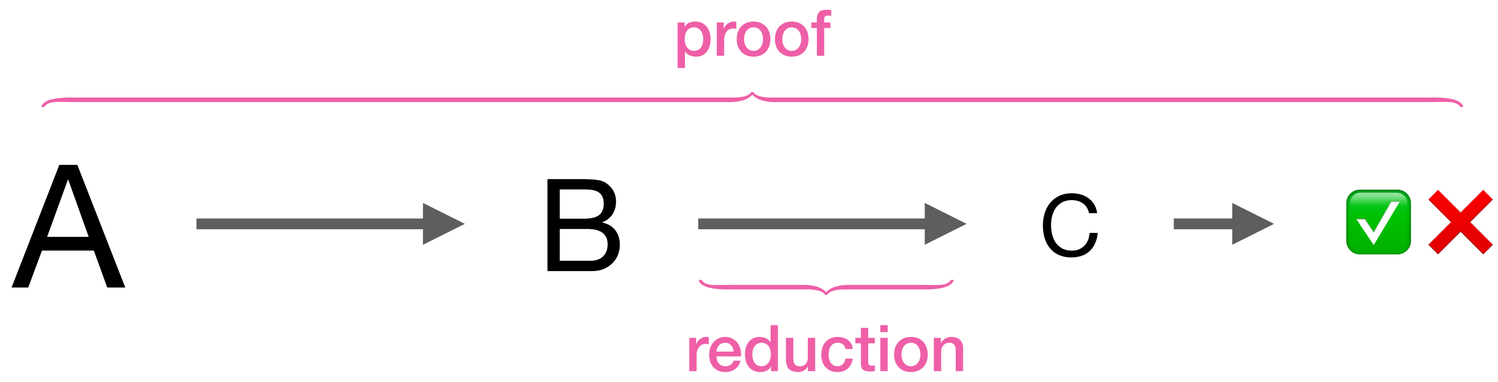
Interactive Oracle Reductions

William Wang (NYU)

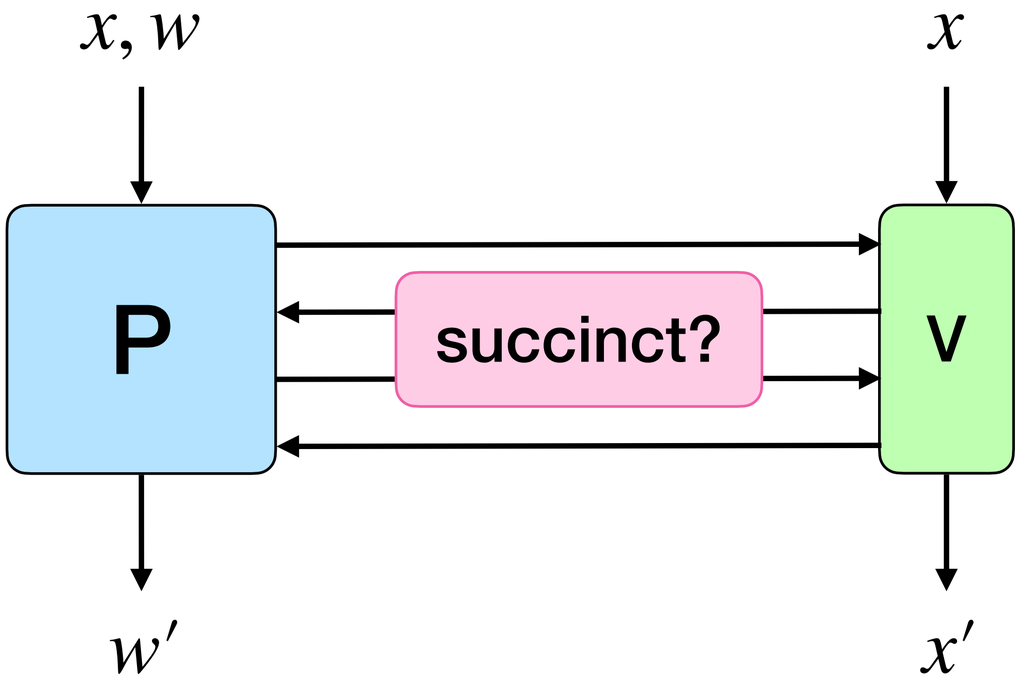
Proofs and reductions



Interactive reductions

[KP22]

