An optimal distance-bounding protocol

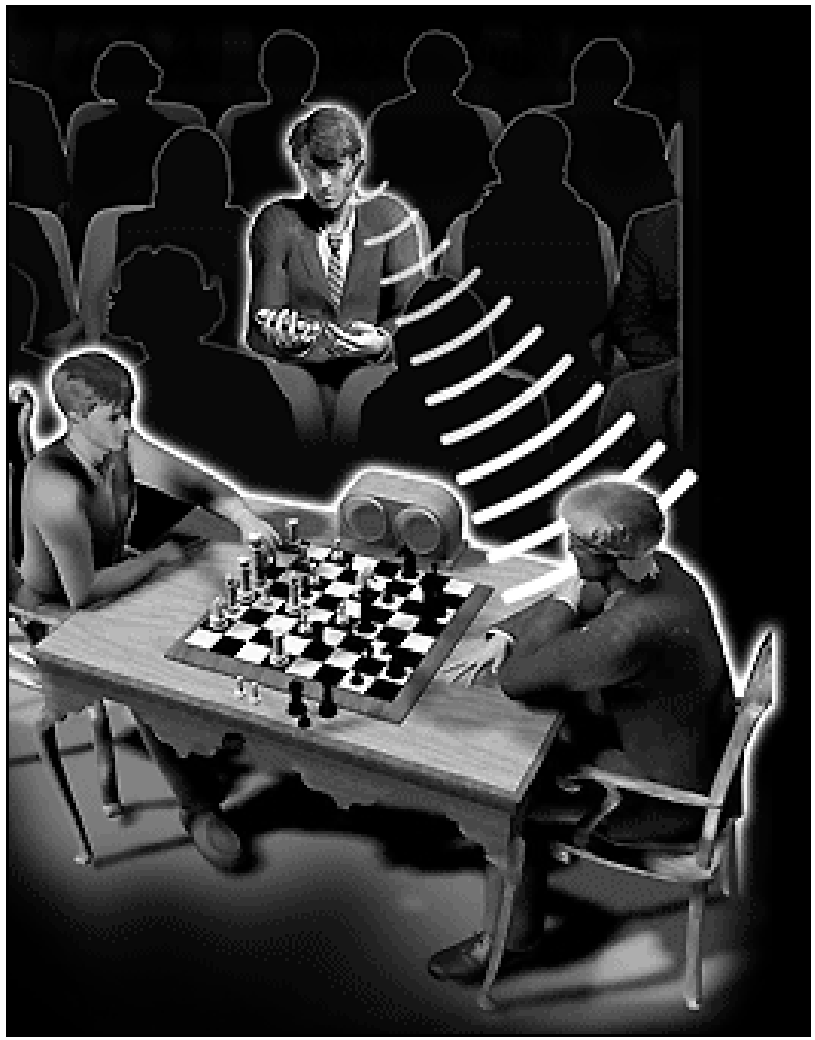
Rolando Trujillo-Rasua
University of luxembourg

(joint work with Sjouke Mauw and Jorge Toro-Pozo)

Euro S&P and RFIDSec, 2016

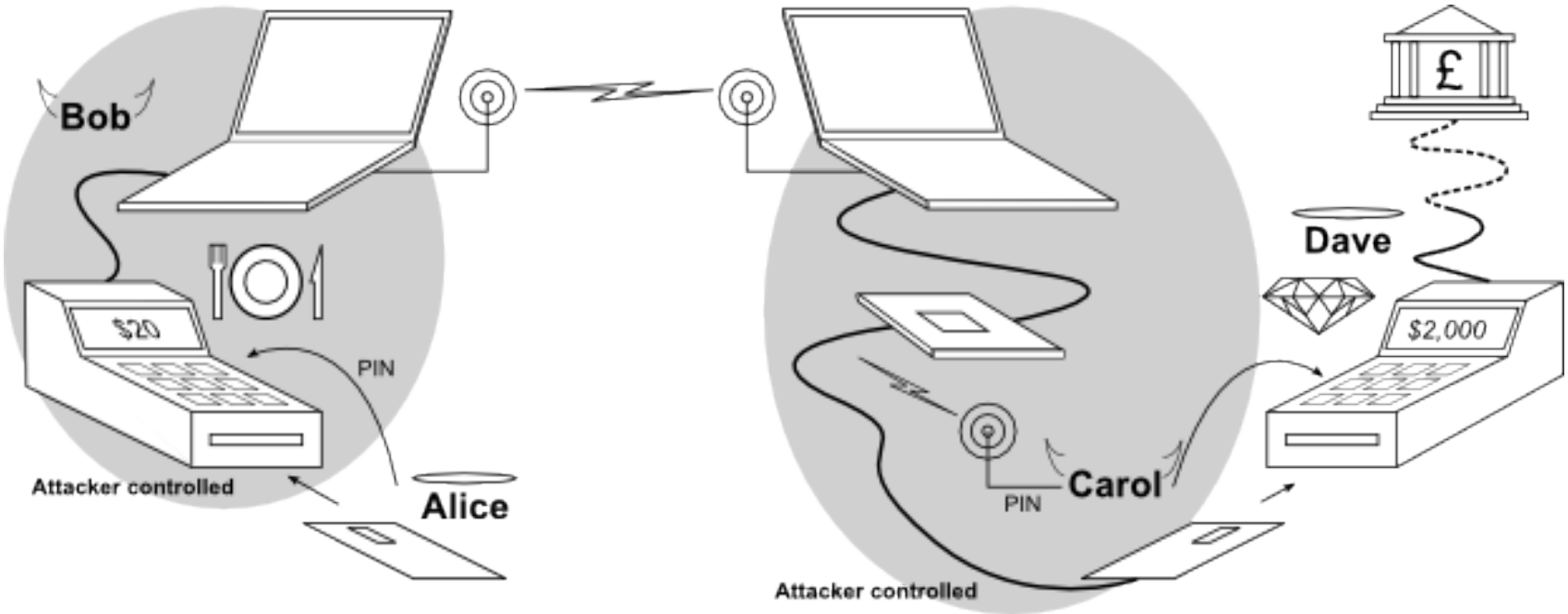
Beating a grand master: is this a relay attack?

Relay attack: is this a relay attack?



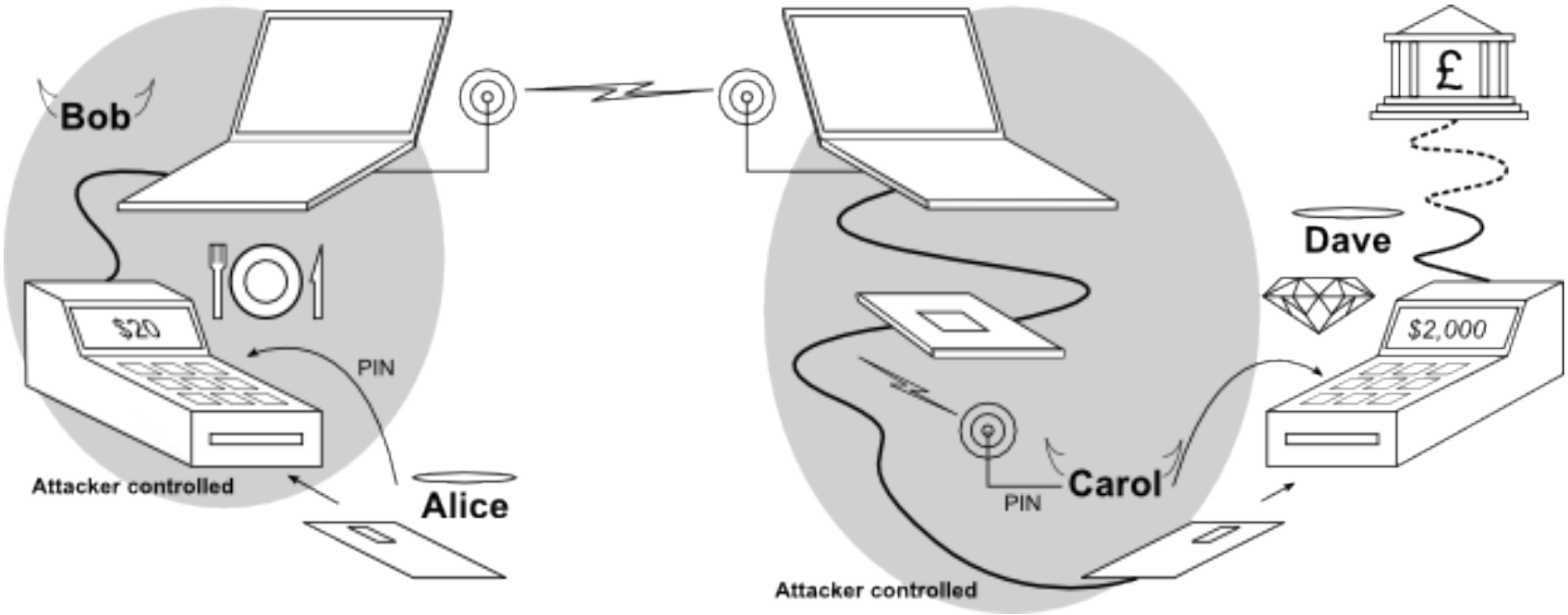
Chip & Pin relay attack

(Murdoch & Drimer 2007)



Chip & Pin relay attack

(Murdoch & Drimer 2007)



Many more practical attacks, e.g.

- Passive keyless entry and start systems used in modern cars (Francillon 2012)
- Google Wallet Relay Attack (Roland 2013)

Solution: distance bounding

- Reader sends a challenge.
- Tag provides correct response.
- Reader measures the round-trip-time and accepts if this is "fast enough".

Solution: distance bounding

- Reader sends a challenge.
- Tag provides correct response.
- Reader measures the round-trip-time and accepts if this is "fast enough".

- RF communication at the speed of light.
- Need very short processing time at the tag (otherwise the adversary could overclock the tag).

Solution: distance bounding

- Reader sends a challenge.
- Tag provides correct response.
- Reader measures the round-trip-time and accepts if this is "fast enough".

- RF communication at the speed of light.
- Need very short processing time at the tag (otherwise the adversary could overclock the tag).

- Slow phase: generation of random values, exchange of parameters, preparation of data structures.
- Fast phase: 1-bit messages, tag performs at most lookup/and/xor/...; repeat this n times.

