Close Enough:
From Linear Codes
To Proximity Testing

Alessandro Chiesa

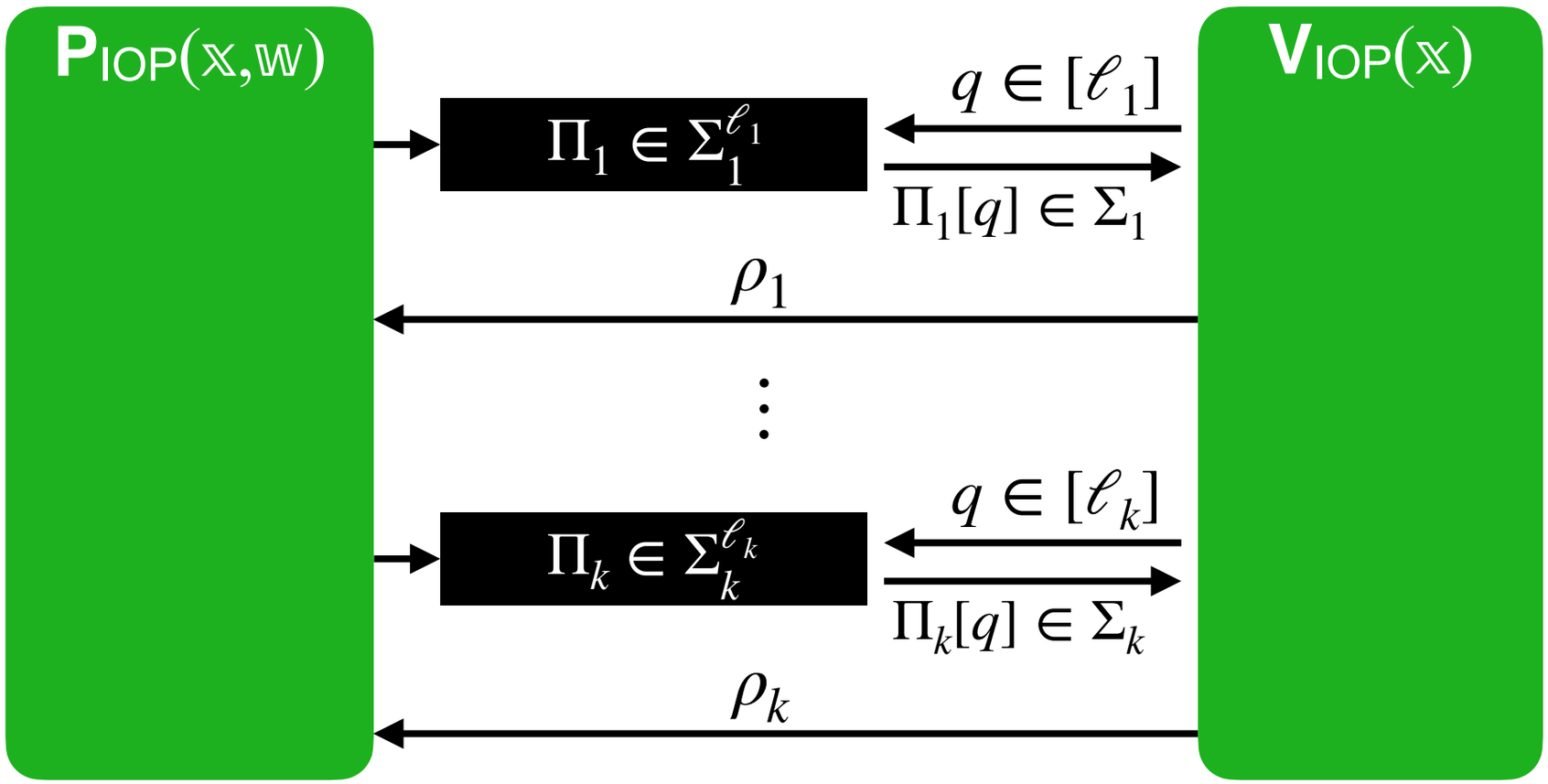
EPFL
StarkWare

2026.05.09

Interactive Oracle Proofs

An **IOP** is an IP where the verifier has query access to prover messages.

We focus on *public-coin* IOPs.

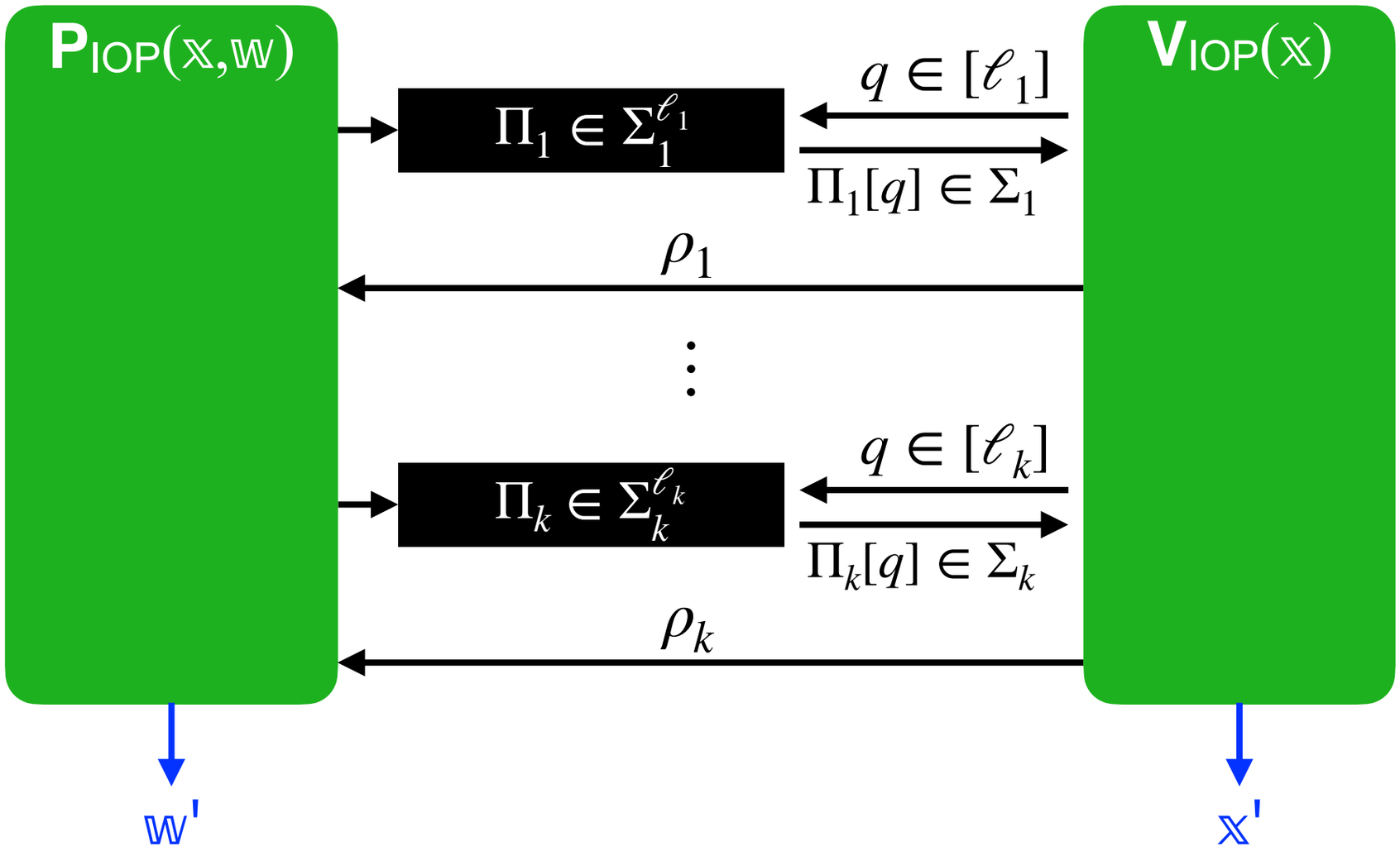


IOPs are essentially equivalent to **hash-based succinct arguments**:

- IOP + collision-resistant hashing ⟶ interactive succinct arguments
- IOP ⟵ succinct arguments based on ideal hashing (i.e. in ROM)

Tool: Interactive Oracle **Reductions**

An **IOR** relaxes the notion of an IOP.



Informally, an IOR from $\mathscr{R}$ to $\mathscr{R}'$ requires:

- $(\mathbb{x},\mathbb{w}) \in \mathscr{R}$ implies $(\mathbb{x}',\mathbb{w}') \in \mathscr{R}'$ (with probability 1)
  - $\mathbb{x} \notin \mathscr{L}(\mathscr{R})$ implies $\mathbb{x}' \notin \mathscr{L}(\mathscr{R}')$ (except for a small error)

Where do IOPs come from?

Established answer:

1. **arithmetize** a (well-chosen) NP problem
2. locally test resulting **polynomial code**

Refined answer:

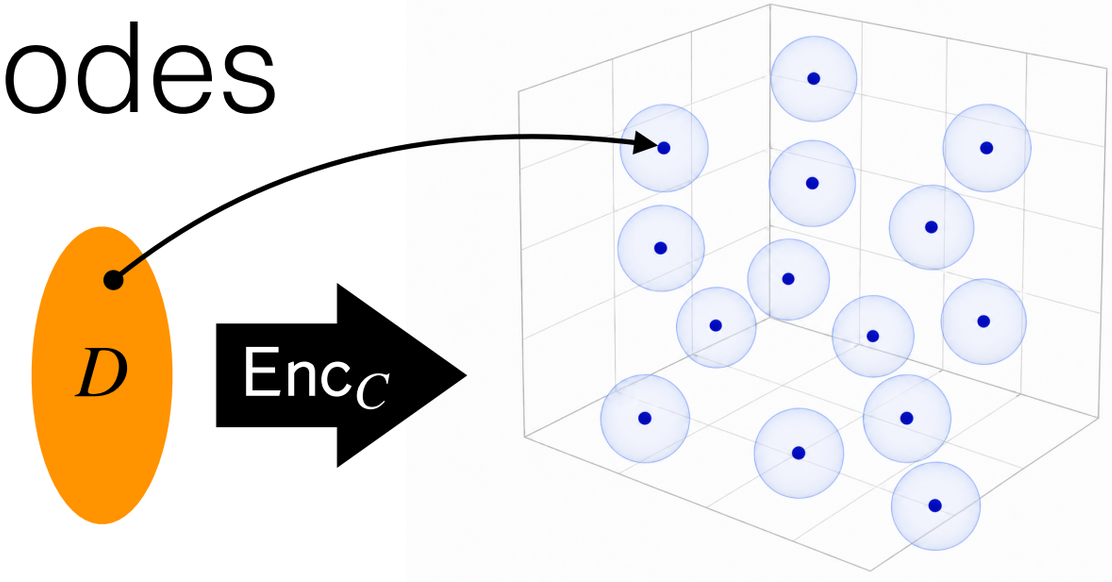
1. pick a **linear code** C and a "structurnig operation" op
2. IOR from an NP problem to $C^* := \mathrm{op}(C)$ with linear constraints
3. IOPP for C^* with linear constraints

Linear Codes

A **code** is a subset $C \subseteq \Sigma^\ell$.

The minimum (relative) distance is

$$\delta(C) := \min_{c_1, c_2 \in C, c_1 \neq c_2} \Delta(c_1, c_2).$$



An **encoding** for C is an injective map $\mathsf{Enc}_C \colon D \to \Sigma^\ell$ s.t. $\mathsf{Enc}_C(D) = C$.

