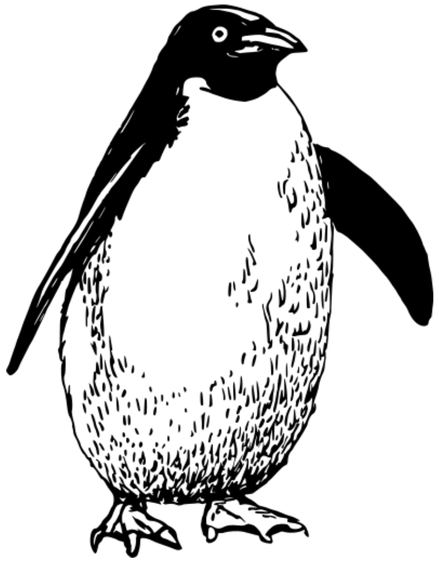
On Valiant's Conjecture

Eurocrypt 2023

Mathias Hall-Andersen, Jesper Buus Nielsen

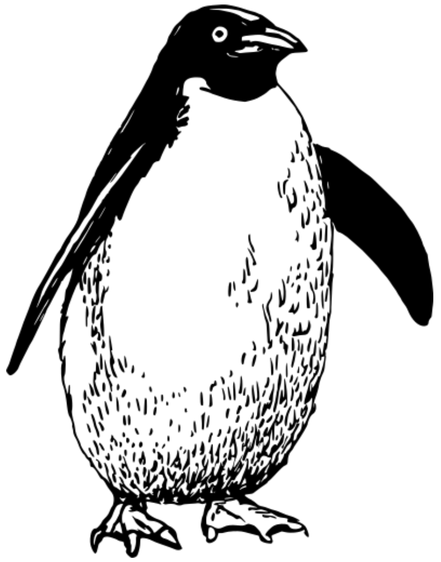
Incrementally Verifiable Computation

Incrementally Verifiable Computation

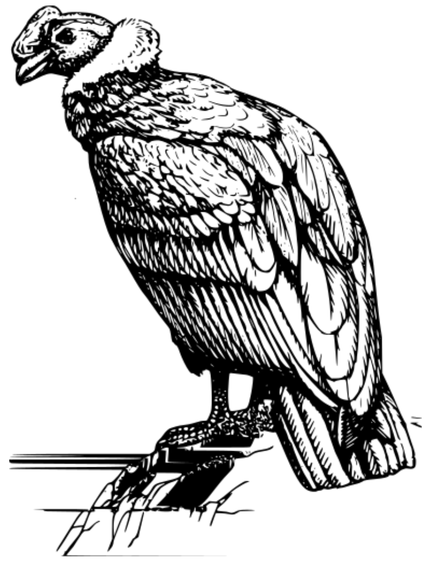


Prover Penguin

Incrementally Verifiable Computation



Prover Penguin

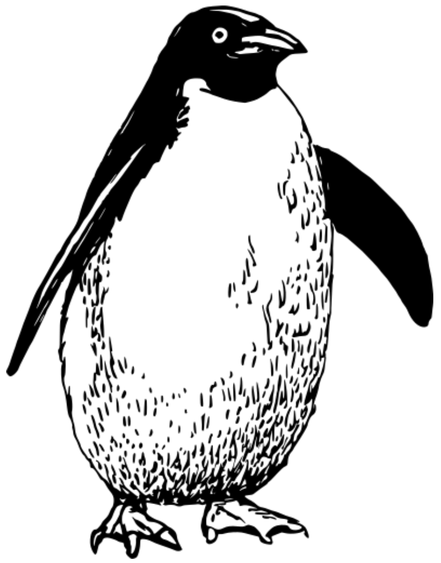


Verification Vulture

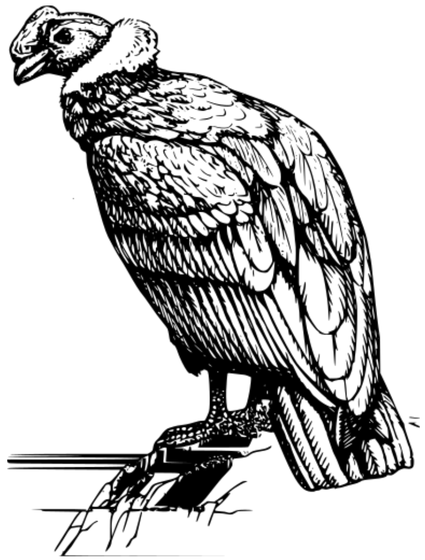
Incrementally Verifiable Computation

"I applied f to st_0 n times:"

$st_n = f^n(st_0)$



Prover Penguin

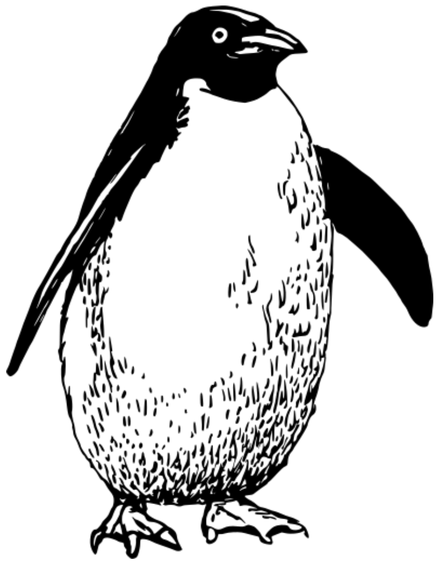


Verification Vulture

Incrementally Verifiable Computation

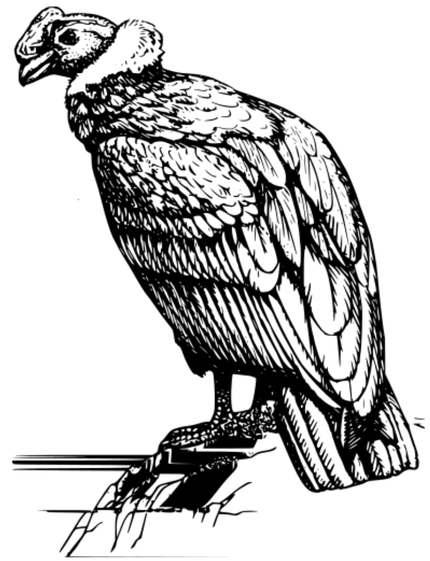
"I applied f to st_0 n times:"

$st_n = f^n(st_0)$



Prover Penguin

"Prove it!"

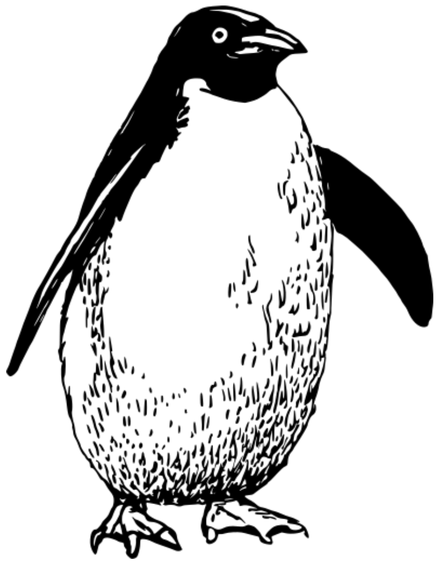


Verification Vulture

Incrementally Verifiable Computation

"I applied f to st_0 n times:"

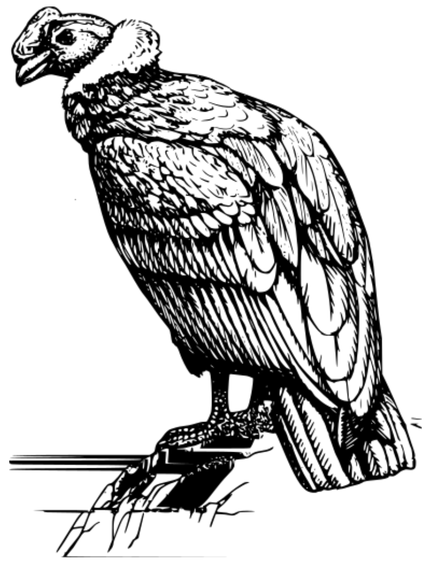
$st_n = f^n(st_0)$



Prover Penguin

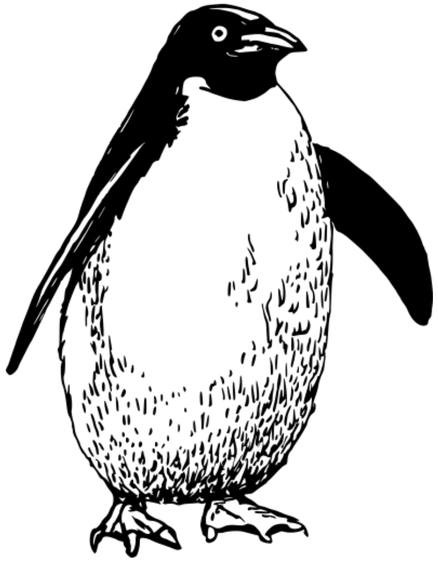
$st_n, \tilde{\pi}_n$

"Prove it!"

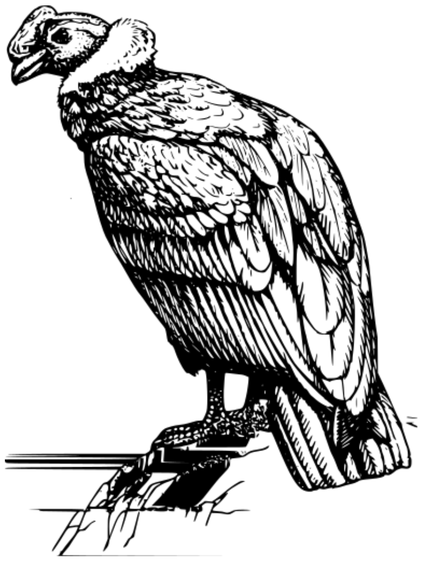


Verification Vulture

Incrementally Verifiable Computation



Prover Penguin

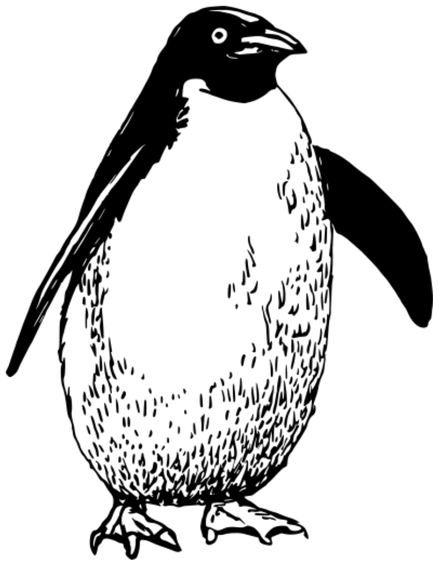


Verification Vulture

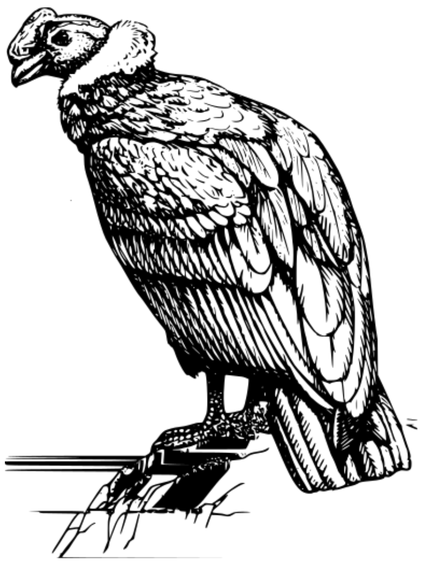
Incrementally Verifiable Computation

"I applied f to st_0 $n+1$ times"

$$st_{n+1} = f^{n+1}(st_0)$$



Prover Penguin

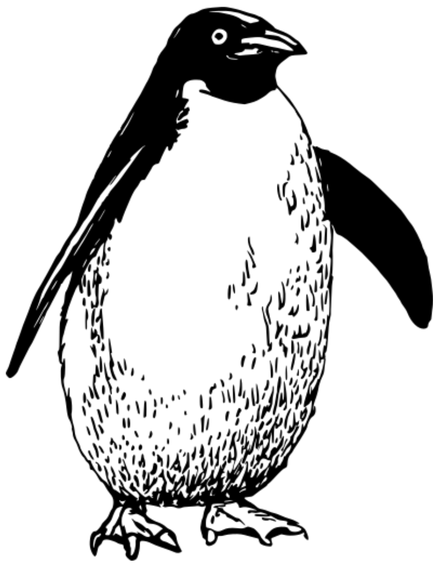


Verification Vulture

Incrementally Verifiable Computation

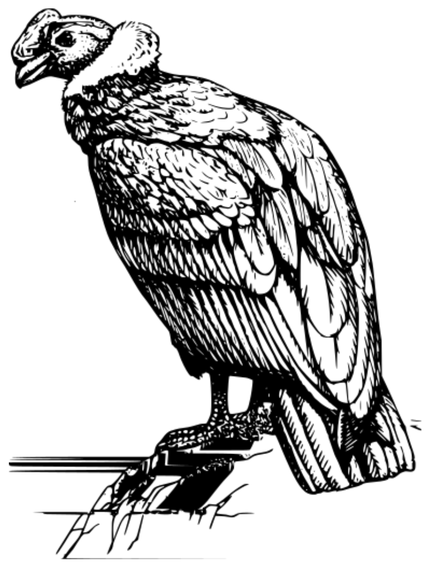
"I applied f to st_0 $n+1$ times"

$st_{n+1} = f^{n+1}(st_0)$



Prover Penguin

"Prove it!"



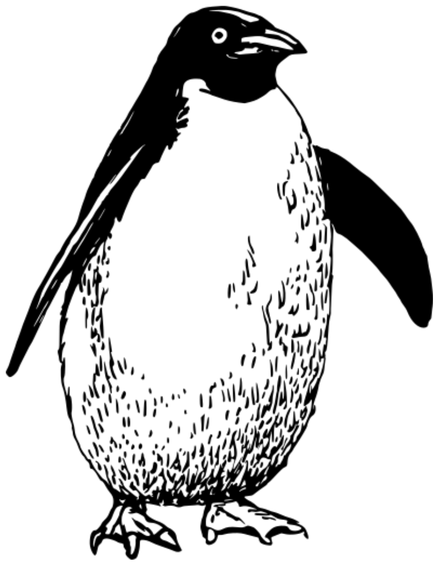
Verification Vulture

Incrementally Verifiable Computation

"I applied f to st_0 $n+1$ times"

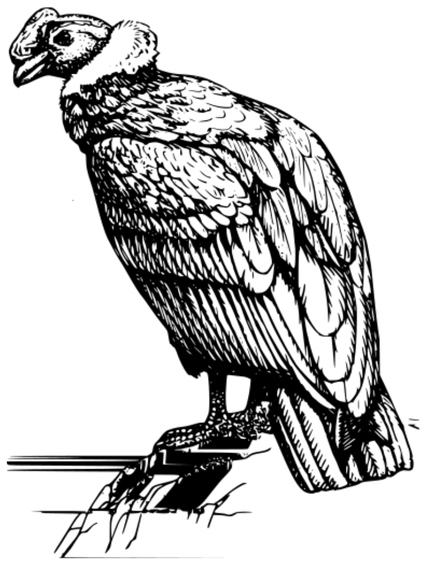
$st_{n+1} = f^{n+1}(st_0)$

"Prove it!"



Prover Penguin

st_{n+1}, π_{n+1}



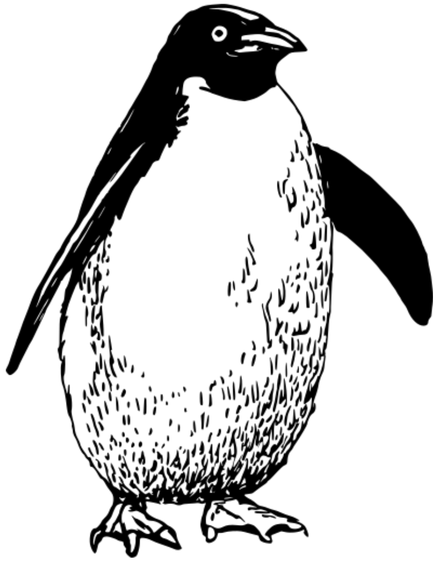
Verification Vulture

Incrementally Verifiable Computation

"I applied f to st_0 $n+1$ times"

$st_{n+1} = f^{n+1}(st_0)$

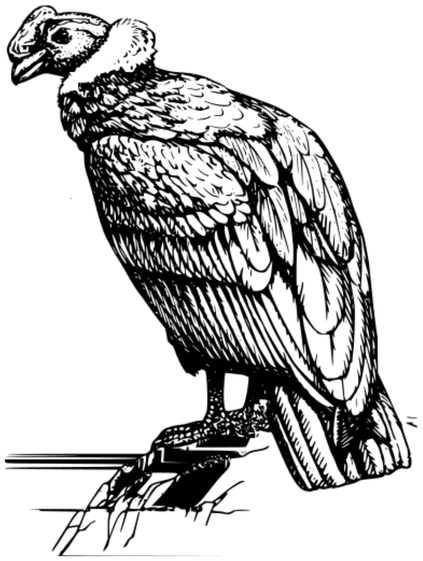
"Prove it!"



Prover Penguin

st_{n+1}, π_{n+1}

$st_{n+1} = f(st_n)$ computed from st_n.



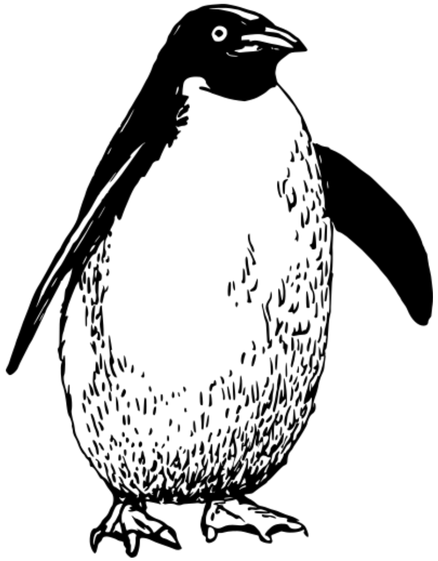
Verification Vulture

Incrementally Verifiable Computation

"I applied f to st_0 $n+1$ times"

$st_{n+1} = f^{n+1}(st_0)$

"Prove it!"



