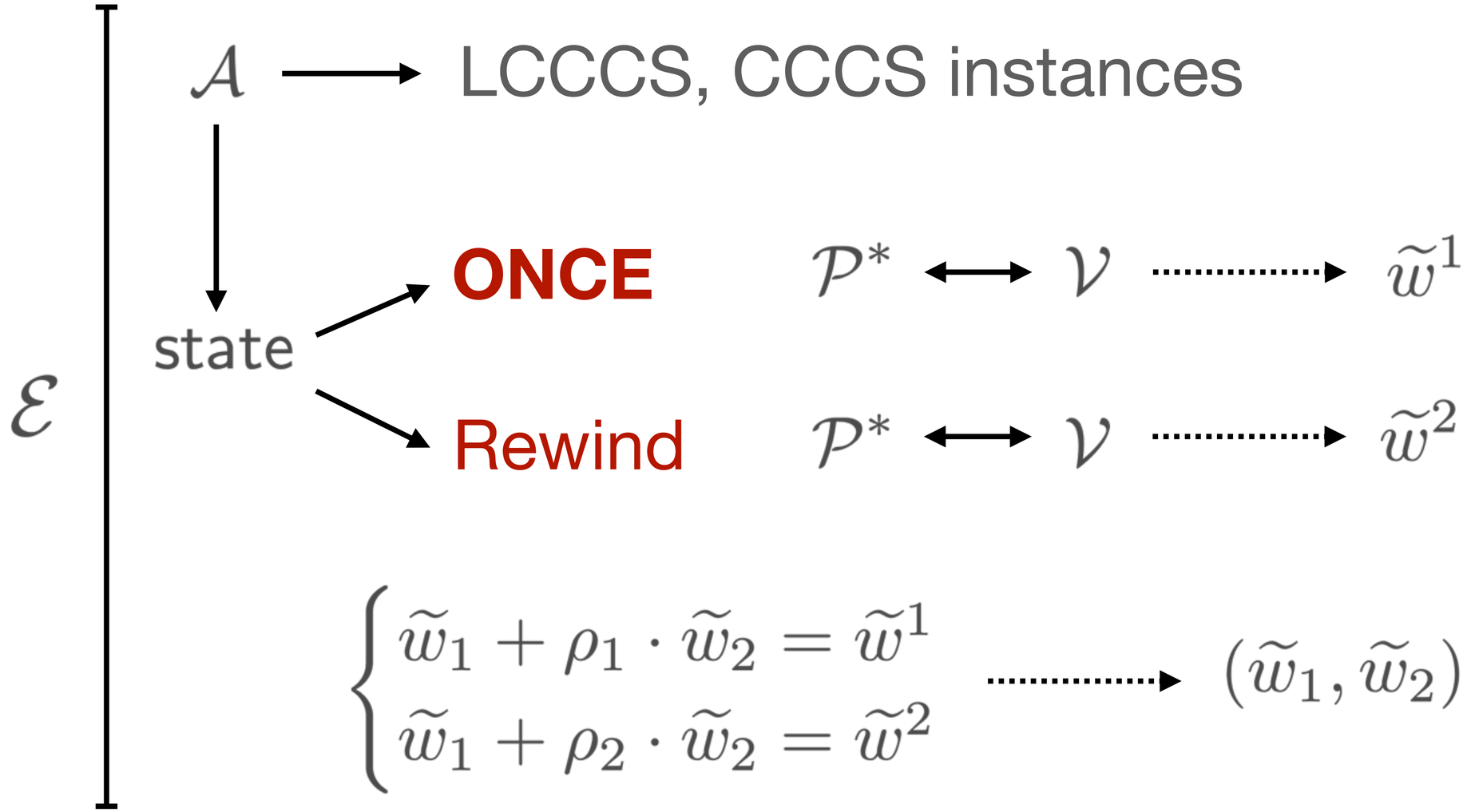
8. $\mathcal{P}$: Output the folded witness $\widetilde{w}' \leftarrow \widetilde{w}_1 + \rho \cdot \widetilde{w}_2$.

Lemma 4 (Perfect Completeness). *Construction 2 satisfies perfect completeness.*

Non-Interactive via Fiat-Shamir on the Sum-Check protocol

NIFS Soundness

Lemma 5 (Knowledge Soundness). *Construction 2 satisfies knowledge soundness.*



the interaction until $\mathcal{P}^*$ succeeds. Thus, the expected number of times $\mathcal{E}$ runs the interaction is

$$1 + \Pr[\text{First call to } \langle \mathcal{P}^*, \mathcal{V} \rangle \text{ succeeds}] \cdot \frac{1}{\Pr[\langle \mathcal{P}^*, \mathcal{V} \rangle \text{ succeeds}]} = 1 + \epsilon \cdot \frac{1}{\epsilon} = 2.$$

Therefore, we have that the extractor runs in expected polynomial-time.