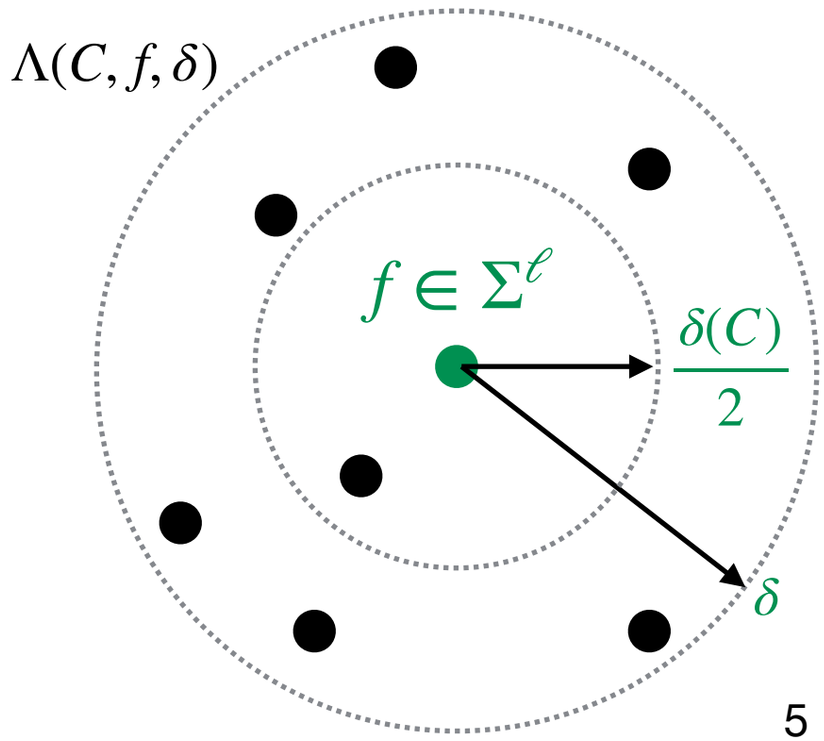
A code $C \subseteq \Sigma^\ell$ is **$\mathbb{F}$-linear** if Σ is an $\mathbb{F}$-linear space (i.e. $\Sigma \cong \mathbb{F}^d$ for some d).

In this case $\delta(C) = \min_{c \in C \setminus \{\mathbf{0}\}} \Delta(c, \mathbf{0})$.

The **list** around $f \in \Sigma^\ell$ at distance $\delta \in [0,1]$ is

$$\Lambda(C, f, \delta) := \{c \in C : \Delta(f, c) \leq c\}.$$

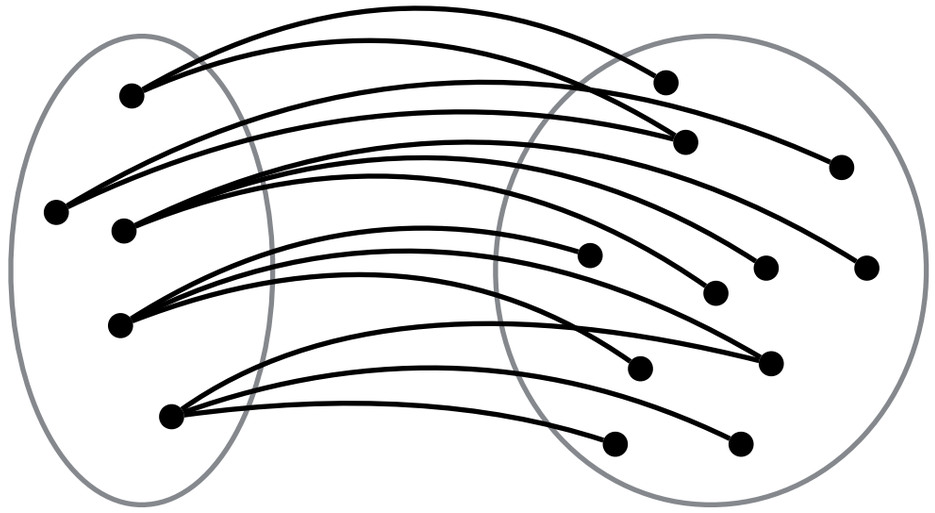
The list size is $|\Lambda(C, \delta)| := \max_{f \in \Sigma^\ell} |\Lambda(C, f, \delta)|$.



Relations

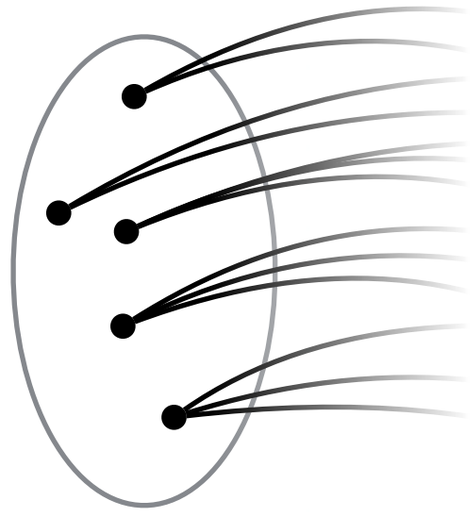
A **relation** is a set of instance-witness pairs

$$\mathscr{R} = \{(\mathbb{x}, \mathbb{w}) : \ldots\}$$



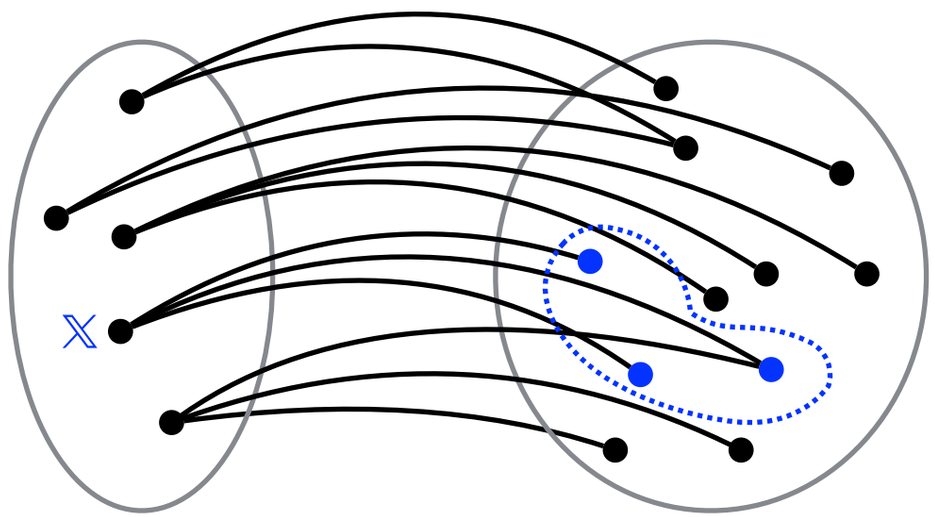
The corresponding **language** is the set of instances

$$\mathscr{L}(\mathscr{R}) := \{\mathbb{x} : \exists\, \mathbb{w} \text{ s.t. } (\mathbb{x}, \mathbb{w}) \in \mathscr{R}\}$$



The **valid witnesses** for an instance $\mathbb{x}$ (if any) are

$$\mathscr{R}[\mathbb{x}] := \{\mathbb{w} : (\mathbb{x}, \mathbb{w}) \in \mathscr{R}\}$$



Blueprint

ENCODE WITNESS

"non-linear" NP-complete relation

$\mathscr{R}_0$

LOCAL TESTING OF LINEAR CONSTRAINTS

"linear" relations

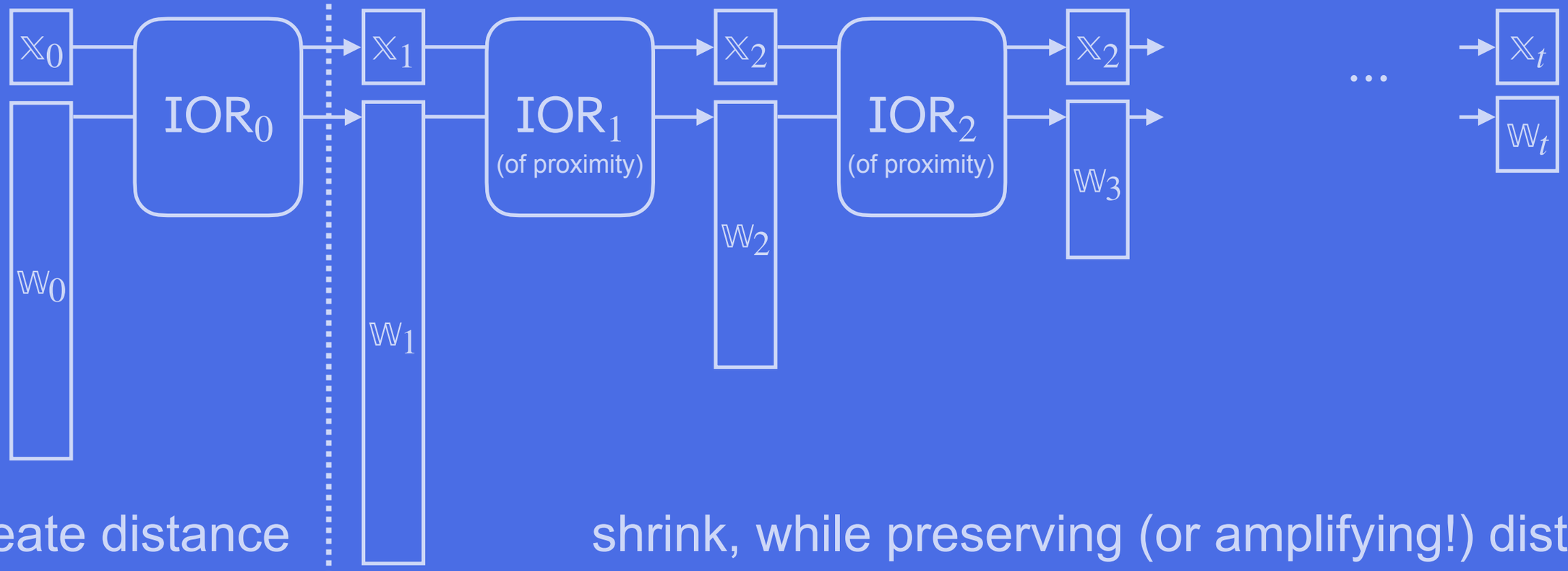
$\mathscr{R}_1$ based on a code C_1

$\mathscr{R}_2$ based on a code C_2

$\mathscr{R}_3$ based on a code C_3

$\cdots$

$\mathscr{R}_t$ based on a code C_t



create distance

shrink, while preserving (or amplifying!) distance

Part 1 Part 2

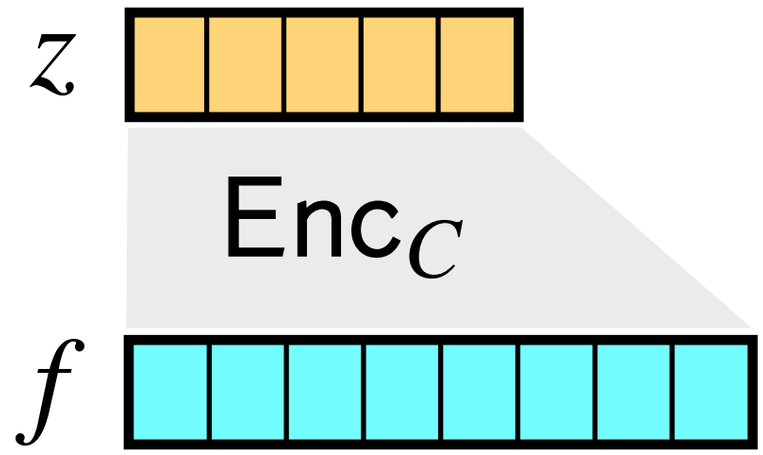
Part 1:
From NP
To Constrained Codes

Creating Distance

Target: Constrained Code Relation

- Code relation:

$$\mathscr{R}_C := \left\{ (\mathbb{x}, \mathbb{w}) : \begin{array}{l} \mathbb{x} = \perp, \mathbb{w} = f \in \Sigma^\ell \\ \exists\, z \in D \text{ s.t. } f = \mathsf{Enc}_C(z) \end{array} \right\}$$