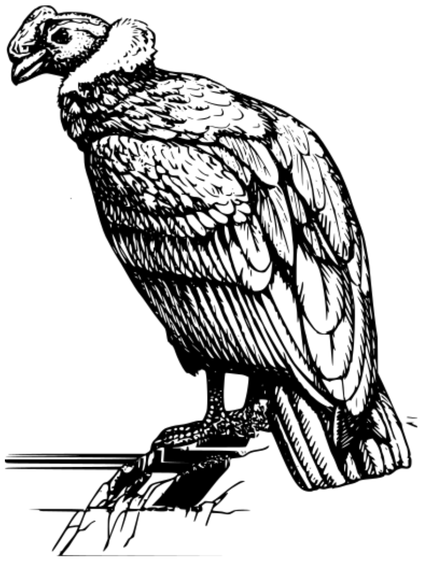
Prover Penguin

st_{n+1}, π_{n+1}

$st_{n+1} = f(st_n)$ computed from st_n.

What about computing π_{n+1} from π_n?

(with polylog(n) computation)



Verification Vulture

Incrementally Verifiable Computation

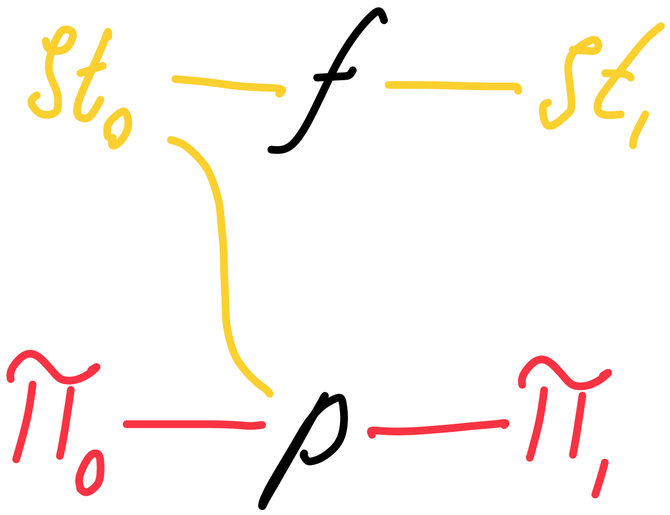
Incrementally Verifiable Computation

st_0

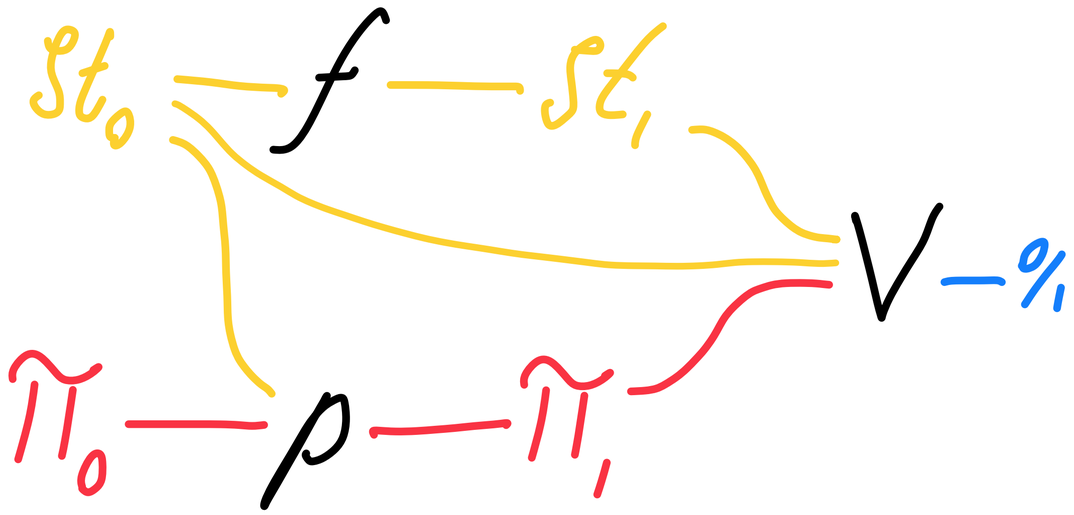
$\tilde{\pi}_0$

Incrementally Verifiable Computation

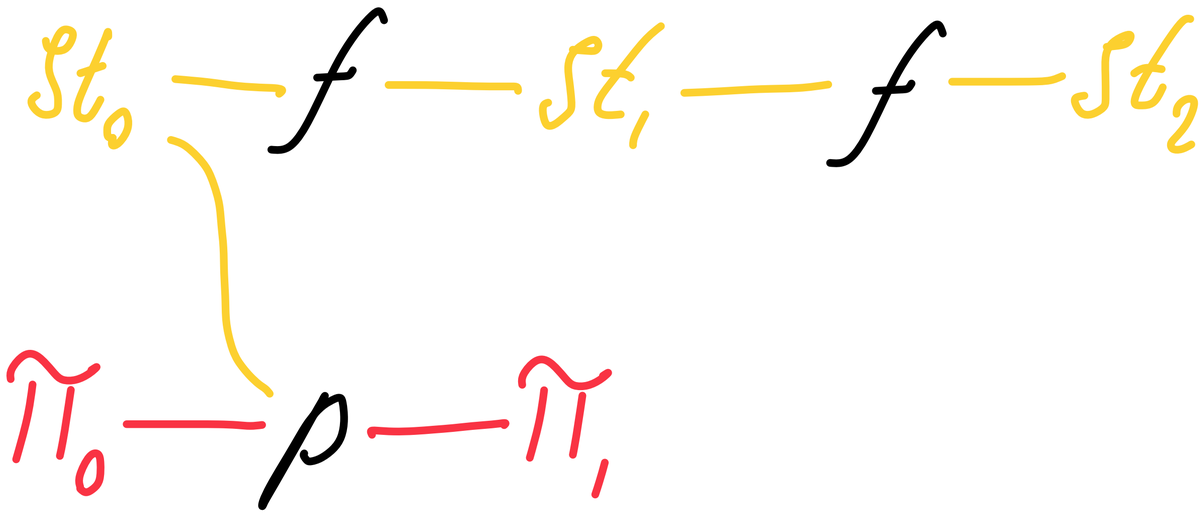
Incrementally Verifiable Computation



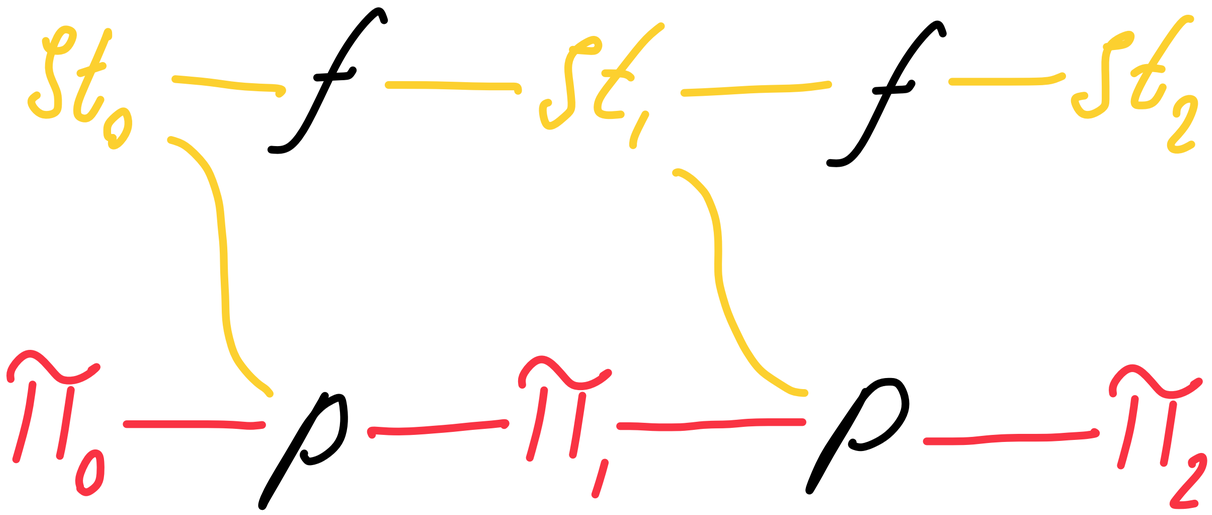
Incrementally Verifiable Computation



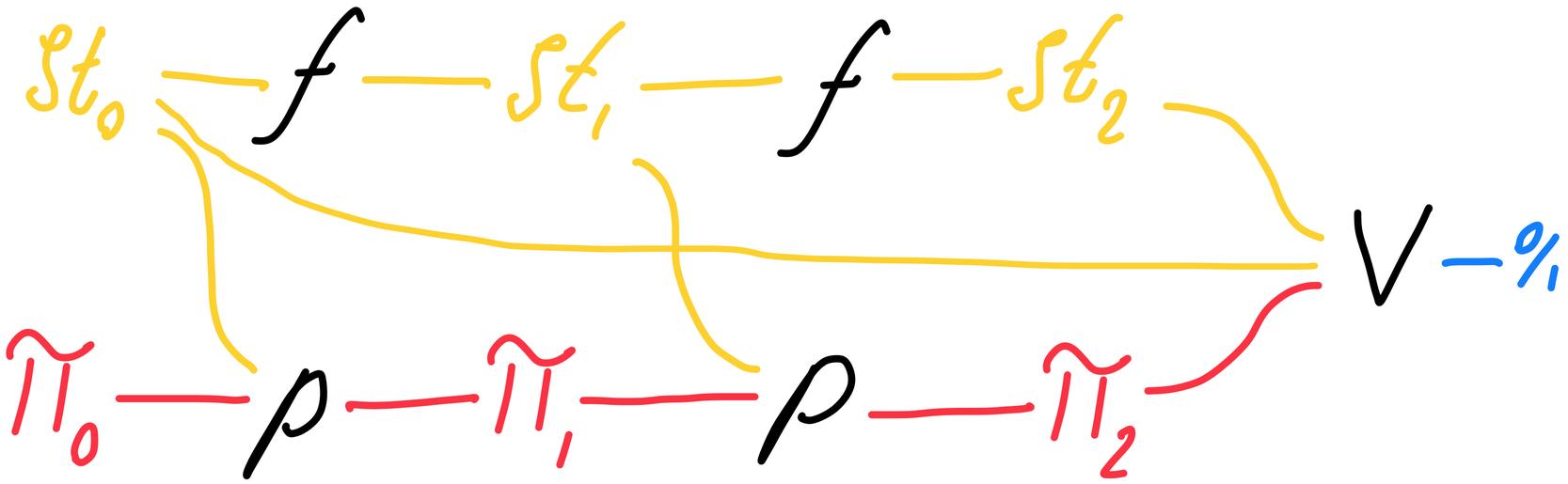
Incrementally Verifiable Computation



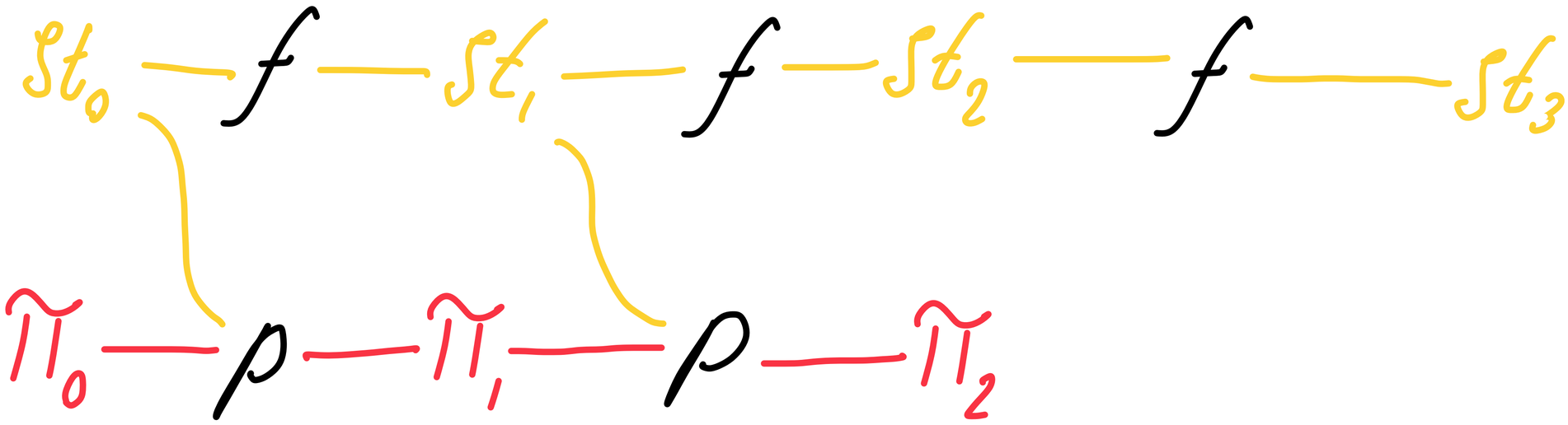
Incrementally Verifiable Computation



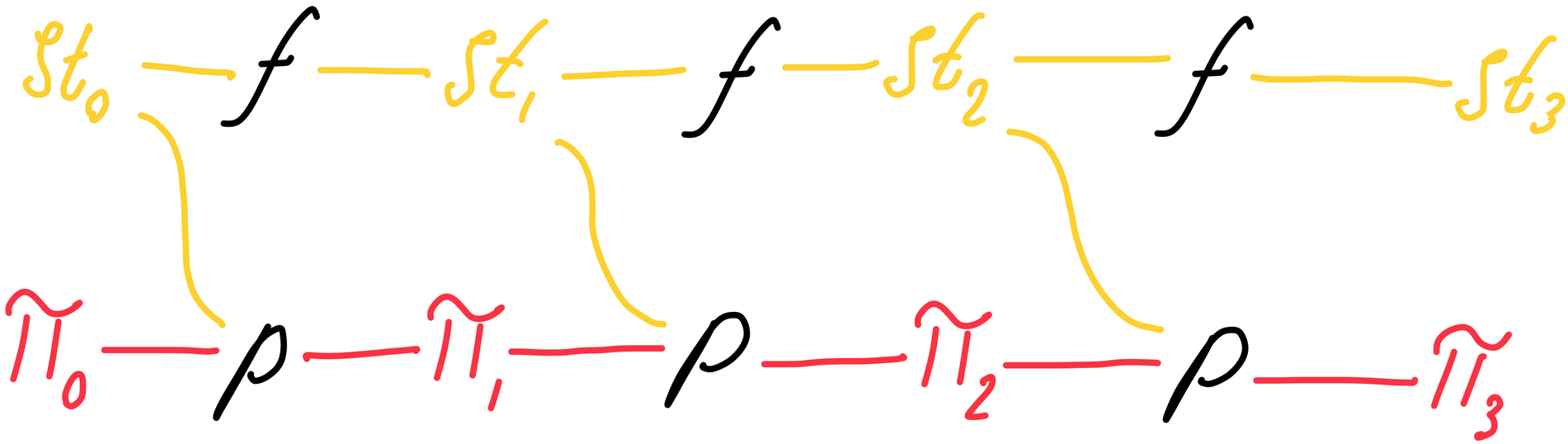
Incrementally Verifiable Computation



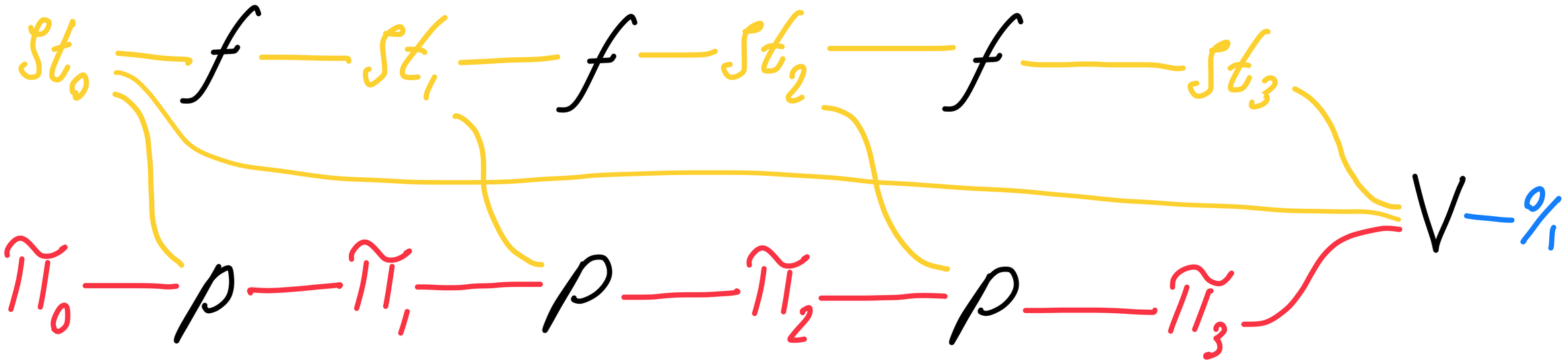
Incrementally Verifiable Computation



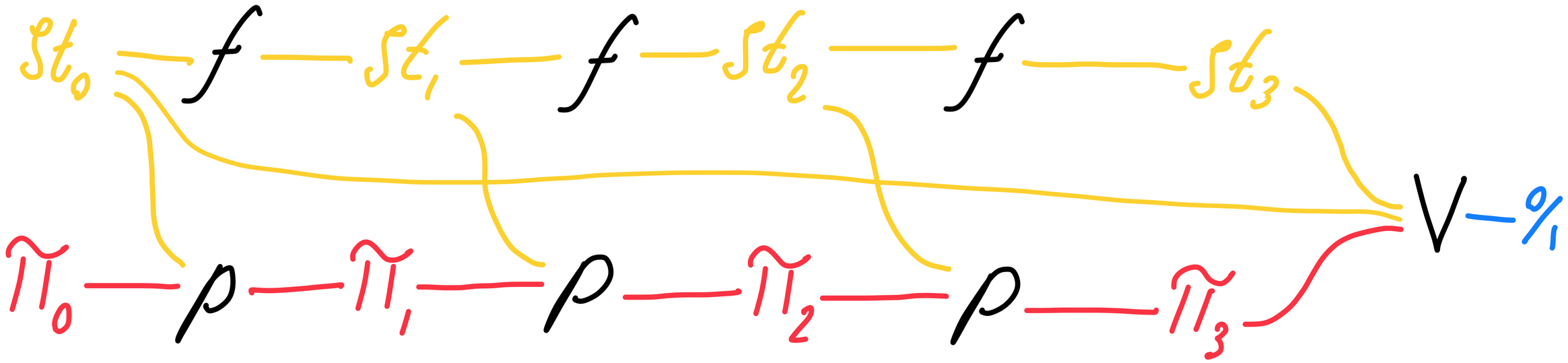
Incrementally Verifiable Computation



Incrementally Verifiable Computation

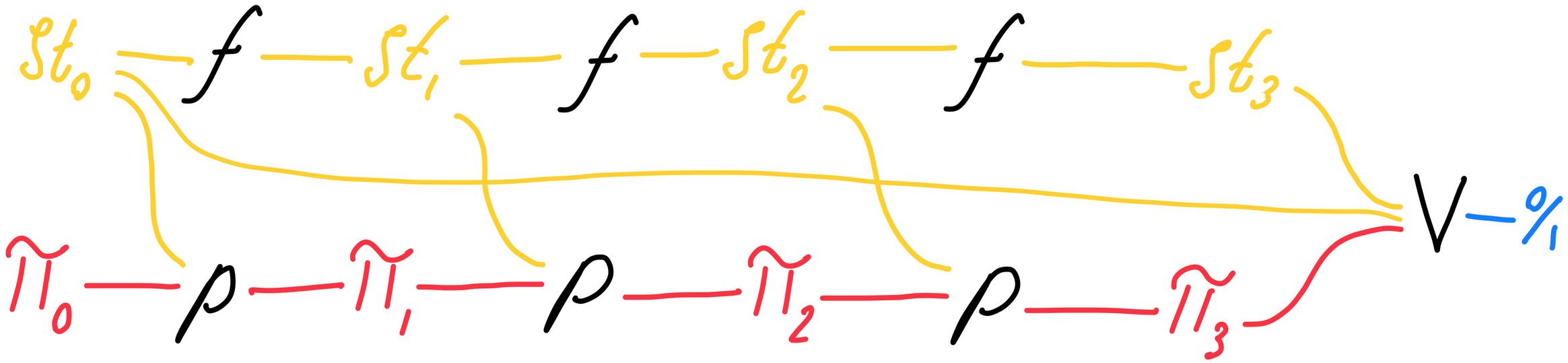