$$(x, w) \stackrel{?}{\in} R$$

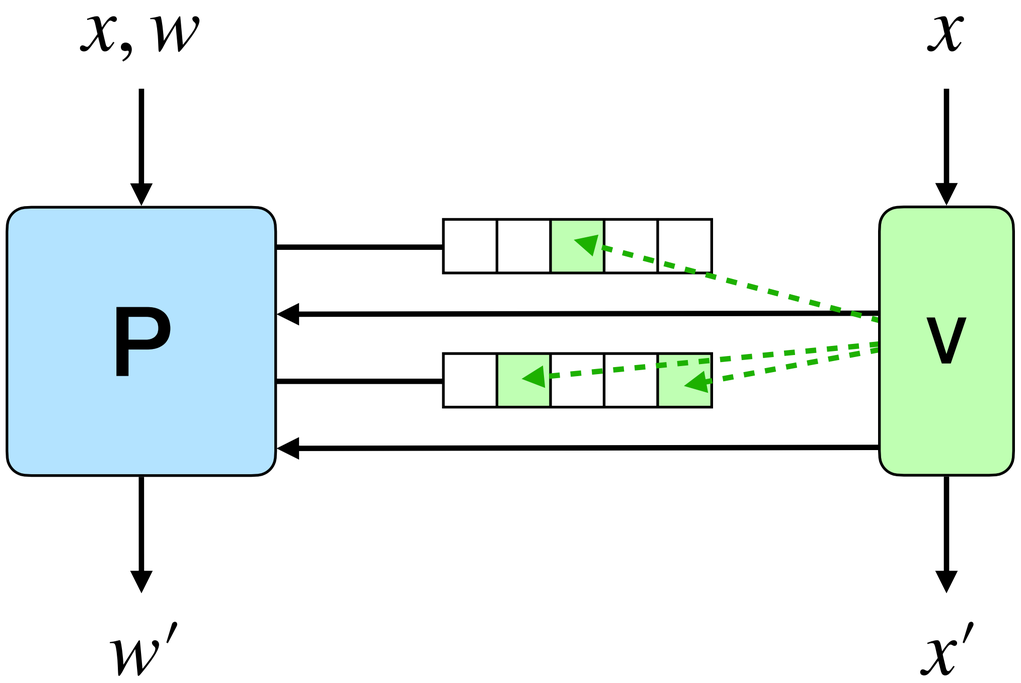


$$(x', w') \stackrel{?}{\in} R'$$

Interactive oracle reductions

A first attempt

$$(x, w) \stackrel{?}{\in} R$$

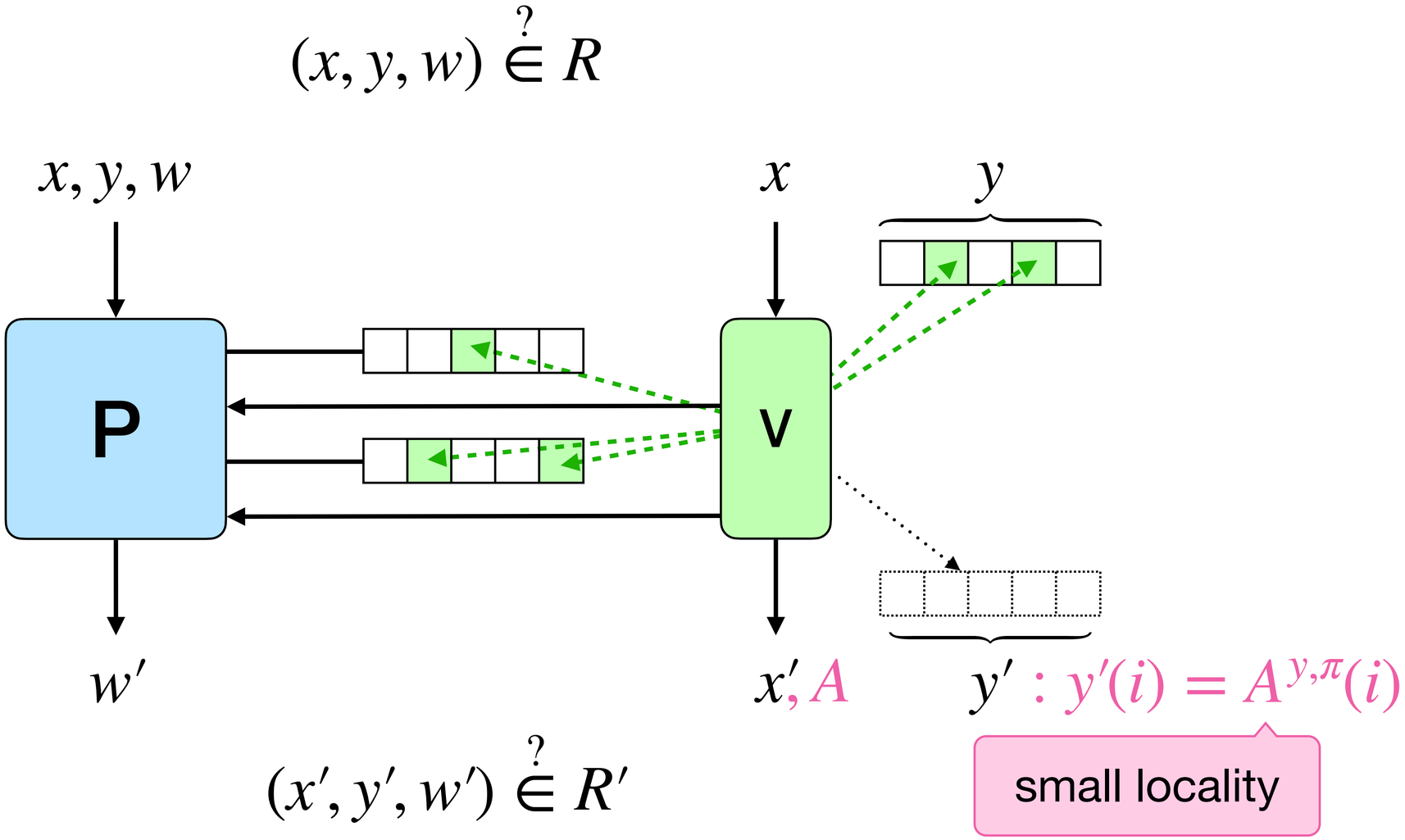


$$(x', w') \stackrel{?}{\in} R'$$

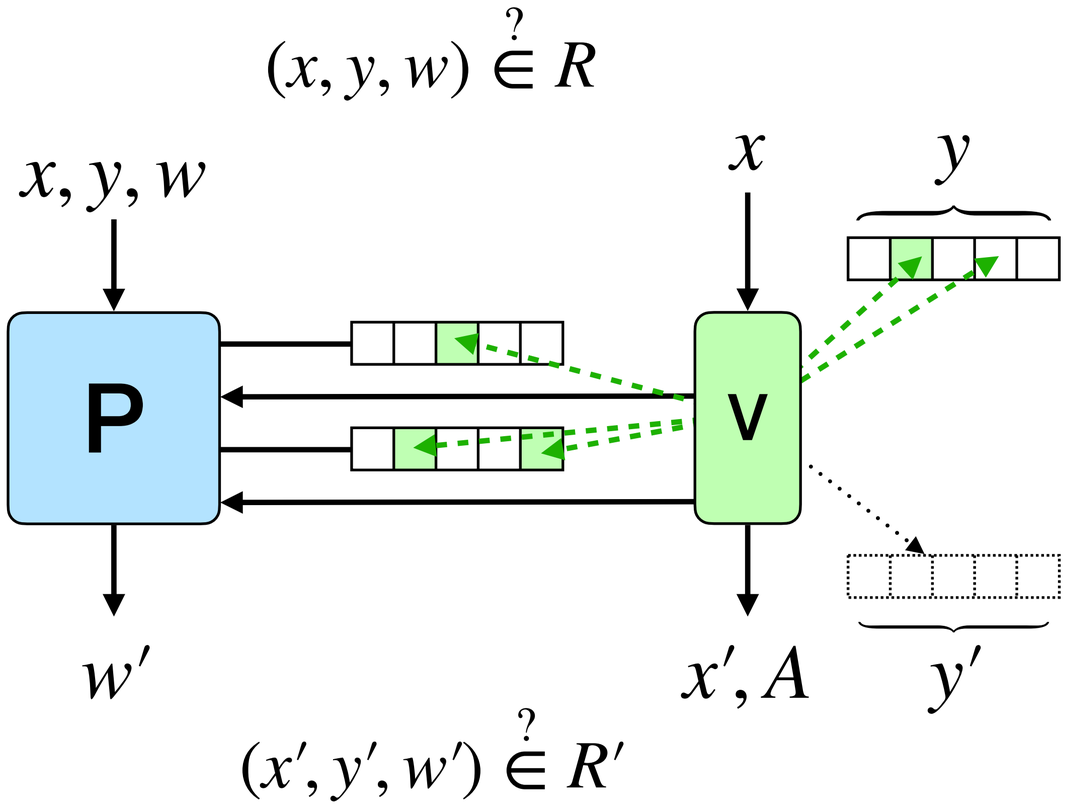
not expressive enough…

Interactive oracle reductions

For real [BCGGRS19, BMNW25]



IOR security



completeness

$$(x, y, w) \in R \implies (x', y', w') \in R'$$

in general, promise relations

Soundness based on proximity.

