This week:

The Evolution of Proofs in Computer Science

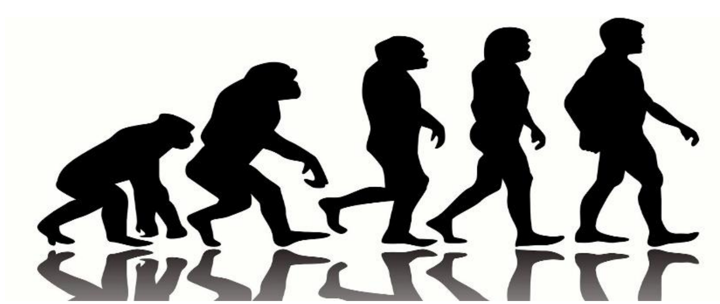
Classical proofs

(Zero-knowledge)
Interactive proofs

Multi-prover
interactive proofs

Probabilistically
checkable proofs (PCPs)

Succinct non-interactive arguments
(SNARGs)



Zero-Knowledge Proofs

[Goldwasser-Micali-Rackoff85]

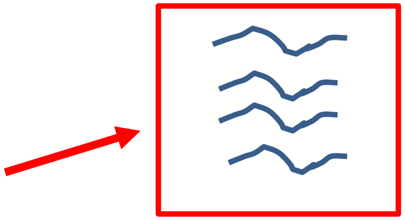
Proofs that **reveal no information** beyond the validity of the statement

Zero-Knowledge Proofs

[Goldwasser-Micali-Rackoff85]

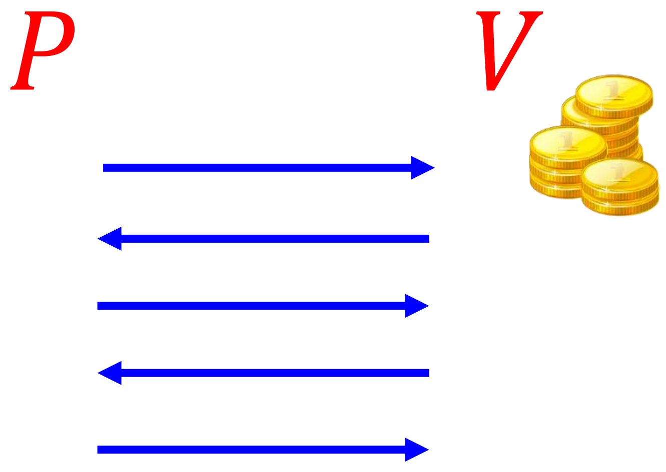
Impossible!

This is information!



Interactive Proofs

[Goldwasser-Micali-Rackoff85]



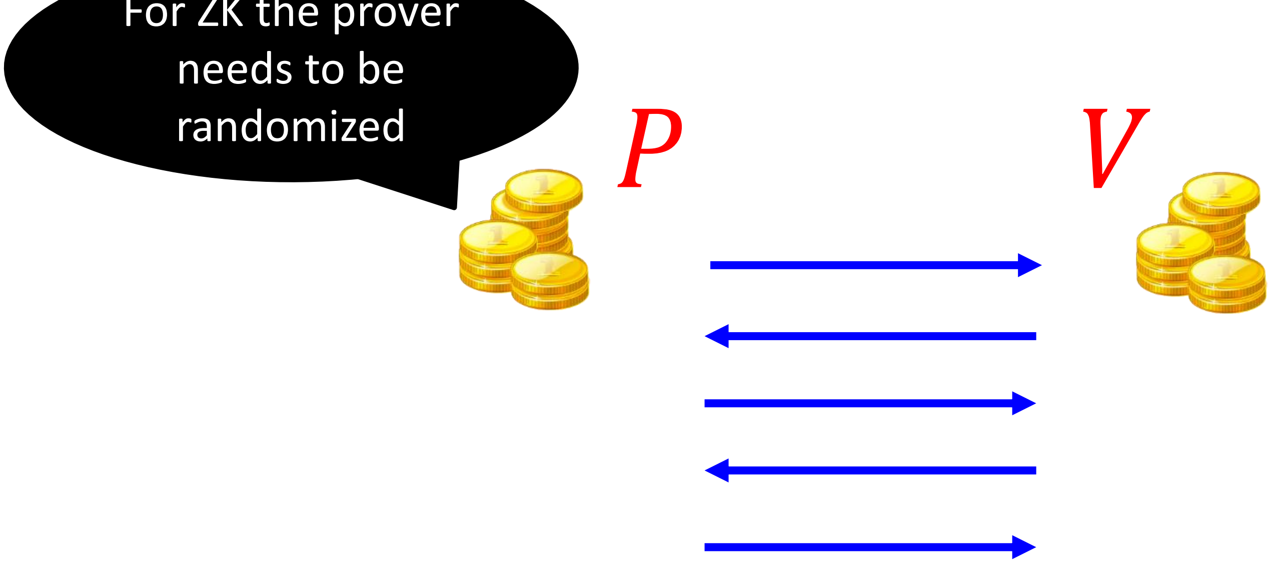
Completeness: $\forall x \in L \;\; \Pr[(P,V)(x) = 1] \geq 2/3$