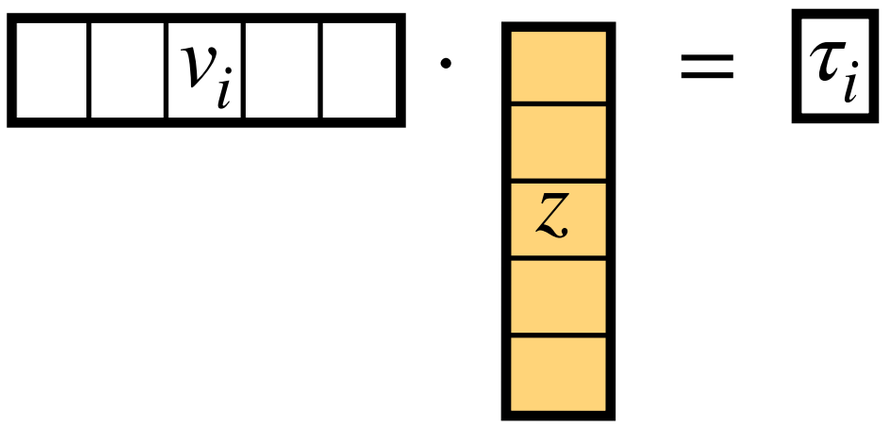


- Code relation with **$\mathbb{F}$-linear constraints**:

$$\mathscr{R}_C^{\text{lin}} := \left\{ (\mathbb{x}, \mathbb{w}) : \begin{array}{l} \mathbb{x} = (v_i, \tau_i)_i, \mathbb{w} = f \in \Sigma^\ell \\ \exists\, z \in \mathbb{F}^m \text{ s.t.} \\ f = \mathsf{Enc}_C(z) \\ \forall\, i, \langle v_i, z \rangle = \tau_i \end{array} \right\}$$

C is $\mathbb{F}$-linear
$\rightarrow \mathscr{L}(\mathscr{R}_C^{\text{lin}}) \in \mathbf{P}$

message space is $D = \mathbb{F}^m$
(Enc_C: $\mathbb{F}^m \rightarrow \Sigma^\ell$)



Source: Nice NP Relation

Rank-1 Constraint Satisfaction

$\mathscr{R}_{\mathrm{R1CS}}$ consists of pairs $(\mathbb{x},\mathbb{w})$ such that

- $\mathbb{x} = (A, B, C, x)$ with $A, B, C \in \mathbb{F}^{M \times N}$ and $x \in \mathbb{F}^{n_x}$
- $\mathbb{w} = w \in \mathbb{F}^{N-n_x}$
- satisfiability condition: $\left(A \cdot \begin{bmatrix} x \\ w \end{bmatrix}\right) \circ \left(B \cdot \begin{bmatrix} x \\ w \end{bmatrix}\right) = C \cdot \begin{bmatrix} x \\ w \end{bmatrix}$

"∘" means entry-wise product

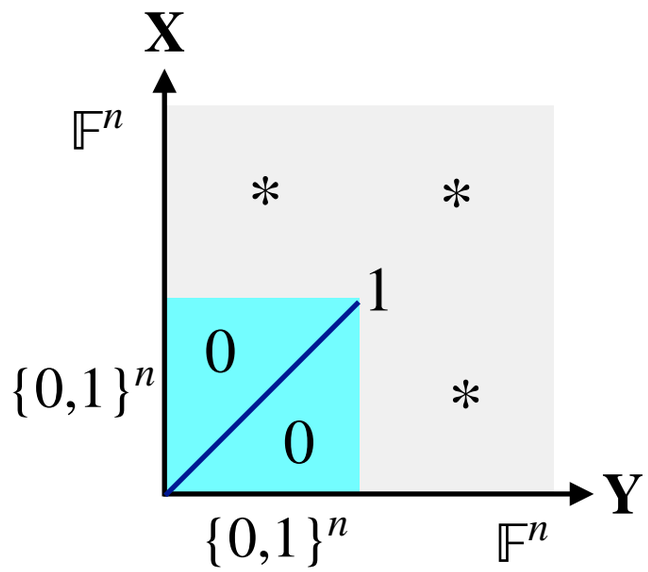
$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,N} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ a_{M,1} & a_{M,2} & \cdots & a_{M,N} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n_x} \\ w_1 \\ \vdots \\ w_{N-n_x} \end{bmatrix} \circ \begin{bmatrix} b_{1,1} & b_{1,2} & \cdots & b_{1,N} \\ b_{2,1} & b_{2,2} & \cdots & b_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ b_{M,1} & b_{M,2} & \cdots & b_{M,N} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n_x} \\ w_1 \\ \vdots \\ w_{N-n_x} \end{bmatrix} = \begin{bmatrix} c_{1,1} & c_{1,2} & \cdots & c_{1,N} \\ c_{2,1} & c_{2,2} & \cdots & c_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ c_{M,1} & c_{M,2} & \cdots & c_{M,N} \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_{n_x} \\ w_1 \\ \vdots \\ w_{N-n_x} \end{bmatrix}$$

$$\begin{bmatrix} z_{A,1} \\ \vdots \\ z_{A,M} \end{bmatrix} \circ \begin{bmatrix} z_{B,1} \\ \vdots \\ z_{B,M} \end{bmatrix} = \begin{bmatrix} z_{C,1} \\ \vdots \\ z_{C,M} \end{bmatrix} \quad \Rightarrow \quad \begin{bmatrix} z_{A,1} \cdot z_{B,1} \\ \vdots \\ z_{A,M} \cdot z_{B,M} \end{bmatrix} = \begin{bmatrix} z_{C,1} \\ \vdots \\ z_{C,M} \end{bmatrix}$$

Multilinear Extensions

The ***n*-variate equality polynomial** is

$$\mathsf{eq}_n(X_1, \ldots, X_n, Y_1, \ldots, Y_n) := \prod_{i \in [n]} (X_i Y_i + (1 - X_i)(1 - Y_i)) \quad .$$

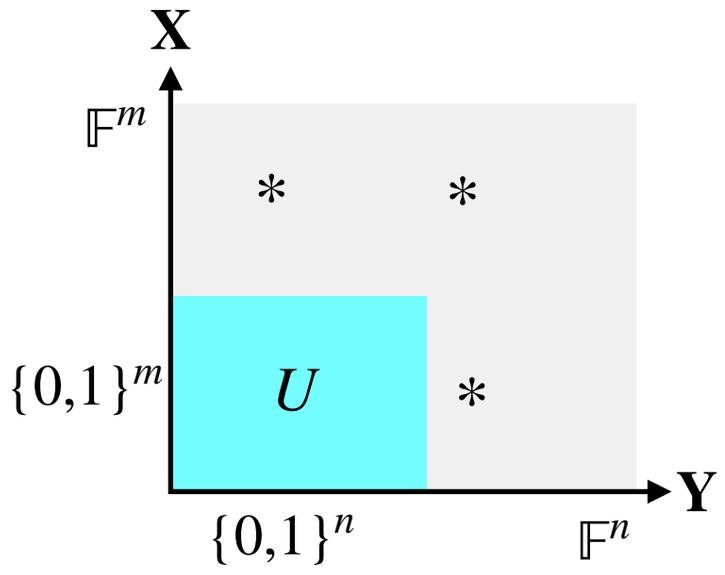


Assume number of rows $M = 2^m$ and number of columns $N = 2^n$.

The **multilinear extension** of a matrix $U \in \mathbb{F}^{M \times N}$ is

$$\hat{U}(\mathbf{X}, \mathbf{Y}) := \sum_{i \in \{0,1\}^m} \sum_{j \in \{0,1\}^n} U[i, j] \cdot \mathsf{eq}_m(\mathbf{X}, i) \cdot \mathsf{eq}_n(\mathbf{Y}, j) \quad .$$

Note that $\hat{U}(\{0,1\}^m, \{0,1\}^n) = U$.



Satisfiability As Multilinear Vanishing

Fix a variable assignment $z \in \mathbb{F}^N = \mathbb{F}^{2^n}$.

For a matrix $U \in \mathbb{F}^{M \times N} = \mathbb{F}^{2^m \times 2^n}$, define the m-variate polynomial

$$Q_{U,z}(\mathbf{X}) := \sum_{j \in \{0,1\}^n} \hat{U}(\mathbf{X}, j) \cdot z[j] = \langle \hat{U}(\mathbf{X}, \{0,1\}^n), z \rangle \quad .$$

Note that $Q_{U,z}(\mathbf{X}) \equiv \widehat{U \cdot z}(\mathbf{X})$.

Claim: $(A \cdot z) \circ (B \cdot z) = C \cdot z$ if and only

$Q_{A,z}(\mathbf{X})Q_{B,z}(\mathbf{X}) - Q_{C,z}(\mathbf{X})$ vanishes (everywhere) on $\{0,1\}^m$

Claim:

- $Q(\mathbf{X})$ vanishes on $\{0,1\}^m \rightarrow \forall \nu \in \mathbb{F}^m \ \sum_{b \in \{0,1\}^m} \mathsf{eq}_m(\nu, b) \cdot Q(b) = 0$
- $Q(\mathbf{X})$ does not vanish on $\{0,1\}^m \rightarrow \Pr_{\nu} \left[\sum_{b \in \{0,1\}^m} \mathsf{eq}_m(\nu, b) \cdot Q(b) = 0 \right] \leq \frac{m}{|\mathbb{F}|}$

Encode then Sumcheck

P$(\mathbb{x},\mathbb{w})$

$z := (x, w) \in \mathbb{F}^N$

$f := \mathsf{Enc}_C(z) \in \mathbb{F}^\ell$

V$(\mathbb{x})$

$\nu \leftarrow \mathbb{F}^m$

sumcheck reduction for the claim

$$\sum_{b\in\{0,1\}^m} \mathsf{eq}_m(\nu, b) \cdot (Q_{A,z}(b)Q_{B,z}(b) - Q_{C,z}(b)) = 0$$

$\rho \leftarrow \mathbb{F}^m$

$$\text{"}\mathsf{eq}_m(\nu, \rho) \cdot (Q_{A,z}(\rho)Q_{B,z}(\rho) - Q_{C,z}(\rho)) = h_m(\rho_m)\text{"}$$