$$L_x := \{y : \exists w, (x, y, w) \in R\}$$

soundness

y far from L_x

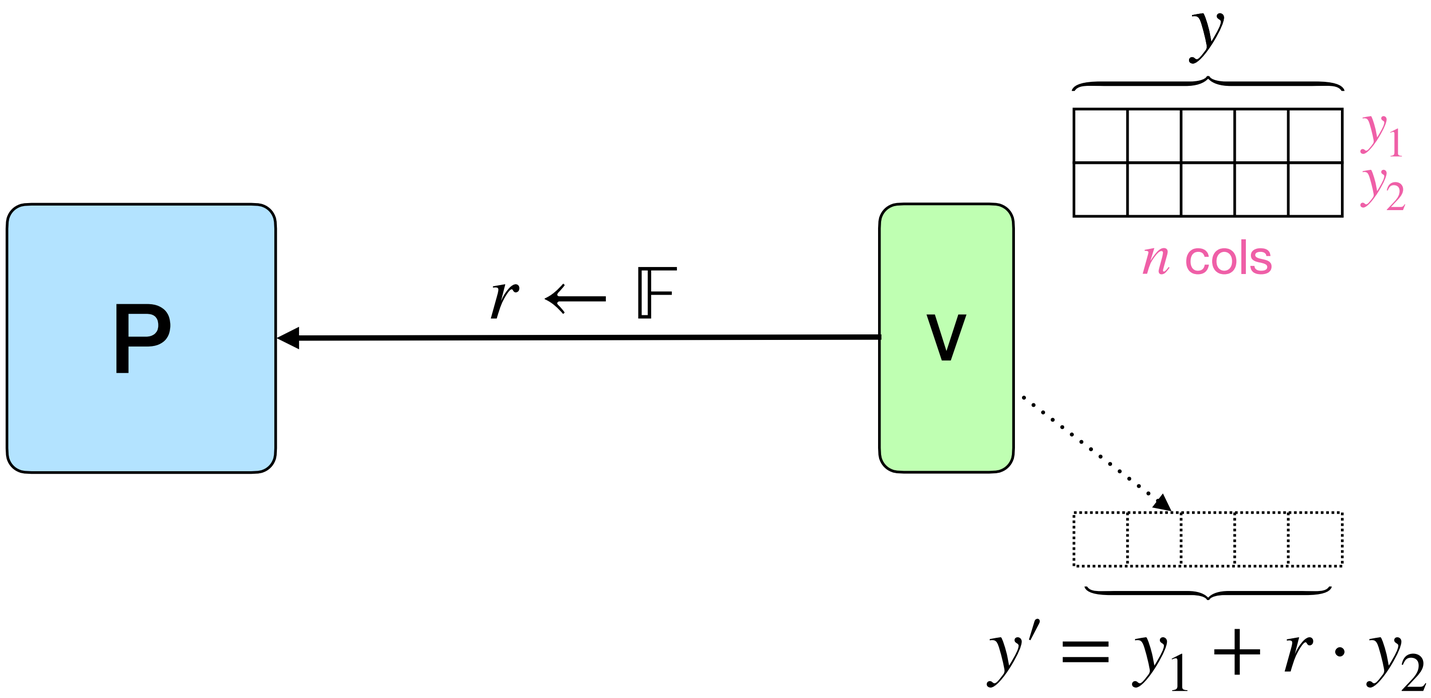
$\implies$ w.h.p. y' far from $L'_{x'}$

IORs in action: interleaved folding

[AHIV17]

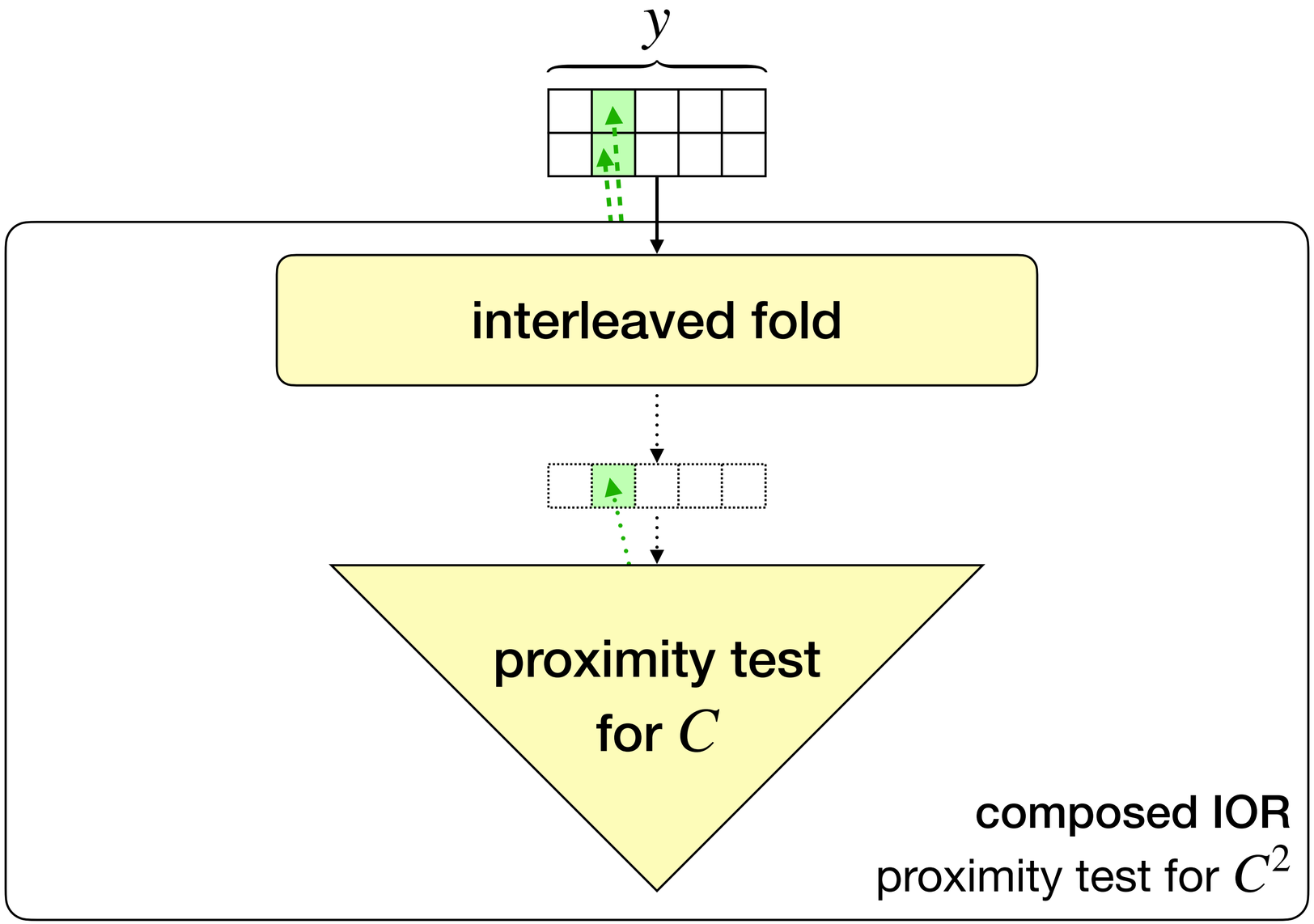
Setup: Code $C \subset \mathbb{F}^n$

y close to **interleaved code** C^2?



y' close to C?