Soundness: $\forall x \notin L, \forall P^* \;\; \Pr[(P^*,V)(x) = 1] \leq 1/3$

Note: By repetition, we can get completeness $1 - 2^{-k}$ and soundness 2^{-k}

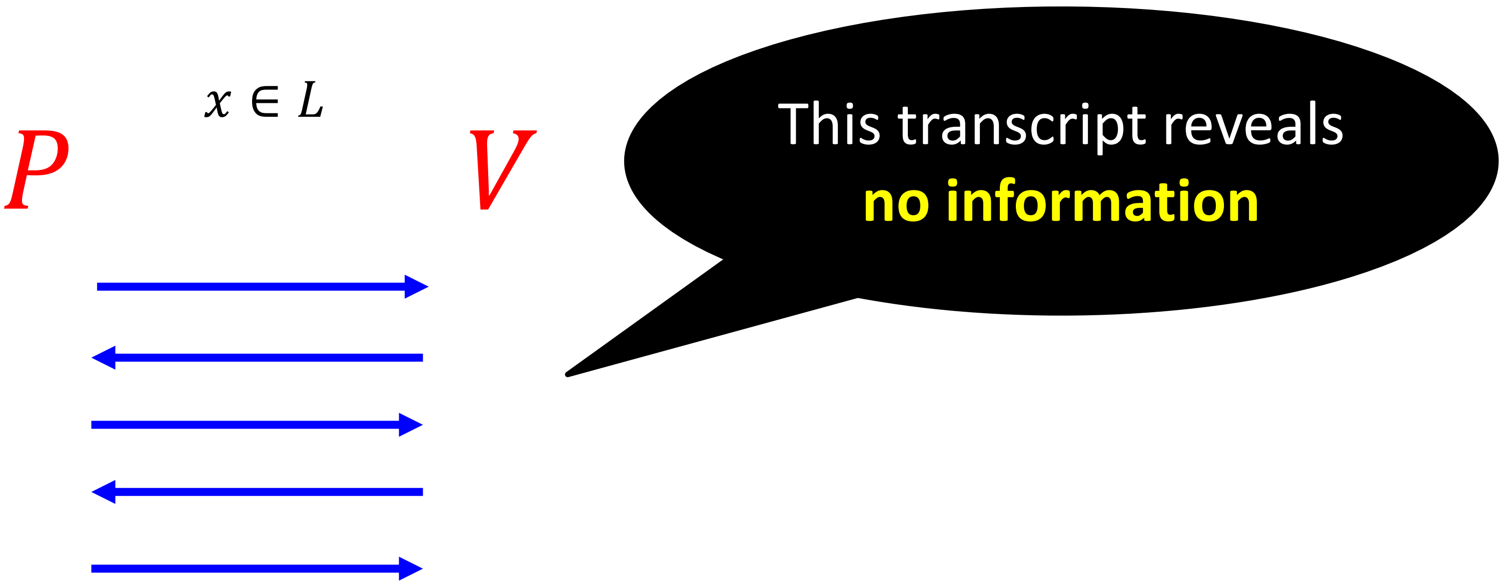
Interactive Proofs

[Goldwasser-Micali-Rackoff85]



[Goldreich-Micali-Wigderson87]: Every statement that has a classical proof has **zero-knowledge (ZK)** interactive proof, assuming one-way functions exist