Hancke and Kuhn's proposal (2005)

P (Tag)
secret x

V (Reader)
secret x

Hancke and Kuhn's proposal (2005)

P (Tag)
secret x

V (Reader)
secret x

slow phase

fast phase

Hancke and Kuhn's proposal (2005)

P (Tag)
secret x

V (Reader)
secret x

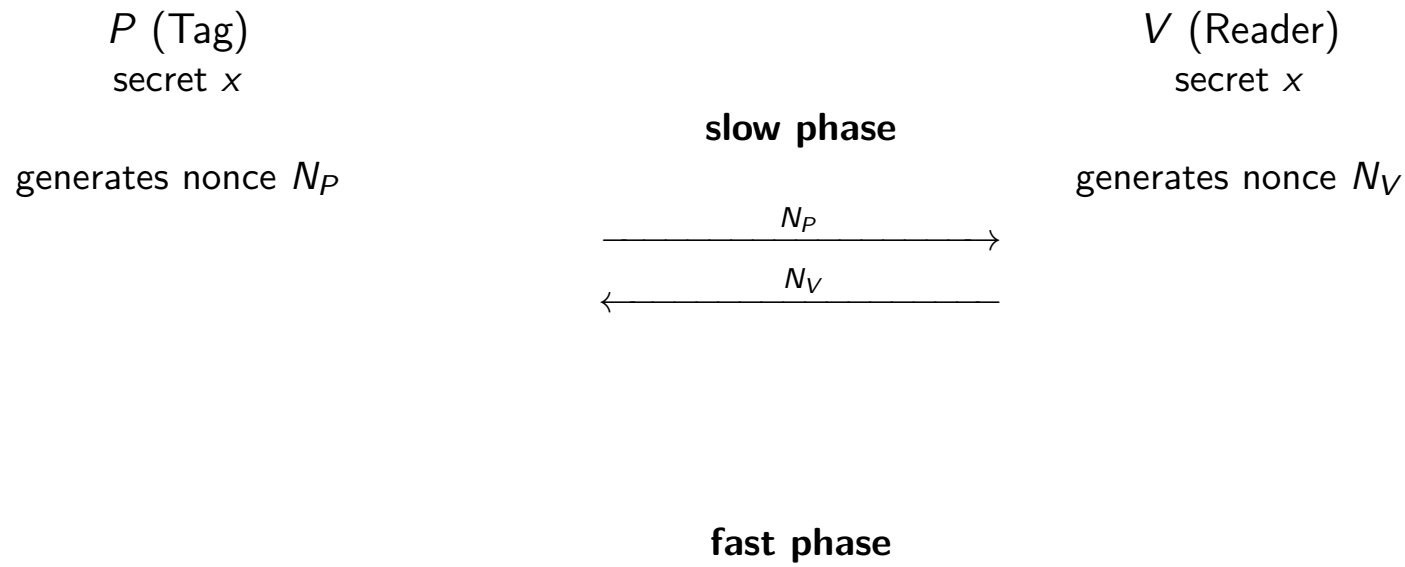
slow phase

fast phase

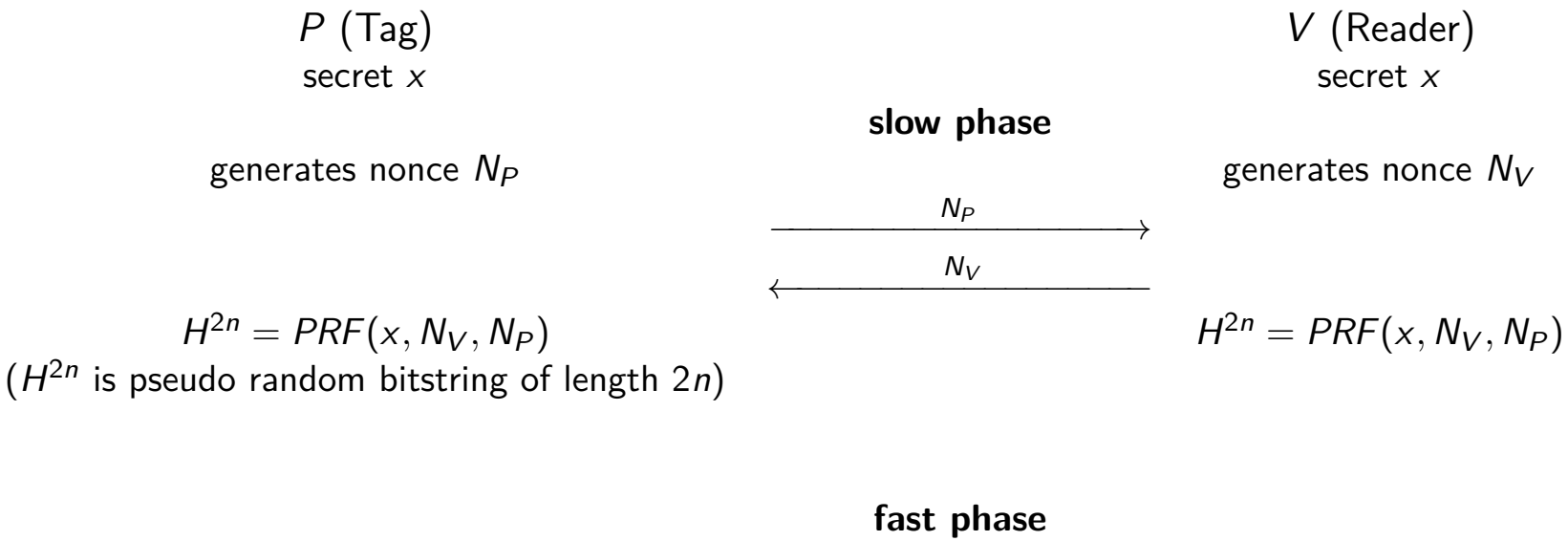
Hancke and Kuhn's proposal (2005)

P (Tag)		V (Reader)
secret x		secret x
	slow phase	
generates nonce N_P		generates nonce N_V
	fast phase	

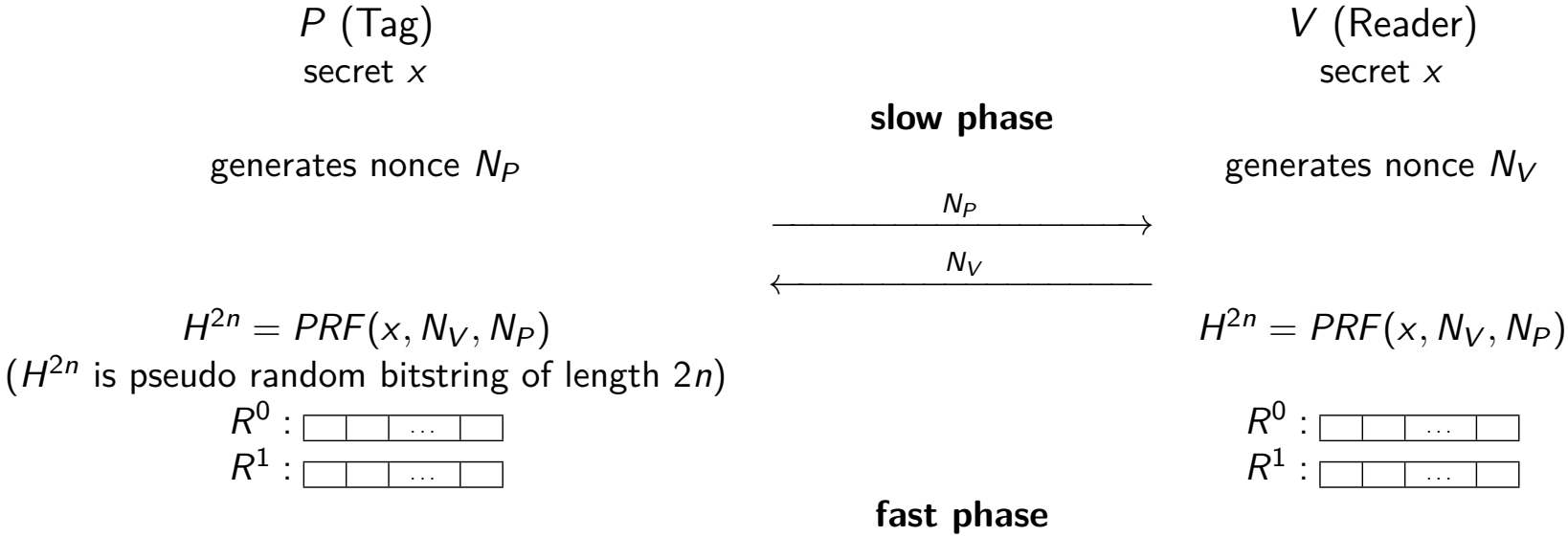
Hancke and Kuhn's proposal (2005)



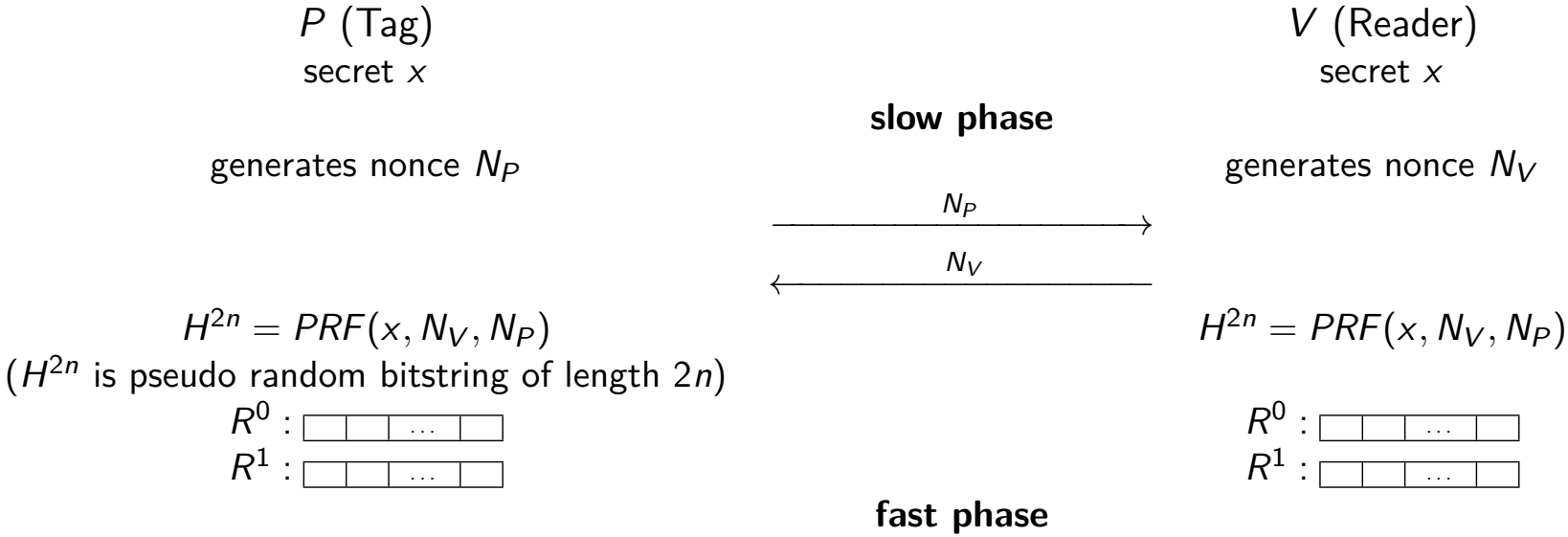
Hancke and Kuhn's proposal (2005)