IORs in action: composition



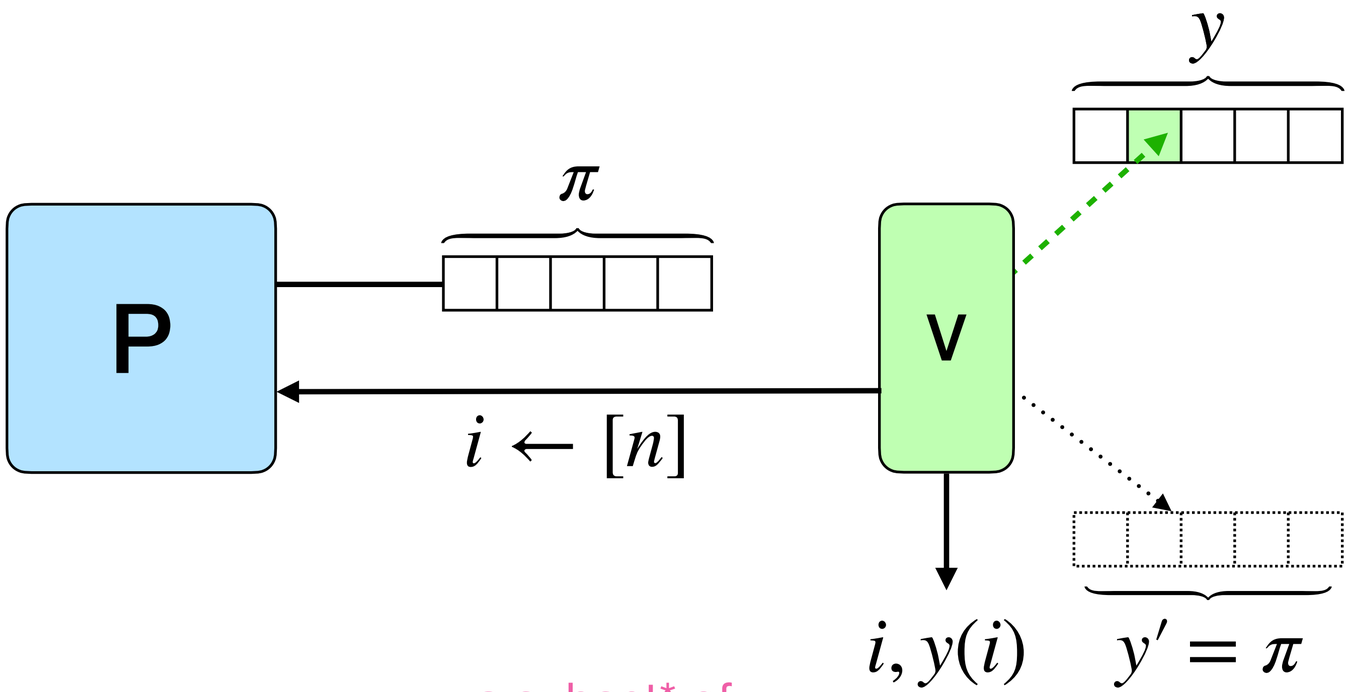
IORs in action: code switching

[RR22]

Setup: codes C, C'

y close to C?

Claim:
$C^{-1}(y) = C'^{-1}(\pi)$



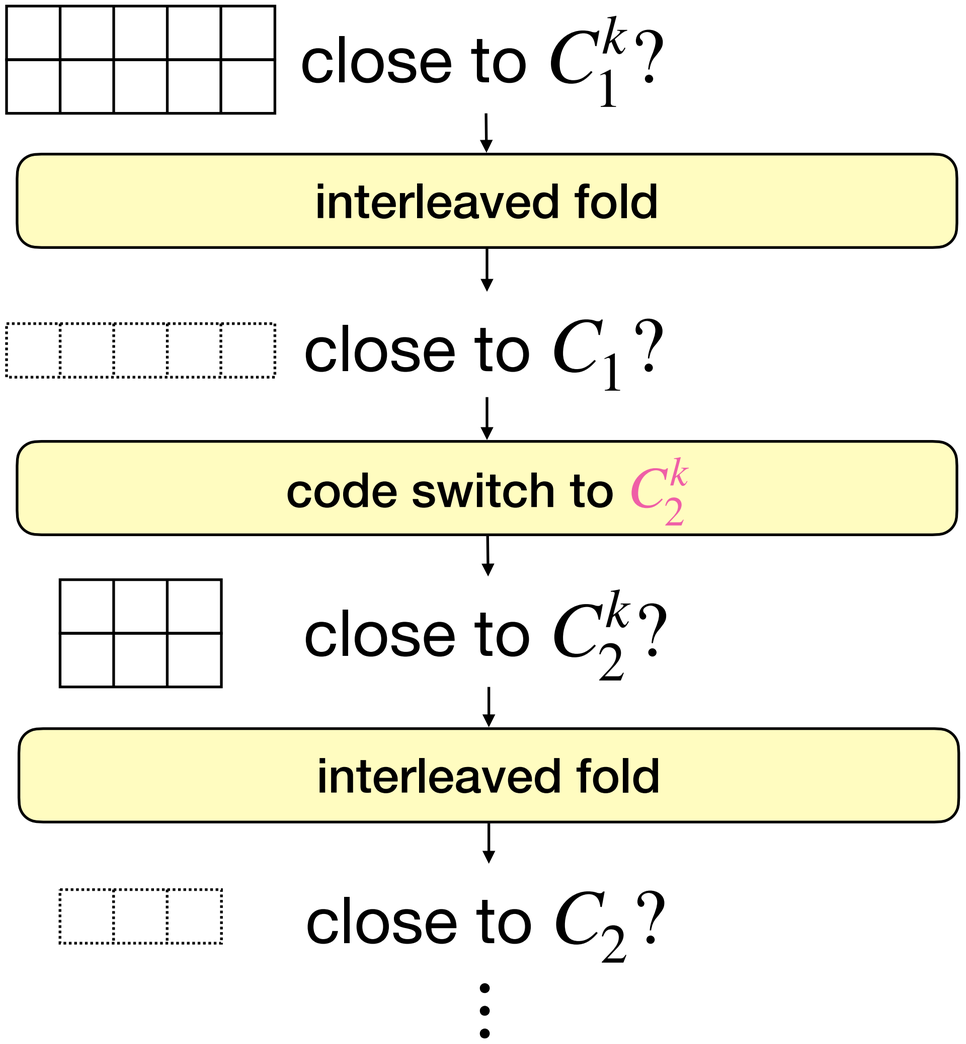
a subset* of

y' close to C'?