$\alpha := Q_{A,z}(\rho), \beta := Q_{B,z}(\rho), \gamma := Q_{C,z}(\rho)$

$\sigma \leftarrow \mathbb{F}^n$

check that

$\mathsf{eq}_m(\nu, \rho) \cdot (\alpha\beta - \gamma) = h_m(\rho_m)$

$v_A := \hat{A}(\rho, \{0,1\}^n)$

$v_B := \hat{B}(\rho, \{0,1\}^n)$

$v_C := \hat{C}(\rho, \{0,1\}^n)$

for $0 \le j < n_x$:
$v_x[j] := \mathsf{eq}_n(\sigma, j)$

for $n_x \le j < N$:
$v_x[j] := 0$

$\tau_x := \langle v_x, x0^{N-n} \rangle$

New instance: $\mathbb{x}' := ((v_A, \alpha), (v_B, \beta), (v_C, \gamma), (v_x, \tau_x))$

New witness: $\mathbb{w}' := f$

I.e., this claim: $\exists\, z \in \mathbb{F}^N$ s.t. $f := \mathsf{Enc}_C(z)$ and

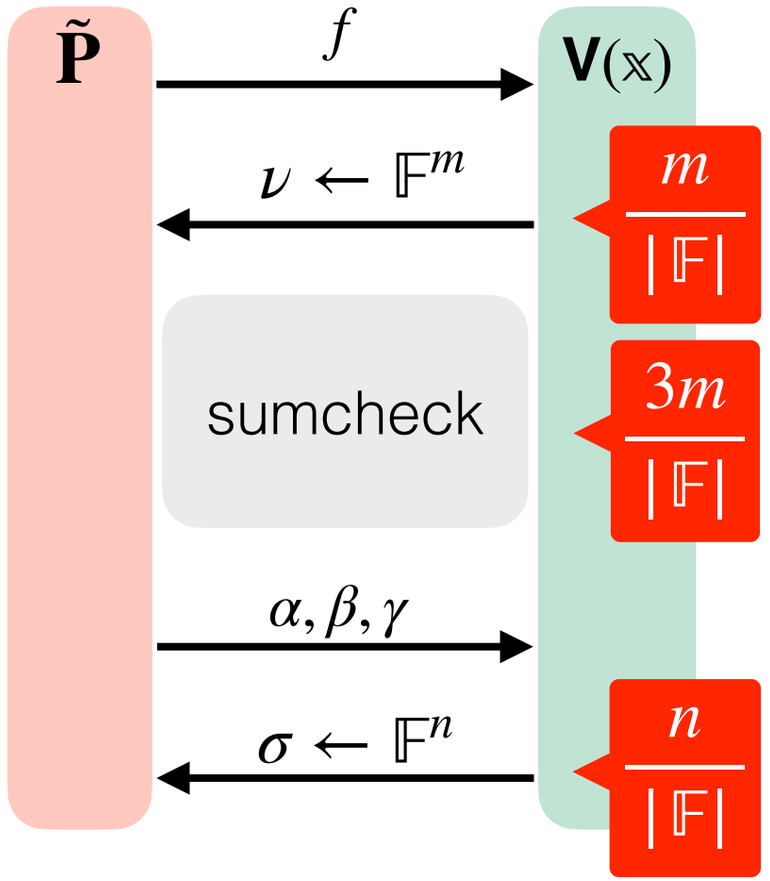
$$\langle v_A, z\rangle = \alpha$$
$$\langle v_B, z\rangle = \beta \quad \langle v_x, z\rangle = \tau_x$$
$$\langle v_C, z\rangle = \gamma$$

Encode then Sumcheck

Lemma: if $\mathbb{x} = (A, B, C, x) \notin \mathscr{L}(\mathscr{R}_{\text{R1CS}})$

then, $\forall\, \tilde{\mathbf{P}}$, $((v_i, \tau_i)_i, f) \notin \mathscr{R}_C^{\text{lin}}$

except with probability $\leq \dfrac{O(m+n)}{|\mathbb{F}|}$



Not enough for local testing of output claim.

BUT can create distance at a **small price** in error:

Lemma: if $\mathbb{x} = (A, B, C, x) \notin \mathscr{L}(\mathscr{R}_{\text{R1CS}})$

then, $\forall\, \delta \in [0,1]$ $\forall\, \tilde{\mathbf{P}}$, f is δ**-far** from $\mathscr{R}_C^{\text{lin}}[(v_i, \tau_i)_i]$

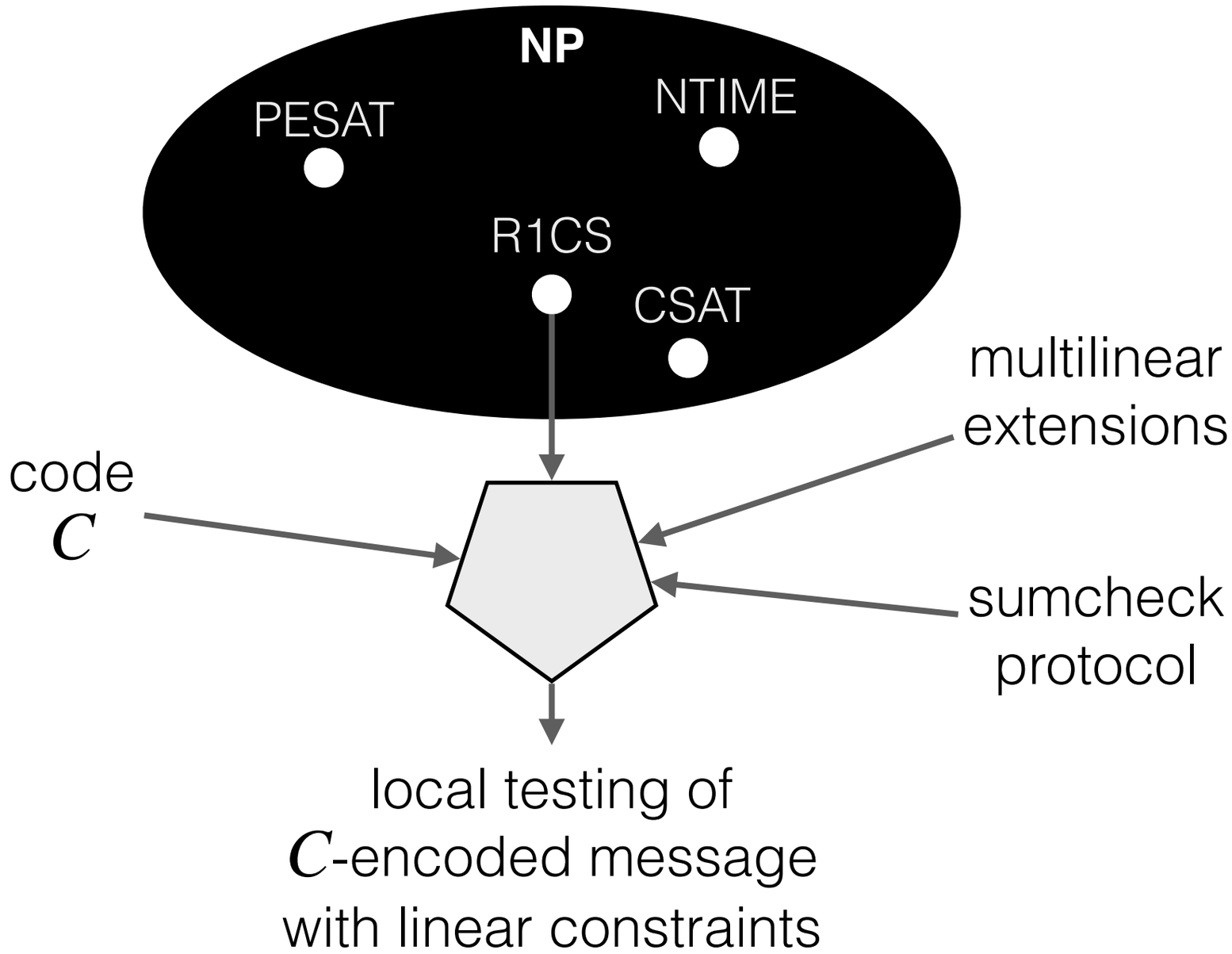
except with probability $\leq |\Lambda(C, \delta)| \cdot \dfrac{O(m+n)}{|\mathbb{F}|}$

For local testing we want $\delta = \Omega(1)$ or even $\delta = 1 - o(1)$.
A good code C ensures that $|\Lambda(C, \delta)|$ is small.

Useful variant:
- t OOD samples on f
- t additional constraints

Achieves error
$\binom{|\Lambda(C,\delta)|}{2} \left(\frac{m}{|\mathbb{F}|}\right)^t + \frac{O(m+n)}{|\mathbb{F}|}$

Takeaway



Can modify (interactive oracle) reduction to:

- achieve **zero knowledge**
- achieve **holography** (preprocess A, B, C matrices or equivalent)
- start from other **nice NP relations**

Blueprint

ENCODE WITNESS

"non-linear" NP-complete relation

$\mathcal{R}_0$

LOCAL TESTING OF LINEAR CONSTRAINTS

"linear" relations

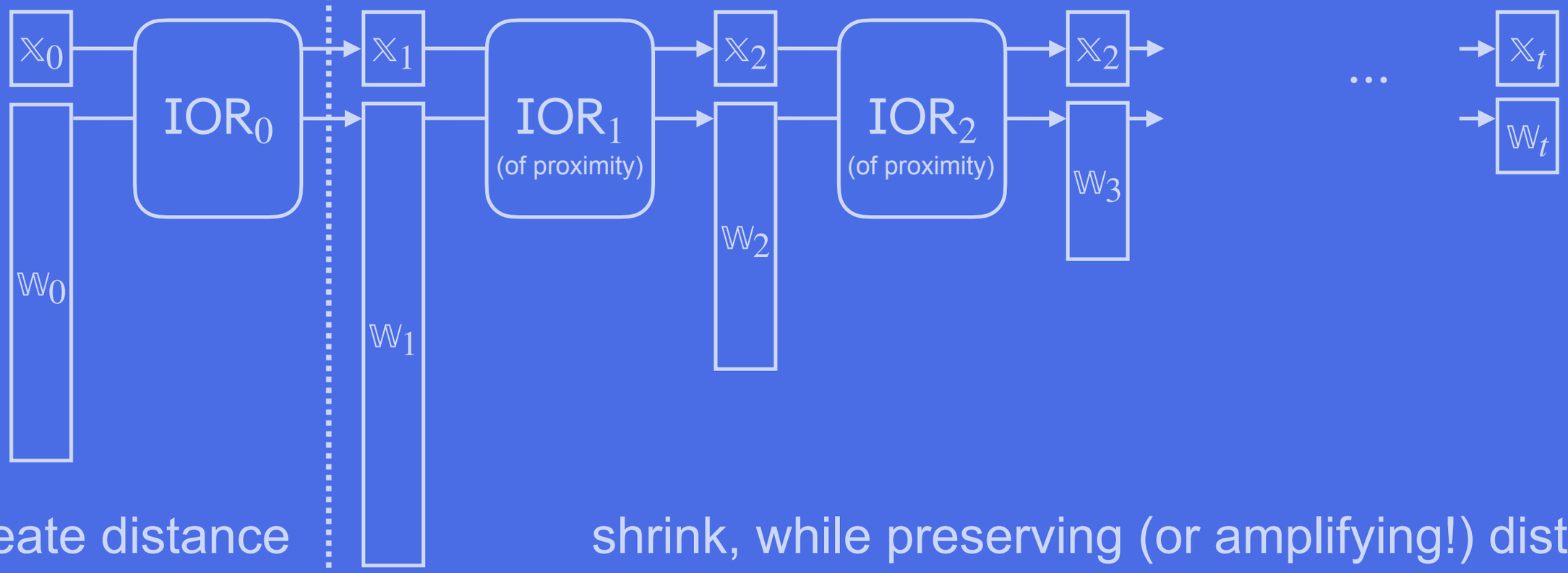
$\mathcal{R}_1$ based on a code C_1

$\mathcal{R}_2$ based on a code C_2

$\mathcal{R}_3$ based on a code C_3

$\cdots$

$\mathcal{R}_t$ based on a code C_t



create distance

shrink, while preserving (or amplifying!) distance

Part 1 Part 2

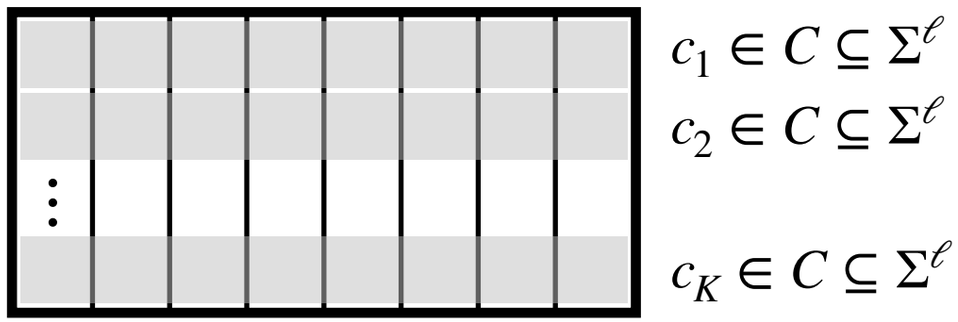
Part 2:
Local Testing of Linear Constraints