Defining Zero-Knowledge



Formally: There exists a PPT algorithm S (called a simulator), such that for every PPT (cheating) verifier $V^*$and for every $x \in L$:

$$S(x) \approx (P, V^*)(x)$$

Denotes the transcript

ZK Proofs for NP

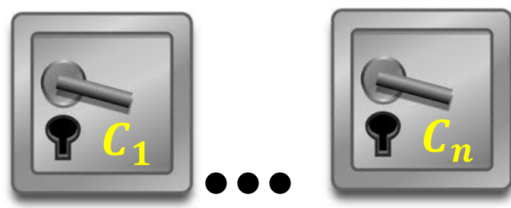
Vertices can be colored by {1,2,3} s.t. no two adjacent vertices are colored by the same color

Focus on the NP-complete language of all 3-colorable graphs:

$$G = (Ver, E)$$

P V

Locked safe, reveals no information about its content



Randomly permute the coloring, to obtain valid coloring $(c_1, \ldots, c_n)$

$P \rightarrow V$: safes $c_1, \ldots, c_n$

Choose a random edge $(i,j) \in E$

$V \rightarrow P$: $(i,j) \in E$

$P \rightarrow V$: Open safes i, j

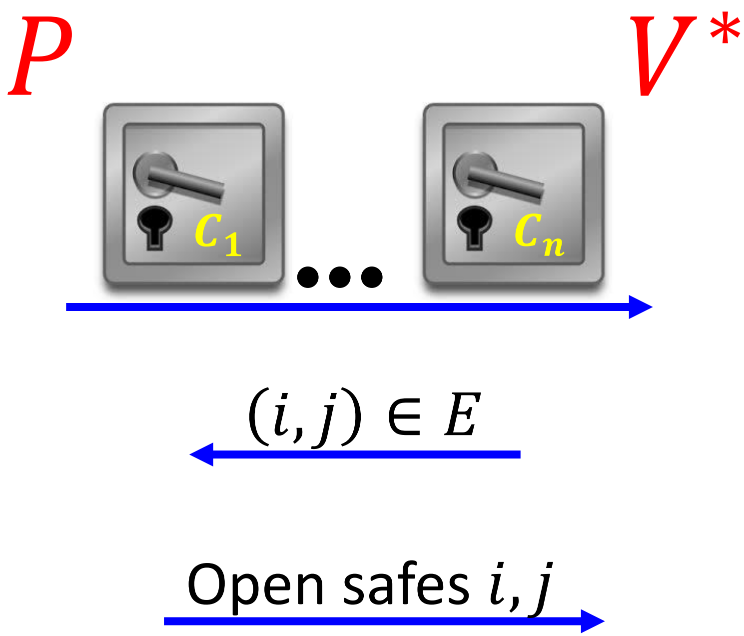
Soundness: Only $1 - \frac{1}{|E|}$ but can be amplified via repetition.

ZK Proofs for NP

Focus on the NP-complete language of all 3-colorable graphs:

For honest V

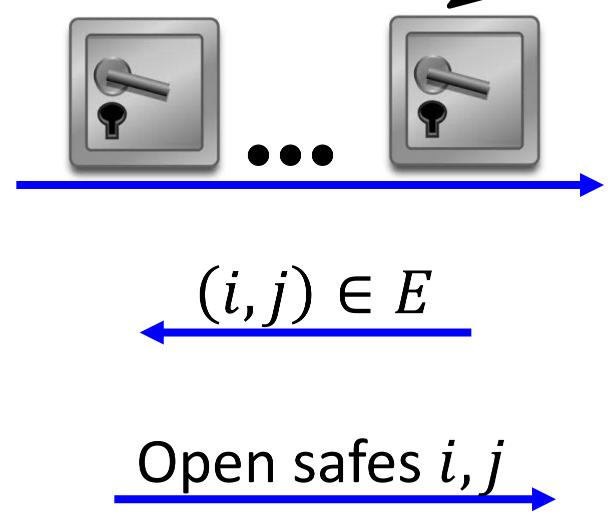
$G = (Ver, E)$



$S(G)$:

1. Choose a random $(i,j) \in E$
2. Choose random distin[...] colors c_i, c_j
3. The simulated transc[...]

safes i, j have values c_i, c_j



Implementing Digital Safes: **Commitment Scheme**

Commitment scheme is a randomized algorithm Com s.t.