*$\{C'(m) : C(m)(i) = y(i)\}$

A simple proximity test

Setup: progressively smaller codes $C_1, C_2, C_3, \ldots$



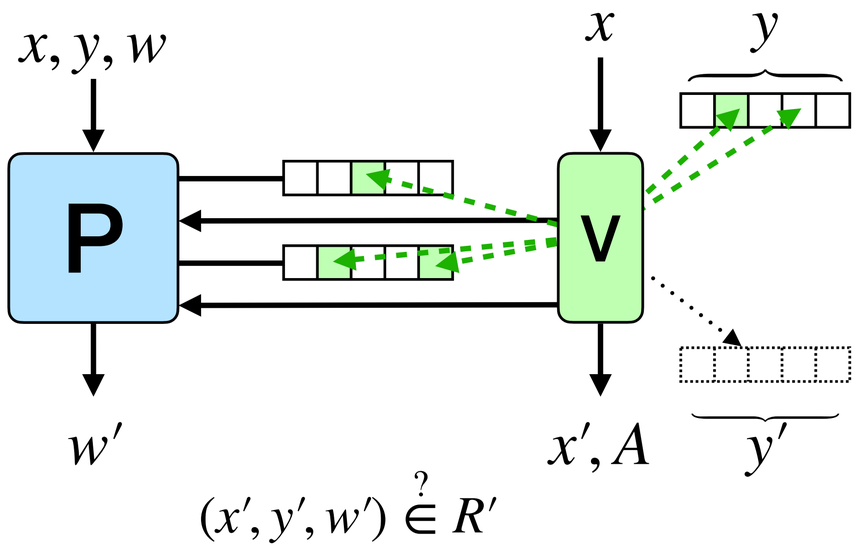
Standalone reductions

interactive oracle reduction

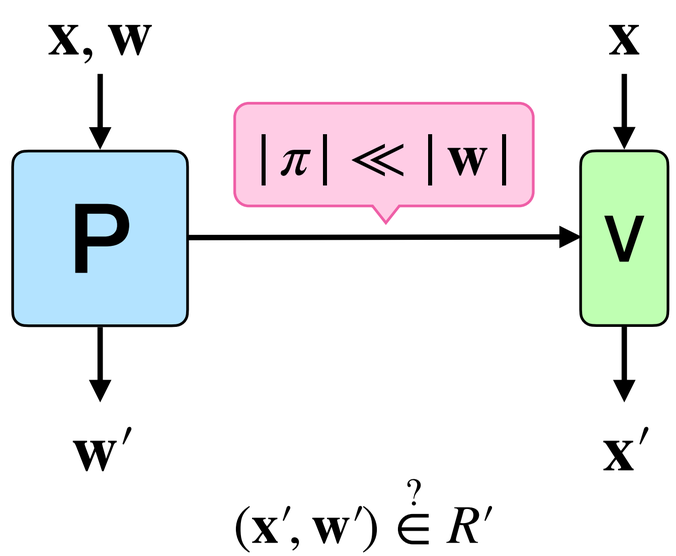
succinct non-interactive reduction

$\mathbf{x} = (x, \mathsf{cm}_y)$

$\mathbf{w} = (y, w)$



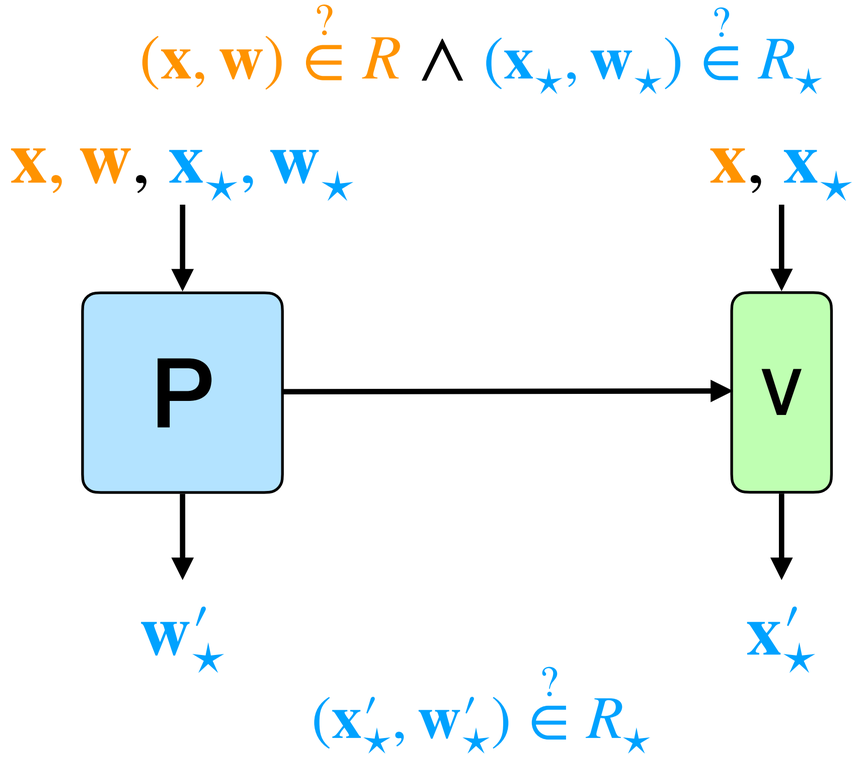
Merkle trees
+
Fiat-Shamir



Accumulation schemes

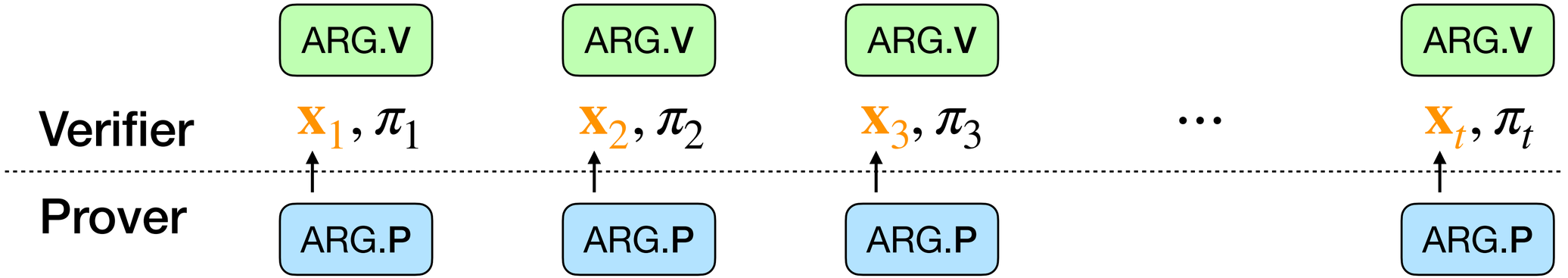
[BCLMS21, KST21]

An accumulation (or folding) scheme for R is a reduction from $R \times R_\star$ to $R_\star$.



Goal: be “lightweight”, compared to (succinct) arguments for R.

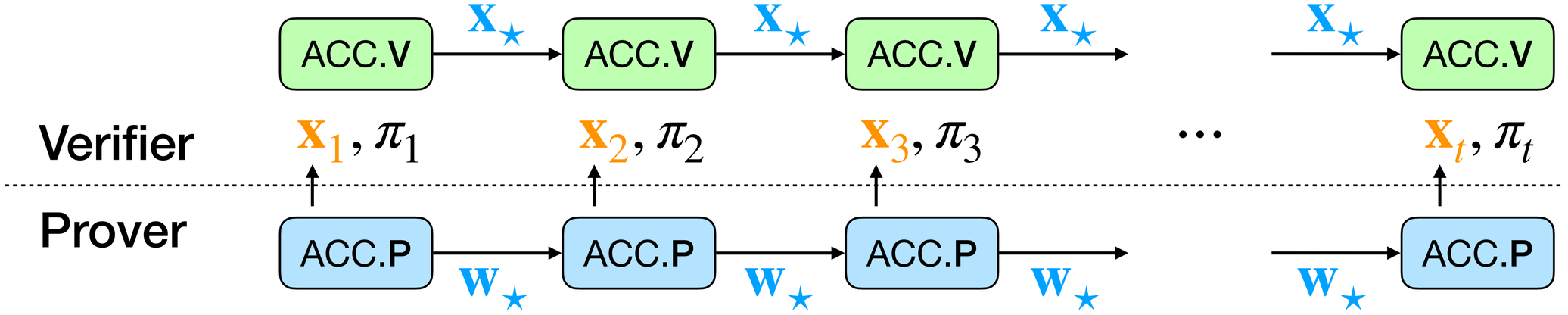
Use case: batch verification



succinct argument

---

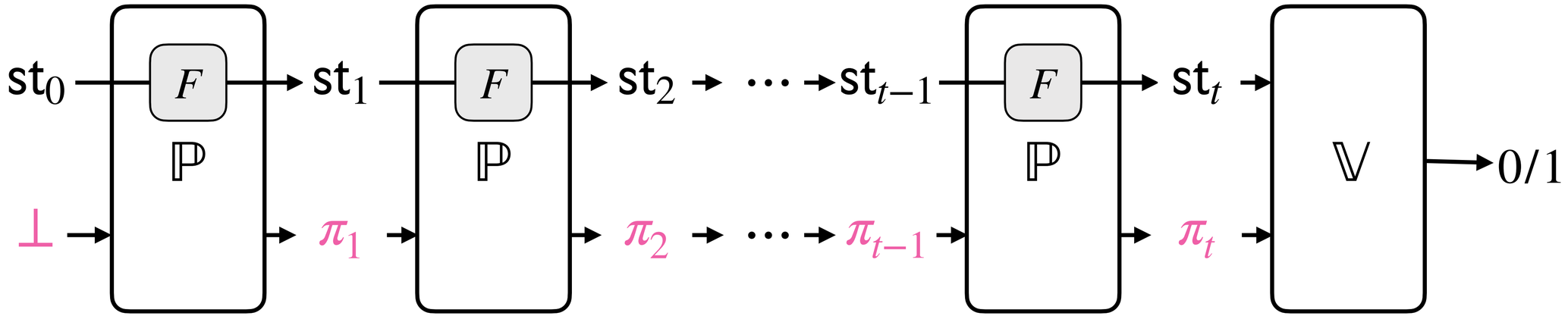
accumulation scheme



Incrementally verifiable computation

[Val08]