Warm Up:
Local Testing
Without Constraints

Interleaved Codes

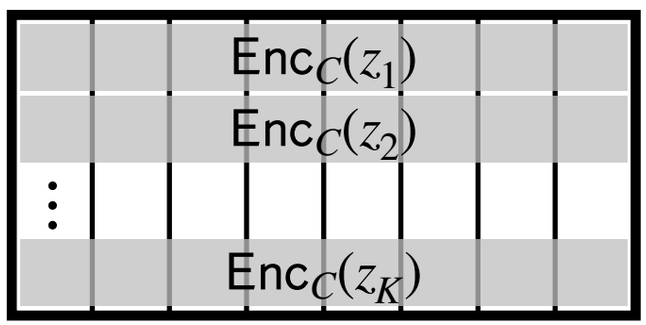
The **K-interleaving** of $C \subseteq \Sigma^\ell$ is the code $C^{\equiv K} \subseteq (\Sigma^K)^\ell$ containing all $c \in (\Sigma^K)^\ell$ s.t. $\exists\, c_1, \ldots, c_K \in C \;\; \forall\, i \in [\ell]\; c[i] = (c_1[i], \ldots, c_K[i]) \in \Sigma^K$.

$c \in C^{\equiv K} \subseteq (\Sigma^K)^\ell$

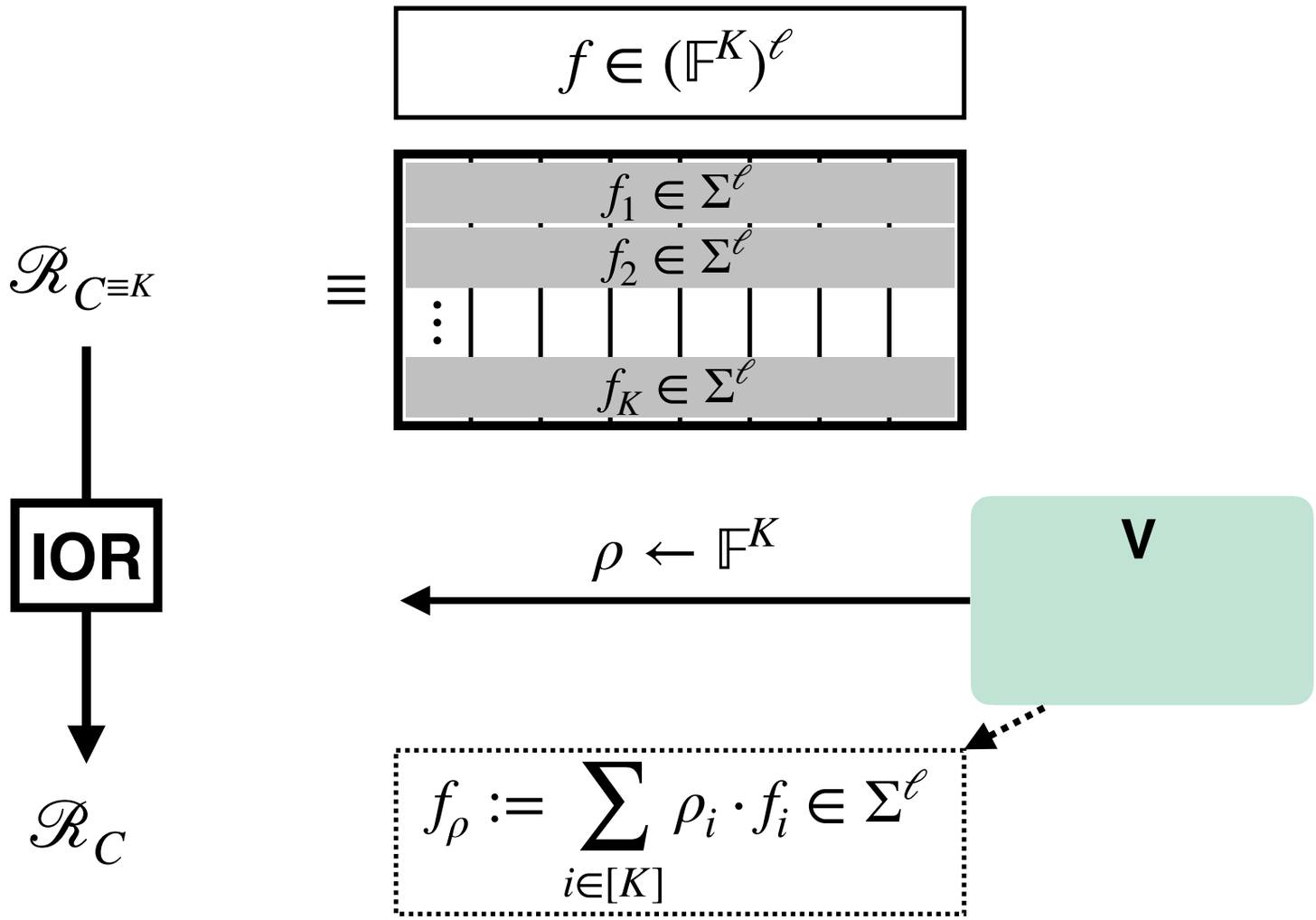


If $C \subseteq \Sigma^\ell$ is $\mathbb{F}$-linear then $C^{\equiv K} \subseteq (\Sigma^K)^\ell$ is $\mathbb{F}$-linear.

If $\mathsf{Enc}_C\colon \mathbb{F}^m \to \Sigma^\ell$ is an encoding for C then an encoding for $C^{\equiv K}$ is $\mathsf{Enc}_{C^{\equiv K}}\colon \mathbb{F}^{Km} \to \Sigma^\ell$ that splits $z \in \mathbb{F}^{Km}$ into pieces $z_1, \ldots, z_K \in \mathbb{F}^m$ and then outputs



From Interleaved Code to Base Code



Easy: if $f \notin C^{\equiv K}$ then $f_\rho \notin C$ except w.p. $\leq 1/|\mathbb{F}|$

Not enough for our purposes. We need **distance preservation**.

Goal: if $\Delta(f, C^{\equiv K}) \geq \delta$ then $\Delta(f_\rho, C) \geq \delta$ except w.p. $\leq \epsilon(C, \delta)$

Distance Preservation

Let $G\colon S \to \mathbb{F}^K$ be a coefficient "**generator**". (Ex: $G\colon \mathbb{F}^K \to \mathbb{F}^K$ is the identity.)

Def: distance preservation for $\mathbb{F}$-linear $C \subseteq \Sigma^\ell$ with error ϵ requires that

$$\forall f = \begin{bmatrix} f_1 \\ \vdots \\ f_K \end{bmatrix} \in (\Sigma^K)^\ell,\ \Delta(f, C^{\equiv K}) \geq \delta \to \Pr_{\sigma \leftarrow S}\left[\Delta\left(\sum_{i\in[K]} G(\sigma)[i]\cdot f_i, C\right) < \delta\right] \leq \epsilon(C,\delta).$$

Later we will need a **stronger notion** of distance perservation.

Def: mutual correlated agreement for $\mathbb{F}$-linear $C \subseteq \Sigma^\ell$ with error ϵ requires that

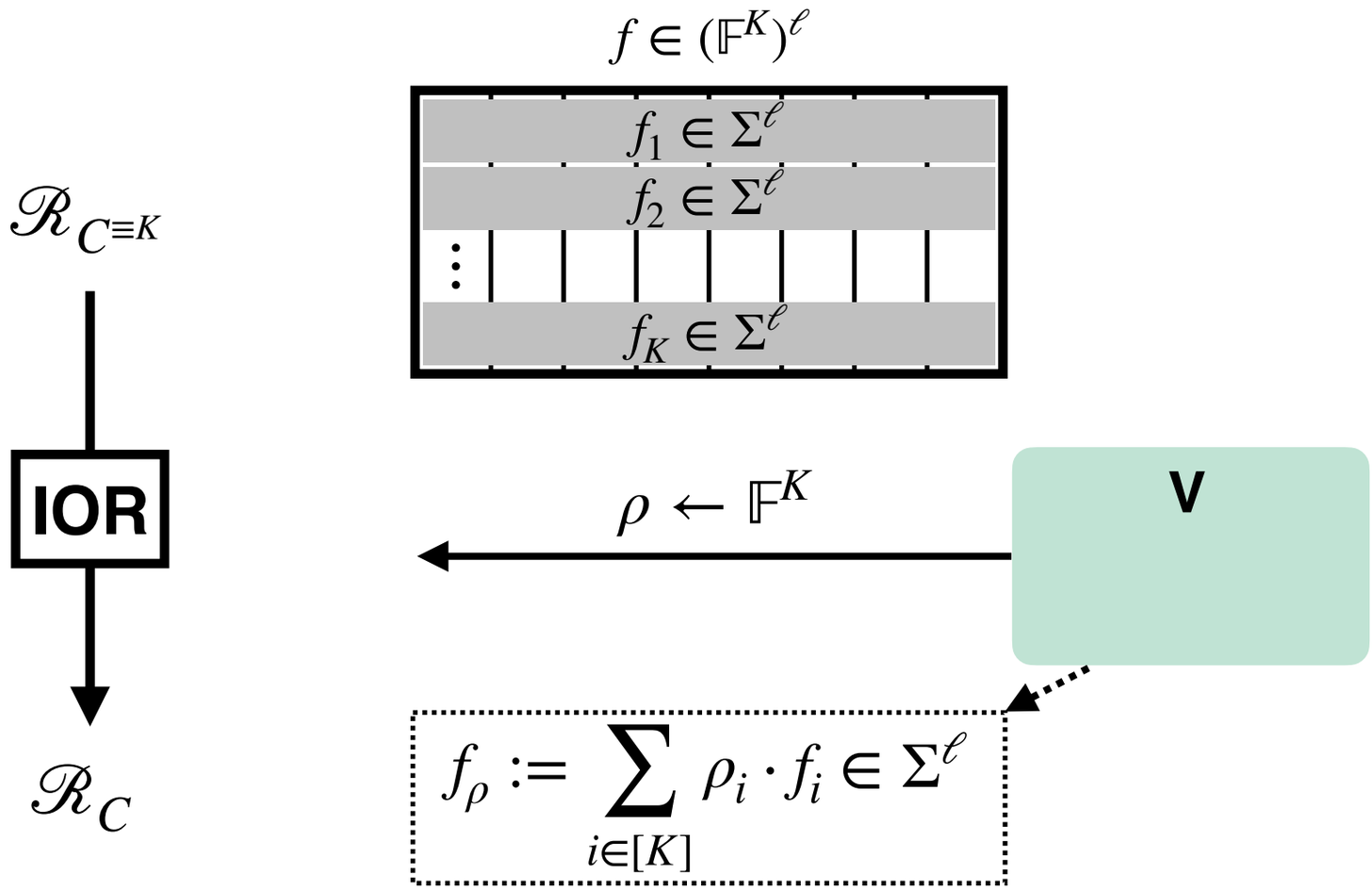
$$\forall f = \begin{bmatrix} f_1 \\ \vdots \\ f_K \end{bmatrix} \in (\Sigma^K)^\ell,\ \Pr_{\sigma \leftarrow S}\left[\begin{array}{l} |T| \geq (1-\delta)\ell \\ \sum_{i\in[K]} G(\sigma)[i]\cdot f_i|_T \in C|_T \\ \exists\, i \in [K] : f_i|_T \notin C|_T \end{array}\right] \leq \epsilon(C,\delta).$$

There are several results on these for various combinations of G and C.