Incrementally Verifiable Computation



P Complexity: O(poly(|f|, log(n)))

Incrementally Verifiable Computation

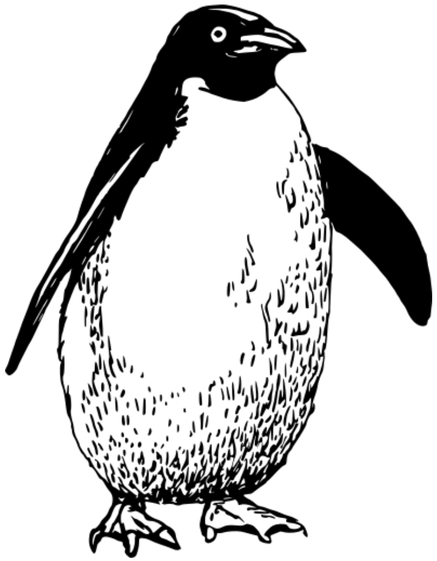


P Complexity: O(poly(|f|, log(n)))

V Complexity: O(poly(|f|, log(n)))

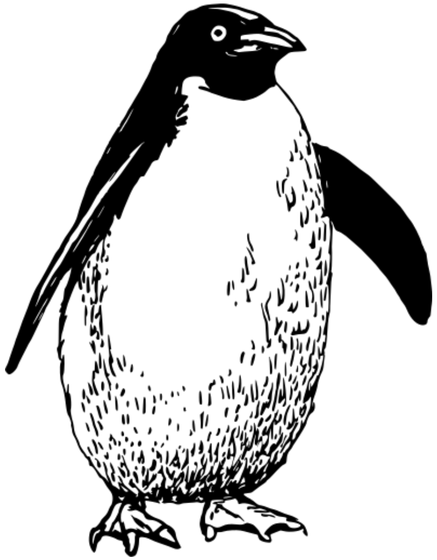
The Random Oracle Model

The Random Oracle Model

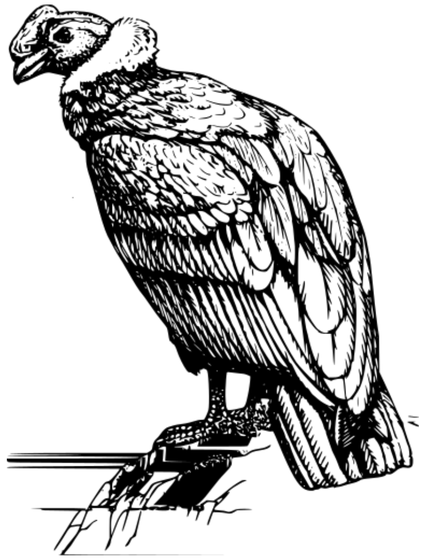


Prover Penguin

The Random Oracle Model



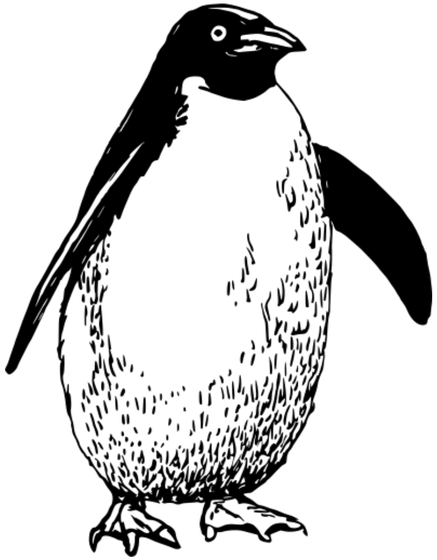
Prover Penguin



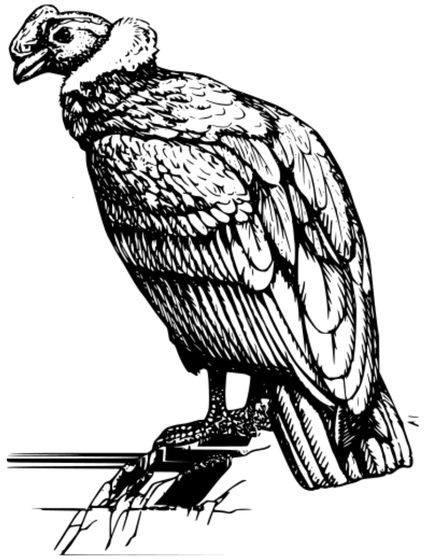
Verification Vulture

The Random Oracle Model

Random Oracle

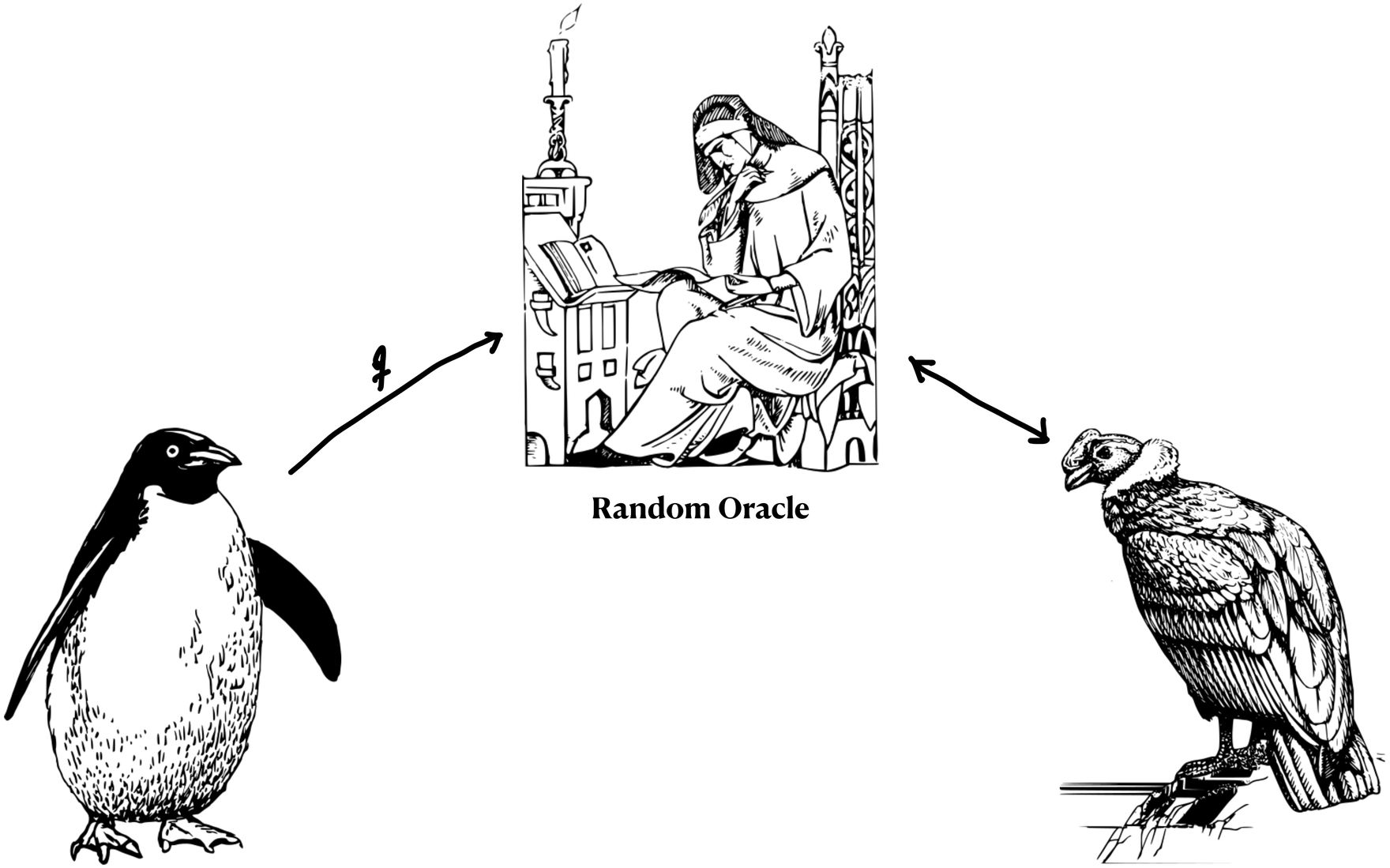


Prover Penguin



Verification Vulture

The Random Oracle Model

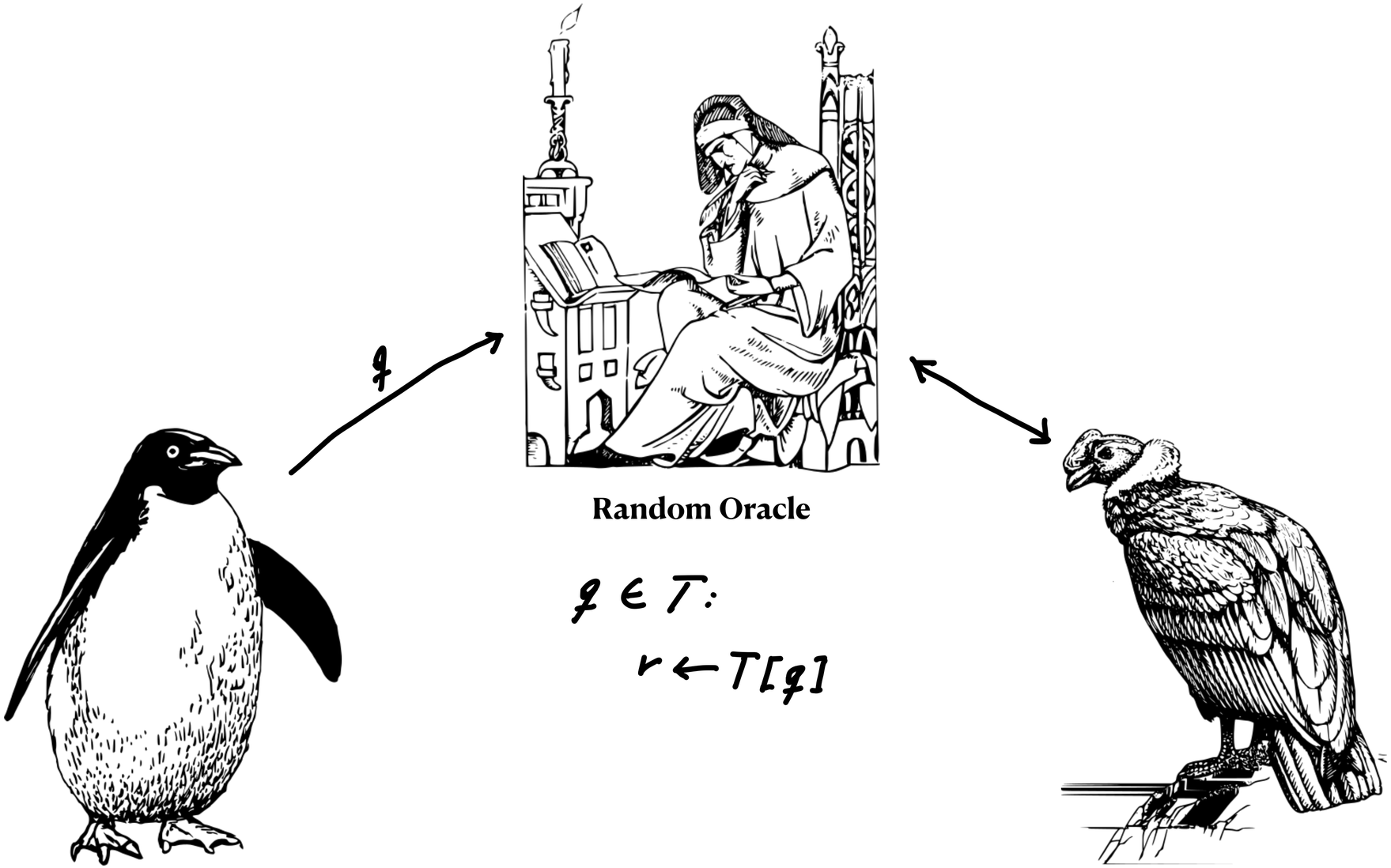


Random Oracle

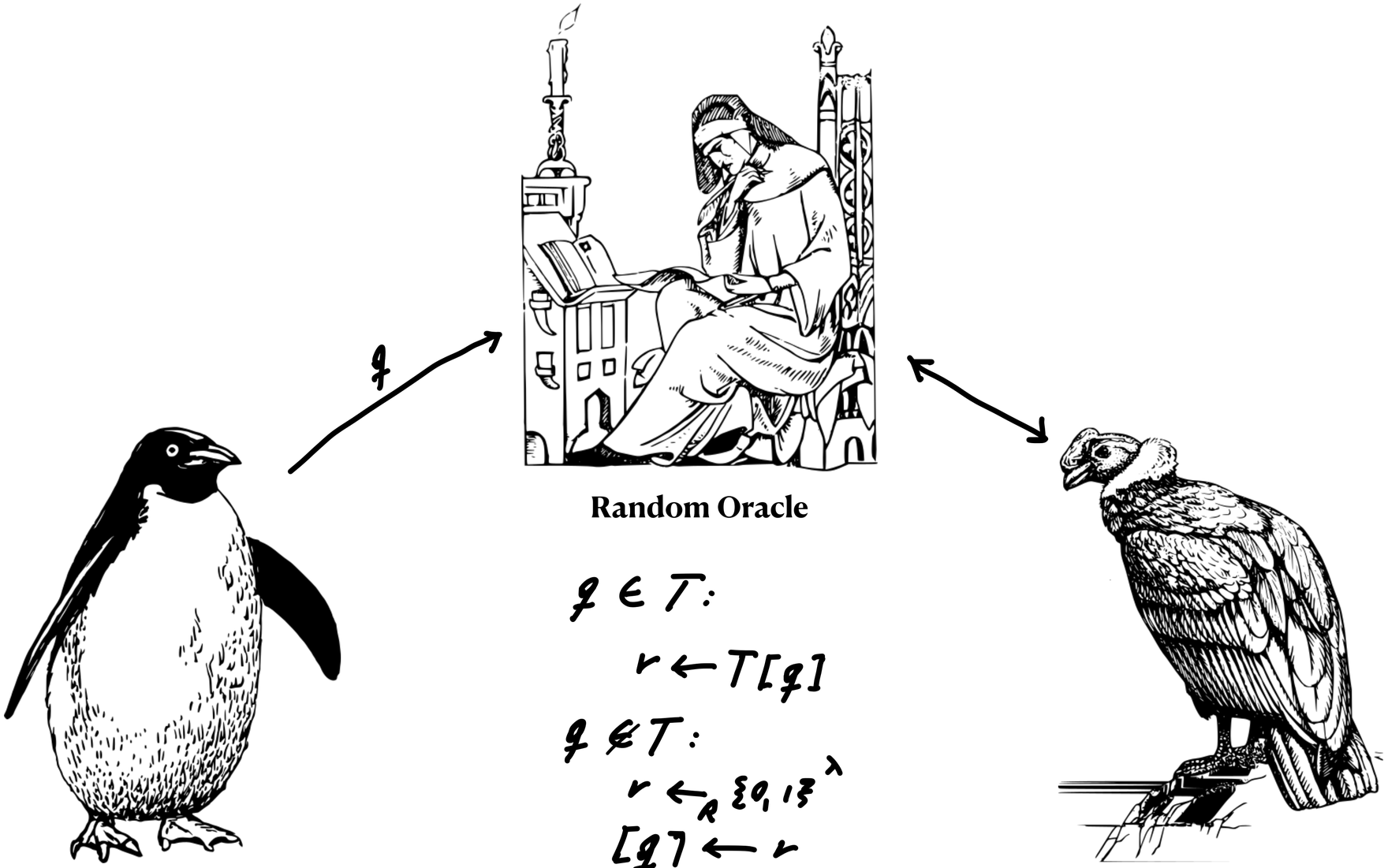
Prover Penguin

Verification Vulture

The Random Oracle Model



The Random Oracle Model



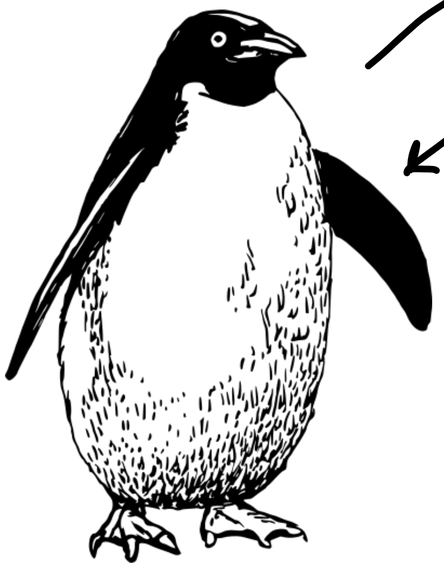
Random Oracle

Prover Penguin

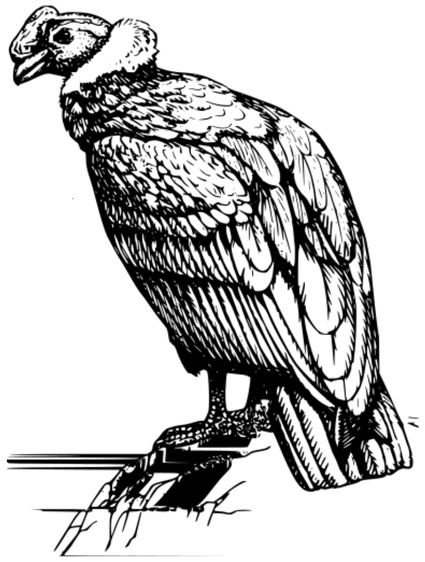
Verification Vulture

The Random Oracle Model

Random Oracle



Prover Penguin



Verification Vulture

q

r

$q \in T$:

$r \leftarrow T[q]$

$q \notin T$:

$r \leftarrow_R \{0,1\}^\lambda$

$[q] \leftarrow r$

Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs

Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs

st_0

Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs

Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs

Valiant's Construction of IVC "in the ROM"

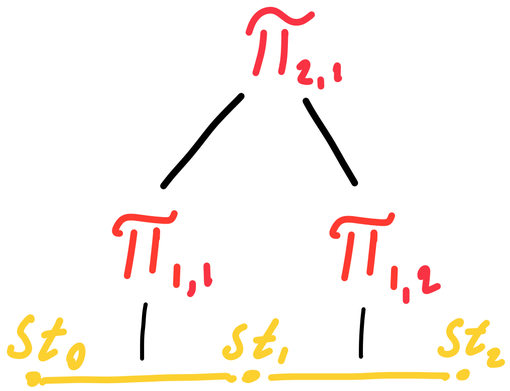
Valiant's Construction [2008] IVC From CS Proofs

Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs

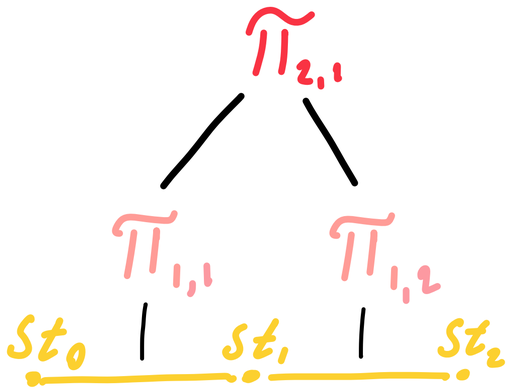
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



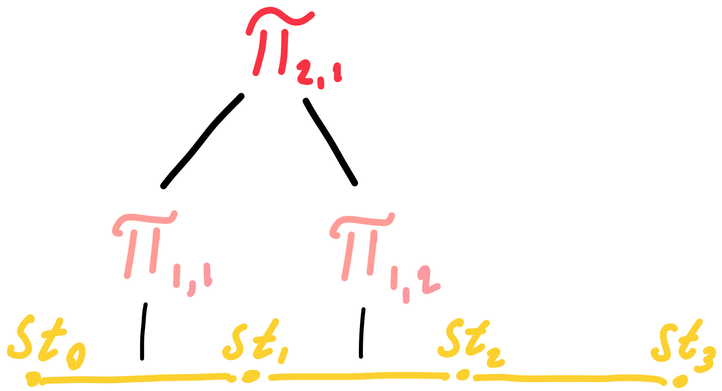
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



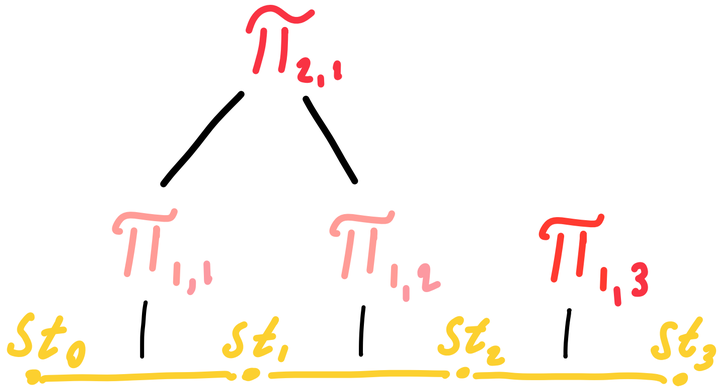
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



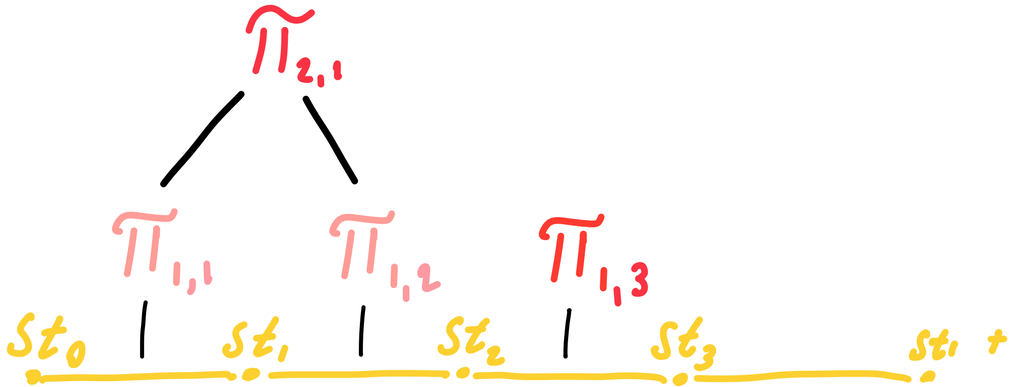
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



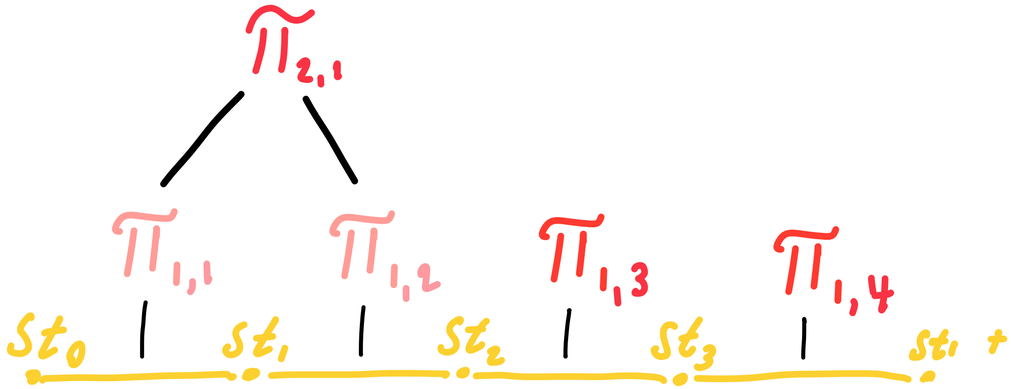
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



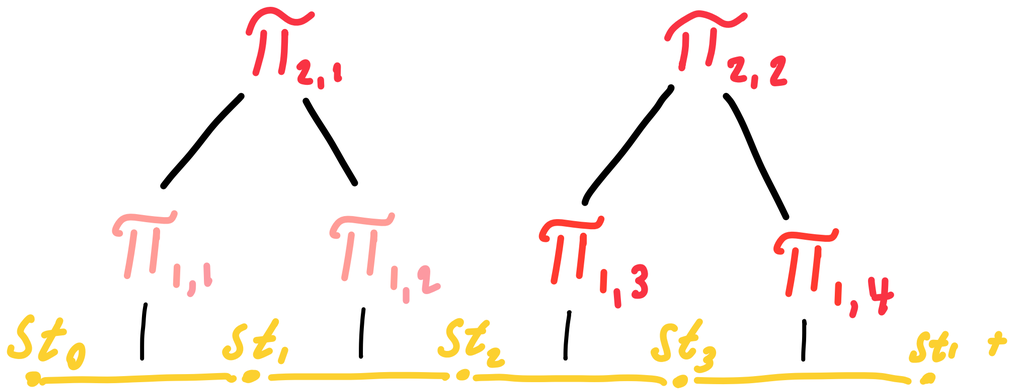
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



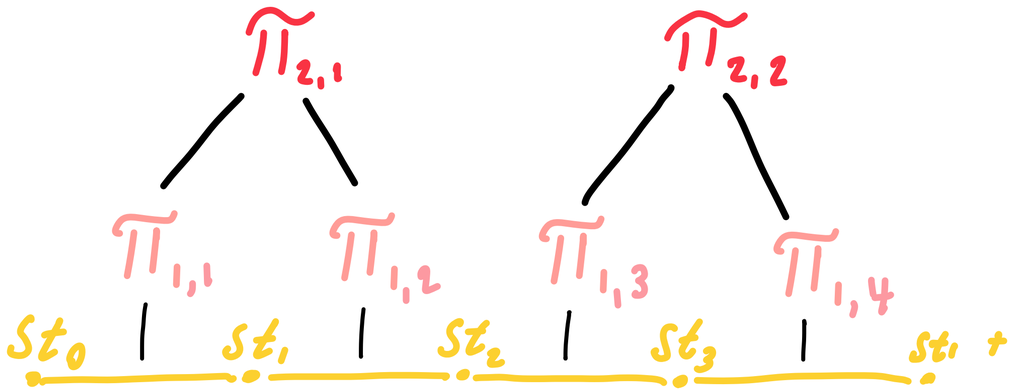
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



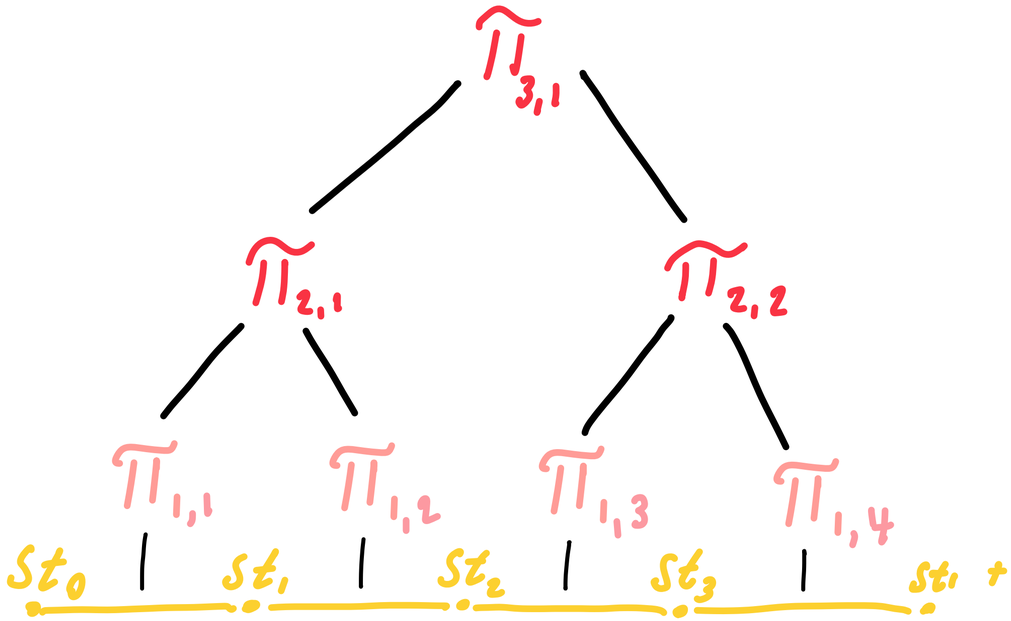
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



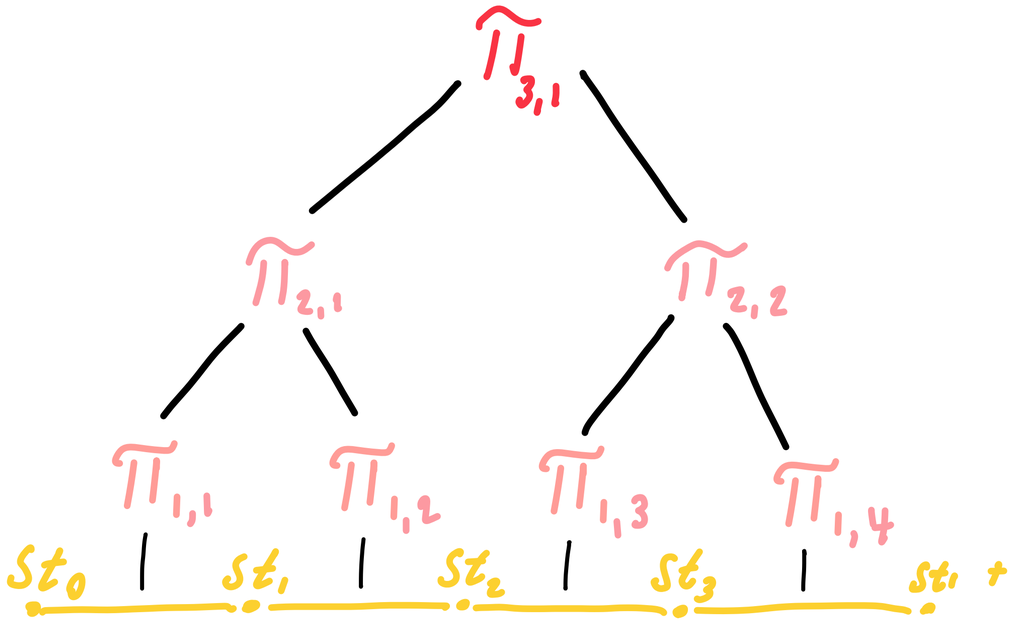
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



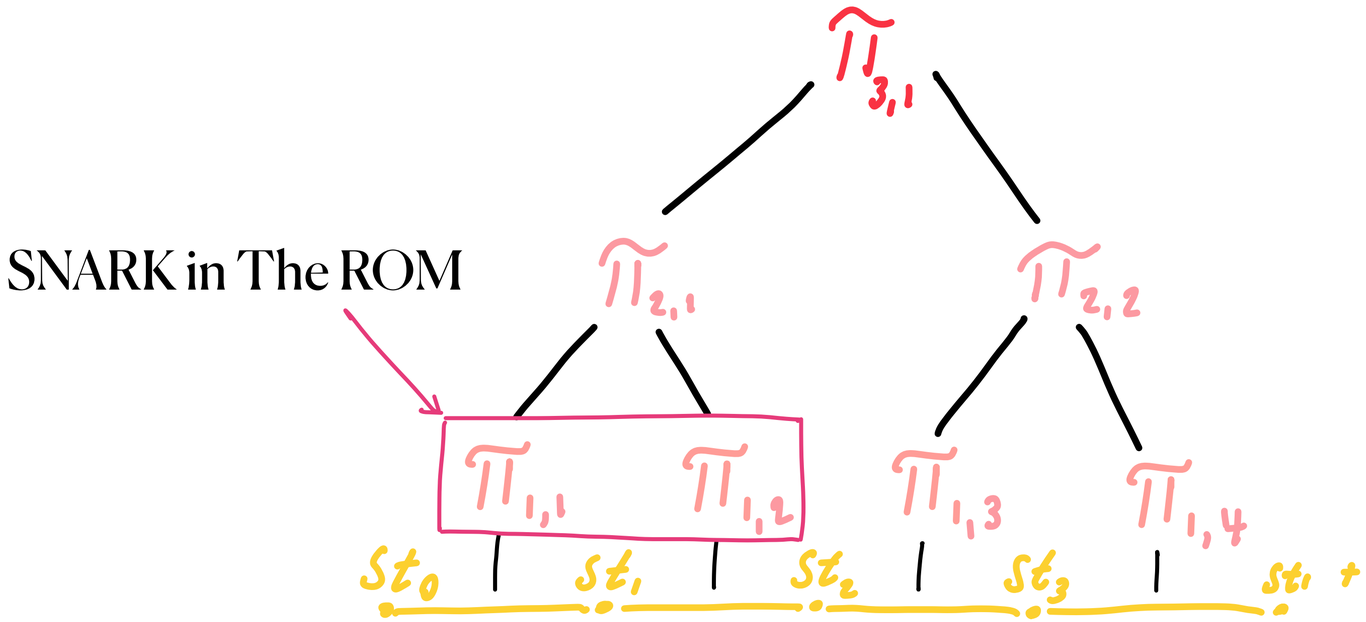
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