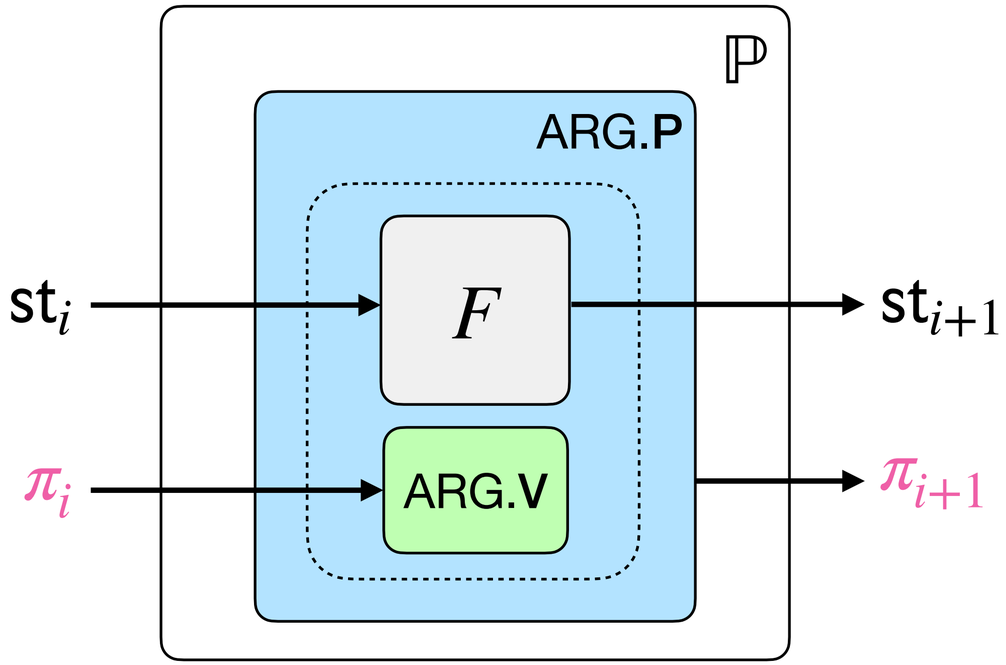
IVC prover $\mathbb{P}$ and verifier $\mathbb{V}$



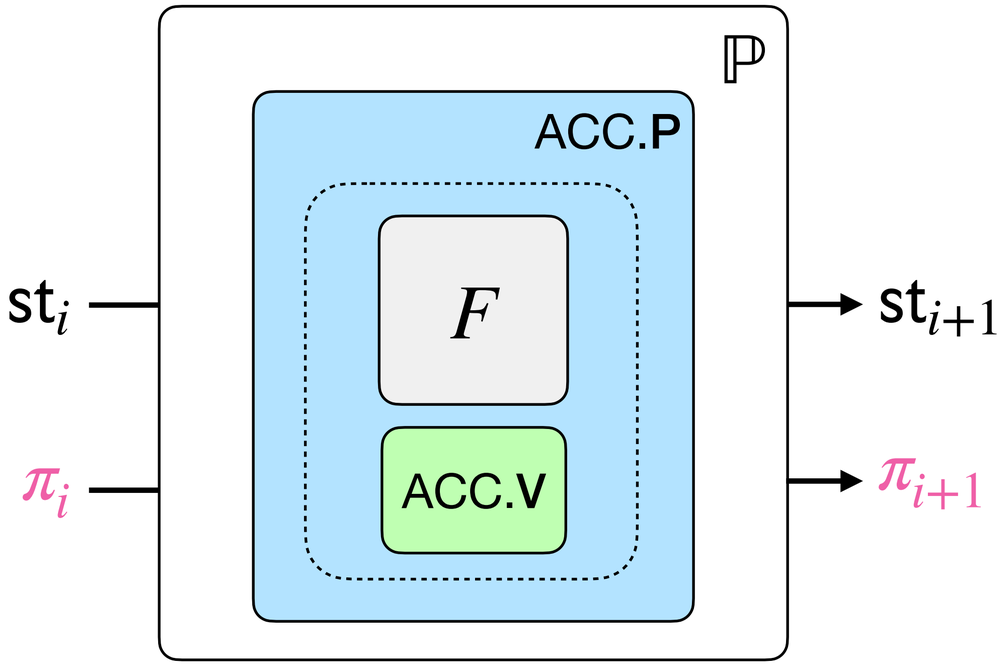
Proof-carrying data (PCD) generalizes IVC to acyclic graphs [CT10]