These results are not a focus of this talk.

What Next?

We can reduce testing an interleaved code to testing its base code.



- Can we also **reduce linear constraints**?
- Even if so, how to **reduce further**?

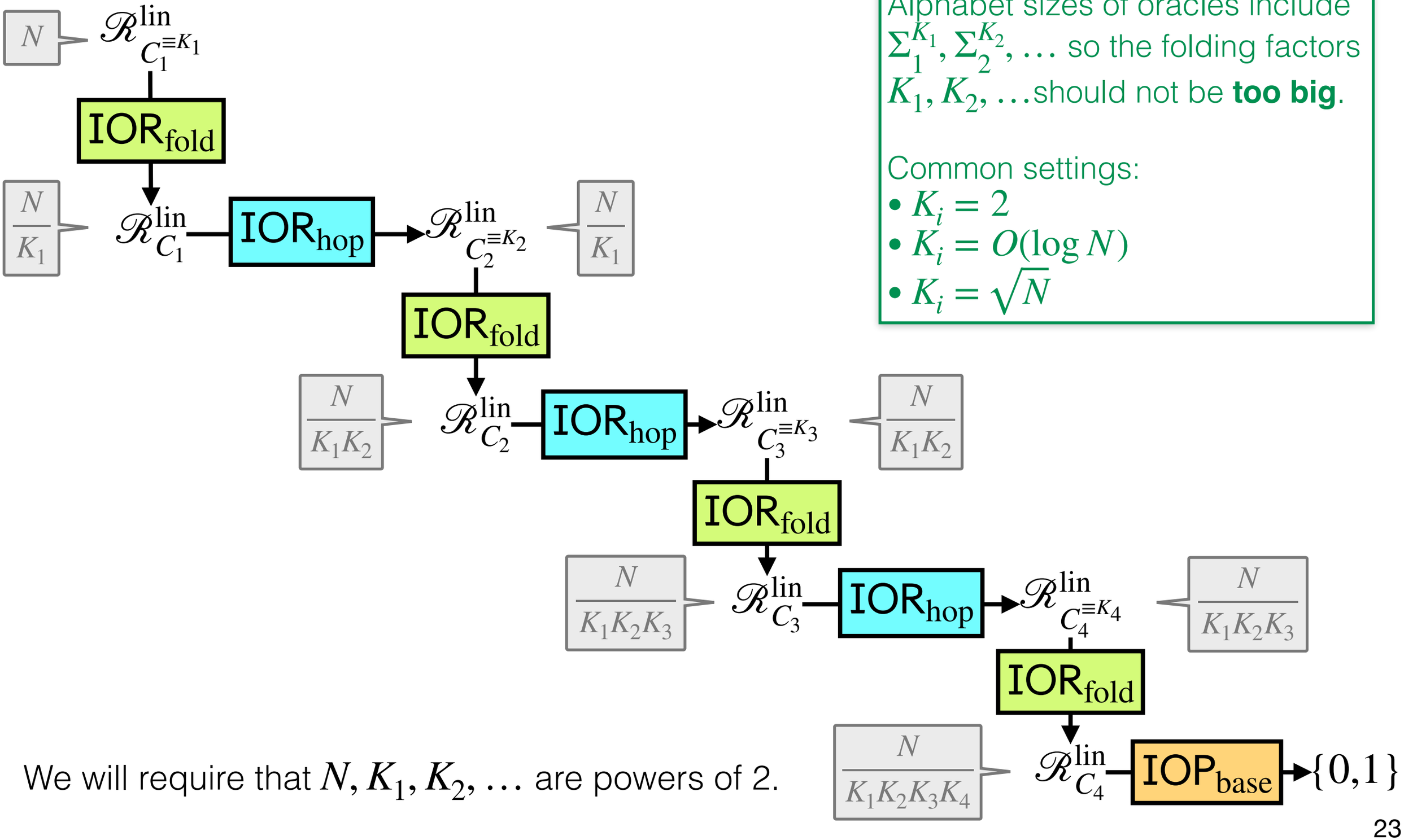
We see now how to fold code+constraints, and then "hop" to another code!

Part 2:
Local Testing of Linear Constraints

Back to Local Testing **With** Constraints

Fold and Hop

Strategy for reducing "problem size" (e.g., encoded message size).



Alphabet sizes of oracles include $\Sigma_1^{K_1}, \Sigma_2^{K_2}, \ldots$ so the folding factors $K_1, K_2, \ldots$should not be **too big**.

Common settings:

- $K_i = 2$
- $K_i = O(\log N)$
- $K_i = \sqrt{N}$

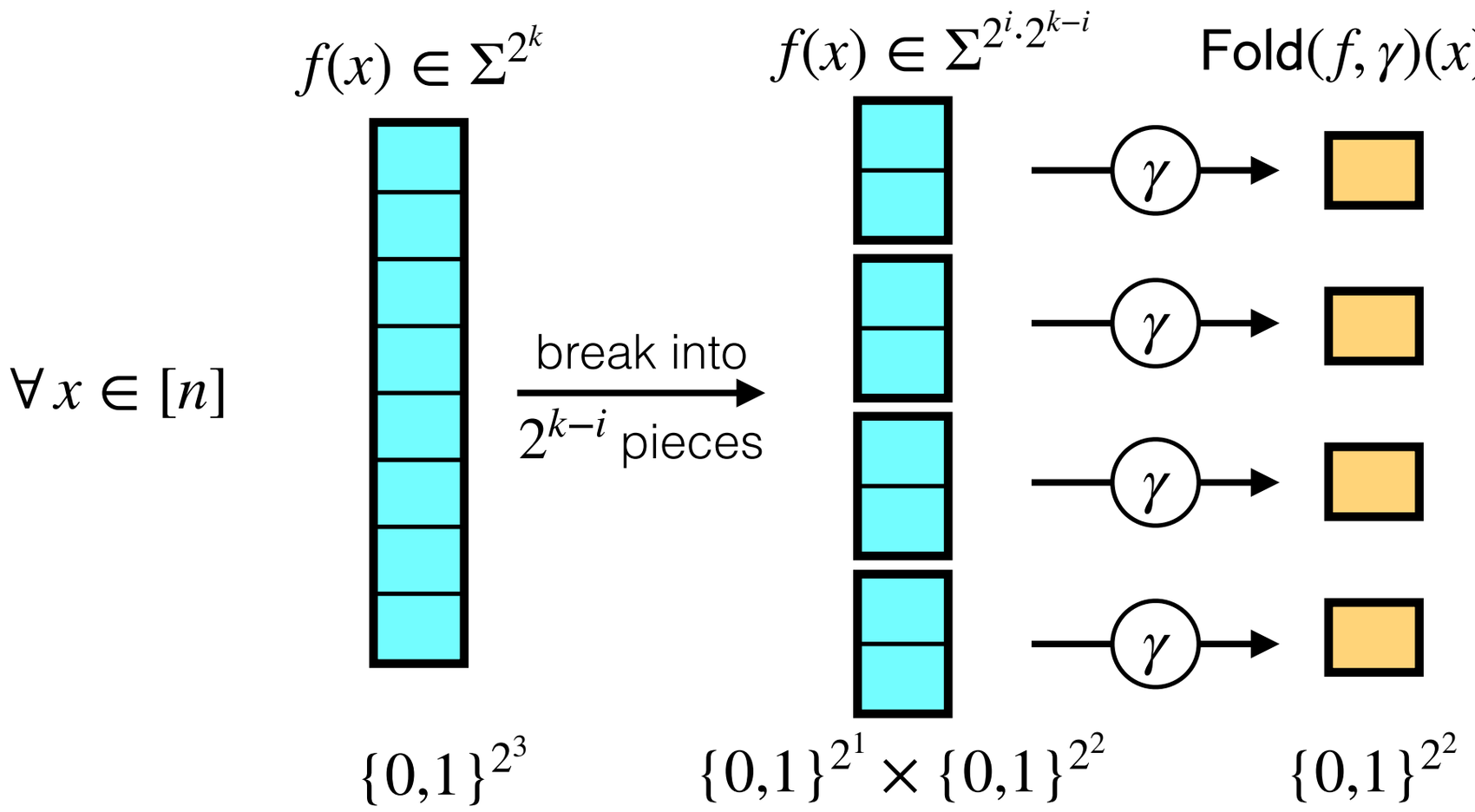
We will require that $N, K_1, K_2, \ldots$ are powers of 2.

Interleaved Fold

The **interleaved fold** of $f\colon [n] \to \Sigma^{2^k}$ at $\gamma \in \mathbb{F}^i$ is the function

$\mathsf{Fold}(f, \gamma)\colon [n] \to \Sigma^{2^{k-i}}$ such that

$$x \mapsto \left(\sum_{b \in \{0,1\}^i} \mathsf{eq}_i(b, \gamma) \cdot f(x)[b, c] \right)_{c \in \{0,1\}^{k-i}}$$



Folding with Encoding and Constraints

The operation **Fold** "**commutes**" with code encoding:

Claim: $\forall\, \gamma \in \mathbb{F}^i$, $\mathsf{Fold}(\mathsf{Enc}_{C\equiv 2^k}(z), \gamma) = \mathsf{Enc}_{C\equiv 2^{k-i}}(\mathsf{Fold}(z, \gamma))$

A linear constraint can be **rewritten as a sumcheck claim**.

Fix $\forall\, v, z \in \mathbb{F}^{2^k m}$.

Define the k-variate multiquadratic $Q_{v,z}(\mathbf{X}) := \langle \mathsf{Fold}(v, \mathbf{X}), \mathsf{Fold}(z, \mathbf{X}) \rangle$.