- **Computationally Hiding:**

$$\forall m, m' \;\; Com(m; r) \approx Com(m'; r')$$

- **Statistically Binding:** $\nexists (m, r), (m', r')$ s.t. $m \neq m'$ and

$$Com(m; r) = Com(m'; r')$$

Constructing a Commitment Scheme

Construction 1:

$(f(r), p(r)) \approx (f(r), U)$

Let $f: \{0,1\}^* \rightarrow \{0,1\}^*$ be an injective **OWF**,
and $p: \{0,1\}^* \rightarrow \{0,1\}$ be a corresponding **hardcore predicate**.

$$Com(b; r) = (f(r), p(r) \oplus b)$$

Binding: Follows from the fact that f is injective

Hiding: Relies on the fact that if **f is one-way** then:

$$(f(r), p(r)) \approx (f(r), U)$$

Constructing a Commitment Scheme

Construction 2: computationally hiding, and statistically binding [Pederson]

Let G be a group of prime order p, let $g \in G$ be any generator, and h be a random group element.

$$\boldsymbol{Com_{g,h}(m,r) = g^m h^r}$$

Hiding: Information theoretically!

Binding: Follows from the Discrete Log assumption.

Perfect ZK Computationally Sound Proofs

For the NP-complete language of all 3-colorable graphs

$\boldsymbol{G = (V, E)}$

P $\qquad$ V

$\boldsymbol{g, h}$ ←

Randomly permute the coloring, to obtain valid coloring $(\boldsymbol{c_1, \ldots, c_n})$

$\boldsymbol{Com_{g,h}(c_1), \ldots, Com_{g,h}(c_n)}$ →

$(i, j) \in E$ ←

Choose a random edge $(\boldsymbol{i, j}) \in \boldsymbol{E}$

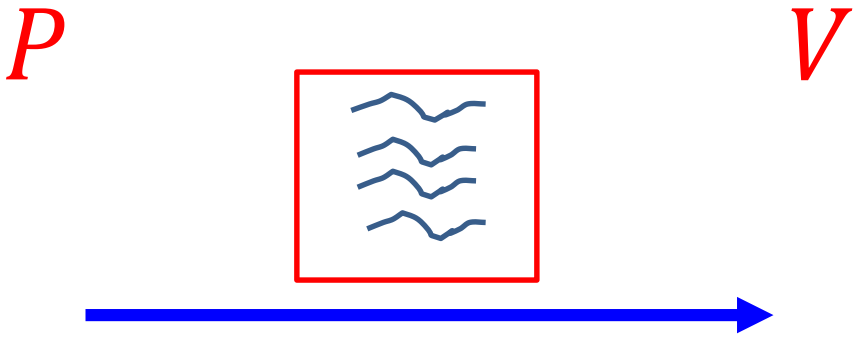
Reveal c_i, c_j, with corresponding randomness →

So Far…

- **Constructed ZK proofs for all of NP**
  - using commitment schemes

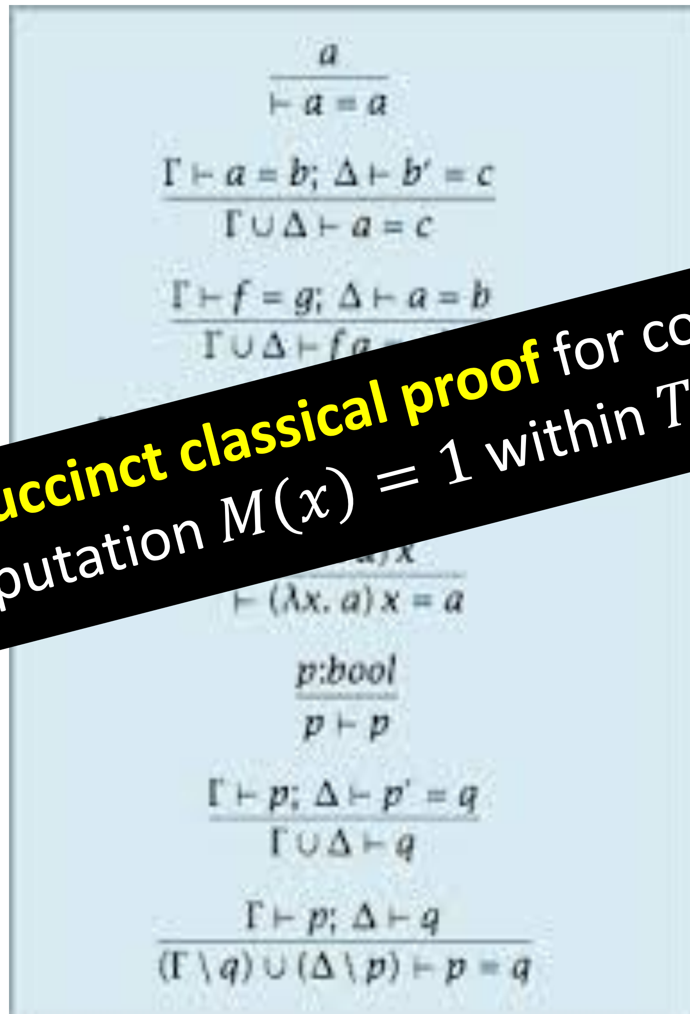
Interactive Proofs are more efficient!