Use case: IVC

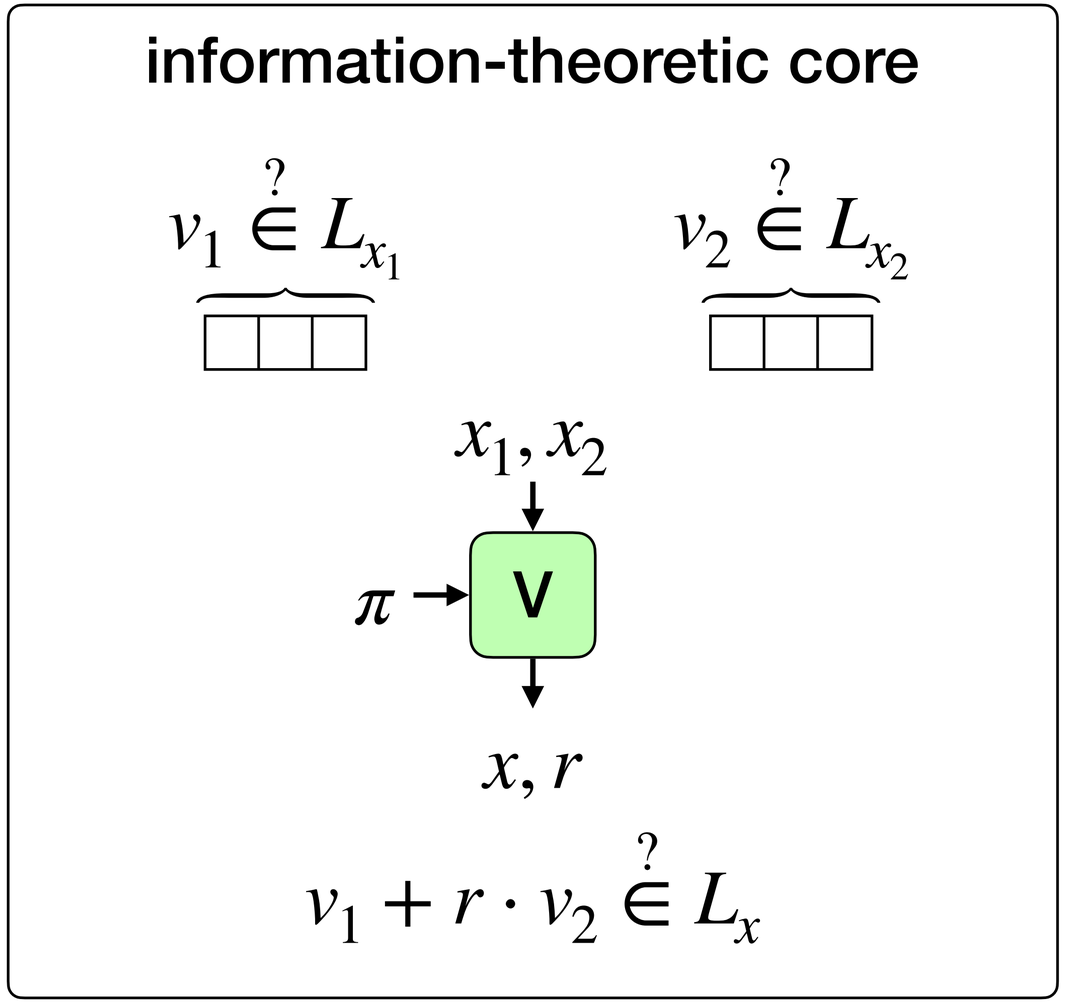
succinct argument



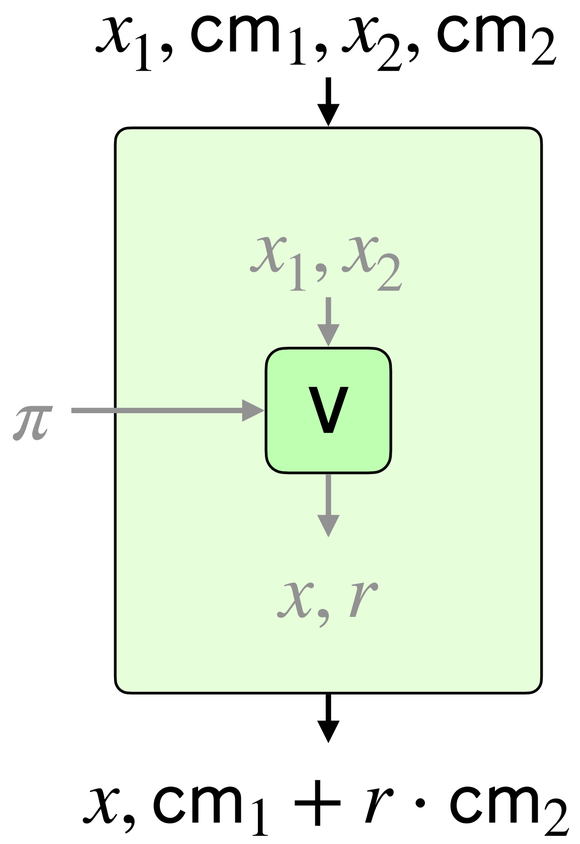
accumulation scheme



Prior accumulation schemes



linearly homomorphic commitment

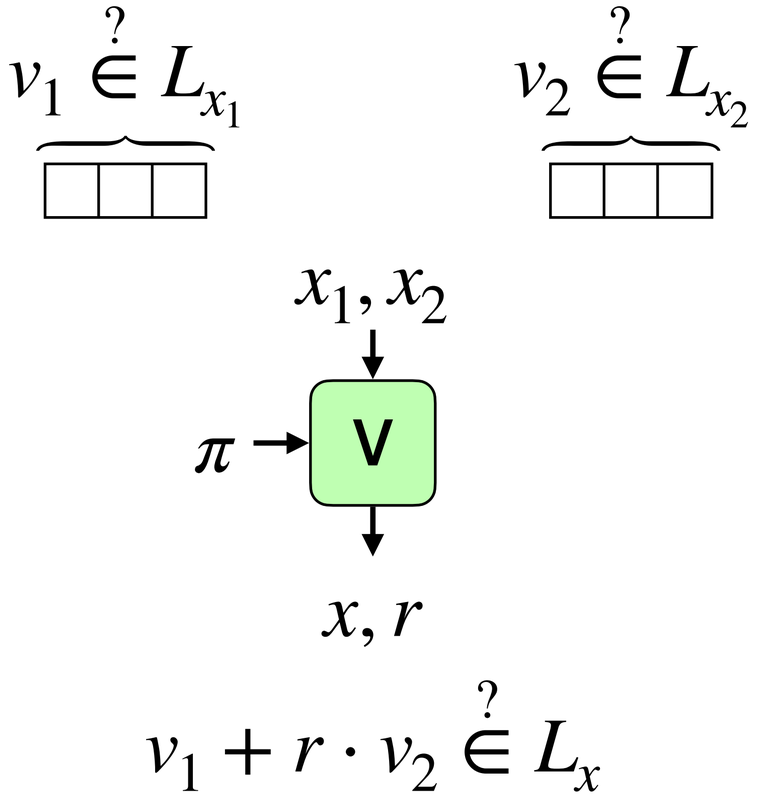


can we use commitments without homomorphism, e.g., **Merkle trees**?

Lifting to an IOR

[BMNW25, BCFW25]

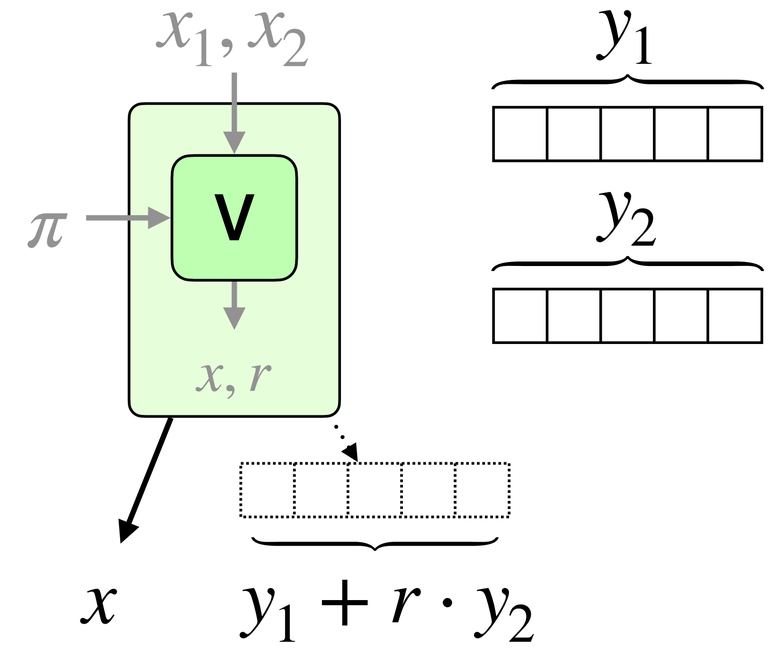
information-theoretic core



input claim

y_1 close to $C(L_{x_1})$ and y_2 close to $C(L_{x_2})$?