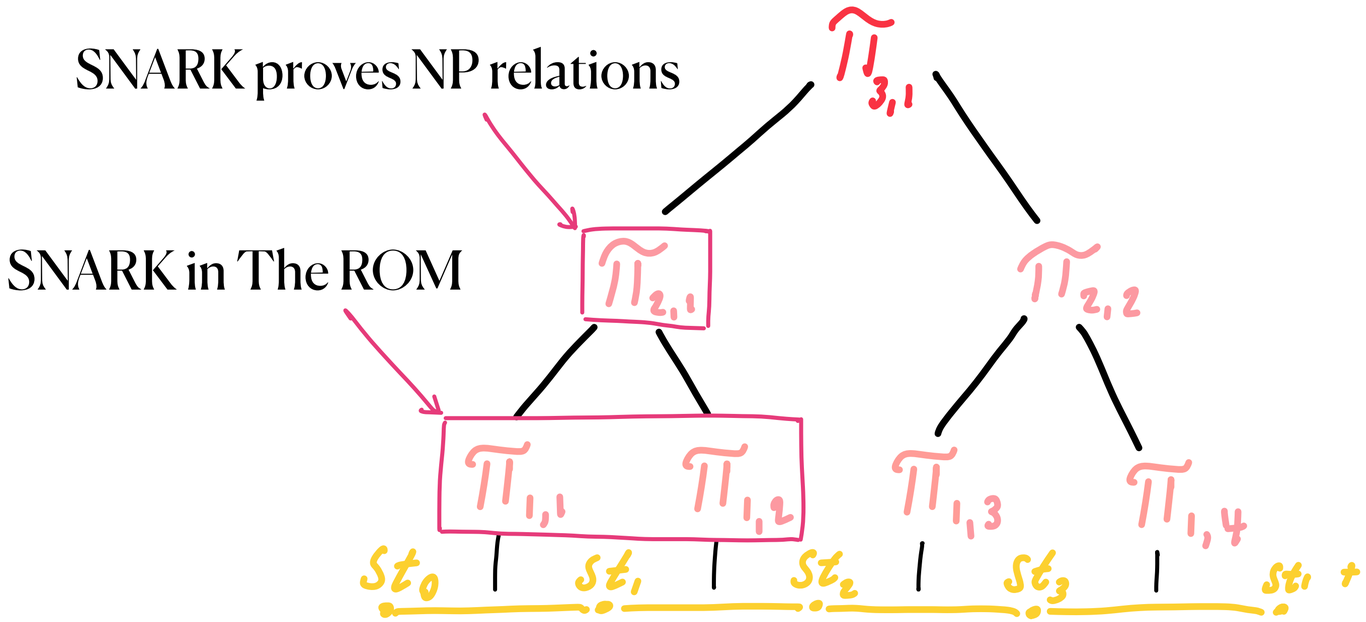
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



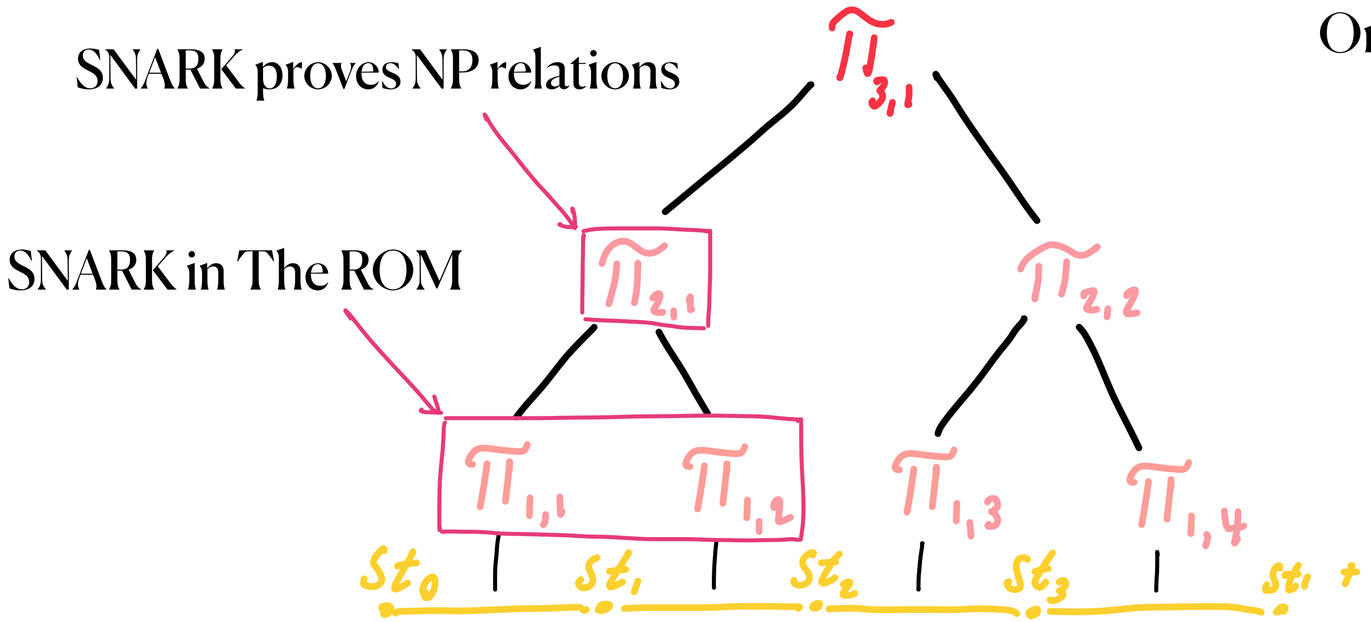
Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



Valiant's Construction of IVC "in the ROM"

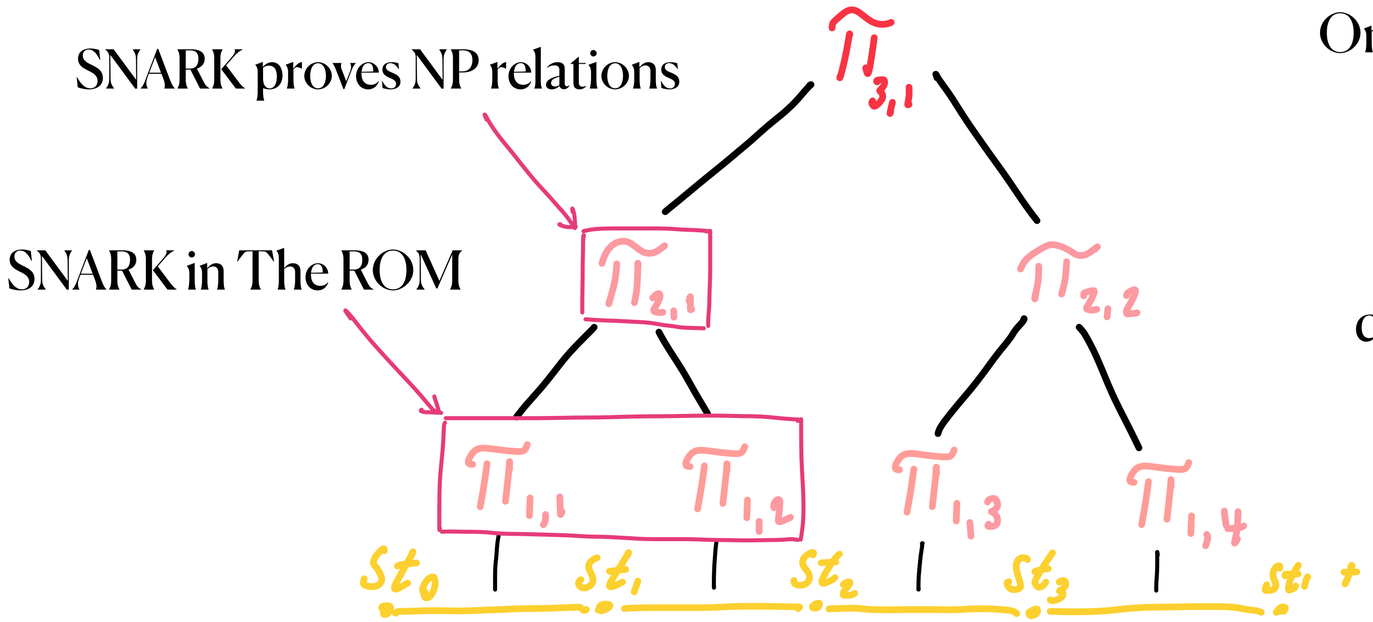
Valiant's Construction [2008] IVC From CS Proofs



Only works when heuristically "instantiating" the ROM

Valiant's Construction of IVC "in the ROM"

Valiant's Construction [2008] IVC From CS Proofs



Only works when heuristically "instantiating" the ROM

The construction does not work in the ROM

Of Course Valiant Noted This.

"... recursion breaks down because, even at the first level of recursion, we are no longer trying to prove statements about classical computation but rather statements of the form "M with oracle access to O accepts the following string..." *Thus standard applications of random oracles do not appear to help*.

– Paul Valiant

"... recursion breaks down because, even at the first level of recursion, we are no longer trying to prove statements about classical computation but rather statements of the form "M with oracle access to O accepts the following string..." *Thus standard applications of random oracles do not appear to help*.

– Paul Valiant

Can we prove this intuition?

Overview of Results

IVC to Non-Deterministic IVC

IVC to Non-Deterministic IVC

Ideally: "IVC is Impossible in the ROM"

IVC to Non-Deterministic IVC

Ideally: "IVC is Impossible in the ROM"

Problem: Hard to Even Rule Out Trivial Schemes.

IVC to Non-Deterministic IVC

Ideally: "IVC is Impossible in the ROM"

Problem: Hard to Even Rule Out Trivial Schemes.

Constructions with CRS: Devadas, Goyal, Kalai and Vaikuntanathan [2022]

IVC to Non-Deterministic IVC

Ideally: "IVC is Impossible in the ROM"

Problem: Hard to Even Rule Out Trivial Schemes.

Constructions with CRS: Devadas, Goyal, Kalai and Vaikuntanathan [2022]

Introduce a Witness:

IVC to Non-Deterministic IVC

Ideally: "IVC is Impossible in the ROM"

Problem: Hard to Even Rule Out Trivial Schemes.

Constructions with CRS: Devadas, Goyal, Kalai and Vaikuntanathan [2022]

Introduce a Witness: "RO Does Not Help Construct non-Deterministic IVC"

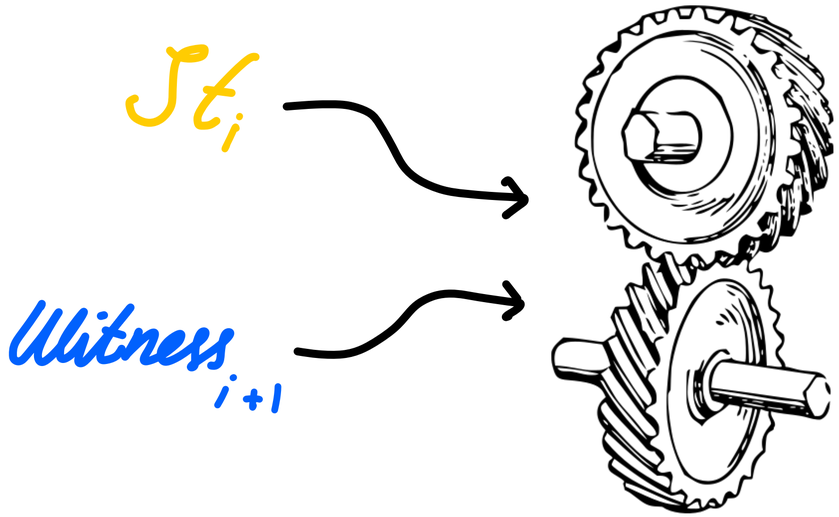
IVC to Non-Deterministic IVC

Ideally: "IVC is Impossible in the ROM"

Problem: Hard to Even Rule Out Trivial Schemes.

Constructions with CRS: Devadas, Goyal, Kalai and Vaikuntanathan [2022]

Introduce a Witness: "RO Does Not Help Construct non-Deterministic IVC"



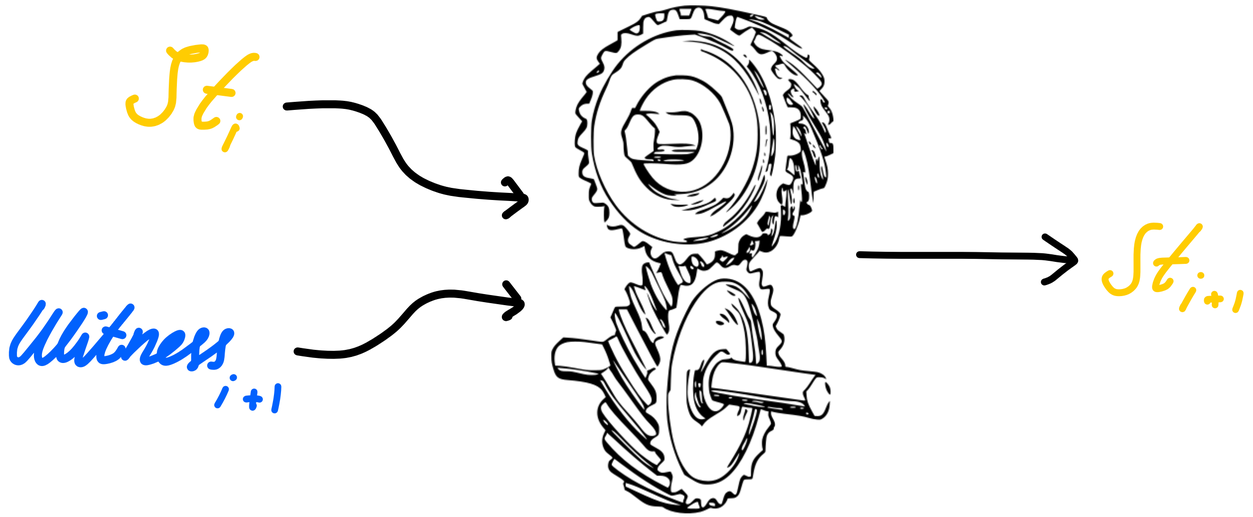
IVC to Non-Deterministic IVC

Ideally: "IVC is Impossible in the ROM"

Problem: Hard to Even Rule Out Trivial Schemes.

Constructions with CRS: Devadas, Goyal, Kalai and Vaikuntanathan [2022]

Introduce a Witness: "RO Does Not Help Construct non-Deterministic IVC"



Non-Deterministic IVC Interesting?

Non-Deterministic IVC Interesting?

- **Covers PCD**: Setting is A Special Case of PCD in The ROM.

Non-Deterministic IVC Interesting?

- **Covers PCD**: Setting is A Special Case of PCD in The ROM.

- **Natural Schemes**: Valiant's Scheme Extended to Non-Deterministic IVC.

Non-Deterministic IVC Interesting?

- **Covers PCD**: Setting is A Special Case of PCD in The ROM.
- **Natural Schemes**: Valiant's Scheme Extended to Non-Deterministic IVC.
- **Justifying New Models**: Contrasts Positive Results in Related Idealised Models.

Non-Deterministic IVC Interesting?

- **Covers PCD**: Setting is A Special Case of PCD in The ROM.
- **Natural Schemes**: Valiant's Scheme Extended to Non-Deterministic IVC.
- **Justifying New Models**: Contrasts Positive Results in Related Idealised Models.
  - zk-SNARKs and PCD for Low-Degree ROM (LDROM) [CCS2022]
  - PCD from AROM [CCS2022]

    (No Such Hope for Regular ROM)

Non-Deterministic IVC Interesting?

- **Covers PCD**: Setting is A Special Case of PCD in The ROM.
- **Natural Schemes**: Valiant's Scheme Extended to Non-Deterministic IVC.
- **Justifying New Models**: Contrasts Positive Results in Related Idealised Models.
  - zk-SNARKs and PCD for Low-Degree ROM (LDROM) [CCS2022]
  - PCD from AROM [CCS2022]

    (No Such Hope for Regular ROM)
- **Contrast with SNARKs in ROM:** Proving Incrementally is Harder.

Overview

Of All The Caveats

Non-Deterministic IVC.

Non-Triviality: Accepting Proofs (for true statements)
can be Generated by Programming the RO

Intuition: Soundness Dependent on RO
Otherwise: a Scheme without ROM

Zero-Knowledge Impossible

Without ZK: Schemes of Certain Structure

(Blackbox Knowledge Extractor)

Overview

Of All The Caveats

Non-Deterministic IVC.

Non-Triviality: Accepting Proofs (for true statements)
can be Generated by Programming the RO

Intuition: Soundness Dependent on RO
Otherwise: a Scheme without ROM

Zero-Knowledge Impossible

Without ZK: Schemes of Certain Structure

(Blackbox Knowledge Extractor)

Overview

Of All The Caveats

Non-Deterministic IVC.

Non-Triviality: Accepting Proofs (for true statements)
can be Generated by Programming the RO

Intuition: Soundness Dependent on RO
Otherwise: a Scheme without ROM

Zero-Knowledge Impossible

Without ZK: Schemes of Certain Structure

(Blackbox Knowledge Extractor)

Takeaway: Adding a RO Does Not Help.

Overview

Of All The Caveats

Non-Deterministic IVC.