Claim: $\langle v, z \rangle = \sum_{b \in \{0,1\}^k} Q_{v,z}(b)$

Proof: $\forall\, b \in \{0,1\}^{2^i}$, $\mathsf{Fold}(v, b) = \sum_{b' \in \{0,1\}^i} \mathsf{eq}_i(b', b) \cdot v[b', c] = v[b, c]$

Hence $\sum_{b \in \{0,1\}^k} \langle \mathsf{Fold}(v, b), \mathsf{Fold}(z, b) \rangle = \sum_{b \in \{0,1\}^k} \sum_{c \in [m]} \mathsf{Fold}(v, b)[c] \cdot \mathsf{Fold}(z, b)[c]$

$$= \sum_{b \in \{0,1\}^k} \sum_{c \in [m]} v[b, c] \cdot z[b, c] = \langle v, z \rangle \,. \quad \blacksquare$$

Folding Code and One Constraint

$\mathscr{R}^{\text{lin}}_{C\equiv 2^k}$

IOR

$\mathscr{R}^{\text{lin}}_{C}$

$z \in \mathbb{F}^{2^k m}$ s.t.
$f = \text{Enc}_C(z)$

P

$f \in (\mathbb{F}^{2^k})^{\ell}$

sumcheck reduction for the claim
$\sum_{b\in\{0,1\}^k} Q_{v,z}(b) = \tau$

$"Q_{v,z}(\rho) = h_k(\rho_k)"$

$f' \in \mathbb{F}^{\ell}$

$v \in \mathbb{F}^{2^k m}$
$\tau \in \mathbb{F}$

V

$\rho \leftarrow \mathbb{F}^k$

$f' := \text{Fold}(f, \rho) \in \mathbb{F}^{\ell}$

$v' := \text{Fold}(v, \rho) \in \mathbb{F}^m$

$\tau' := h_k(\rho) \in \mathbb{F}$

$v' \in \mathbb{F}^m$
$\tau' \in \mathbb{F}$

Lemma: $\forall\, \delta \in [0,1]$ if f is δ-**far** from $\mathscr{R}^{\text{lin}}_{C\equiv 2^k}[(v,\tau)]$

then, $\forall\, \tilde{\mathbf{P}}$, f' is δ-**far** from $\mathscr{R}^{\text{lin}}_{C}[(v',\tau')]$

except with probability $\leq \sum_{j\in[k]} \left(\varepsilon_{\text{mca}}(C^{\equiv 2^{k-j}}, \delta) + |\Lambda(C^{\equiv 2^{k-j+1}}, \delta)| \cdot \frac{2}{|\mathbb{F}|} \right)$

Handling Multiple Constraints

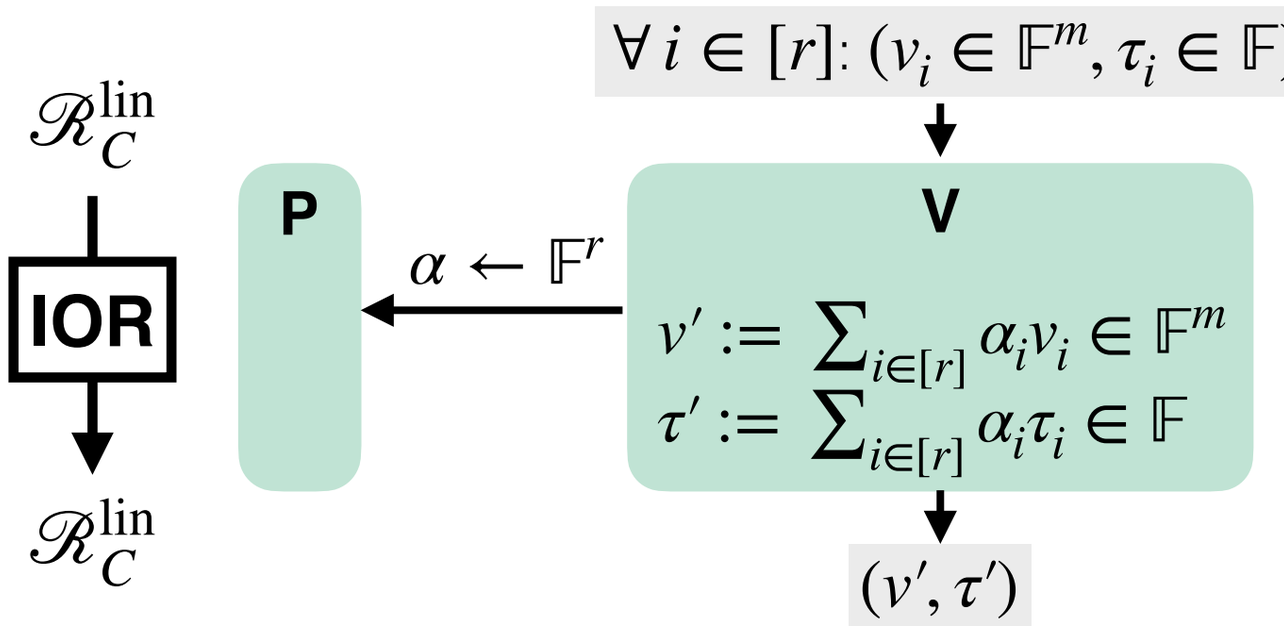
What if there are multiple constraints $(v_i, \tau_i)_i$?

Option 1: parallel sumchecks with shared randomness

Minor changes to prior slide.

Downside: communication grows with number of constraints.

Option 2: batch the linear constraints (immediately prior to folding)



Lemma: $\forall\, \delta \in [0,1]$
if f is δ-**far** from $\mathcal{R}_C^{\text{lin}}[(v_i, \tau_i)_i]$
then f is δ-**far** from $\mathcal{R}_C^{\text{lin}}[(v', \tau')]$

except with probability $\leq |\Lambda(C, \delta)| \cdot \frac{1}{|\mathbb{F}|}$

The Hop: From One Code to Another

Two codes $C_1 \subseteq \Sigma_1^{\ell_1}$ and $C_2 \subseteq \Sigma_2^{\ell_2}$.

Two encodings $\mathbf{Enc}_{C_1} : \mathbb{F}^m \to \Sigma_1^{\ell_1}$ and $\mathbf{Enc}_{C_2} : \mathbb{F}^m \to \Sigma_2^{\ell_2}$. (Same message space.)

GOAL is to hop from C_1 to C_2: $\mathscr{R}^{\text{lin}}_{C_1}$ — IOR_{hop} → $\mathscr{R}^{\text{lin}}_{C_2}$

(no shrinking the problem size)

Suppose C_1 is $\mathbb{F}$-linear, i.e., Σ_1 is an $\mathbb{F}$-linear space $\mathbb{F}^{d_1}$.

Let $G_1 \in \mathbb{F}^{d_1 \ell_1 \times m}$ be the generator matrix for $\mathbf{Enc}_{C_1}$.

Each block of d_1 rows defines a symbol of the encoding:

Def: $\forall\, i \in [\ell_1]\ \forall\, j \in [d_1]$ let $G_1[(i,j), \cdot\,] \in \mathbb{F}^m$ be row of $G_1 \in \mathbb{F}^{d_1 \ell_1 \times m}$ such that $\langle G_1[(i,j), \cdot\,], z \rangle$ is the j-th entry of the i-th symbol of $\mathbf{Enc}_{C_1}(z)$.

Takeaway: we can express every field element in $\mathbf{Enc}_{C_1}(z)$ as a linear constraint $\langle v, z \rangle$ (for suitable v)!

Generic Code Switching

$\mathscr{R}^{\text{lin}}_{C_1}$

IOR_{hop}

$\mathscr{R}^{\text{lin}}_{C_2}$

$z \in \mathbb{F}^m$ s.t.
- $f_1 = \text{Enc}_{C_1}(z)$
- $\forall\, i \in [r], \langle v_i, z \rangle = \tau_i$

P

$f_2 := \text{Enc}_{C_2}(z)$

$f_1 \in \Sigma_1^{\ell_1}$

$f_2 \in \Sigma_2^{\ell_2}$

$\forall\, i \in [r]$
$v_i \in \mathbb{F}^m$
$\tau_i \in \mathbb{F}$

V

$\forall\, i \in [t]$:
- $q_i \leftarrow [\ell_1]$
- $a_i := f_1(q_i) \in \Sigma_1 \equiv \mathbb{F}^{d_1}$
- $\forall\, j \in [d_1]$:

$v_{i,j} := G_1[(q_i, j), \cdot\,] \in \mathbb{F}^m$

Lemma: $\forall\, \delta \in [0,1]$ if f_1 is **δ-far** from $\mathscr{R}^{\text{lin}}_{C_1}[(v_i, \tau_i)_i]$

then, $\forall\, \tilde{\mathbf{P}}$, f_2 is **δ-far** from $\mathscr{R}^{\text{lin}}_{C_2}[(v'_i, \tau'_i)_i]$

except with probability $\leq |\Lambda(C_2, \delta)| \cdot (1-\delta)^t$

$\forall\, i \in [r]$	$\forall\, (i,j) \in [t] \times [d_1]$
$v_i \in \mathbb{F}^m$	$v_{i,j} \in \mathbb{F}^m$
$\tau_i \in \mathbb{F}$	$a_i[j] \in \mathbb{F}$

Useful variant:
- s OOD samples on f_2
- s additional constraints

Achieves error $\binom{|\Lambda(C_2, \delta)|}{2} \left(\frac{m}{|\mathbb{F}|}\right)^s + (1-\delta)^s$

Verifier Efficiency

The IOR verifiers that we saw create/transform linear constraints.

Created contraints:

- $\langle \hat{U}(\rho, \{0,1\}^n), z \rangle = \tau$ for an R1CS matrix $U \in \mathbb{F}^{2^m \times 2^n}$
- $\langle G[(q,j), \cdot\,], z \rangle = \tau$ for a code generator matrix $G \in \mathbb{F}^{d\ell \times m}$

Transformed contraints:

- $(\langle v_i, z \rangle = \tau_i)_i \rightarrow \langle \sum_i \alpha_i v_i, z \rangle = \sum_i \alpha_i \tau_i$
- $\langle v, z \rangle = \tau \rightarrow \langle \mathsf{Fold}(v, \rho), z \rangle = \tau'$

All vectors can be computed in **linear time** (in the vector size).

Is **sublinear verification** possible via this approach?

Succinct Verification

For "structured" codes $C_1, C_2, \ldots$ one can achieve **succinct verification**.

Def: A **succinct linear form** sl of dimension m is an algorithm that outputs a vector in $\mathbb{F}^m$ that we denote $[\mathsf{sl}]$.

Modification: replace constraints $\langle v, z \rangle = \tau$ with $\langle [\mathsf{st}], z \rangle = \tau$.

(the verifier input/outputs are pairs of the form (sl, τ))

Interleaved folding is linear: $\mathsf{Fold}(\alpha_1 v_1 + \alpha_2 v_2, \rho) = \alpha_1 \mathsf{Fold}(v_1, \rho) + \alpha_2 \mathsf{Fold}(v_2, \rho)$.

And it is cumulative: $\mathsf{Fold}(\mathsf{Fold}(v, \rho_1), \rho_2) = \mathsf{Fold}(v, \rho_1 \| \rho_2)$.

How does that help?

- Can delay batching vectors until the end.
- Efficiency is dictated by total **Fold** of initial vector.

The total **Fold** of "nice" codes is efficiently computable. Eg:

Claim: Let $\mathsf{RS}[\mathbb{F}, D, \ell]$ be "nice". There is a generator matrix G s.t.

$\forall\, k \in [\log \ell]\ \forall\, \rho \in \mathbb{F}^k\ \forall\, q \in [\ell]$

$\mathsf{Fold}(G[q, \cdot\,], \rho)$ is computable in time $O(\ell/2^k + k + \log |D|)$.