encode with C



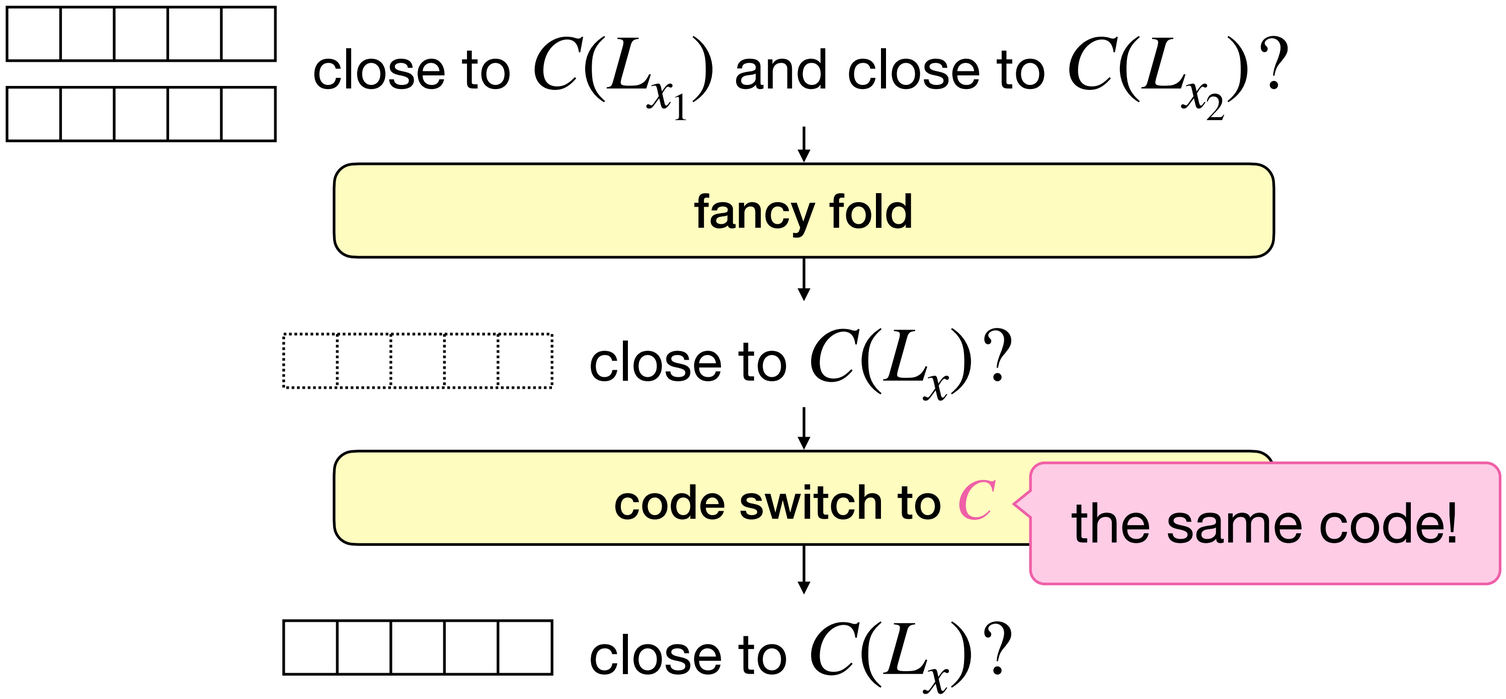
output claim

$y_1 + r \cdot y_2$ close to $C(L_x)$?

Lifting to an IOR

[BMNW25, BCFW25]

Setup: code C



Summary: IORs for IOPs

Takeaway: IORs enable modular constructions of IOPs.

- (functional) IORs are the "central information-theoretic object" underlying **ArkLib**.

overall soundness error $\leq \varepsilon_1 + \varepsilon_2 + \varepsilon_3$

Every IOP with Fiat-Shamir security obeys this structure.

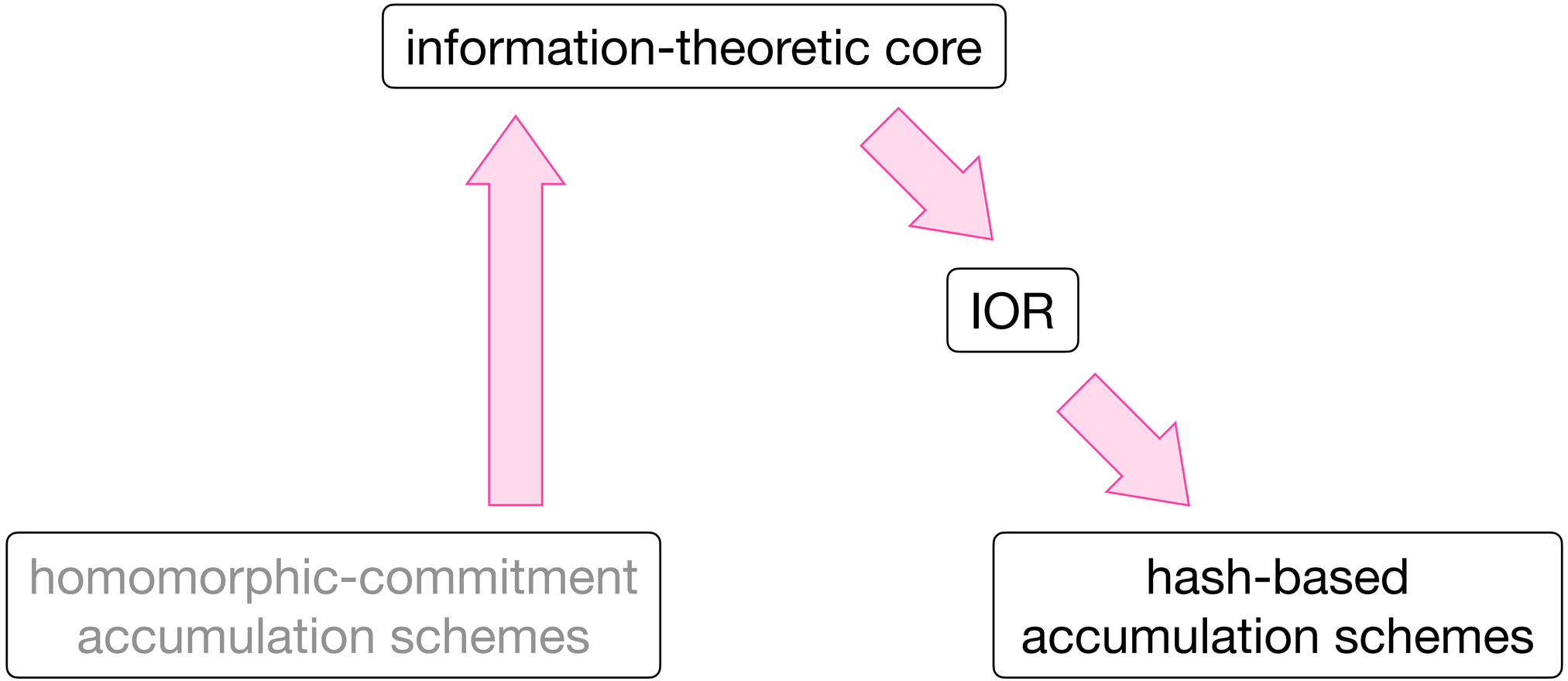
IOP is round-by-round sound [CCHLRR18]

⇳

IOP decomposes into one-round IORs

Summary: IORs for accumulation

Takeaway: IORs yield hash-based accumulation schemes.



Thanks!