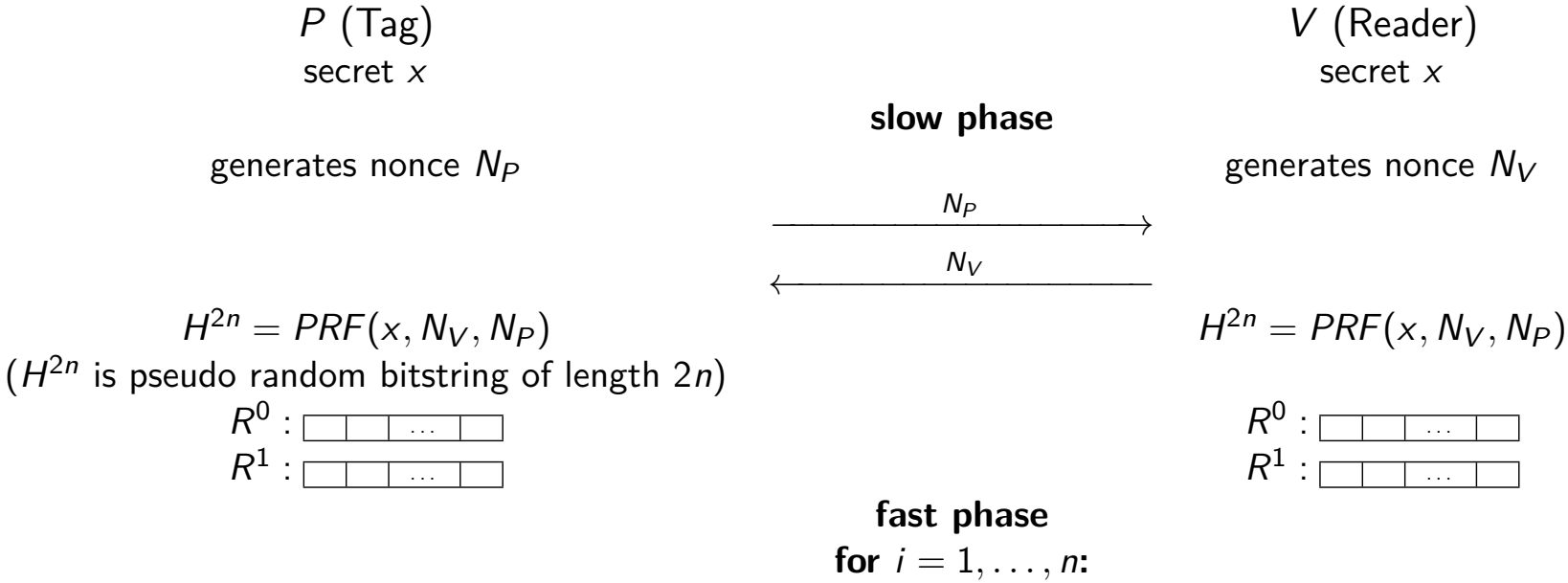
Hancke and Kuhn's proposal (2005)



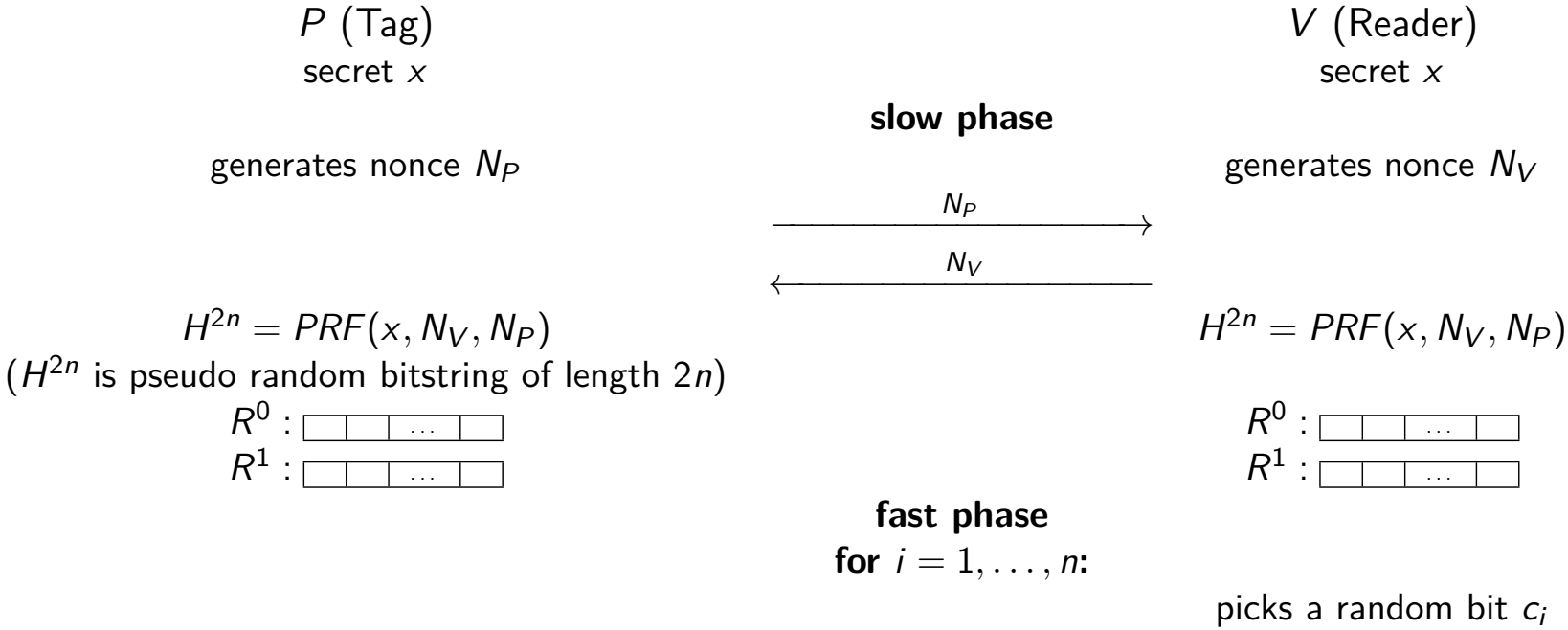
Hancke and Kuhn's proposal (2005)



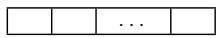
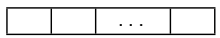
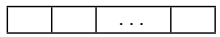
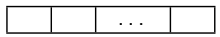
Hancke and Kuhn's proposal (2005)



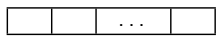
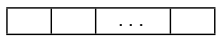
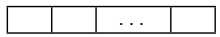
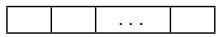
Hancke and Kuhn's proposal (2005)



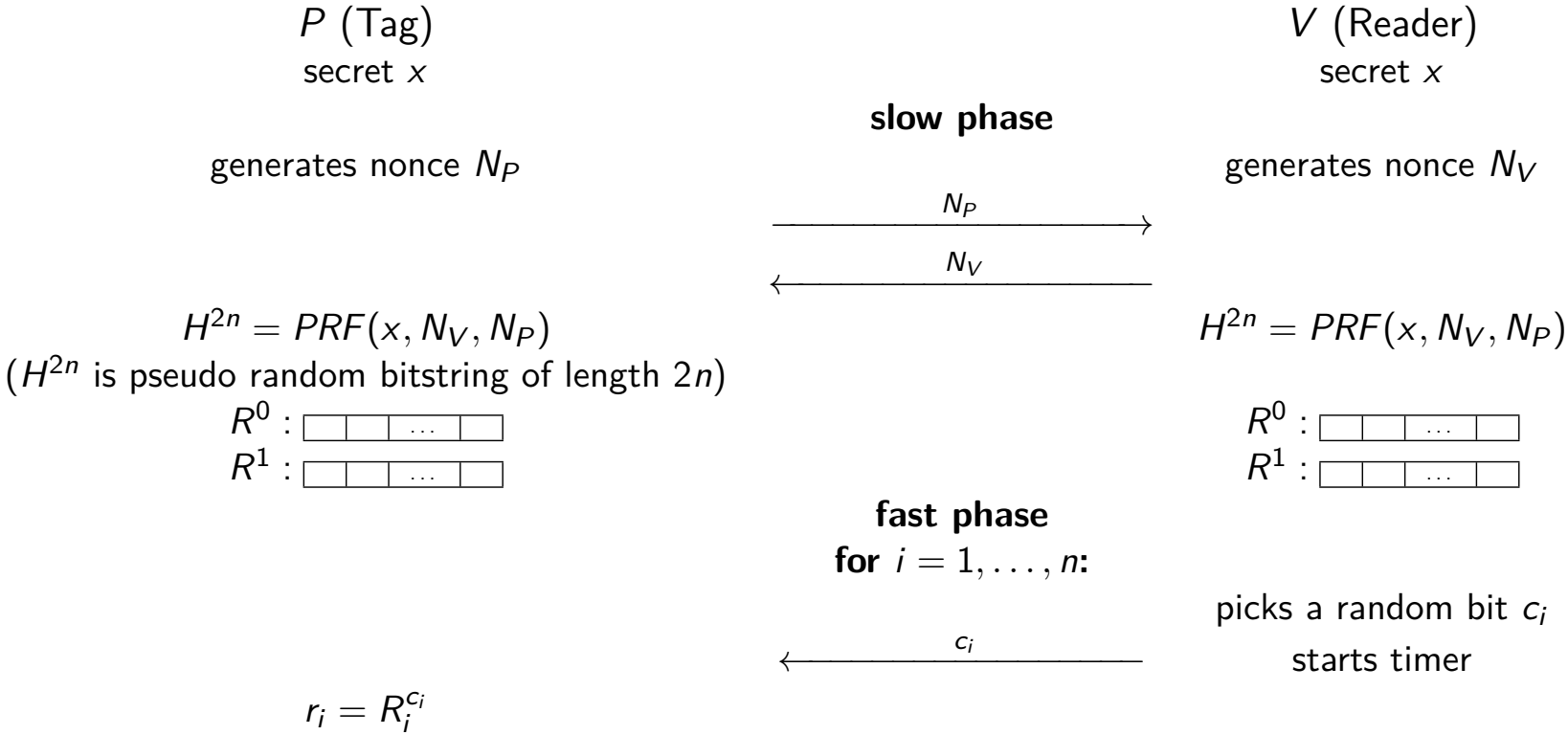
Hancke and Kuhn's proposal (2005)