"succinctly" code-switchable

Landscape

Interleaved codes underlie several IOPP constructions:

	supported codes	**supported constraints**	**soundness regime**
Ligero [AHIV17]	RS	none	unique decoding
Brakedown [GLSTW23]	linear	multilinear	unique decoding
WHIR [ACFY25]	RS	sumcheck	list decoding
Ligerito [NA25]	linear	sumcheck	unique decoding
ZOOK [CFW26]	linear	linear	list decoding

Interleaving is also used as a subroutine (e.g., as in Blaze, Bolt).

Other IOPPs rely on tensor product codes: [BCG20], TensorSwitch, [BMMS26].

Earlier IOPPs such as FRI/STIR/Basefold exploit structure of RS codes (& co).

TODAY's lesson: even the modest structure of interleaved codes yields remarkably efficient (and useful!) IOPPs

Blueprint

A significant **post-quantum** expansion of `arkworks` is in progress.

Message me (alessandro.chiesa@epfl.ch) if you are interested to get involved!

ENCODE WITNESS

"non-linear" NP-complete relation

$\mathscr{R}_0$

LOCAL TESTING OF LINEAR CONSTRAINTS

"linear" relations

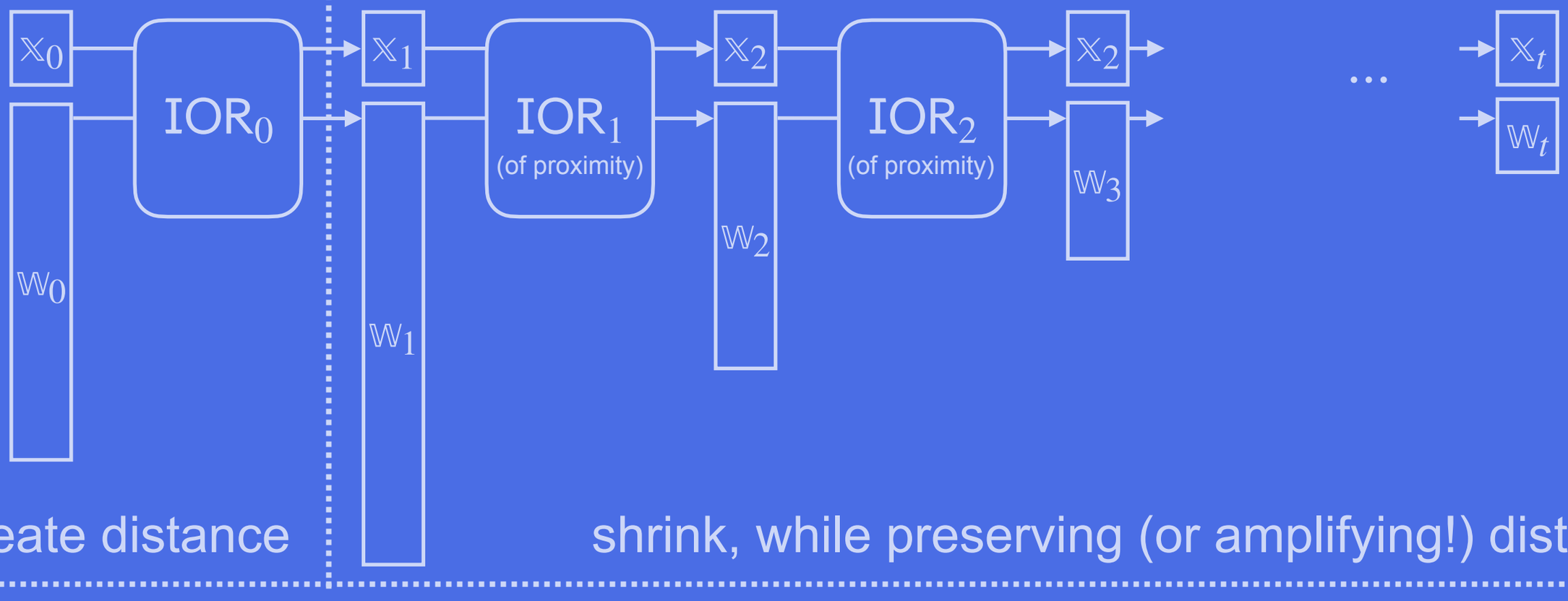
$\mathscr{R}_1$ based on a code C_1

$\mathscr{R}_2$ based on a code C_2

$\mathscr{R}_3$ based on a code C_3

$\cdots$

$\mathscr{R}_t$ based on a code C_t

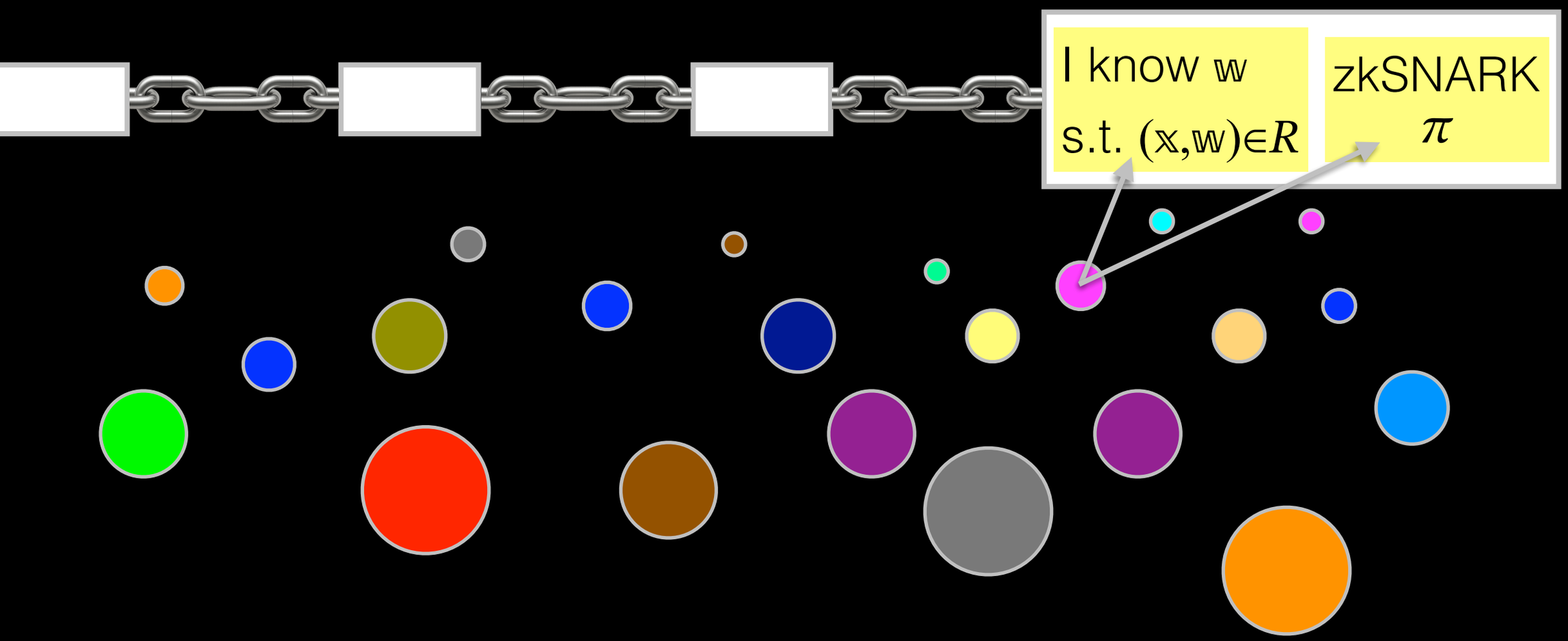


create distance

shrink, while preserving (or amplifying!) distance

Part 1 Part 2

Thanks!



Backup: Out-Of-Domain Sampling

Let $C \subseteq \Sigma^\ell$ be a code with encoding $\mathsf{Enc}_C \colon \mathbb{F}^m \to \Sigma^\ell$.

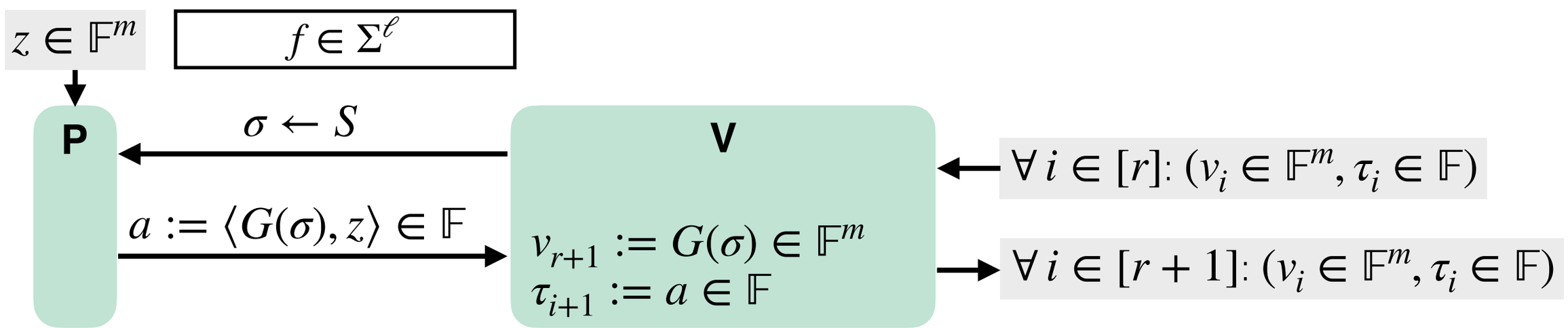
Let $G \colon S \to \mathbb{F}^m$ be such that $\max_{v \in \mathbb{F}^m \setminus \{0^m\}} \Pr_{\sigma \leftarrow S} [\langle G(\sigma), v \rangle = 0] \leq \epsilon$.

Eg $\sigma \in \mathbb{F}^{\log m} \mapsto (\mathsf{eq}_{\log}(\sigma, b))_{b \in \{0,1\}^{\log m}}$ has $\epsilon = \frac{\log m}{|\mathbb{F}|}$.

Lemma: $\forall\, \delta \in [0,1]\ \forall f \in \Sigma^\ell$,

$$\Pr_{\sigma \leftarrow S} \left[\begin{array}{l} \exists\, c_1, c_2 \in \Lambda(C, f, \delta) : \\ z_1 \neq z_2 \wedge \langle G(\sigma), z_1 \rangle = \langle G(\sigma), z_2 \rangle \\ \text{where } z_1 := \mathsf{Enc}_C^{-1}(c_1), z_2 := \mathsf{Enc}_C^{-1}(c_2) \end{array} \right] \leq \binom{|\Lambda(C, \delta)|}{2} \cdot \epsilon$$

One can use such linear constraints to "trim" lists:



Building Cryptographic Proofs from Hash Functions

Alessandro Chiesa & **Eylon Yogev**

Comprehensive and rigorous treatment of SNARGs in the ROM.

Part I
What are Cryptographic Proofs?

Part II
NARGs Based on SPs
- FS protocol for SPs

Part III
NARGs Based on IPs
- FS protocol for IPs
- "fast" FS protocol for IPs

Part IV
Commitment Schemes
- basic commitment
- Merkle commitment

Part V
SNARGs Based on PCPs
- Kilian protocol
- Micali protocol

Part VI
SNARGs Based on IOPs
- iBCS protocol
- BCS protocol

Part VII
Practical Considerations
- setting parameters
- Merkle optimizations
- RBR soundness
- special soundness

`snargsbook.org`