Non-Triviality: Accepting Proofs (for true statements)
can be Generated by Programming the RO

Intuition: Soundness Dependent on RO
Otherwise: a Scheme without ROM

Zero-Knowledge Impossible

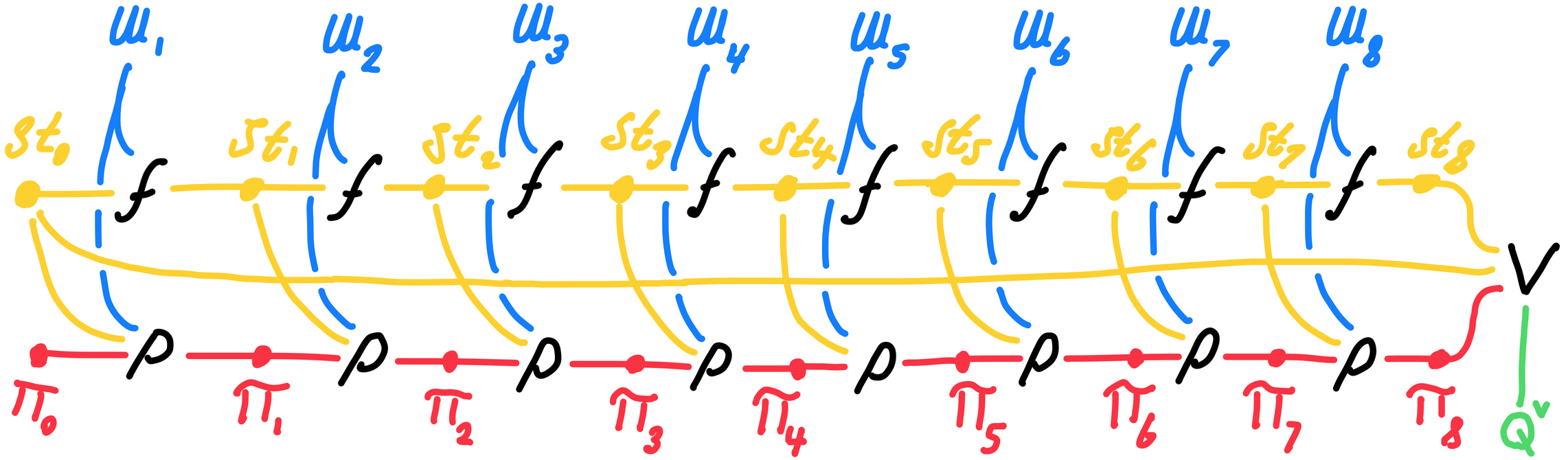
Without ZK: Schemes of Certain Structure

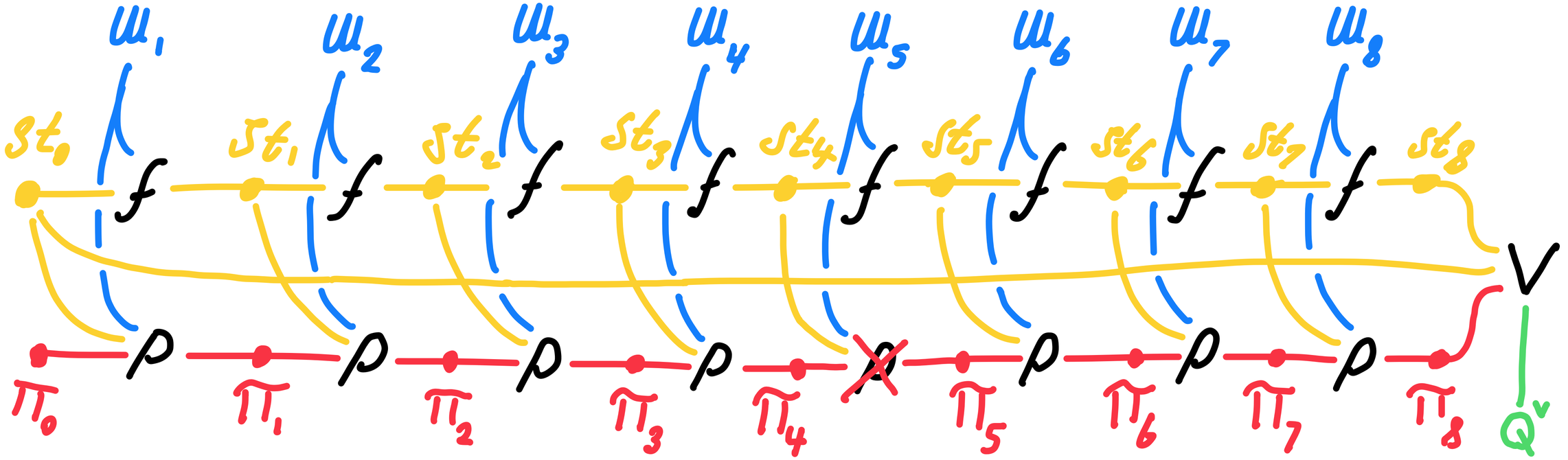
(Blackbox Knowledge Extractor)

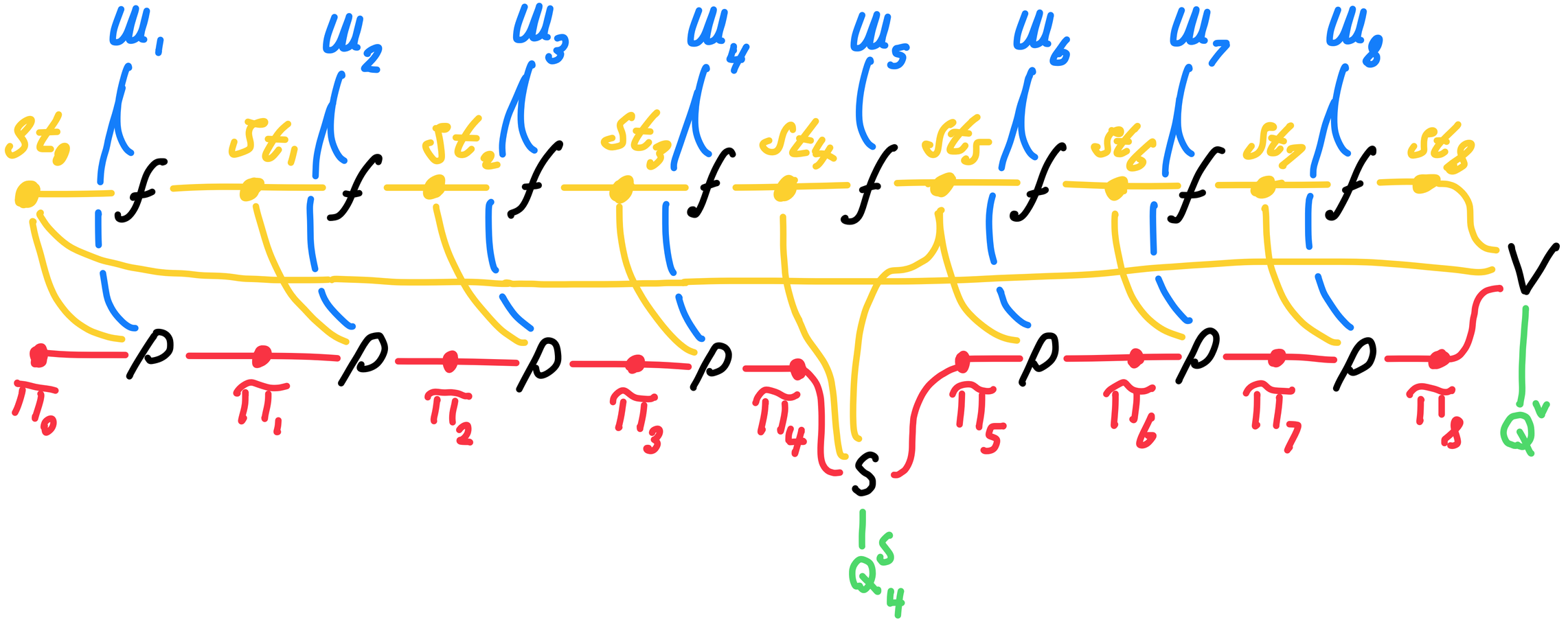
Takeaway: Adding a RO Does Not Help.

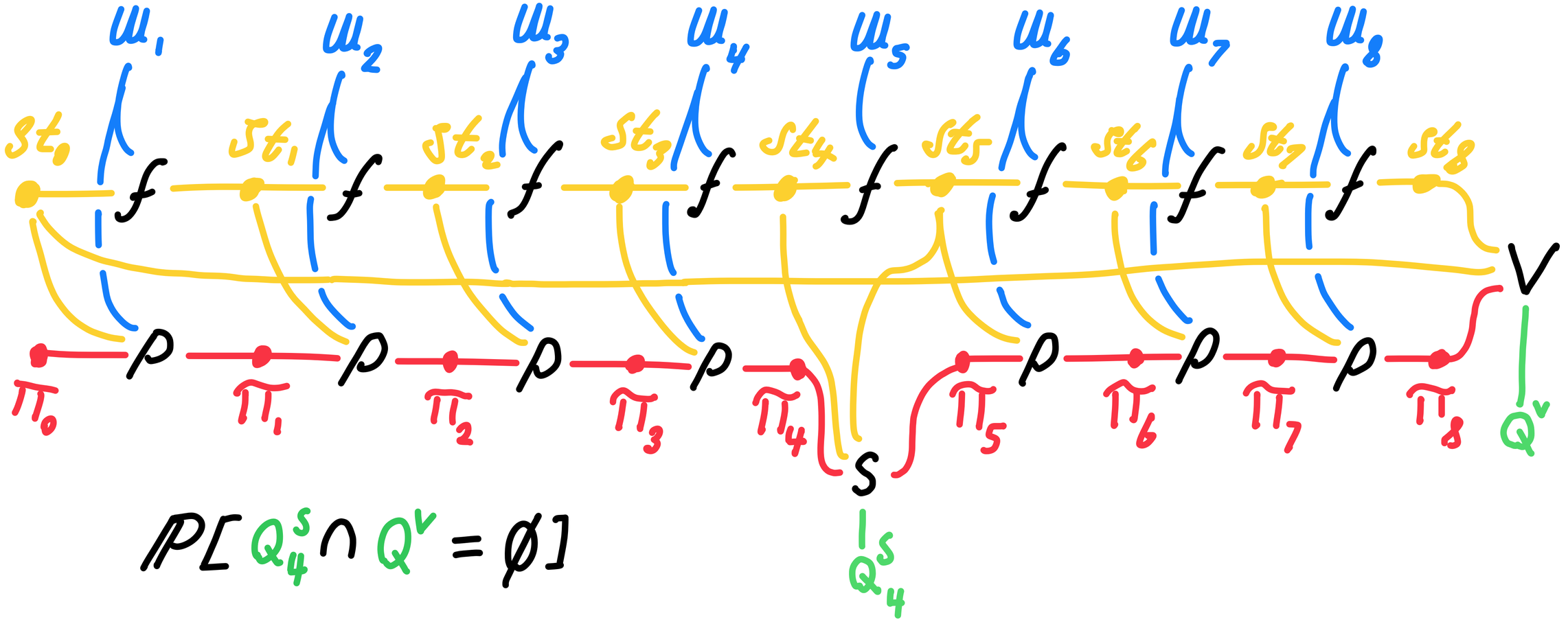
Note: Non-Deterministic IVC + CRH → SNARKs

Proof Sketch

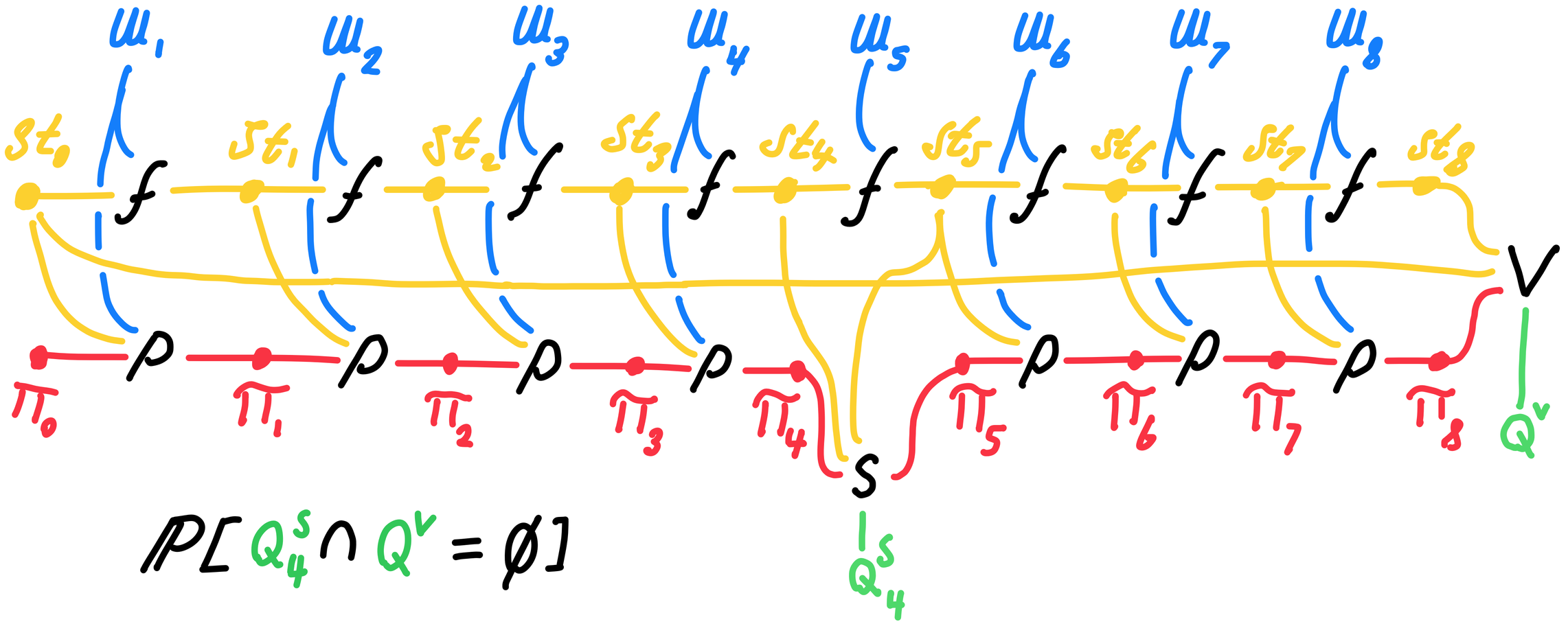

w_1
w_2
w_3
w_4
w_5
w_6
w_7
w_8
st_0
st_1
st_2
st_3
st_4
st_5
st_6
st_7
st_8
f
ρ
π_0
π_1
π_2
π_3
π_4
π_5
π_6
π_7
π_8
V
Q^V


w_1
w_2
w_3
w_4
w_5
w_6
w_7
w_8
st_0
st_1
st_2
st_3
st_4
st_5
st_6
st_7
st_8
f
ρ
π_0
π_1
π_2
π_3
π_4
π_5
π_6
π_7
π_8
V
$Q^\vee$


w_1
w_2
w_3
w_4
w_5
w_6
w_7
w_8
st_0
st_1
st_2
st_3
st_4
st_5
st_6
st_7
st_8
f
ρ
π_0
π_1
π_2
π_3
π_4
π_5
π_6
π_7
π_8
V
Q^V
S
Q^S_4

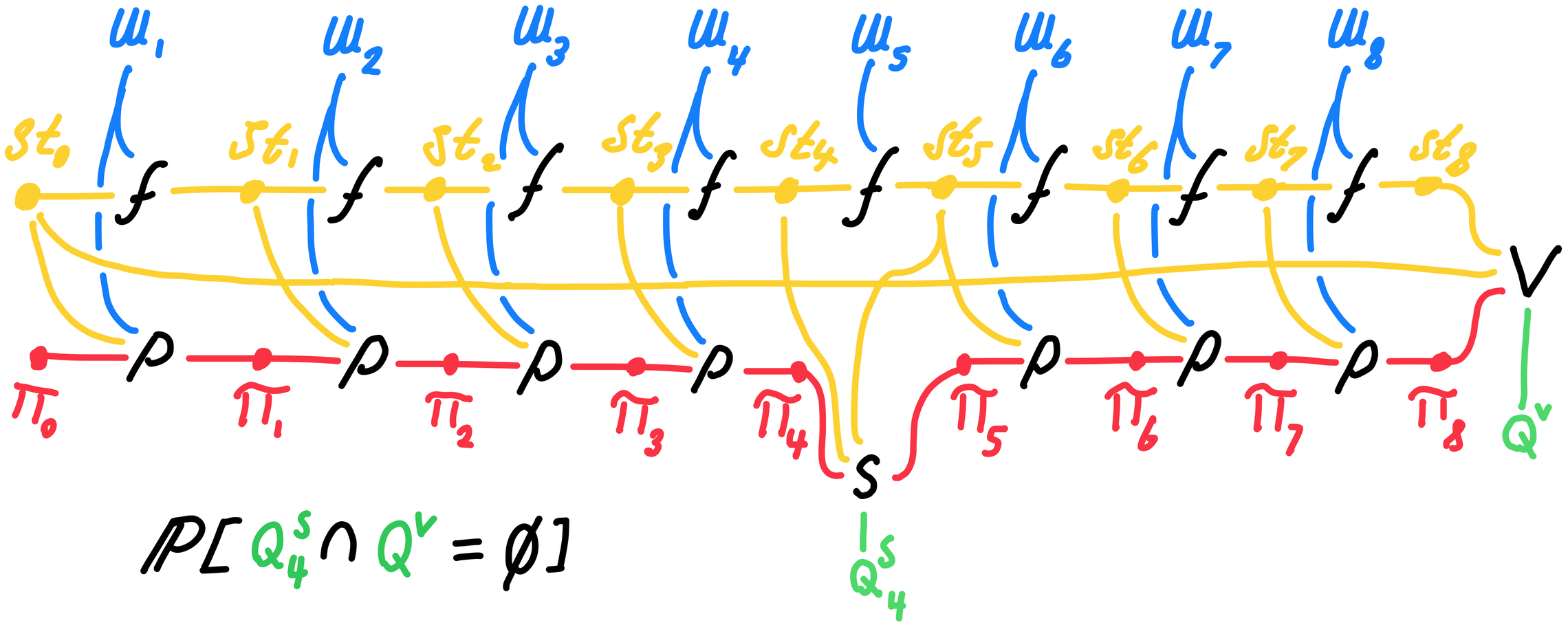


$$\mathbb{P}[\,Q_4^S \cap Q^V = \emptyset\,]$$



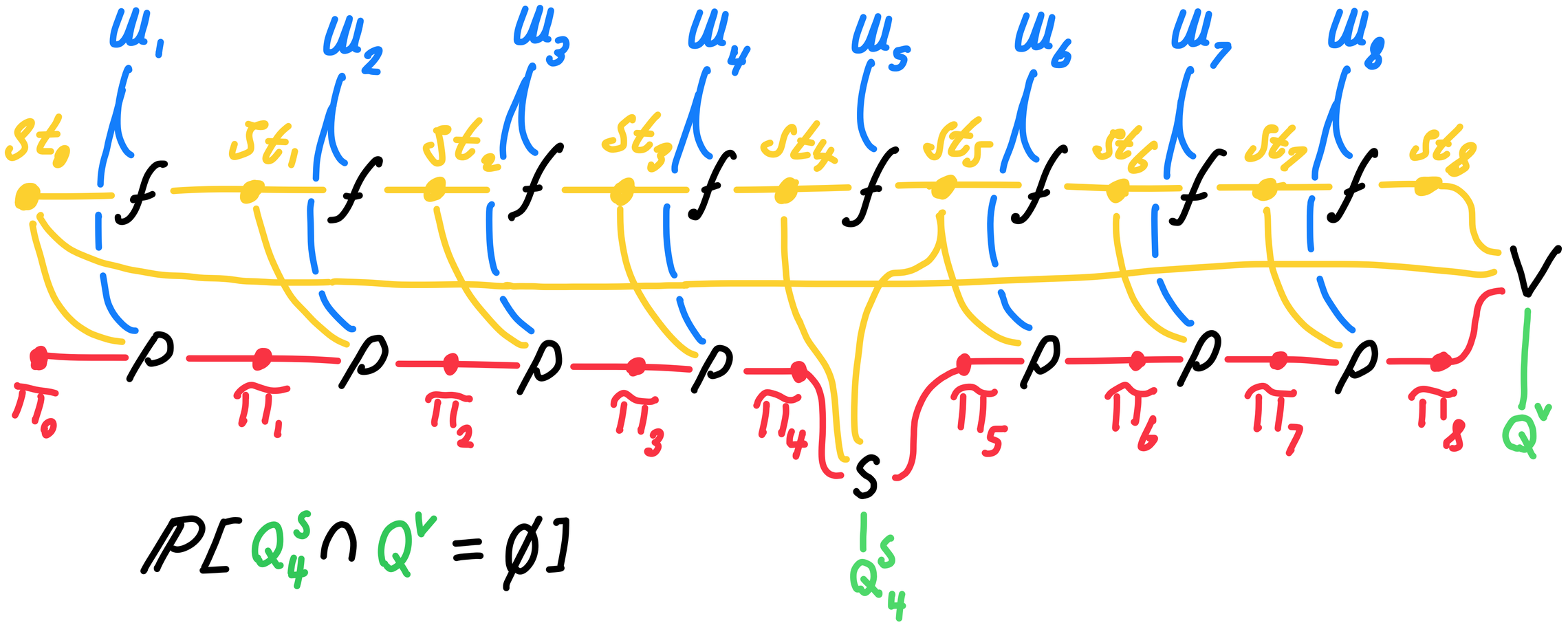
$$\mathbb{P}[Q_4^S \cap Q^V = \emptyset]$$

Can be made 1/p(n) for arbitrary poly. p.