P (Tag)		V (Reader)
secret x		secret x
	slow phase	
generates nonce N_P	$\xrightarrow{N_P}$	
	$\xleftarrow{N_V}$	generates nonce N_V
$H^{2n} = PRF(x, N_V, N_P)$		$H^{2n} = PRF(x, N_V, N_P)$
(H^{2n} is pseudo random bitstring of length $2n$)		
R^0 : 		R^0 : 
R^1 : 		R^1 : 
	fast phase	
	for $i = 1, \ldots, n$**:**	
		picks a random bit c_i
		starts timer

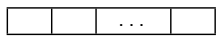
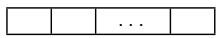
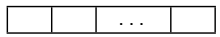
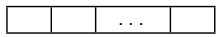
Hancke and Kuhn's proposal (2005)

P (Tag)		V (Reader)
secret x		secret x
	slow phase	
generates nonce N_P		generates nonce N_V
	$N_P \longrightarrow$	
	$\longleftarrow N_V$	
$H^{2n} = PRF(x, N_V, N_P)$		$H^{2n} = PRF(x, N_V, N_P)$
(H^{2n} is pseudo random bitstring of length $2n$)		
R^0 : 		R^0 : 
R^1 : 		R^1 : 
	fast phase	
	for $i = 1, \ldots, n$**:**	
	$c_i \longleftarrow$	picks a random bit c_i
		starts timer

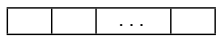
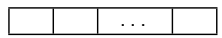
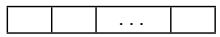
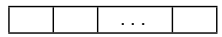
Hancke and Kuhn's proposal (2005)



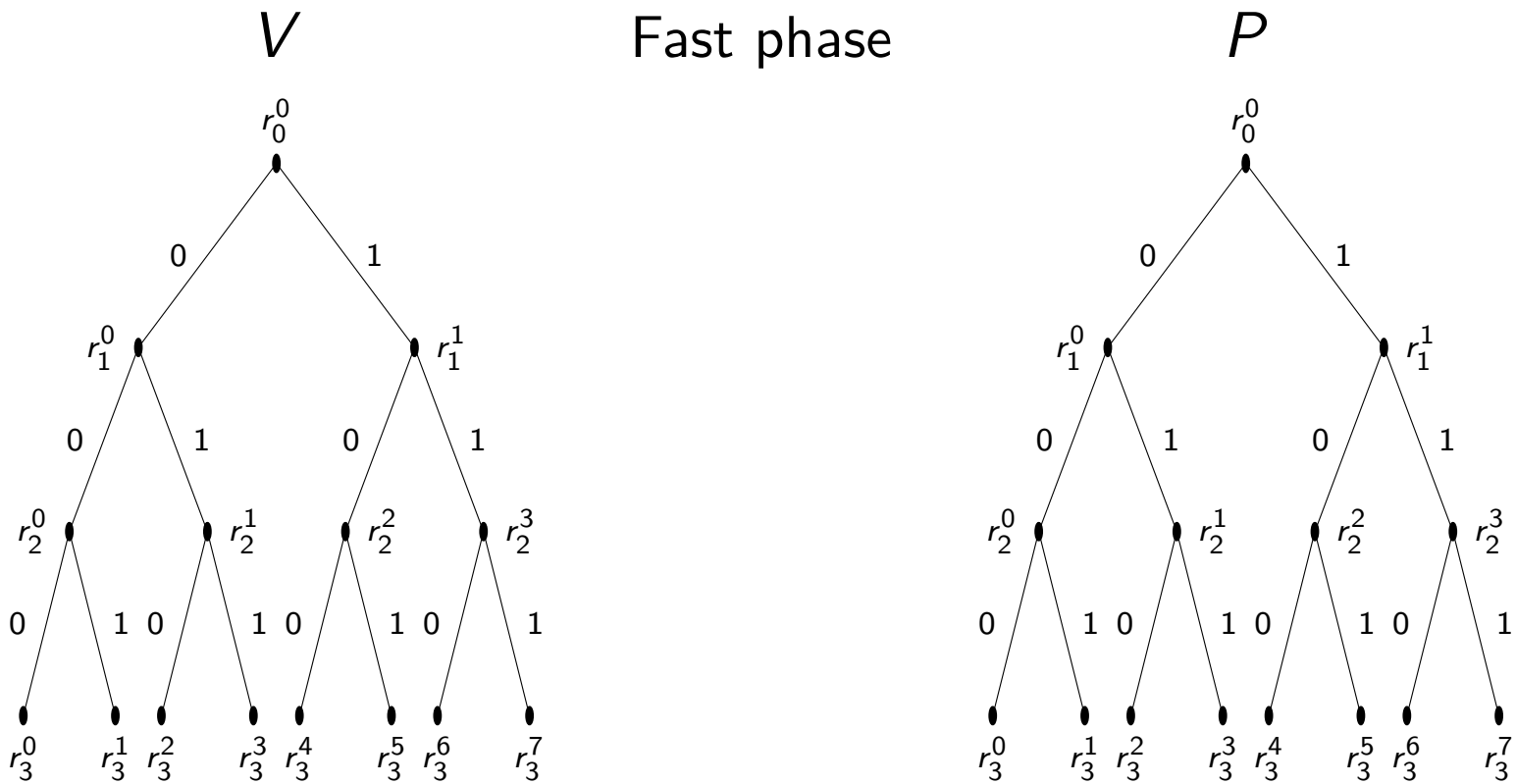
Hancke and Kuhn's proposal (2005)

P (Tag)		V (Reader)
secret x		secret x
	slow phase	
generates nonce N_P		generates nonce N_V
	$N_P \longrightarrow$	
	$\longleftarrow N_V$	
$H^{2n} = PRF(x, N_V, N_P)$		$H^{2n} = PRF(x, N_V, N_P)$
(H^{2n} is pseudo random bitstring of length $2n$)		
R^0 : 		R^0 : 
R^1 : 		R^1 : 
	fast phase	
	for $i = 1, \ldots, n$**:**	
		picks a random bit c_i
	$\longleftarrow c_i$	starts timer
	$r_i \longrightarrow$	
$r_i = R_i^{c_i}$		

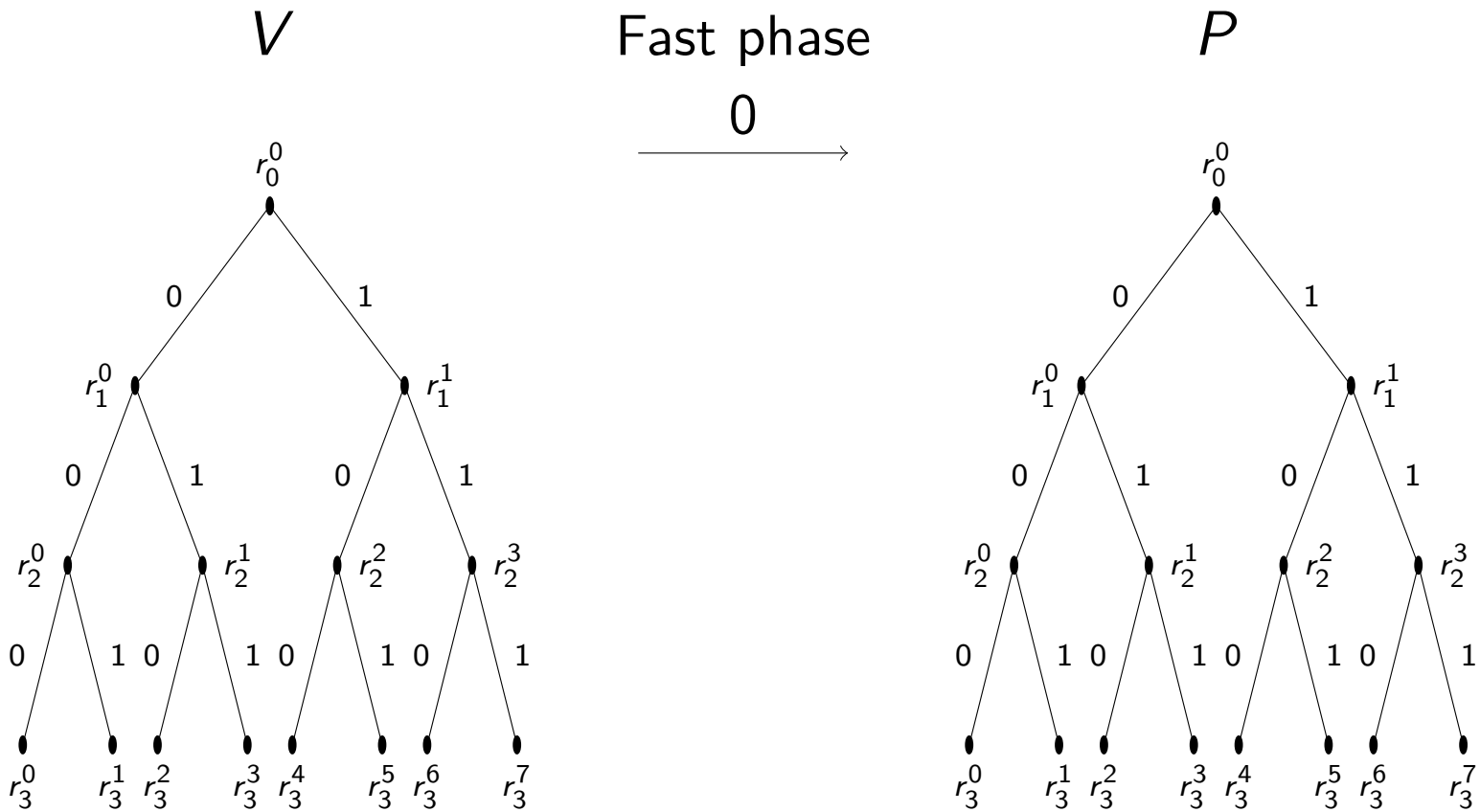
Hancke and Kuhn's proposal (2005)

P (Tag)		V (Reader)
secret x		secret x
	slow phase	
generates nonce N_P		generates nonce N_V
	$N_P \longrightarrow$	
	$\longleftarrow N_V$	
$H^{2n} = PRF(x, N_V, N_P)$		$H^{2n} = PRF(x, N_V, N_P)$
(H^{2n} is pseudo random bitstring of length $2n$)		
R^0 : 		R^0 : 
R^1 : 		R^1 : 
	fast phase	
	for $i = 1, \ldots, n$**:**	
		picks a random bit c_i
	$\longleftarrow c_i$	
		starts timer
	$r_i \longrightarrow$	
$r_i = R_i^{c_i}$		stops timer