Expected number of interactions

that $\rho^{(1)} \neq \rho^{(2)}$ with probability $1/|\mathbb{F}|$. Thus, we have that the probability the extractor successfully produces some output in under $|\mathbb{F}|$ rewinding steps is

$$\left(1 - \frac{2}{|\mathbb{F}|}\right) \cdot \epsilon \cdot \left(1 - \frac{1}{|\mathbb{F}|}\right) = \epsilon - \mathsf{negl}(\lambda).$$

Rewind does not abort in more than field size steps

Success first interaction

Random $\rho_2 \neq \rho_1$

Markov's Inequality $\Pr(X \geq |A|) \leq \frac{\mathbb{E}(X)}{|A|}$

Indeed, for $i \in \{1,2\}$, let $\varphi^{(i)} = (C^{(i)}, u^{(i)}, \mathsf{x}^{(i)}, r_x^{(i)}, v_1^{(i)}, \ldots, v_t^{(i)})$.

$$
\begin{aligned}
&\mathsf{Commit}(\mathsf{pp}, \widetilde{w}_1) + \rho^{(i)} \cdot \mathsf{Commit}(\mathsf{pp}, \widetilde{w}_2) \\
&= \mathsf{Commit}(\mathsf{pp}, \widetilde{w}_1 + \rho^{(i)} \cdot \widetilde{w}_2) && \text{By additive homomorphism.} \\
&= \mathsf{Commit}(\mathsf{pp}, \widetilde{w}^{(i)}) && \text{By Equation (1).} \\
&= C^{(i)} && \text{Witness } \widetilde{w}^{(i)} \text{ is a satisfying opening.} \\
&= C_1 + \rho^{(i)} \cdot C_2 && \text{By the verifier's computation.}
\end{aligned}
$$

Interpolating, we have that

$$\mathsf{Commit}(\mathsf{pp}, \widetilde{w}_1) = C_1 \tag{2}$$
$$\mathsf{Commit}(\mathsf{pp}, \widetilde{w}_2) = C_2. \tag{3}$$

Now, because $\widetilde{w}^{(i)}$ is a satisfying witness, for $i \in \{1,2\}$ we have for all $j \in [t]$ that

$$v_j^{(i)} = \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(r'_x, y) \cdot \widetilde{z}^{(i)}(y),$$

where $\widetilde{z}^{(i)} = (w^{(i)}, \widetilde{u^{(i)}}, \mathsf{x}^{(i)})$.

However, by Equations (1) and (4), for $i \in \{1,2\}$ and $j \in [t]$, this implies that

$$\sigma_j + \rho^{(i)} \cdot \theta_j = \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(r'_x, y) \cdot \widetilde{z}_1(y) + \rho^{(i)} \cdot \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(r'_x, y) \cdot \widetilde{z}_2(y),$$

where $\widetilde{z}_1 = (\widetilde{w_1, u}, \mathsf{x}_1)$ and $\widetilde{z}_2 = (\widetilde{w_2, 1}, \mathsf{x}_2)$. Interpolating, we have that, for all $j \in [t]$

$$\sigma_j = \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(r'_x, y) \cdot \widetilde{z}_1(y)$$
$$\theta_j = \sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(r'_x, y) \cdot \widetilde{z}_2(y)$$

By hom. linearity of PolyCom witnesses are correct openings

$\widetilde{\omega}^{(i)}$ are correct witnesses

$$v_j^{(i)}, \rho^{(i)}, \sigma_j, \theta_j$$

$$\begin{cases} \widetilde{w}_1 + \rho_1 \cdot \widetilde{w}_2 = \widetilde{w}^1 \\ \widetilde{w}_1 + \rho_2 \cdot \widetilde{w}_2 = \widetilde{w}^2 \end{cases}$$

$$\downarrow$$

$$(\sigma_j, \theta_j) \text{ w.r.t. } (\widetilde{z}_1, \widetilde{z}_2)$$

$$\downarrow$$

$$c = g(r'_x)$$

Indeed, for $i \in \{1,2\}$, let $\varphi^{(i)} = (C^{(i)}, u^{(i)}, \mathsf{x}^{(i)}, r_x^{(i)}, v_1^{(i)}, \ldots, v_t^{(i)})$.

by the soundness of the sum-check protocol, this implies that with probability $1 - O(d \cdot s)/|\mathbb{F}| = 1 - \mathsf{negl}(\lambda)$ over the choice of r'_x,

$$\sum_{j \in [t]} \gamma^j \cdot v_j + \gamma^{t+1} \cdot 0 = \sum_{x \in \{0,1\}^s} g(x)$$

$$= \sum_{x \in \{0,1\}^s} \left(\sum_{j \in [t]} \gamma^j \cdot L_j(x) + \gamma^{t+1} \cdot Q(x) \right)$$

$$= \sum_{j \in [t]} \gamma^j \cdot \left(\sum_{x \in \{0,1\}^s} L_j(x) \right) + \gamma^{t+1} \cdot \sum_{x \in \{0,1\}^s} Q(x)$$

Sum-Check soundness

By the Schwartz-Zippel lemma [Sch80], this implies that with probability $1 - O(t)/|\mathbb{F}| = 1 - \mathsf{negl}(\lambda)$ over the choice of γ, we have

$$v_j = \sum_{x \in \{0,1\}^s} L_j(x)$$

for all $j \in [t]$ and

$$0 = \sum_{x \in \{0,1\}^s} Q(x).$$

Schwartz-Zippel lemma

Prob. of randomly getting a root of a polynomial

$$\mathsf{Pr}_{r_i \in_U A}\big(g(r_1, \ldots, r_l)\big) \leq \frac{\partial(g)}{|A|}$$

Indeed, for $i \in \{1,2\}$, let $\varphi^{(i)} = (C^{(i)}, u^{(i)}, \mathsf{x}^{(i)}, r_x^{(i)}, v_1^{(i)}, \ldots, v_t^{(i)})$.