Classical Proofs



Classical Proofs

P V

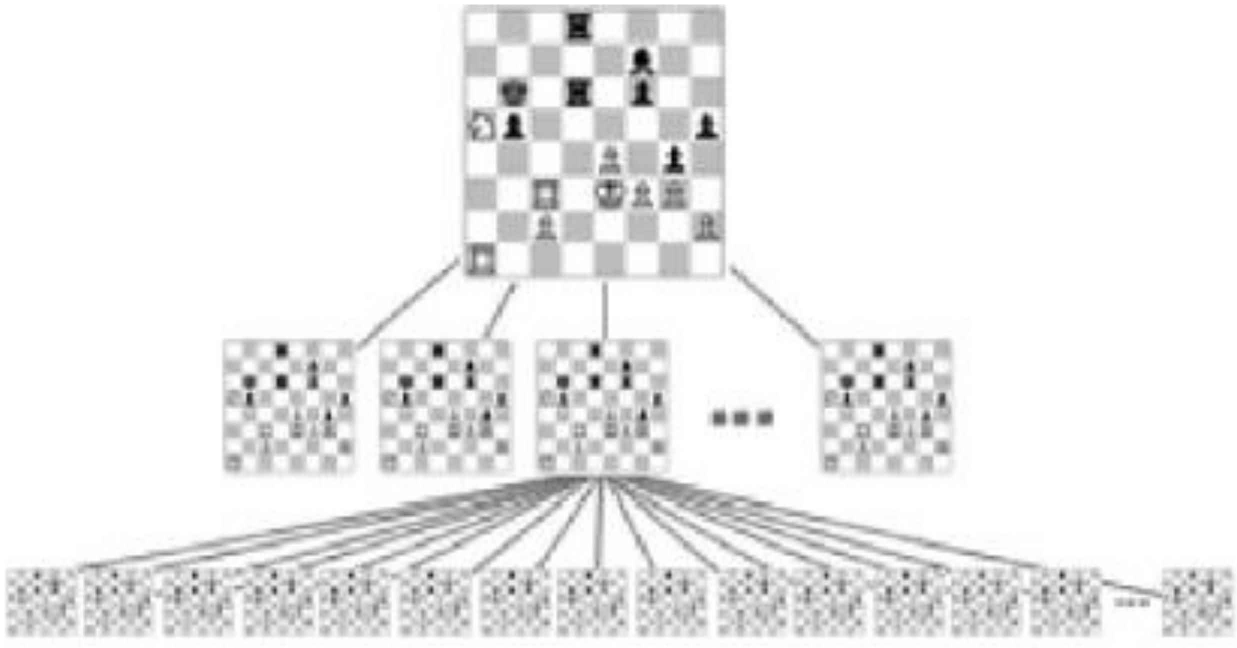


Conjecture: $\nexists$ **succinct classical proof** for correctness of any computation $M(x) = 1$ within T steps

Interactive Proofs are More Efficient!

[Lund-Fortnow-Karloff-Nissan90, Shamir90]

Example: Chess



Interactive Proofs are More Efficient!

[Lund-Fortnow-Karloff-Nissan90, Shamir90]

correctness of any computation can be proved:

Time to verify

$\approx$

Space required to do the computation

Interactive Proof

$$IP = PSPACE$$

Interactive Proofs are More Efficient!