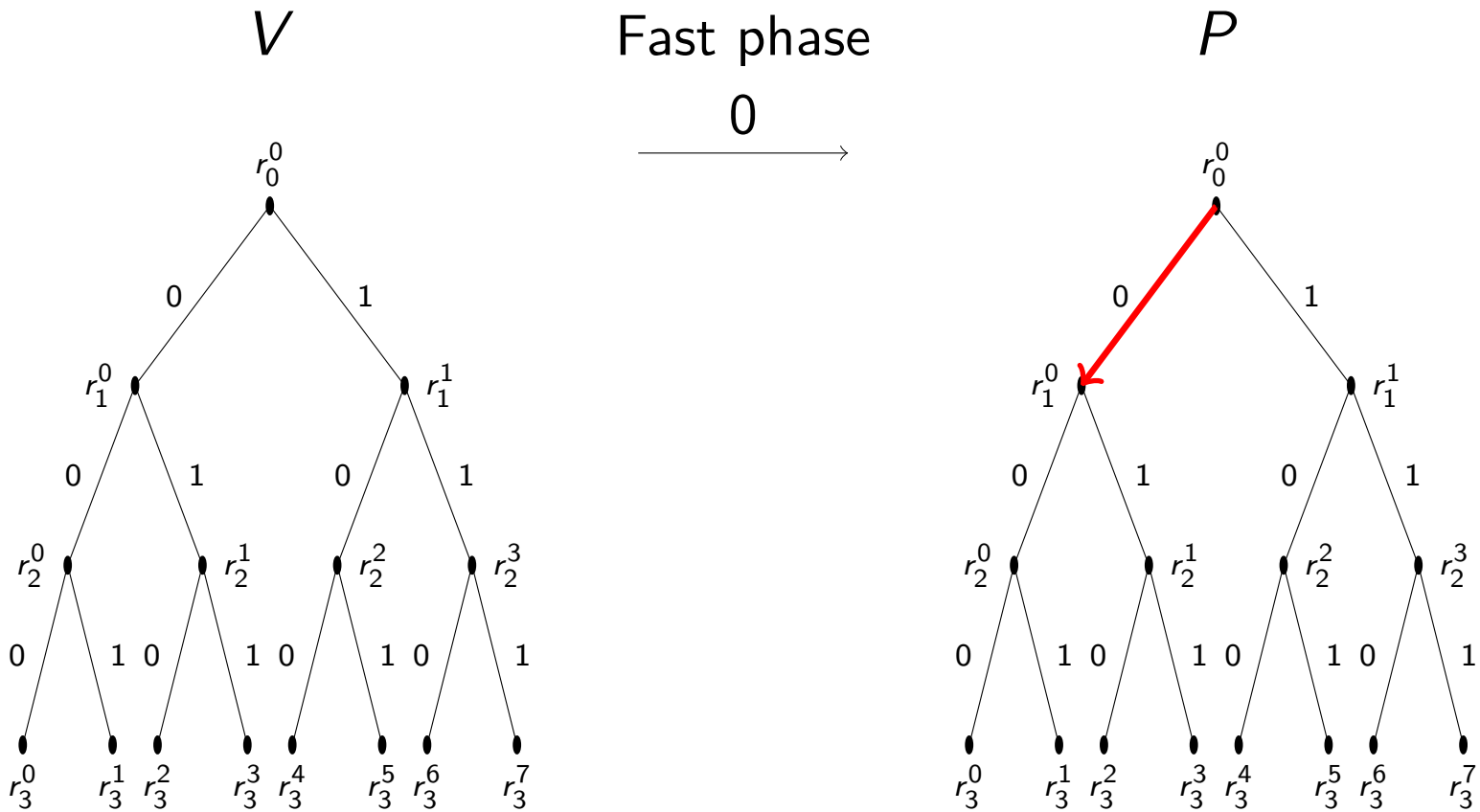
Avoine and Tchamkerten's protocol (2009)



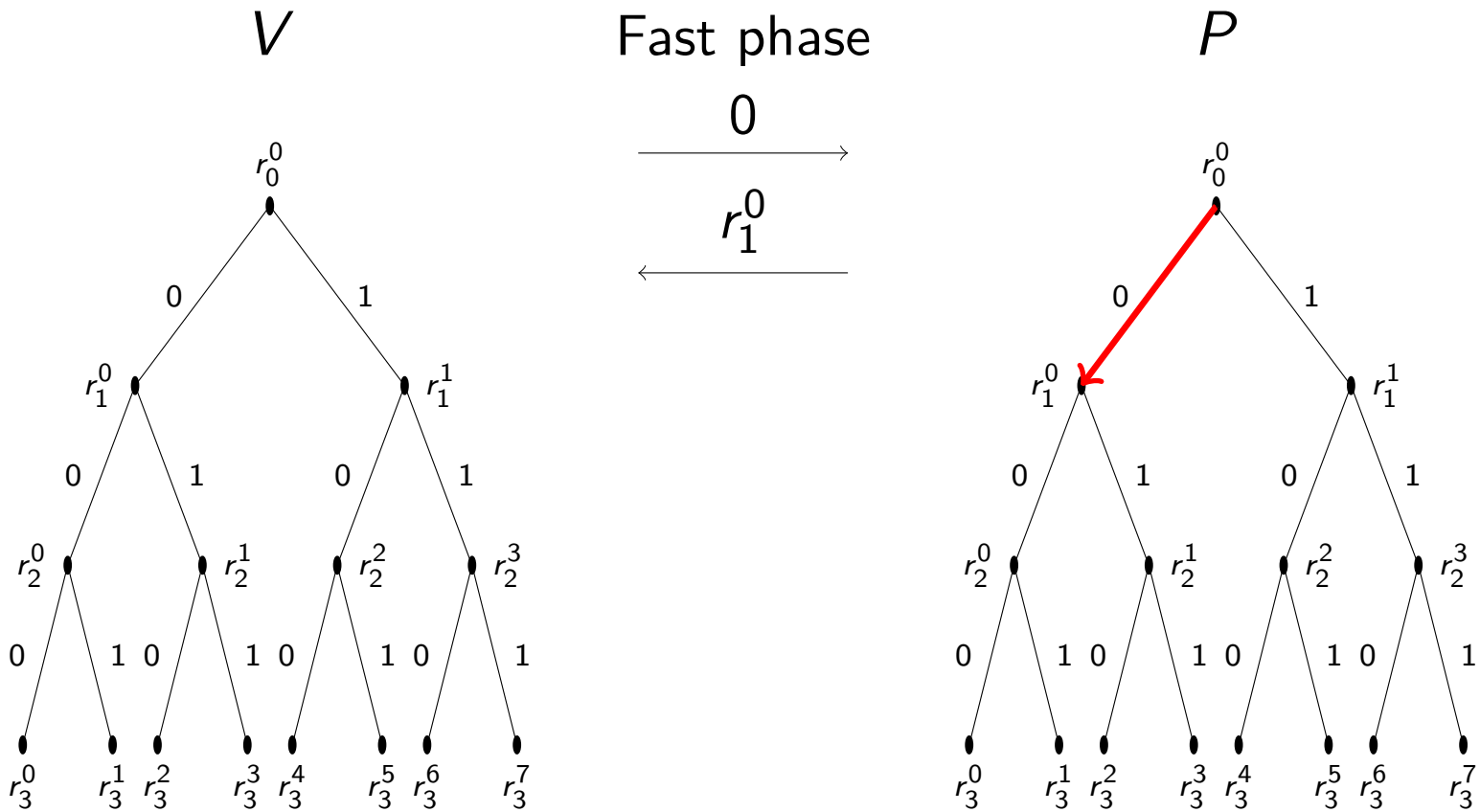
Avoine and Tchamkerten's protocol (2009)



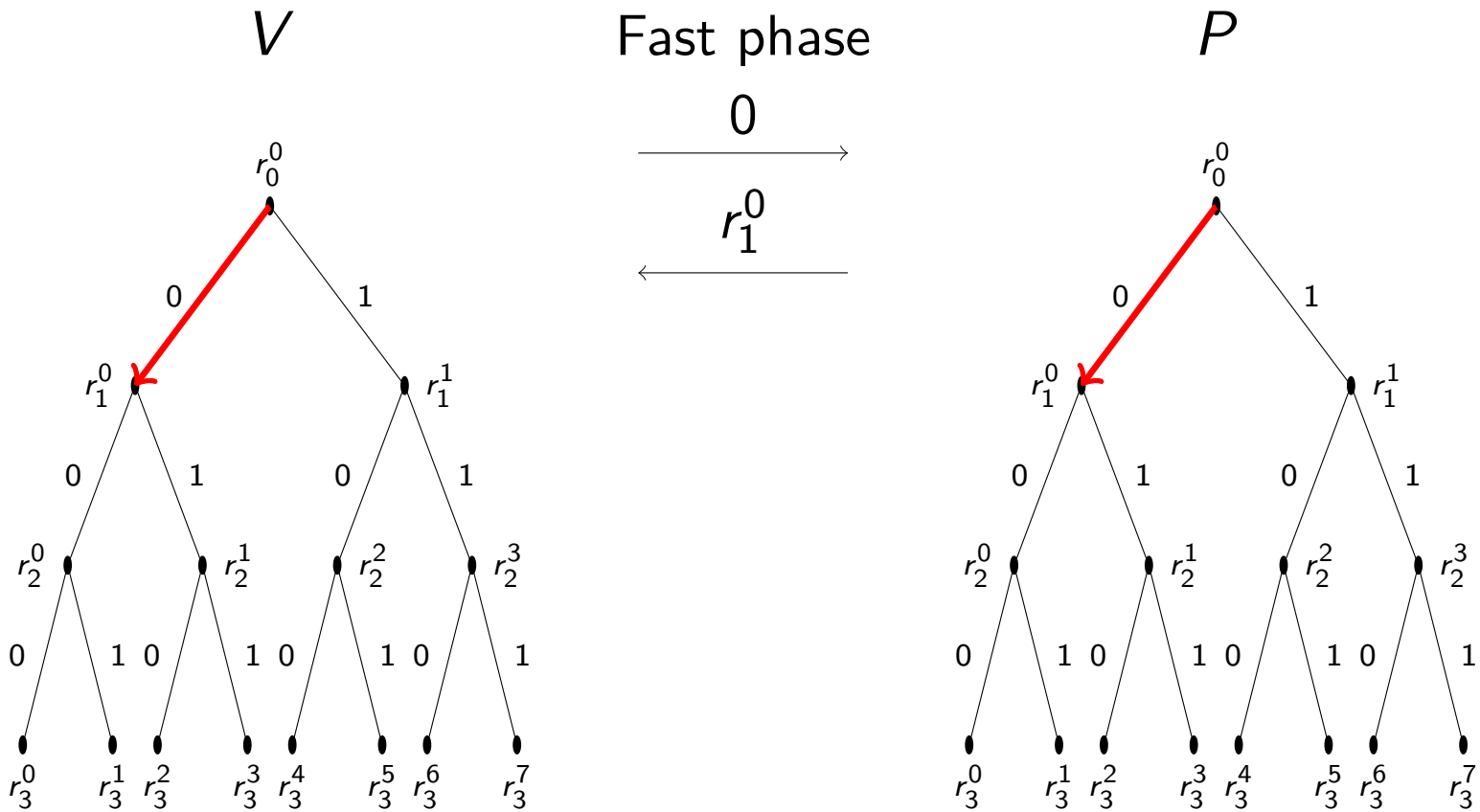
Avoine and Tchamkerten's protocol (2009)