Now, for all $j \in [t]$, we have

$$\begin{aligned} v_j &= \sum_{x \in \{0,1\}^s} L_j(x) \\ &= \sum_{x \in \{0,1\}^s} \widetilde{eq}(r_x, x) \cdot \left(\sum_{y \in \{0,1\}^{s'}} M_j(x,y) \cdot \widetilde{z}_1(y) \right) \\ &= \sum_{y \in \{0,1\}^{s'}} M_j(r_x, y) \cdot \widetilde{z}_1(y) \end{aligned}$$

Finally, we have that

$$\begin{aligned} 0 &= \sum_{x \in \{0,1\}^s} Q(x) \\ &= \sum_{x \in \{0,1\}^s} \widetilde{eq}(\beta, x) \cdot \left(\sum_{i=1}^{q} c_i \cdot \prod_{j \in S_i} \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x,y) \cdot \widetilde{z}_2(y) \right) \right) \\ &= \sum_{i=1}^{q} c_i \cdot \prod_{j \in S_i} \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(\beta, y) \cdot \widetilde{z}_2(y) \right) \end{aligned}$$

By the Schwartz-Zippel lemma, this implies that with probability $1 - s/|\mathbb{F}| = 1 - \mathsf{negl}(\lambda)$ over the choice of β, we have that for all $x \in \{0,1\}^s$

$$0 = \sum_{i=1}^{q} c_i \cdot \prod_{j \in S_i} \left(\sum_{y \in \{0,1\}^{s'}} \widetilde{M}_j(x,y) \cdot \widetilde{z}_2(y) \right)$$

Good $L_j(x), Q(x)$

Definition of v_j L_j

Definition of $Q(x)$

What about β ?

Schwartz-Zippel Lemma

HyperNova

HyperNova: IVC from multi-folding schemes

Direct Construction

6.1 A direct approach

This section provides a direct construction of IVC using the non-interactive multi-folding scheme. Our construction below is an adaptation of Nova's IVC scheme to the case of multi-folding schemes.

Nova as a Black-Box

6.2 A simple approach to build HyperNova: Use Nova as a black box

We design a step circuit for Nova that runs the verifier's logic in the non-interactive multi-folding scheme for $\mathcal{R}_{\mathsf{LCCCS}}$ and $\mathcal{R}_{\mathsf{CCCS}}$. The step circuit is encoded with R1CS (a popular NP-complete constraint system [GGPR13]) and proven incrementally with Nova, but the step circuit is only in charge of running the verifier of the non-interactive multi-folding scheme, in addition to simple bookkeeping. As a result, this provides an IVC scheme, where each step of the incremental computation is expressed with committed CCS. Furthermore, we achieve this with a black box use of an IVC scheme for R1CS.

Consider a polynomial-time function F that takes non-deterministic input and a cryptographic hash function hash. We define our augmented function F', where all input arguments are taken as non-deterministic advice, as follows.

$F'(\mathsf{vk}, \mathsf{U}_i, \mathsf{u}_i, (i, z_0, z_i), \omega_i, \pi) \to \mathsf{x}$:

1. If $i = 0$, output $\mathsf{hash}(\mathsf{vk}, 1, z_0, F(z_0, \omega_0), \mathsf{u}_\perp)$
2. Otherwise:
   (a) Check that $\mathsf{u}_i.\mathsf{x} = \mathsf{hash}(\mathsf{vk}, i, z_0, z_i, \mathsf{U}_i)$, where $\mathsf{u}_i.\mathsf{x}$ is the public IO of u_i
   (b) Compute $\mathsf{U}_{i+1} \leftarrow \mathsf{NIFS.V}(\mathsf{vk}, \mathsf{U}_i, \mathsf{u}_i, \pi)$
   (c) Output $\mathsf{hash}(\mathsf{vk}, i+1, z_0, F(z_i, \omega_i), \mathsf{U}_{i+1})$

$$\mathsf{H}(\mathsf{vk}, 1, z_0, F(z_0, \omega_0), u_\perp)$$

$$\mathsf{NIFS.V}(\mathsf{vk}, U_1, u_1, \pi_1)$$

$$\mathsf{H}(\mathsf{vk}, 2, z_1, F(z_1, \omega_1), U_1)$$

$$\mathsf{NIFS.V}(\mathsf{vk}, U_2, u_2, \pi_2)$$

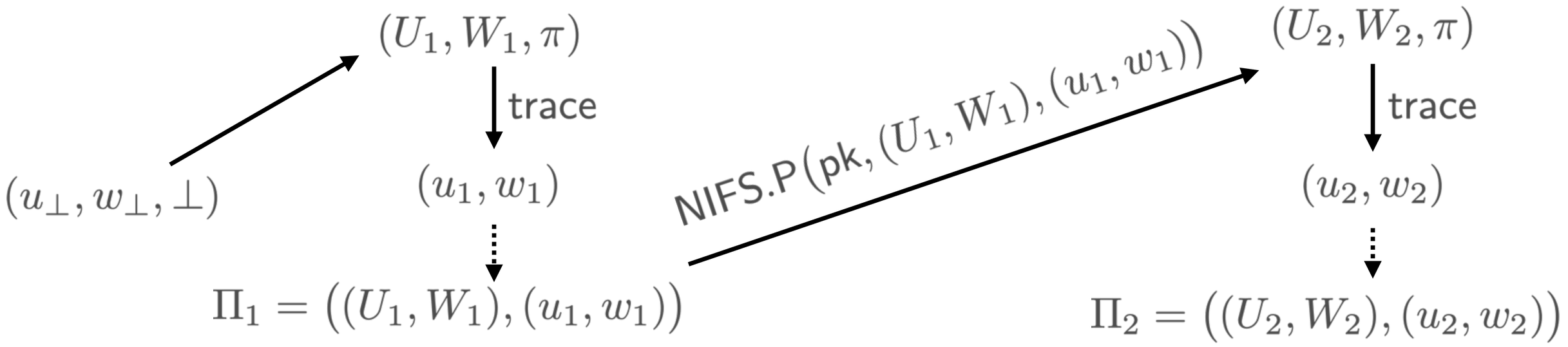
$$\mathsf{H}(\mathsf{vk}, 3, z_2, F(z_2, \omega_2), U_2)$$