Can be made 1/p(n) for arbitrary poly. p.

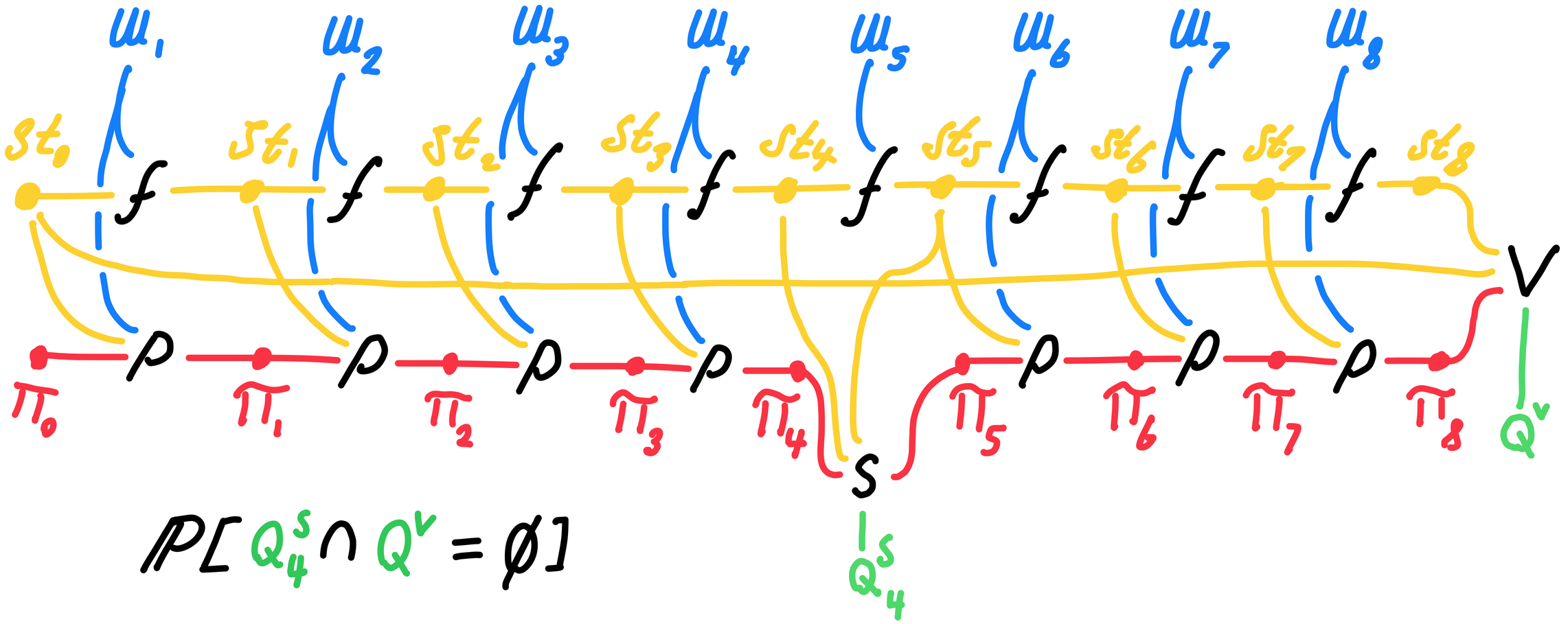
Then final proof is accepting under real oracle.



Can be made 1/p(n) for arbitrary poly. p.

Where:

Then final proof is accepting under real oracle.



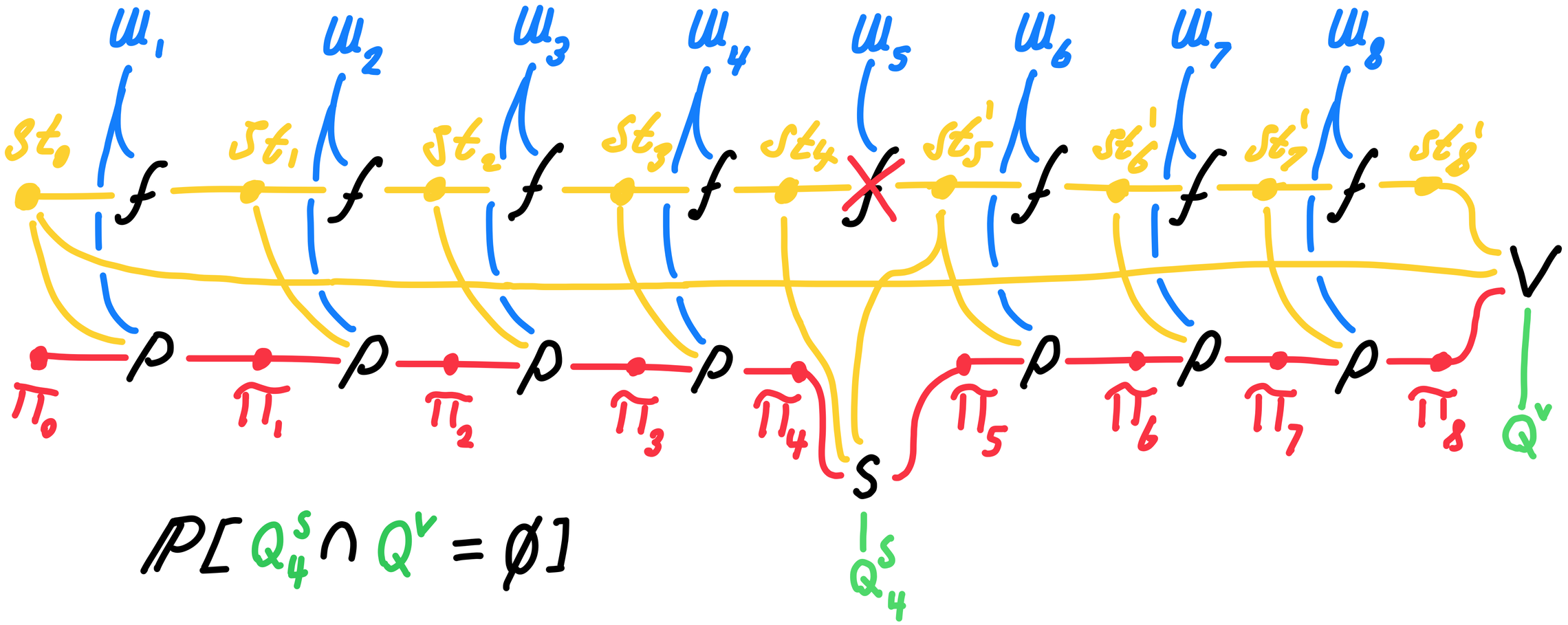
$$\mathbb{P}[Q_4^S \cap Q^V = \emptyset]$$

Can be made 1/p(n) for arbitrary poly. p.

Then final proof is accepting under real oracle.

Affects prob. by negl(λ)

Where:



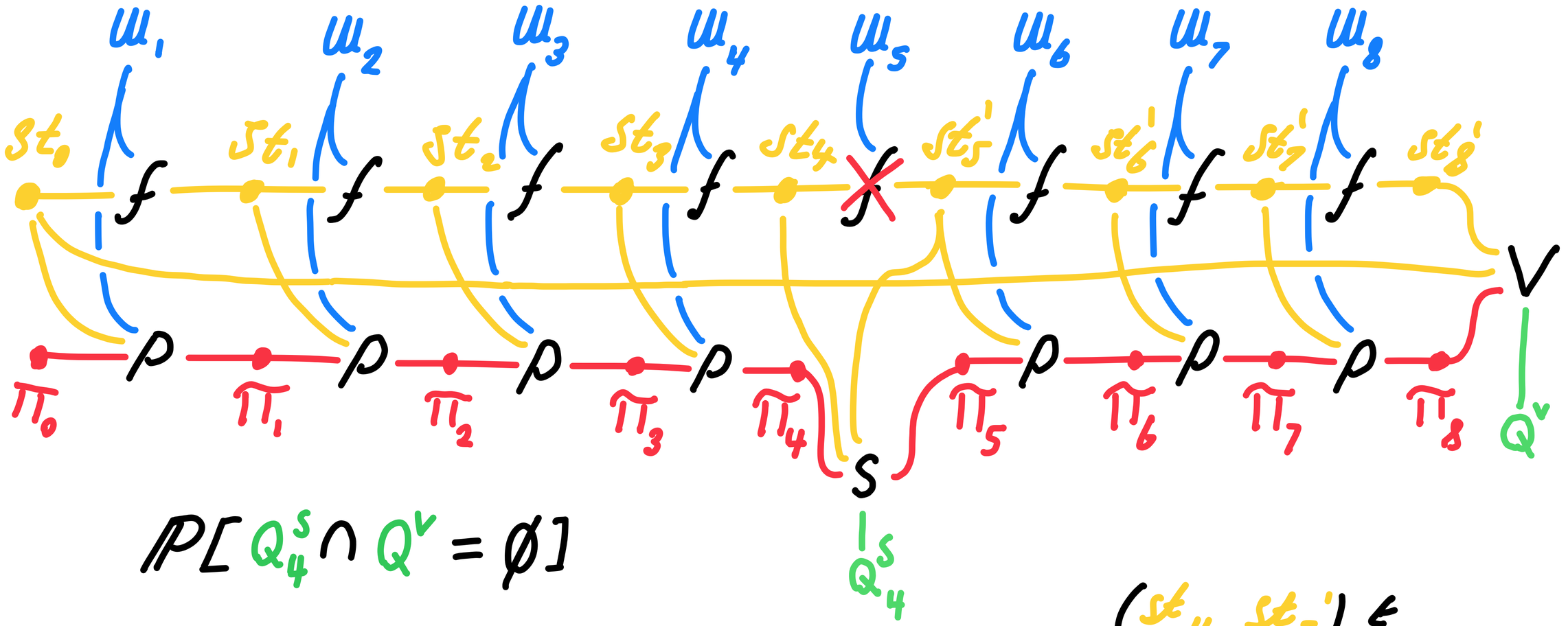
$$\mathbb{P}[Q_4^S \cap Q^V = \emptyset]$$

Can be made 1/p(n) for arbitrary poly. p.

Then final proof is accepting under real oracle.

Affects prob. by negl(λ)

Where:



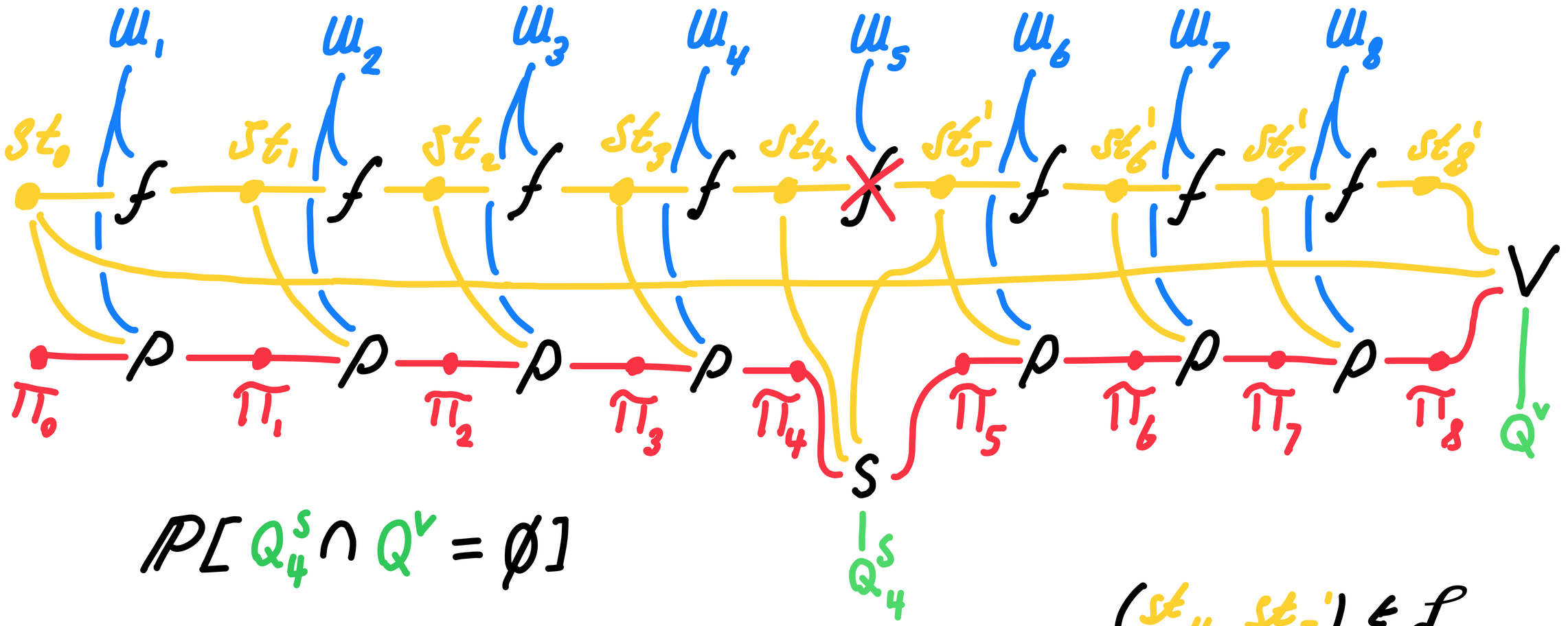
$$\mathbb{P}[Q_4^S \cap Q^V = \emptyset]$$

Can be made 1/p(n) for arbitrary poly. p.

Then final proof is accepting under real oracle.

Affects prob. by negl(λ)

Where: $(st_4, st_5') \in \mathcal{L}$



$$\mathbb{P}[Q_4^S \cap Q^V = \emptyset]$$

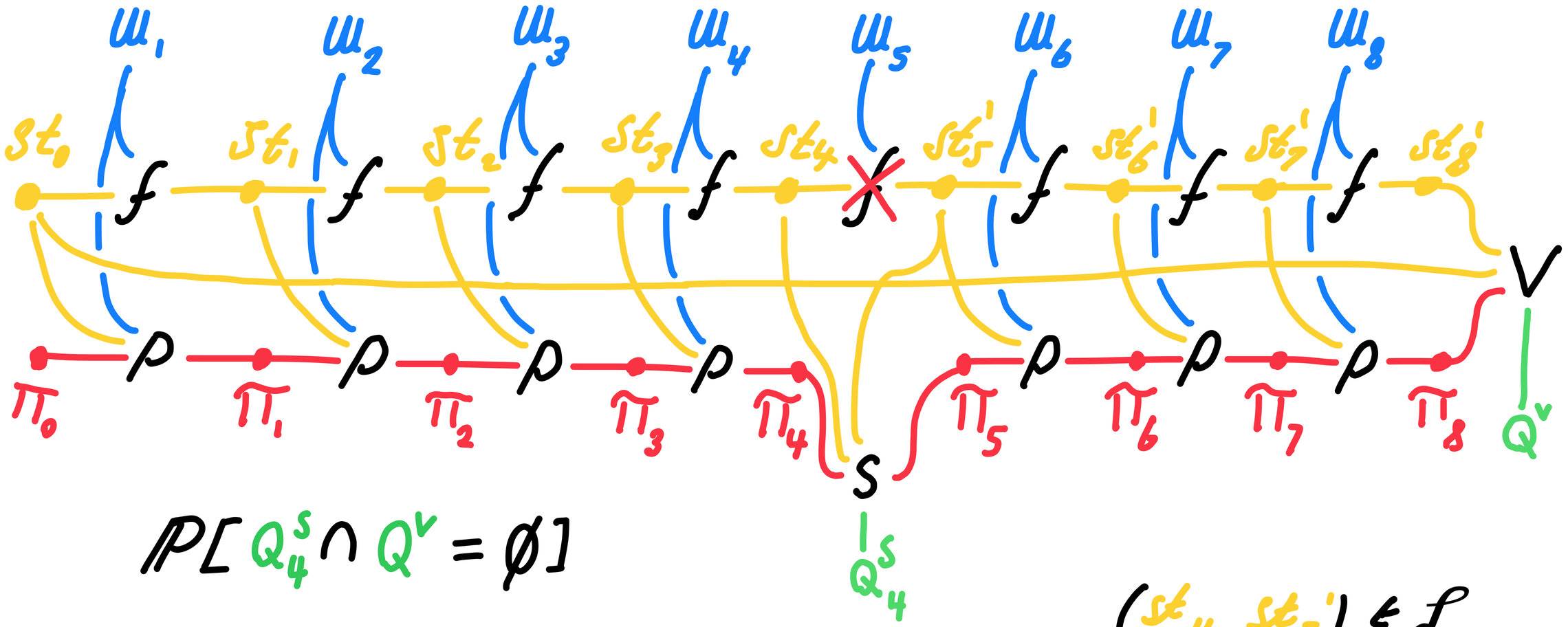
Can be made 1/p(n) for arbitrary poly. p.