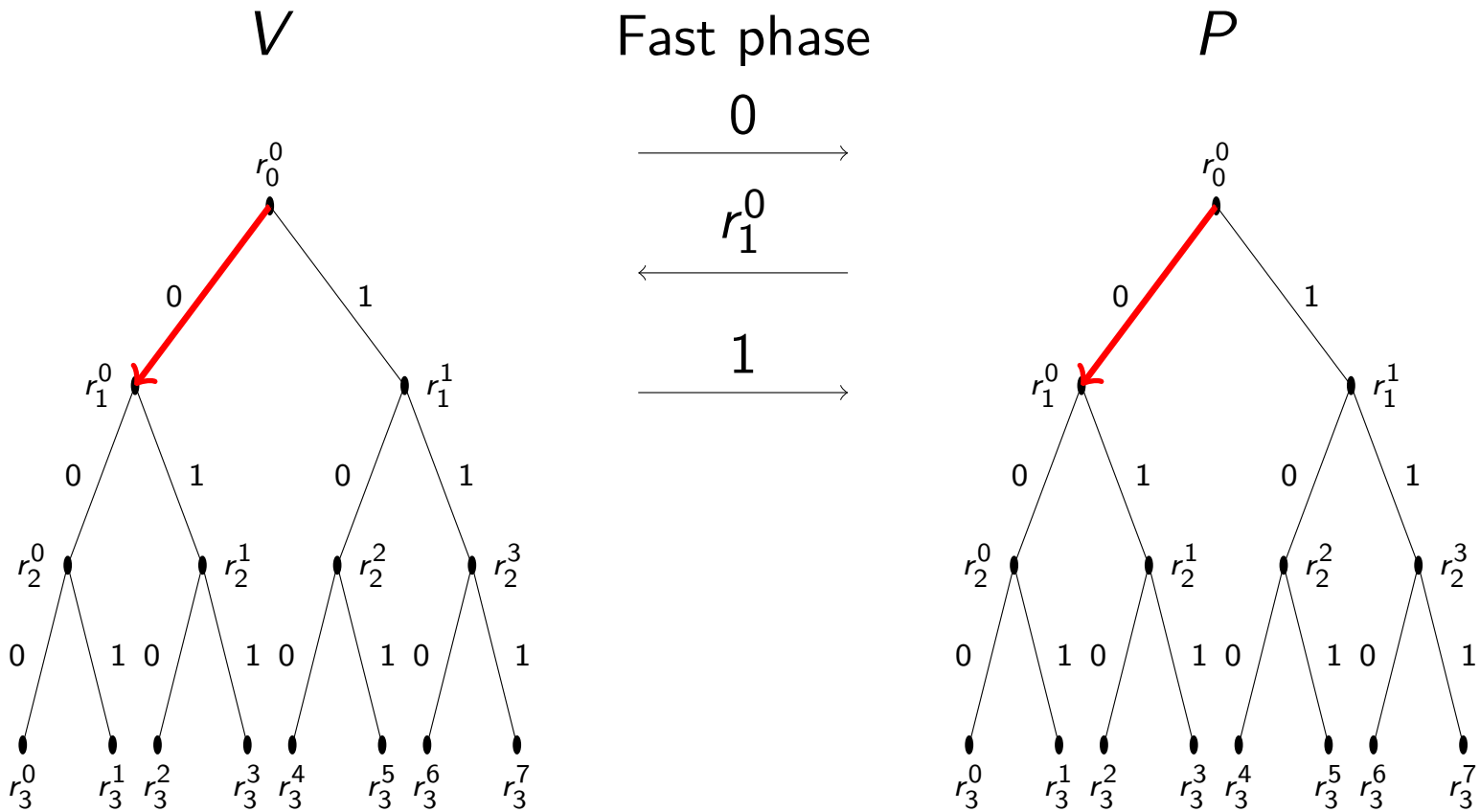
Avoine and Tchamkerten's protocol (2009)



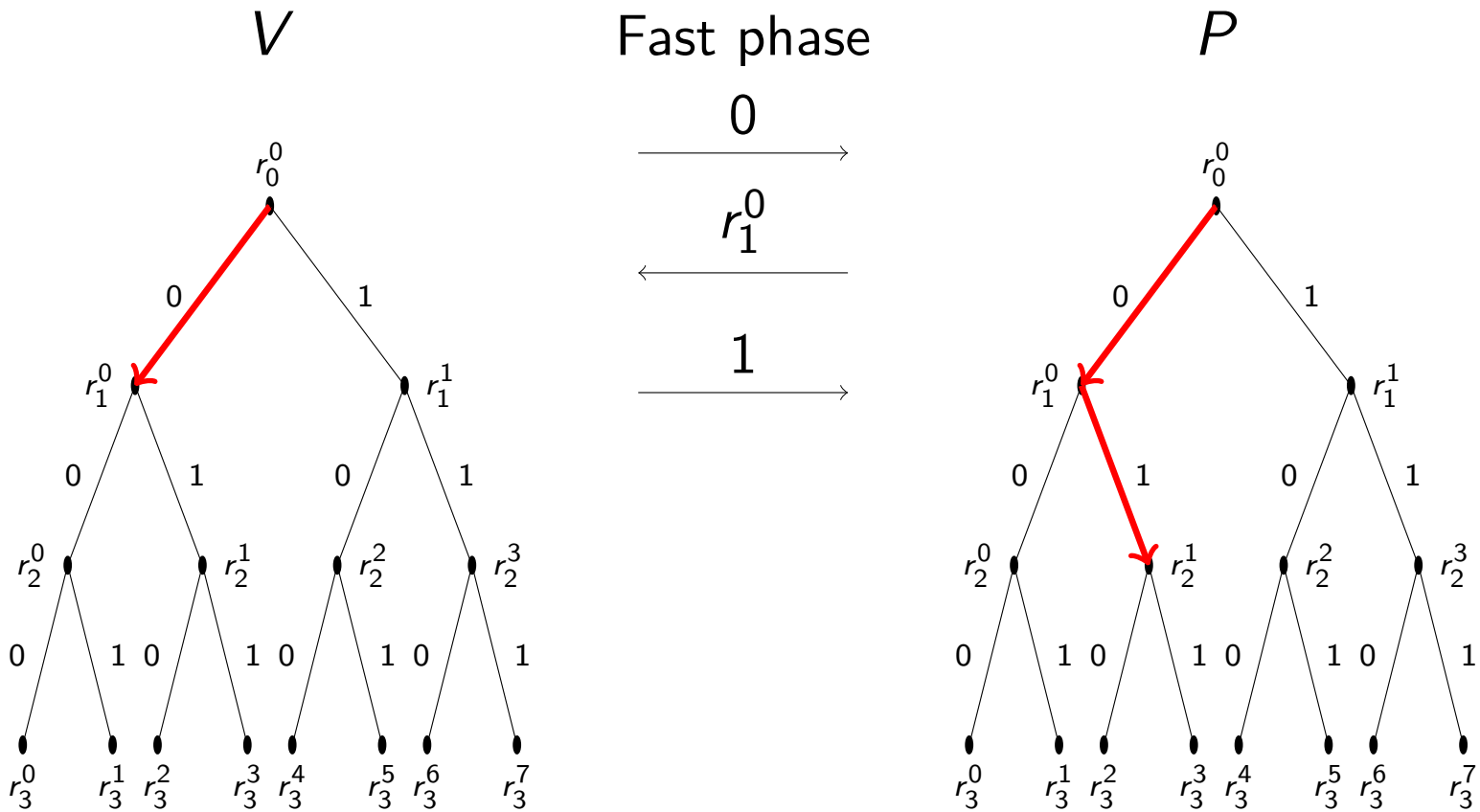
Avoine and Tchamkerten's protocol (2009)