[Lund-Fortnow-Karloff-Nissan90, Shamir90]

correctness of any computation can be proved:

Time to verify

≈

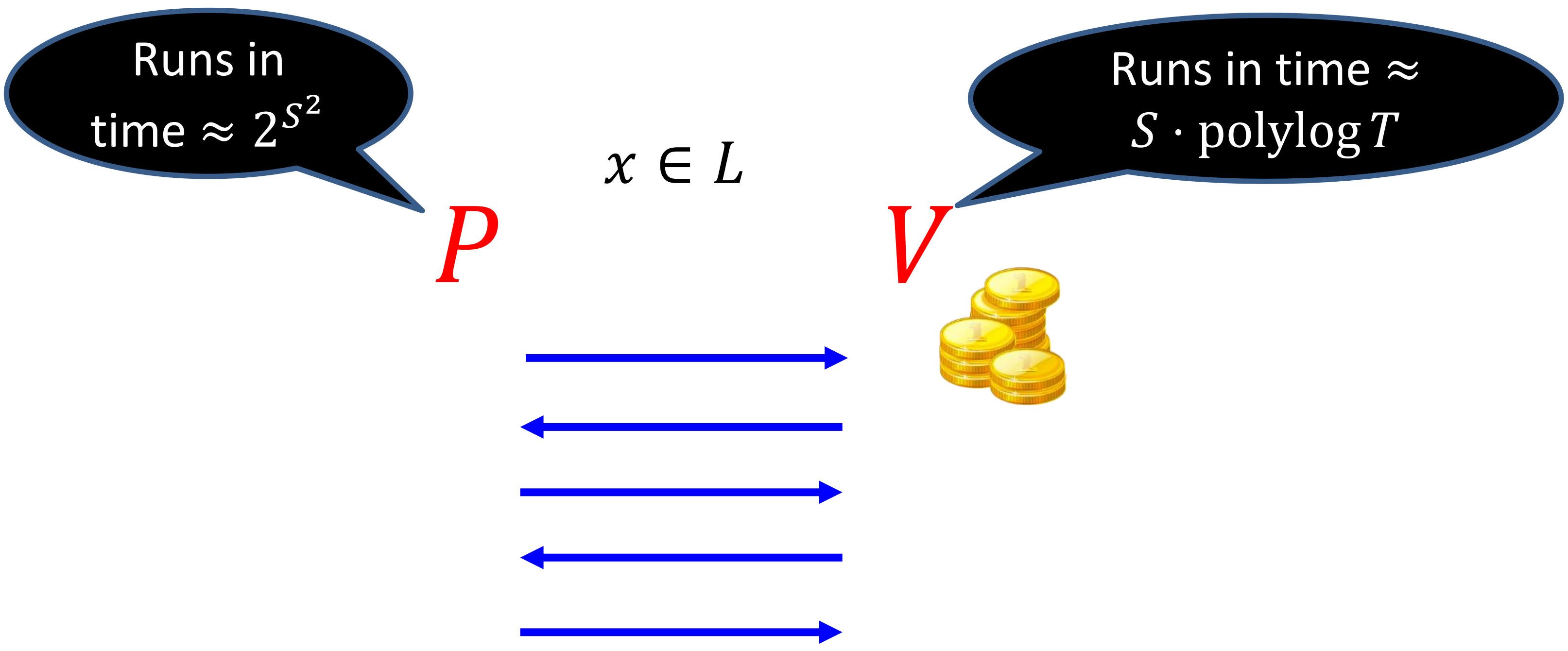
Space required to do the computation

Succinct space → **succinct interactive proof**

Interactive Proofs are More Efficient!

[Lund-Fortnow-Karloff-Nissan90, Shamir90]