Because F' can be computed in polynomial time, it can be represented as a $\mathcal{R}_{\mathsf{CCCS}}$ structure. Let

$$(\mathsf{u}_{i+1}, \mathsf{w}_{i+1}) \leftarrow \mathsf{trace}(F', (\mathsf{vk}, \mathsf{U}_i, \mathsf{u}_i, (i, z_0, z_i), \omega_i, \pi))$$

denote the satisfying $\mathcal{R}_{\mathsf{CCCS}}$ instance-witness pair $(\mathsf{u}_{i+1}, \mathsf{w}_{i+1})$ for the execution of F' on non-deterministic advice $(\mathsf{vk}, \mathsf{U}_i, \mathsf{u}_i, (i, z_0, z_i), \omega_i, \pi)$.

$$F' \in \mathcal{R}_{\mathsf{CCCS}}$$



$\mathcal{G}(1^\lambda) \to \mathsf{pp}$: Output $\mathsf{NIFS.G}(1^\lambda)$.

$\mathcal{K}(\mathsf{pp}, F) \to (\mathsf{pk}, \mathsf{vk})$:

1. Compute $(\mathsf{pk}_{\mathsf{fs}}, \mathsf{vk}_{\mathsf{fs}}) \leftarrow \mathsf{NIFS.K}(\mathsf{pp}, F')$
2. Output $(\mathsf{pk}, \mathsf{vk}) \leftarrow ((F, \mathsf{pk}_{\mathsf{fs}}), (F, \mathsf{vk}_{\mathsf{fs}}))$.

$\mathcal{P}(\mathsf{pk}, (i, z_0, z_i), \omega_i, \Pi_i) \to \Pi_{i+1}$:

1. Parse Π_i as $((\mathsf{U}_i, \mathsf{W}_i), (\mathsf{u}_i, \mathsf{w}_i))$
2. $(\mathsf{U}_{i+1}, \mathsf{W}_{i+1}, \pi) \leftarrow$ if $i = 0$ $\{(\mathsf{u}_\perp, \mathsf{w}_\perp, \perp)\}$ else $\{\mathsf{NIFS.P}(\mathsf{pk}, (\mathsf{U}_i, \mathsf{W}_i), (\mathsf{u}_i, \mathsf{w}_i))\}$
3. Compute $(\mathsf{u}_{i+1}, \mathsf{w}_{i+1}) \leftarrow \mathsf{trace}(F', (\mathsf{vk}, \mathsf{U}_i, \mathsf{u}_i, (i, z_0, z_i), \omega_i, \pi))$
4. Output $\Pi_{i+1} \leftarrow ((\mathsf{U}_{i+1}, \mathsf{W}_{i+1}), (\mathsf{u}_{i+1}, \mathsf{w}_{i+1}))$

$$\Pi_1 = \big((U_1, W_1), (u_1, w_1)\big)$$

$$u_1.x \stackrel{?}{=} \mathsf{H}(\mathsf{vk}, 1, z_0, z_1, U_1) \longleftarrow F' \text{ evaluation!}$$

$$\begin{aligned} \mathsf{u}_{i+1}.\mathsf{x} &= \mathsf{hash}(\mathsf{vk}, \mathsf{u}_i.i + 1, \mathsf{u}_i.z_0, F(z_i, w_i), \mathsf{NIFS.V}(\mathsf{vk}, \mathsf{U}_i, \mathsf{u}_i, \pi)) \\ &= \mathsf{hash}(\mathsf{vk}, \mathsf{u}_i.i + 1, \mathsf{u}_i.z_0, z_{i+1}, \mathsf{U}_{i+1}) \end{aligned} \tag{5}$$

$\mathcal{V}(\mathsf{vk}, (i, z_0, z_i), \Pi_i) \to \{0, 1\}$:

1. If $i = 0$, check that $z_i = z_0$
2. Otherwise:
   (a) Parse Π_i as $((\mathsf{U}_i, \mathsf{W}_i), (\mathsf{u}_i, \mathsf{w}_i))$
   (b) Check that $\mathsf{u}_i.\mathsf{x} = \mathsf{hash}(\mathsf{vk}, i, z_0, z_i, \mathsf{U}_i)$
   (c) Check that W_i and w_i are satisfying witnesses to U_i and u_i with respect to the structure corresponding to F'.

W_1, w_1 witnesses of U_1, u_1 w.r.t. F'

HyperNova Soundness

By induction:

$\exists\ \mathcal{E}_i$

$\widetilde{\mathcal{P}}_{i-1}$

Interactive adversary against NIFS

$\widetilde{\mathcal{E}}_{i-1}$

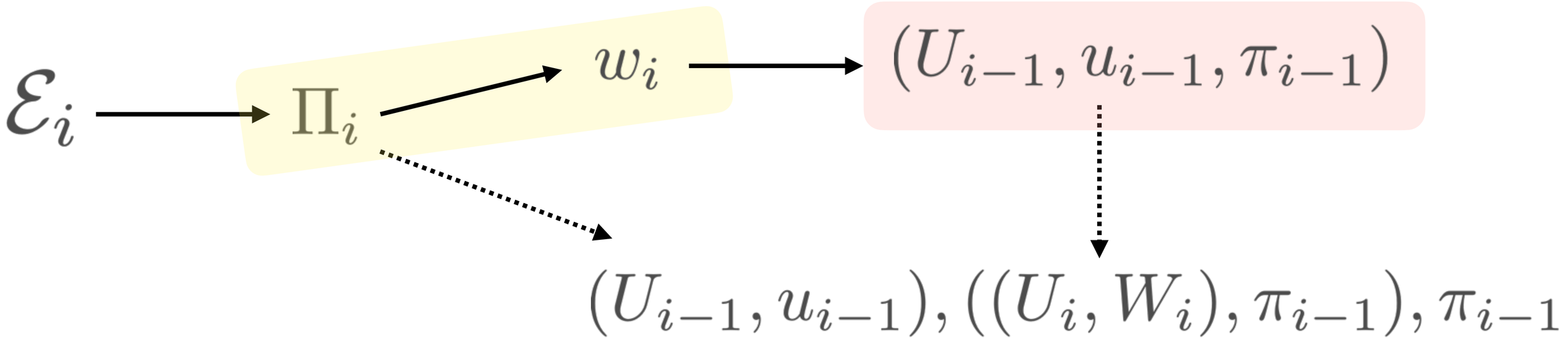
Interactive adversary against HyperNova

$\mathcal{E}_{i-1}$

Non Interactive

In the base case, for $i = n$, let $\mathcal{E}_n(\mathsf{pp}; \mathsf{r})$ output $(\perp, \perp, \Pi_n)$ where Π_n is the output of $\mathcal{P}^*(\mathsf{pp}; \mathsf{r})$. By the premise, $\mathcal{E}_n$ succeeds with probability ϵ in expected polynomial-time.

Induction base case



$\widetilde{\mathcal{P}}_{i-1}(\mathsf{pp}; \mathsf{r})$:

1. Let $((z_i, \ldots, z_{n-1}), (\omega_i, \ldots, \omega_{n-1}), \Pi_i) \leftarrow \mathcal{E}_i(\mathsf{pp}; \mathsf{r})$.
2. Parse Π_i as $((\mathsf{U}_i, \mathsf{W}_i), (\mathsf{u}_i, \mathsf{w}_i))$.
3. Parse w_i to retrieve $(\mathsf{U}_{i-1}, \mathsf{u}_{i-1}, \pi_{i-1})$.
4. Output the common structure corresponding to F' unfolded instances $(\mathsf{U}_{i-1}, \mathsf{u}_{i-1})$, folded instance witness pairs $((\mathsf{U}_i, \mathsf{W}_i), \pi_{i-1})$, and folding proof π_{i-1}.

Therefore, we have that w_i is indeed a satisfying assignment for F' (and not just a trivially satisfying witness). Then, by the construction of F' and the binding property of the hash function, we have that

$$\mathsf{U}_i = \mathsf{NIFS.V}(\mathsf{vk}, \mathsf{U}_{i-1}, \mathsf{u}_{i-1}, \pi_{i-1})$$

with probability $\epsilon - \mathsf{negl}(\lambda)$. Thus, $\widetilde{\mathcal{P}}_{i-1}$ succeeds in producing an accepting folded instance-witness pair $(\mathsf{U}_i, \mathsf{W}_i)$, for instances $(\mathsf{U}_{i-1}, \mathsf{u}_{i-1})$, with probability $\epsilon - \mathsf{negl}(\lambda)$ in expected polynomial-time.

Then, by the knowledge soundness of the underlying non-interactive multi-folding scheme (Assumption 1) there exists an extractor $\widetilde{\mathcal{E}}_{i-1}$ that outputs $(\mathsf{w}_{i-1}, \mathsf{W}_{i-1})$ such that $(\mathsf{u}_{i-1}, \mathsf{w}_{i-1})$ and $(\mathsf{U}_{i-1}, \mathsf{W}_{i-1})$ satisfy F' with probability $\epsilon - \mathsf{negl}(\lambda)$ in expected polynomial-time.