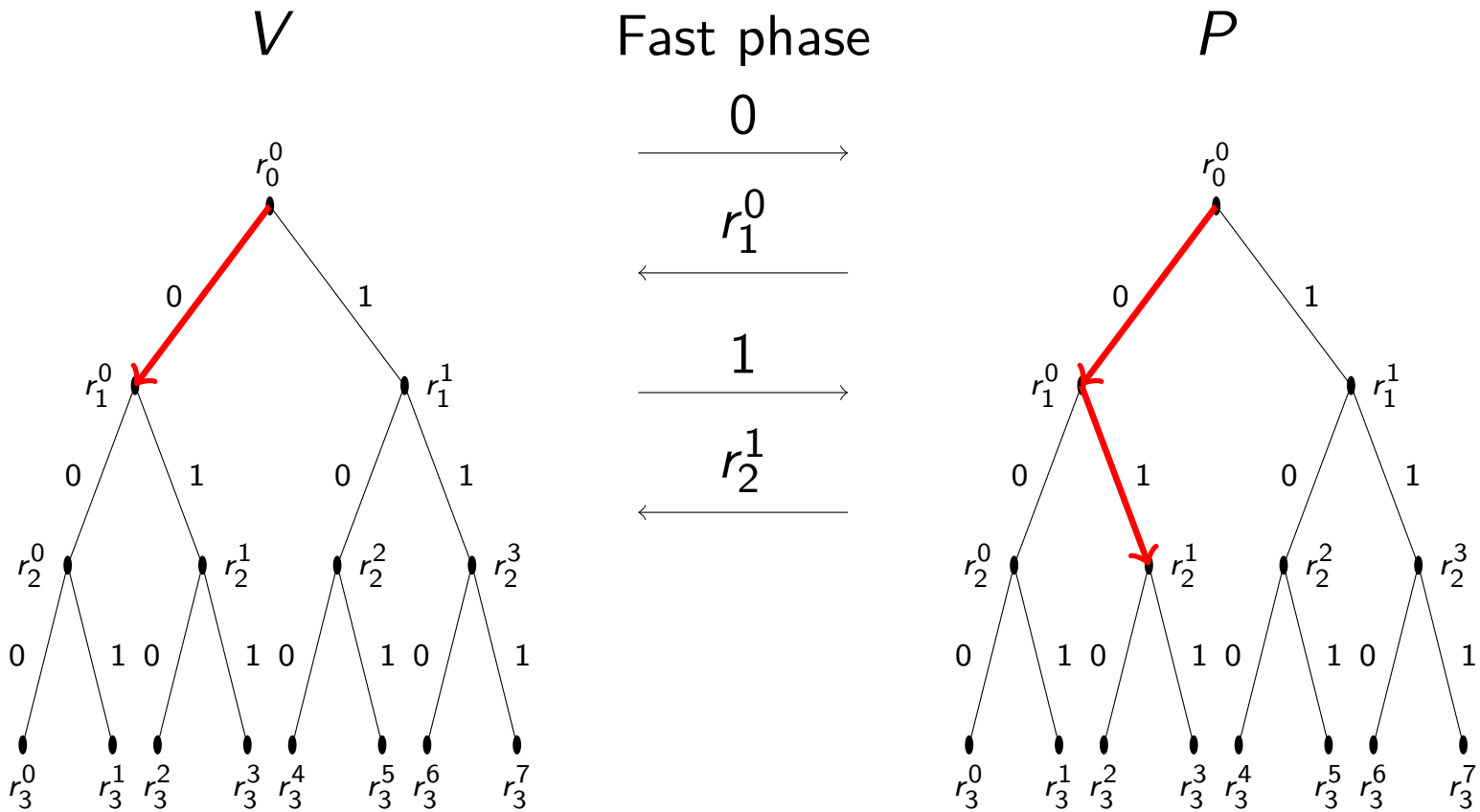
Avoine and Tchamkerten's protocol (2009)



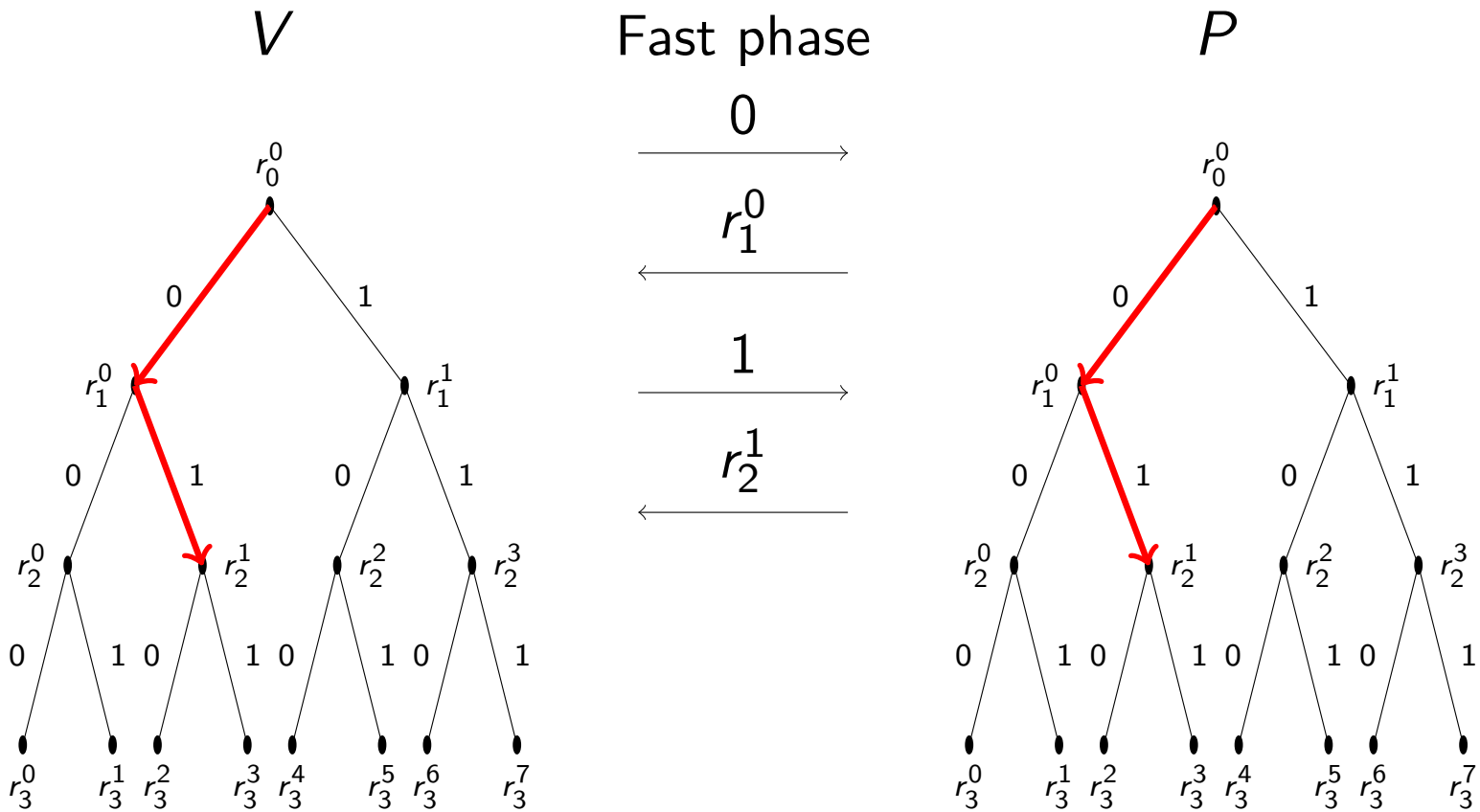
Avoine and Tchamkerten's protocol (2009)



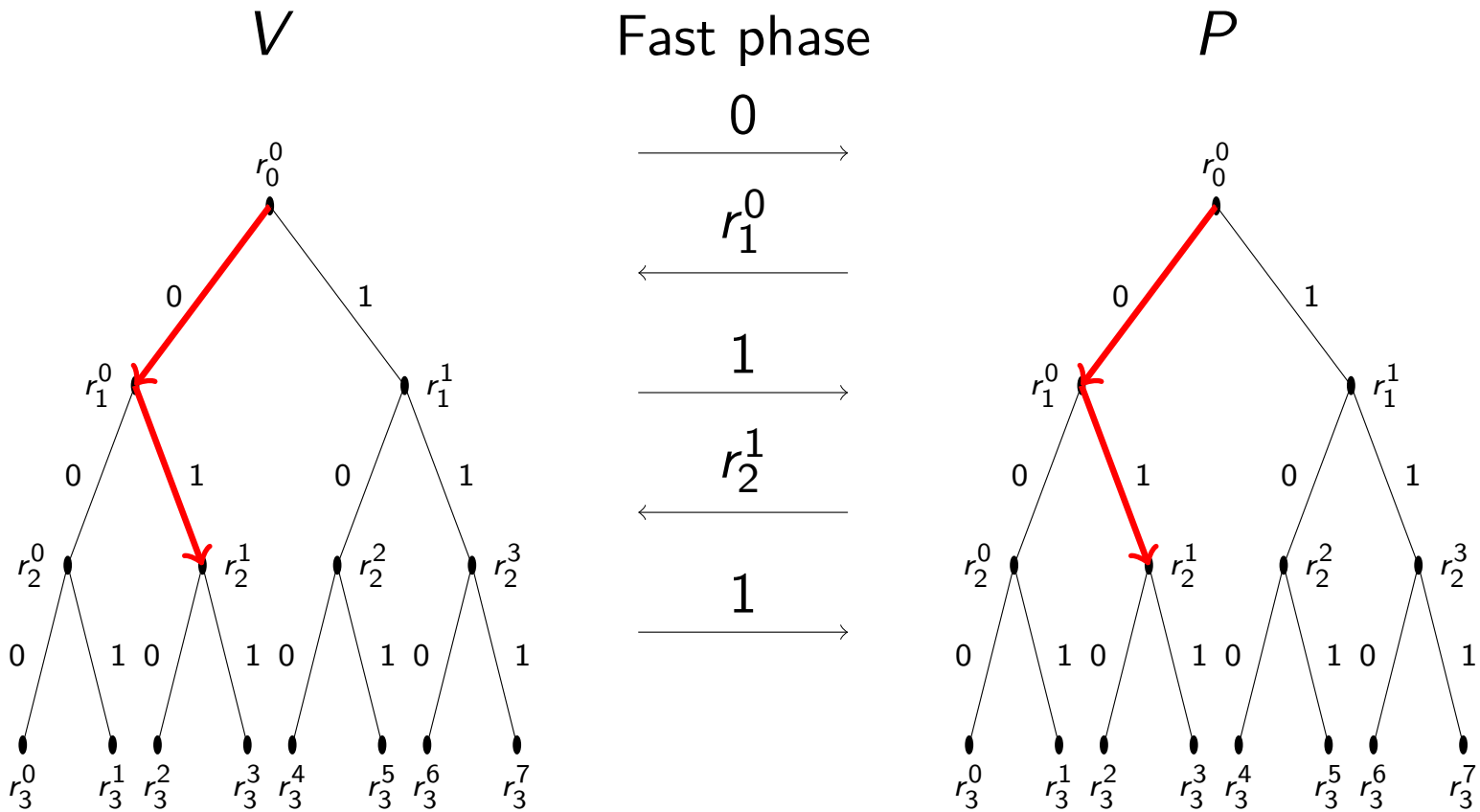
Avoine and Tchamkerten's protocol (2009)