Fix any language L computable in time T and space S

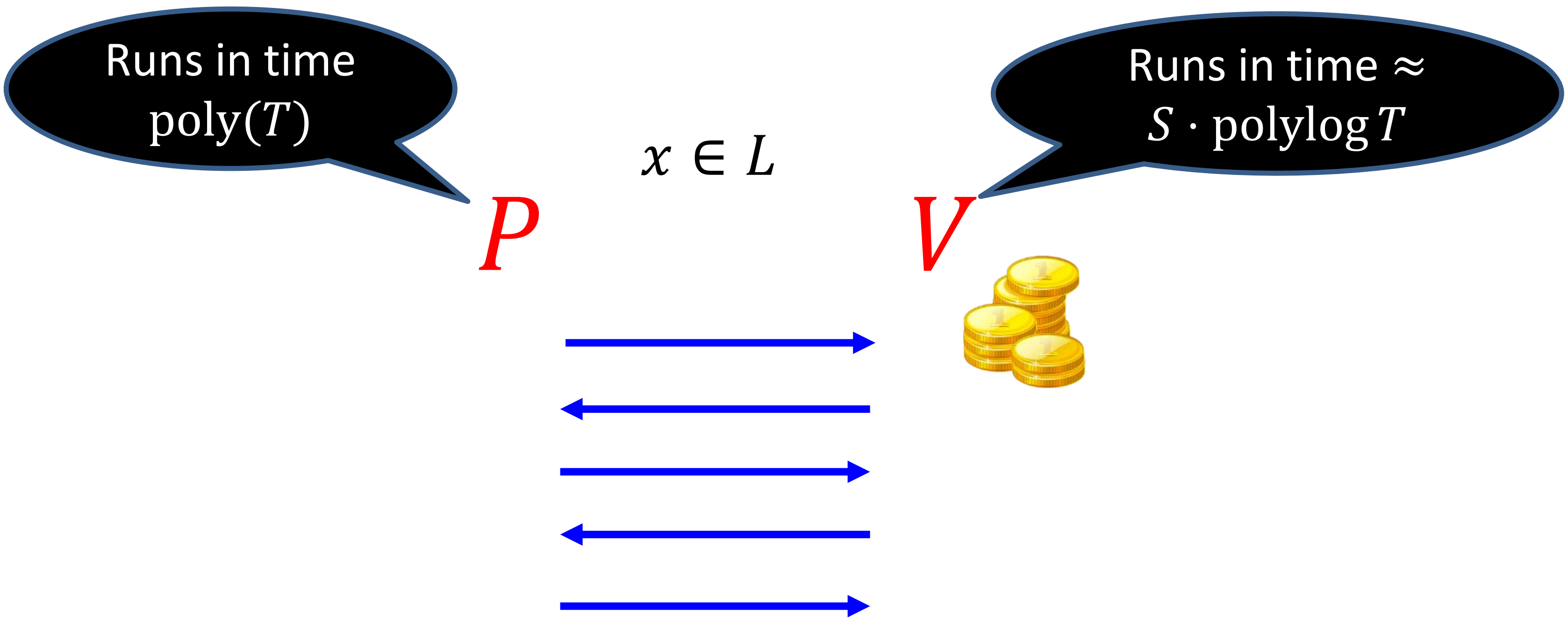


Open Problem:

Is proving harder than computing??

Does there exist an interactive proof for any time-T space-S computation where the verifier runs in time $\approx S \cdot polylog(T)$ and the prover runs in time $poly(T)$?

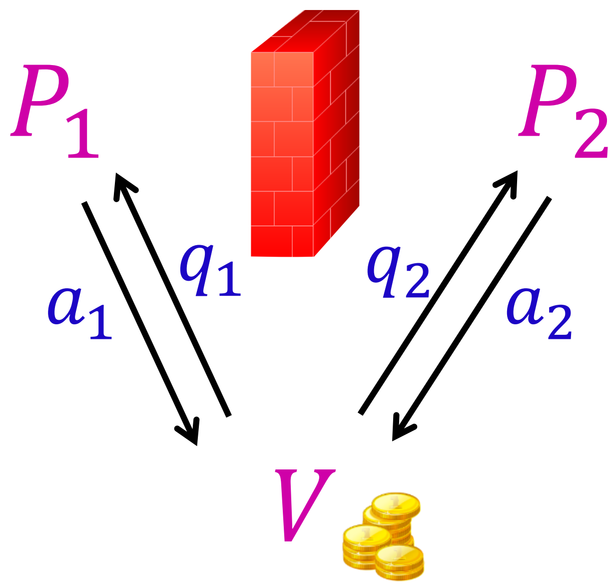
Open Problem:



Multi-Prover Interactive Proofs

[BenOr-Goldwasser-Kilian-Wigderson88]