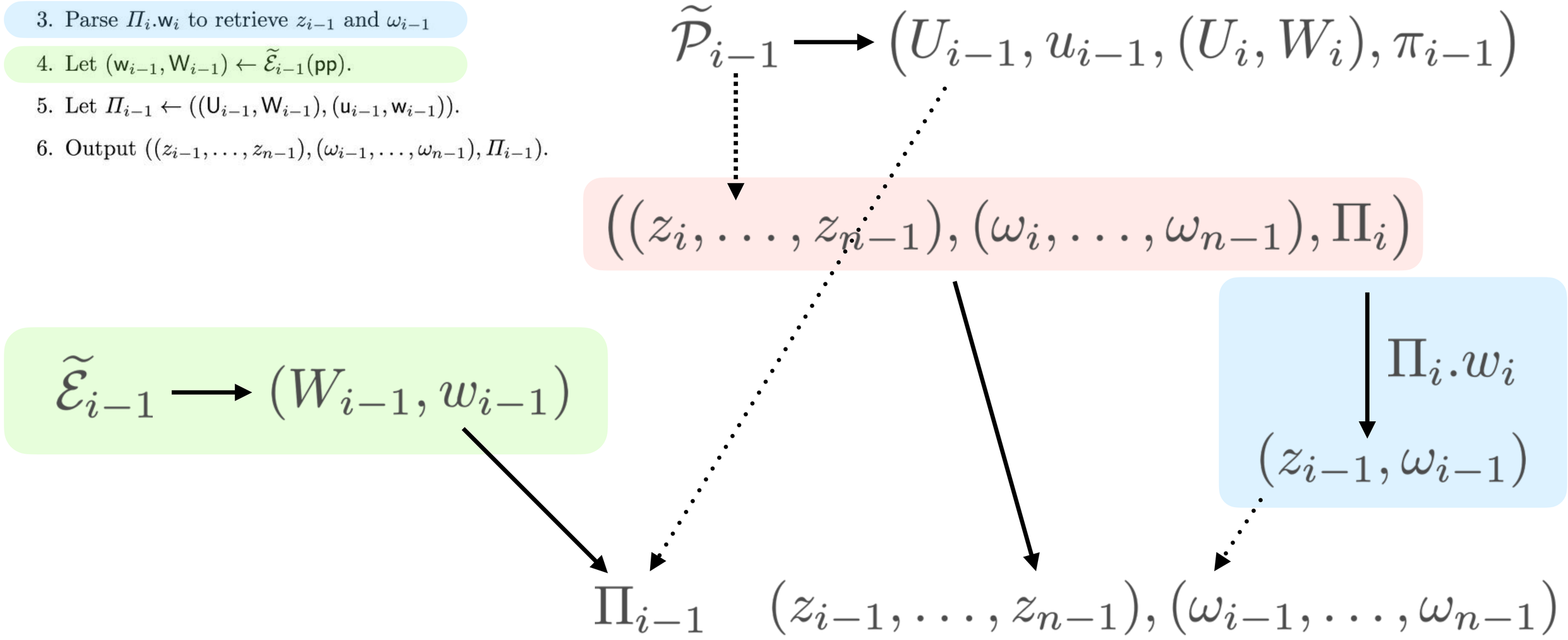
F' hash function, binds the values

$$U_i \longrightarrow U_{i-1}, u_{i-1}, \pi_{i-1}$$

By knowledge soundness on the NIFS, we get $\widetilde{\mathcal{E}}_{i-1}$

$\underline{\mathcal{E}_{i-1}(\mathsf{pp};\mathsf{r})}$:

1. $((\mathsf{U}_{i-1},\mathsf{u}_{i-1}),(\mathsf{U}_i,\mathsf{W}_i),\pi_{i-1}) \leftarrow \widetilde{\mathcal{P}}_{i-1}(\mathsf{pp};\mathsf{r})$
2. Retrieve $((z_i,\ldots,z_{n-1}),(\omega_i,\ldots,\omega_{n-1}),\Pi_i)$ from the internal state of $\widetilde{\mathcal{P}}_{i-1}$.
3. Parse $\Pi_i.\mathsf{w}_i$ to retrieve z_{i-1} and ω_{i-1}
4. Let $(\mathsf{w}_{i-1},\mathsf{W}_{i-1}) \leftarrow \widetilde{\mathcal{E}}_{i-1}(\mathsf{pp})$.
5. Let $\Pi_{i-1} \leftarrow ((\mathsf{U}_{i-1},\mathsf{W}_{i-1}),(\mathsf{u}_{i-1},\mathsf{w}_{i-1}))$.
6. Output $((z_{i-1},\ldots,z_{n-1}),(\omega_{i-1},\ldots,\omega_{n-1}),\Pi_{i-1})$.



F' definition

H binding

$F(z_{i-1}, \omega_{i-1}) = z_i$

(u_i, w_i) valid instance

Extract U_{i-1} from w_i

$\mathsf{u}_{i-1}.\mathsf{x} = \mathsf{hash}(\mathsf{vk}, i-1, z_0, z_{i-1}, \mathsf{U}_{i-1})$

Π_{i-1} valid

HyperNova Efficiency

Lemma 8 (Efficiency). *When instantiated with the Pedersen commitment scheme, we have that* $|F'| = |F| + o(\mathsf{G} + 2 \cdot \mathsf{H} + d \cdot \log m \cdot \mathsf{F} + \log m \cdot \mathsf{R})$, *where* $|F|$ *denotes the number of CCS constraints to encode a function* F, G *is the number of constraints required to encode a group scalar multiplication,* H *is the number of constraints required to encode* hash, F *is the number of constraints to encode field operations, and* R *is the number of constraints to encode the RO* ρ.

 # CCS constrains for

$\lvert F \rvert$	F
G	group scalar multiplication
$2 \cdot \mathsf{H}$	hash
SumCheck $d \log(m) \cdot \mathsf{F}$	field operation
SumCheck $\log(m) \cdot \mathsf{R}$	random oracle calls
d	degree polynomial / size of multiset
$\log(m)$	matrix output size in bit

HyperNova via Nova

We first define a non-deterministic polynomial-time function step, represented as an R1CS structure, that iteratively folds instances expressed in $\mathcal{R}_{\mathsf{CCCS}}$.

$\mathsf{step}(\mathsf{vk}, \mathsf{U}_i, z_i; (\mathsf{u}, \pi)) \to (\mathsf{vk}, \mathsf{U}_{i+1}, z_{i+1})$

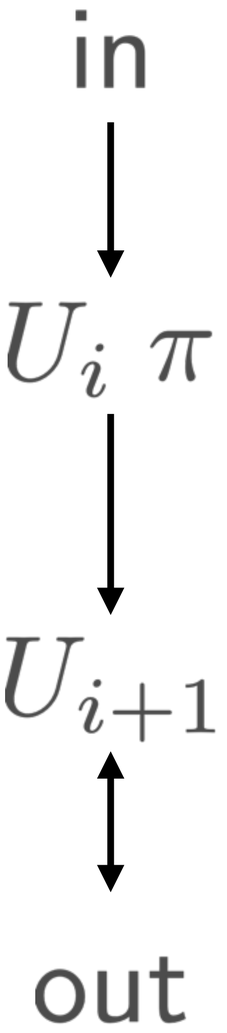
1. Parse $\mathsf{u.x}$ as $(\mathsf{in}, \mathsf{out})$
2. Check that $\mathsf{in} = z_i$
3. Compute $\mathsf{U}_{i+1} \leftarrow \mathsf{NIFS.V}(\mathsf{vk}, \mathsf{U}_i, \mathsf{u}, \pi)$
4. Output $(\mathsf{vk}, \mathsf{U}_{i+1}, \mathsf{out})$

Given F, expressed as an $\mathcal{R}_{\mathsf{CCCS}}$ structure, we define the corresponding IVC scheme $(\mathcal{G}, \mathcal{K}, \mathcal{P}, \mathcal{V})$, which uses Nova in a black-box manner.