Then final proof is accepting under real oracle.

Affects prob. by negl(λ)

Where: $(st_4, st_5') \in \mathcal{L}$

$(st_0, st_8') \notin \mathcal{L}$



$$\mathbb{P}[Q_4^S \cap Q^V = \emptyset]$$

Can be made 1/p(n) for arbitrary poly. p.

Then final proof is accepting under real oracle.

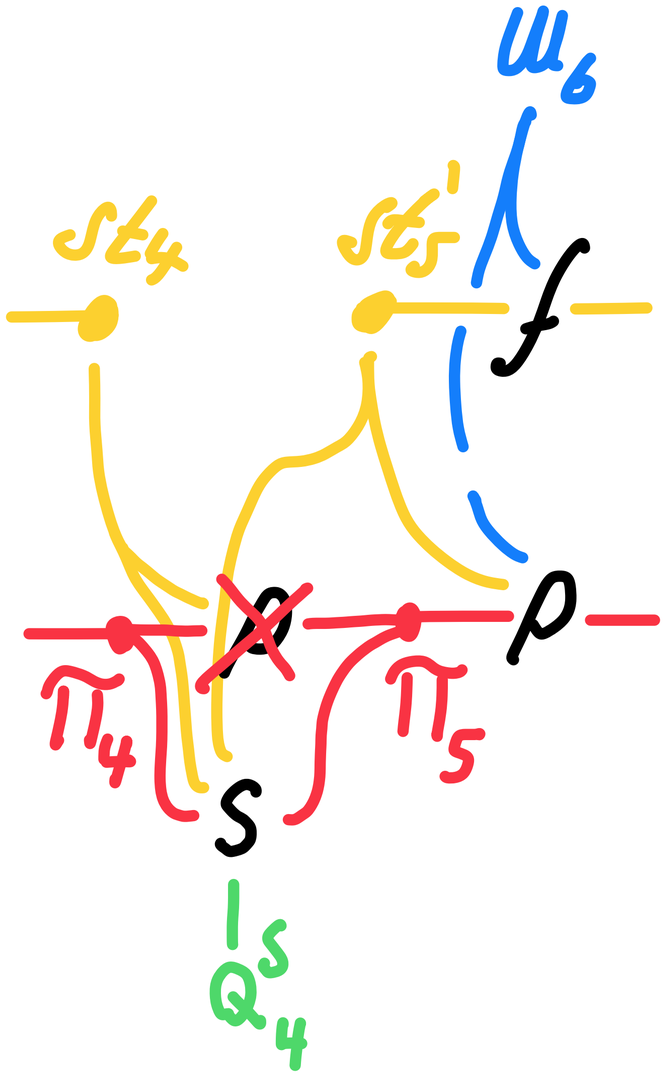
Affects prob. by negl(λ)

Where: $(st_4, st_5') \in \mathcal{L}$

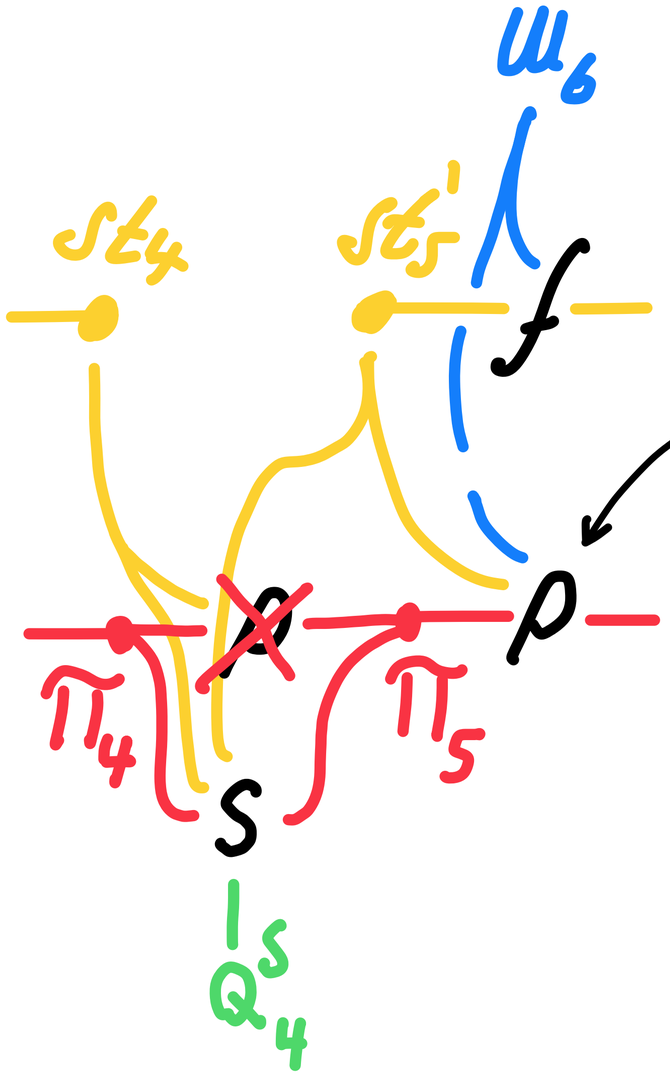
$(st_0, st_8') \notin \mathcal{L}$

Computationally Indistinguishable from $\mathcal{L}$

Zero-Knowledge Requirement

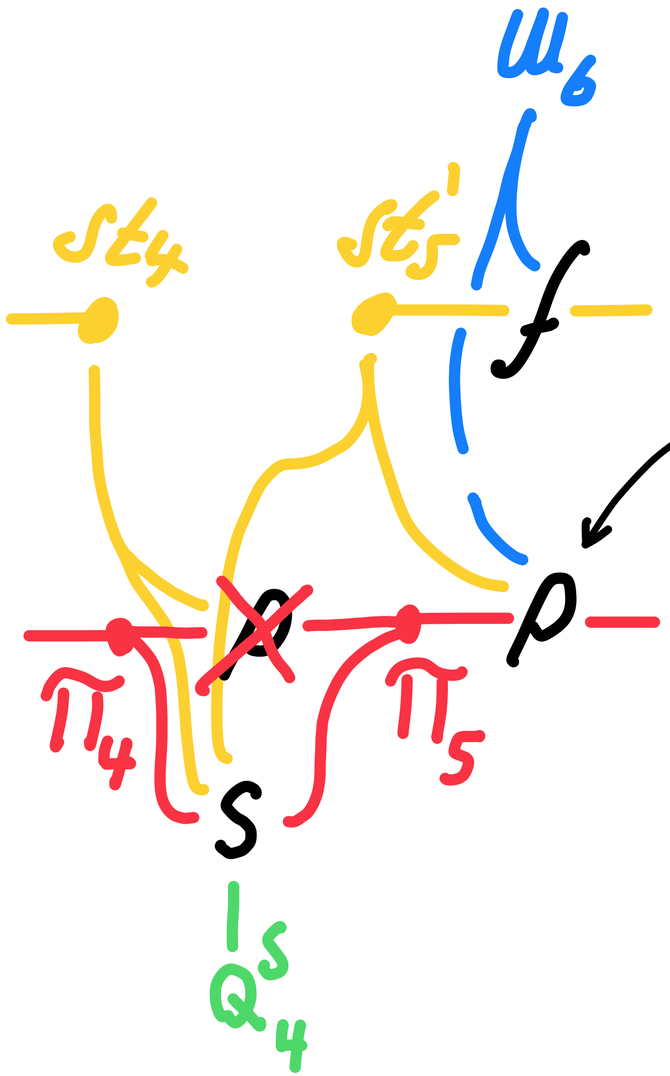


Zero-Knowledge Requirement



ZK Ensures P acts "Normally" on Simulated Proof with Programmed RO

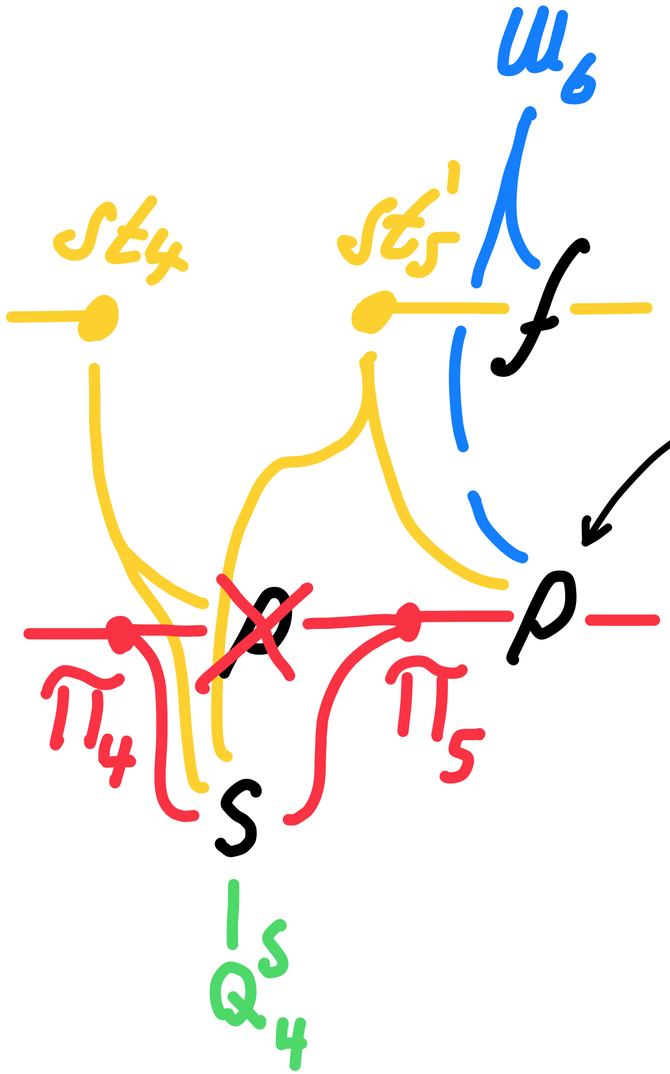
Zero-Knowledge Requirement



ZK Ensures P acts "Normally" on Simulated Proof with Programmed RO

Remove Indistinguishibility of Accepting Proof from Programmed RO

Zero-Knowledge Requirement

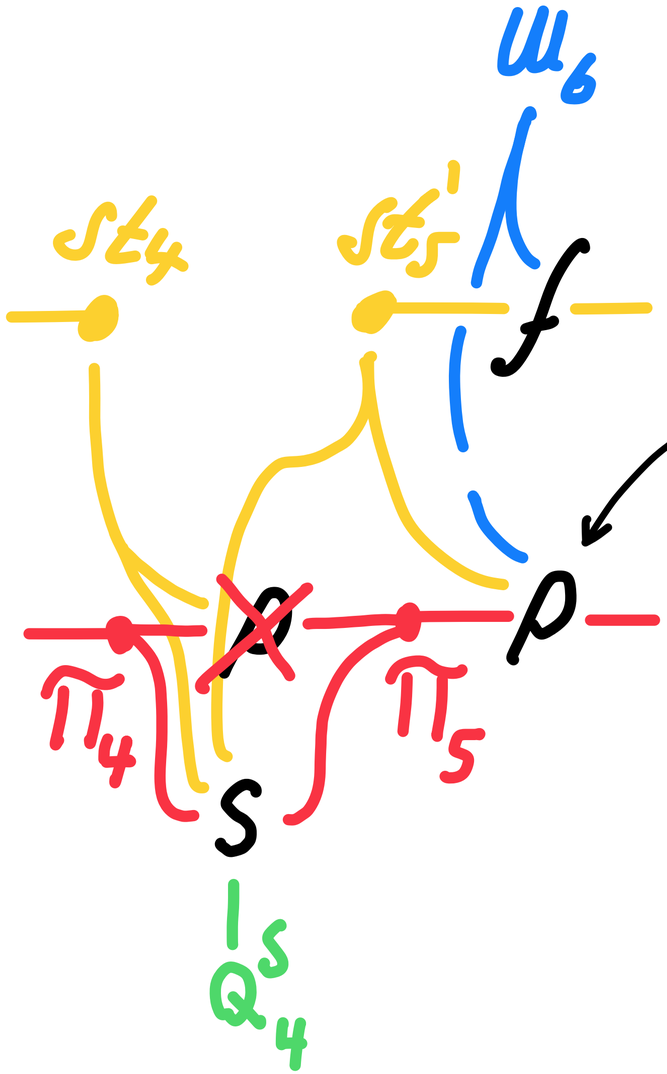


ZK Ensures P acts "Normally" on Simulated Proof with Programmed RO

Remove Indistinguishibility of Accepting Proof from Programmed RO

Rules Out Schemes In Which:

Zero-Knowledge Requirement



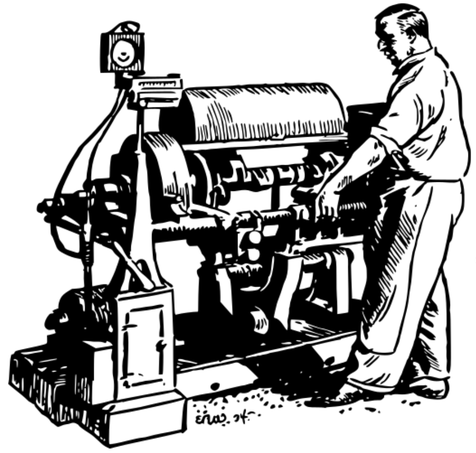
ZK Ensures P acts "Normally" on Simulated Proof with Programmed RO

$st, st', \pi \longrightarrow$

$\rho \longrightarrow$

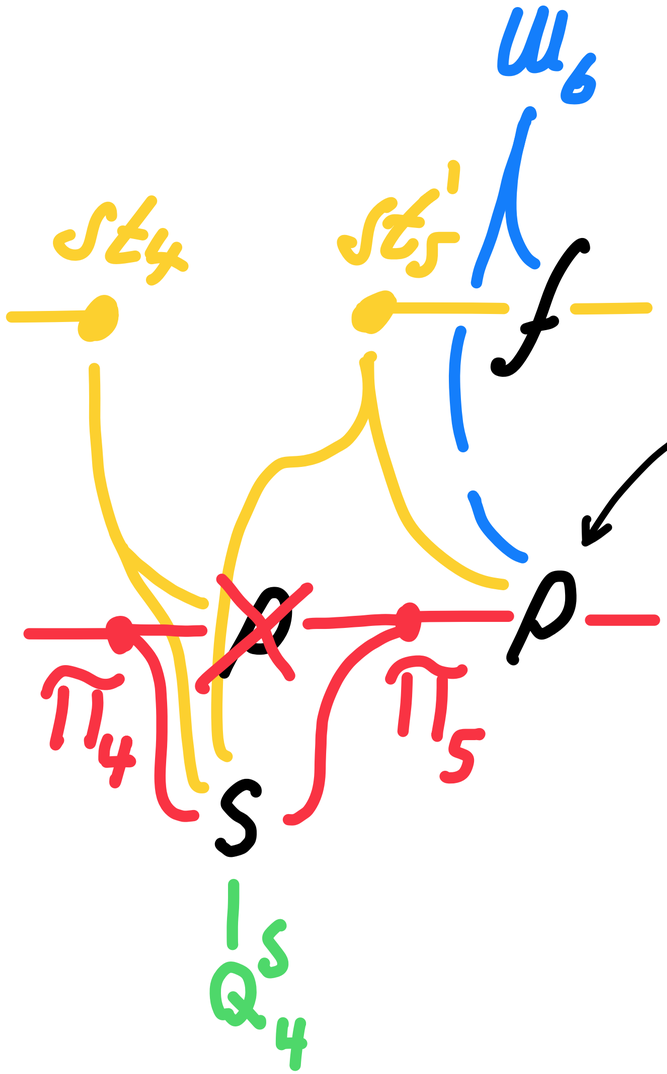
Remove Indistinguishibility of Accepting Proof from Programmed RO

Rules Out Schemes In Which:



Query Was Made by P previously

Zero-Knowledge Requirement



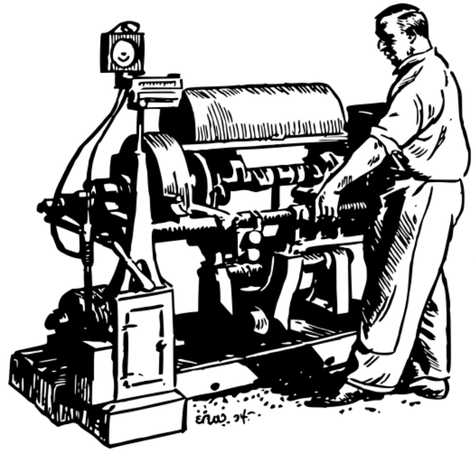
ZK Ensures P acts "Normally" on Simulated Proof with Programmed RO

$st, st', \pi \longrightarrow$

$\rho \longrightarrow$

Remove Indistinguishibility of Accepting Proof from Programmed RO

Rules Out Schemes In Which:



Query Was Made by P previously

Allowed to fail with arbitrary 1/p(n) Prob.

Questions?

"On Valiant's Conjecture"

Mathias Hall-Andersen
Jesper Buus Nielsen

Paintings By Bartholomeus Johannes van Hove