motivated by constructing **perfect ZK proofs**



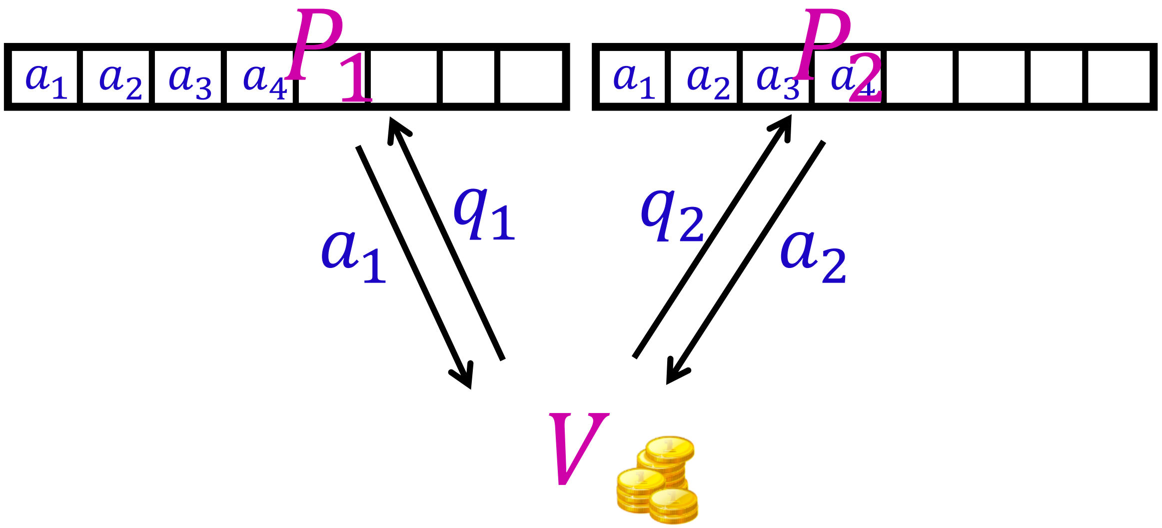
$\forall f$ computable in time T:

2-provers can convince verifier that $f(x) = y$,

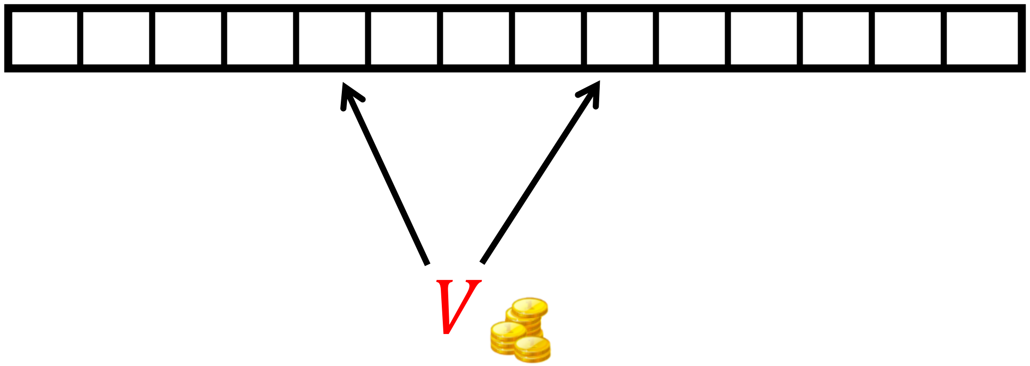
where the **runtime** of the **verifier** is only $|x| \cdot polylog(T)$

and the **communication** is $polylog(T)$

[Fortnow-Rompel-Sipser88]:



Probabilistically Checkable Proofs



[Feige-Goldwasser-Lovasz-Safra-Szegedy91, Babai-Fortnow-Levin-Szegedy91, Arora-Safra92, Arora-Lund-Mutwani-Sudan-Szegedy92]

Read only **3 bits** of the proof, and obtain soundness 1/8

Classical proofs

↓

(Zero-knowledge)
Interactive proofs

↓

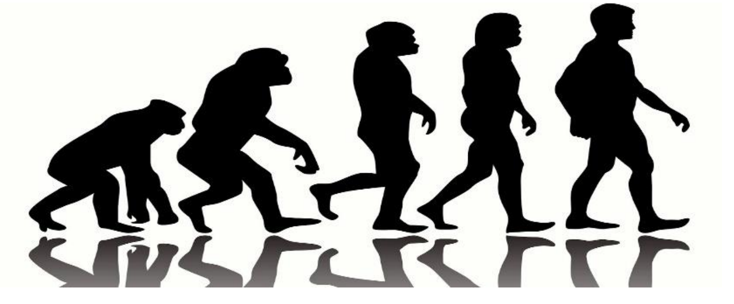
Multi-prover
interactive proofs

↓

Probabilistically
checkable proofs (PCPs)

↓

Succinct non-interactive arguments
(SNARGs)




THANK
YOU