$\mathcal{G}(1^\lambda) \to \mathsf{pp}$: Output $(\mathsf{NIFS.G}(1^\lambda), \mathsf{IVC.G}(1^\lambda))$

$\mathcal{K}((\mathsf{pp}_{\mathsf{NIFS}}, \mathsf{pp}_{\mathsf{IVC}}), F) \to (\mathsf{pk}, \mathsf{vk})$:

1. Compute $(\mathsf{pk}_{\mathsf{NIFS}}, \mathsf{vk}_{\mathsf{NIFS}}) \leftarrow \mathsf{NIFS.K}(\mathsf{pp}_{\mathsf{NIFS}}, F)$
2. Compute $(\mathsf{pk}_{\mathsf{IVC}}, \mathsf{vk}_{\mathsf{IVC}}) \leftarrow \mathsf{IVC.K}(\mathsf{pp}_{\mathsf{IVC}}, \mathsf{step})$
3. Output $(\mathsf{pk}, \mathsf{vk}) \leftarrow ((F, \mathsf{pk}_{\mathsf{NIFS}}, \mathsf{vk}_{\mathsf{NIFS}}, \mathsf{pk}_{\mathsf{IVC}}), (\mathsf{step}, \mathsf{pp}_{\mathsf{NIFS}}, \mathsf{vk}_{\mathsf{NIFS}}, \mathsf{vk}_{\mathsf{IVC}}))$.



$\mathsf{NIFS.P}\big(\mathsf{pk}, (U_i, W_i), (u_i, w_i)\big)$ → $U_{i+1}, W_{i+1}, \pi_{i+1}$

$\mathsf{IVC.P}\big(\mathsf{pk}, (\mathsf{vk}, u_\perp, z_0), (\mathsf{vk}, U_i, z_i), (u_i, \pi_{i+1}), \Pi'_i\big)$ → Π'_{i+1}

$\mathcal{P}(\mathsf{pk}, (i, z_0, z_i), \omega_i, \Pi_i) \to \Pi_{i+1}$:

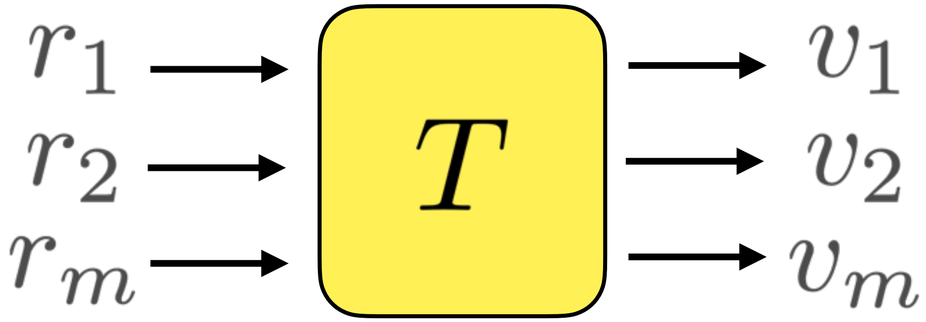
1. Parse Π_i as $(\Pi'_i, \mathsf{U}_i, \mathsf{W}_i)$
2. Compute $(\mathsf{u}_i, \mathsf{w}_i) \leftarrow \mathsf{trace}(F, (z_i, \omega_i))$ (where trace is defined as in Construction 3)
3. Compute $(\mathsf{U}_{i+1}, \mathsf{W}_{i+1}, \pi_{i+1}) \leftarrow \mathsf{NIFS.P}(\mathsf{pk}_{\mathsf{NIFS}}, (\mathsf{U}_i, \mathsf{W}_i), (\mathsf{u}_i, \mathsf{w}_i))$
4. Compute $\Pi'_{i+1} \leftarrow \mathsf{IVC.P}(\mathsf{pk}_{\mathsf{IVC}}, i, (\mathsf{vk}_{\mathsf{NIFS}}, \mathsf{u}_\perp, z_0), (\mathsf{vk}_{\mathsf{NIFS}}, \mathsf{U}_i, z_i), (\mathsf{u}_i, \pi_{i+1}), \Pi'_i)$
5. Output $\Pi_{i+1} = (\Pi'_{i+1}, \mathsf{U}_{i+1}, \mathsf{W}_{i+1})$

$\mathcal{V}(\mathsf{vk}, (i, z_0, z_i), \Pi_i) \to \{0, 1\}$:

1. Parse Π_i as $(\Pi'_i, \mathsf{U}_i, \mathsf{W}_i)$.
2. Check that $\mathsf{IVC.V}(\mathsf{vk}_{\mathsf{IVC}}, i, (\mathsf{vk}_{\mathsf{NIFS}}, \mathsf{u}_\perp, z_0), (\mathsf{vk}_{\mathsf{NIFS}}, \mathsf{U}_i, z_i), \Pi'_i) = 1$
3. Check that $(\mathsf{pp}_{\mathsf{NIFS}}, \mathsf{step}, \mathsf{U}_i, \mathsf{W}_i) \in \mathcal{R}_{\mathsf{LCCCS}}$

nLookup

m lookup in a table of size n



$$m \ll n = 2^l$$

$$T : \{0,1\}^l \to \mathbb{F} \Longrightarrow \widetilde{T}$$

multilinear extended polynomial

Definition 15 (Polynomial Evaluation Relation). *We define the polynomial evaluation relation $\mathcal{R}_{\mathsf{poly}}$ as follows. Let the public parameters consist of size parameter $\ell \in \mathbb{N}$. An $\mathcal{R}_{\mathsf{poly}}$ structure consists of $\widetilde{T}$, a multilinear polynomial in ℓ variables. An $\mathcal{R}_{\mathsf{poly}}$ instance is $(r, v) \in (\mathbb{F}^\ell, \mathbb{F})$ where r is an evaluation point and v is a claimed evaluation. An $\mathcal{R}_{\mathsf{poly}}$ witness is $\perp$. We define $\mathcal{R}_{\mathsf{poly}}$ as follows.*

$$\mathcal{R}_{\mathsf{poly}} = \left\{ ((\ell, \widetilde{T}), (r, v), \perp) \;\middle|\; \begin{array}{l} \ell \in \mathbb{N}, \widetilde{T} \in \mathbb{F}^1[X_1, \ldots, X_\ell], (r, v) \in (\mathbb{F}^\ell, \mathbb{F}) \\ \widetilde{T}(r) = v \end{array} \right\}.$$

Definition 16 (Lookup Relation). *We define the lookup relation $\mathcal{R}_{\mathsf{lookup}}$ as follows. Let the public parameters consist of size parameter $\ell \in \mathbb{N}$. For vector $T \in \mathbb{F}^n$ (where $n = 2^\ell$), an $\mathcal{R}_{\mathsf{lookup}}$ structure consists of the corresponding multilinear extension in ℓ variables, $\widetilde{T}$. An $\mathcal{R}_{\mathsf{lookup}}$ instance consists of value $v \in \mathbb{F}$. An $\mathcal{R}_{\mathsf{lookup}}$ witness consists of index $q \in \{0,1\}^\ell$. We define $\mathcal{R}_{\mathsf{lookup}}$ as follows.*

$$\mathcal{R}_{\mathsf{lookup}} = \left\{ ((\ell, \widetilde{T}), v, q) \;\middle|\; \begin{array}{l} \ell \in \mathbb{N}, \widetilde{T} \in \mathbb{F}^1[X_1, \ldots, X_\ell], v \in \mathbb{F}, q \in \{0,1\}^\ell \\ \widetilde{T}(q) = v \end{array} \right\}.$$

Construction 5 (A multi-folding scheme for lookup instances). We construct a multi-folding scheme for $(\mathcal{R}_{\mathsf{poly}}, \mathcal{R}_{\mathsf{lookup}}, \mathsf{compat}, \mu = 1, \nu)$ for arbitrary $\nu \in \mathbb{N}$.

m lookup in a table of size n

Sum-Check

$\mathcal{O}(m \log(n))$ Multiplications

$\mathcal{O}(\log(n))$ Hash inside a circuit

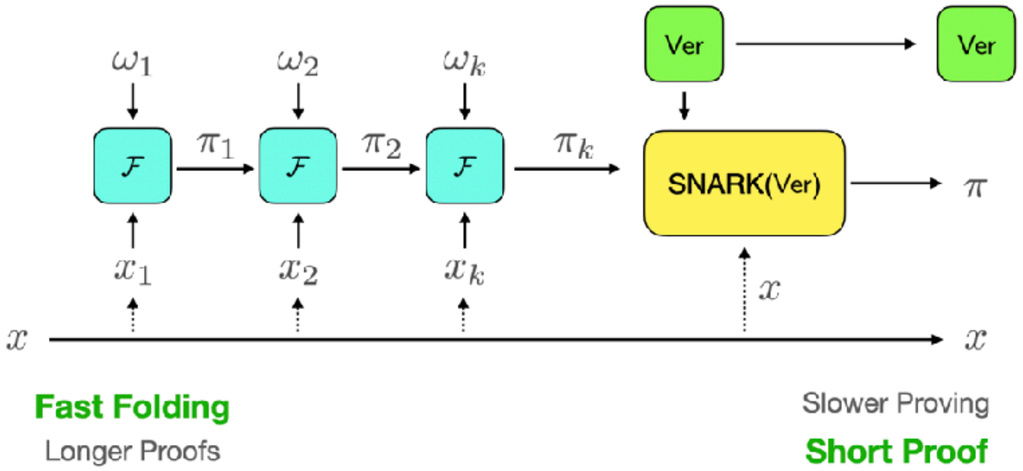
$\mathcal{O}(n)$ FF operations (prover)

Conclusions

HyperNova: Recursive arguments for customizable constraint systems

Abhiram Kothapalli[†] Srinath Setty[⋆]

[†]Carnegie Mellon University [⋆]Microsoft Research



Lemma 8 (Efficiency). *When instantiated with the Pedersen commitment scheme, we have that* $|F'| = |F| + o(\mathsf{G} + 2 \cdot \mathsf{H} + d \cdot \log m \cdot \mathsf{F} + \log m \cdot \mathsf{R})$*, where* $|F|$ *denotes the number of CCS constraints to encode a function* F*,* G *is the number of constraints required to encode a group scalar multiplication,* H *is the number of constraints required to encode* hash*,* F *is the number of constraints to encode field operations, and* R *is the number of constraints to encode the RO* ρ*.*