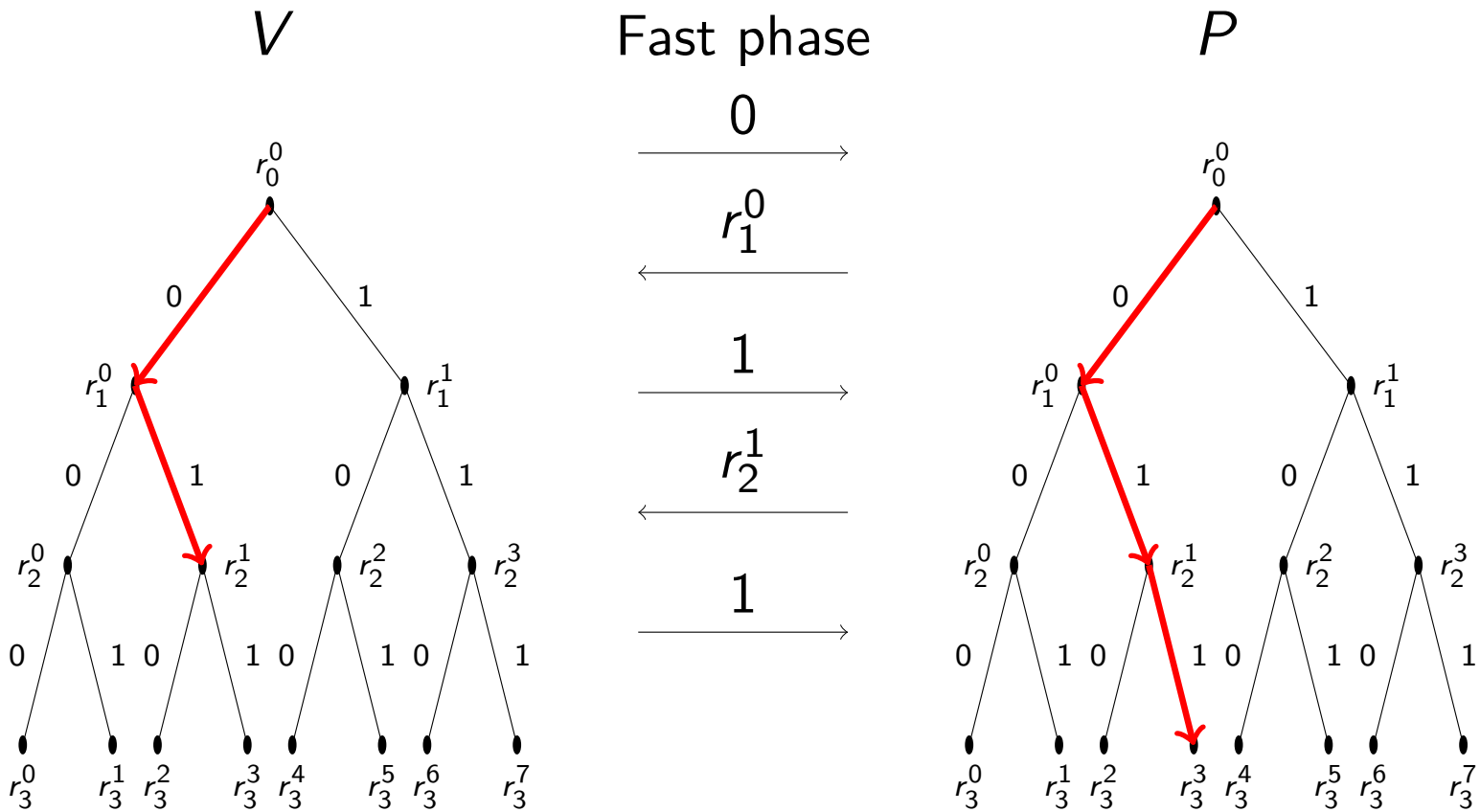
Avoine and Tchamkerten's protocol (2009)



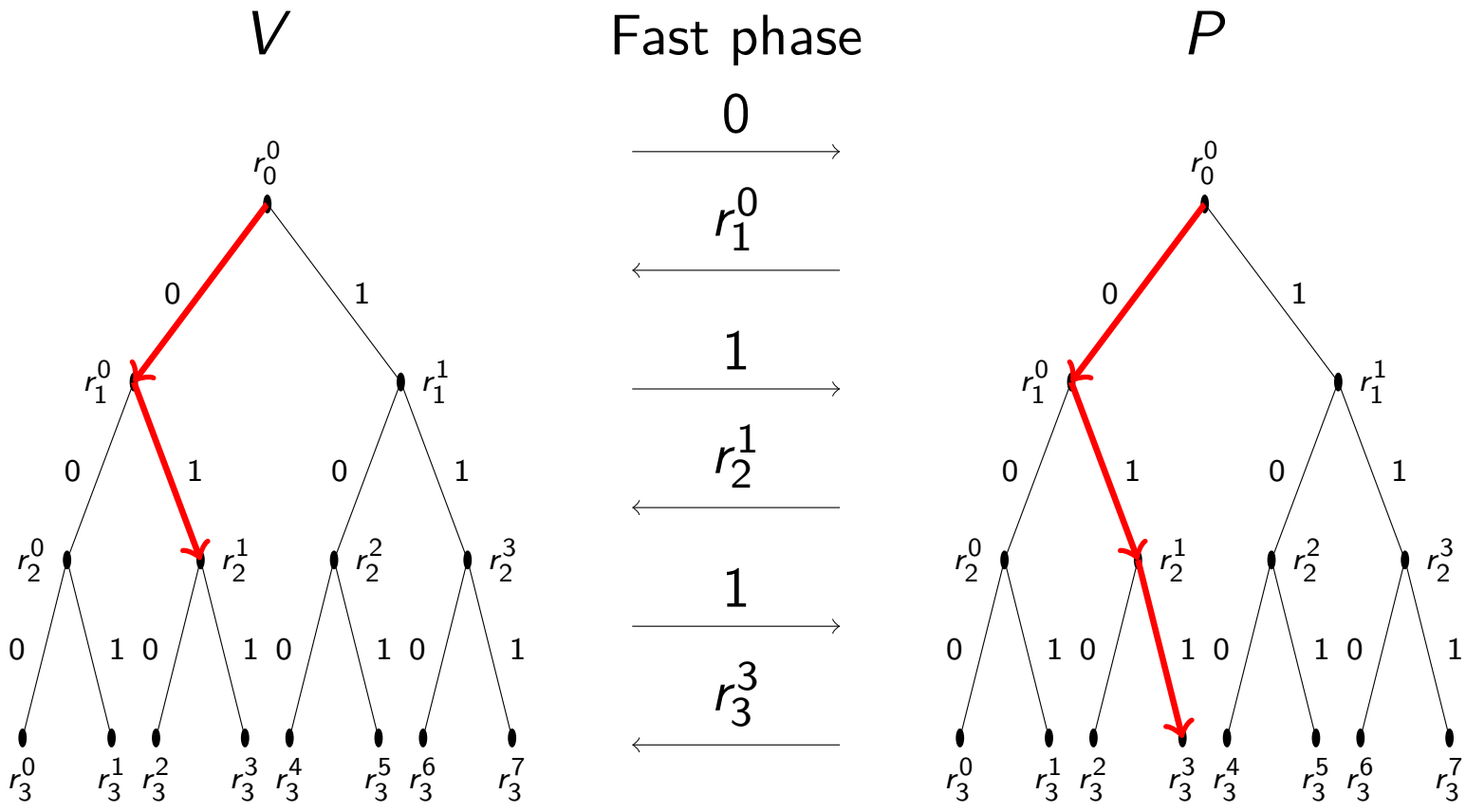
Avoine and Tchamkerten's protocol (2009)



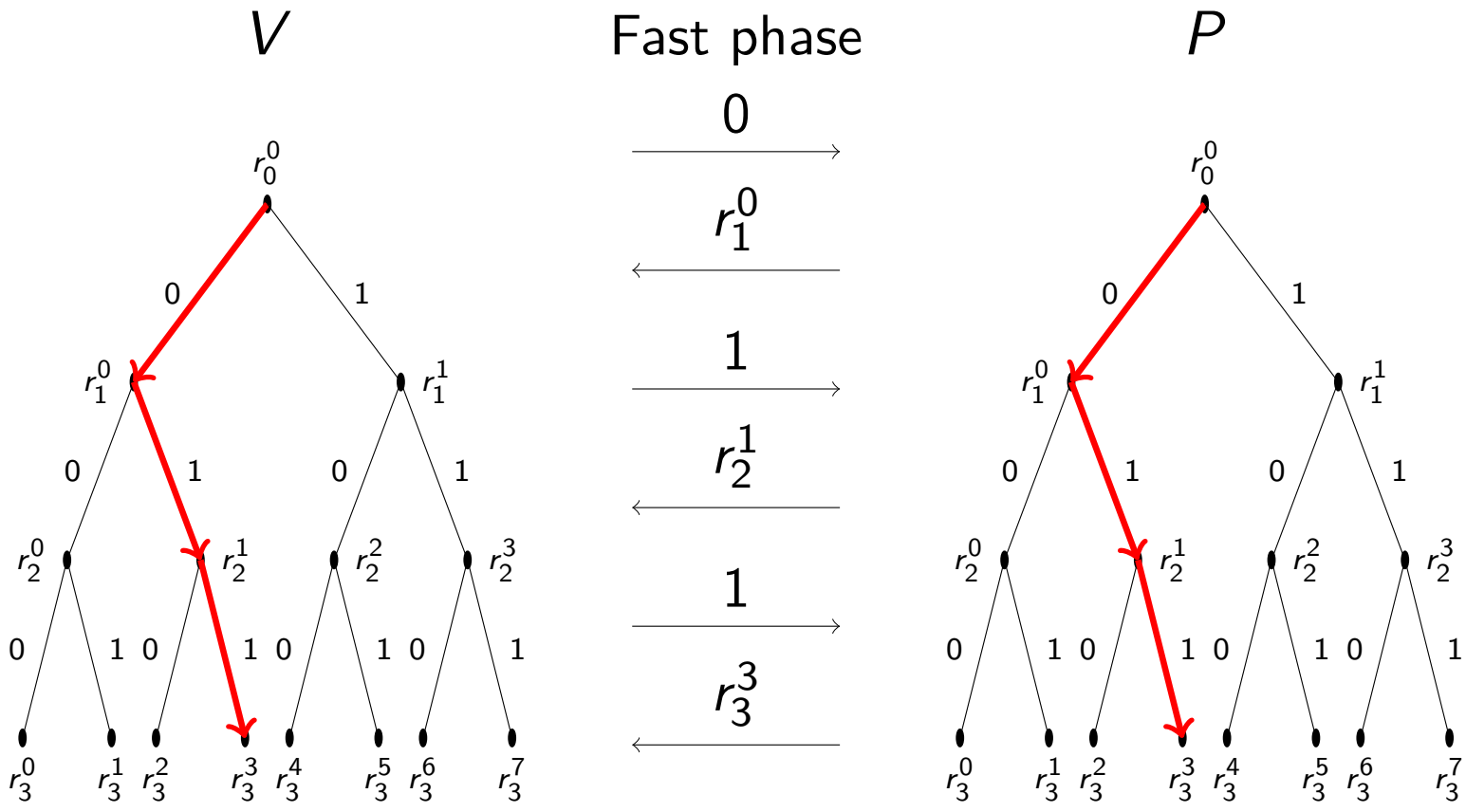
Avoine and Tchamkerten's protocol (2009)



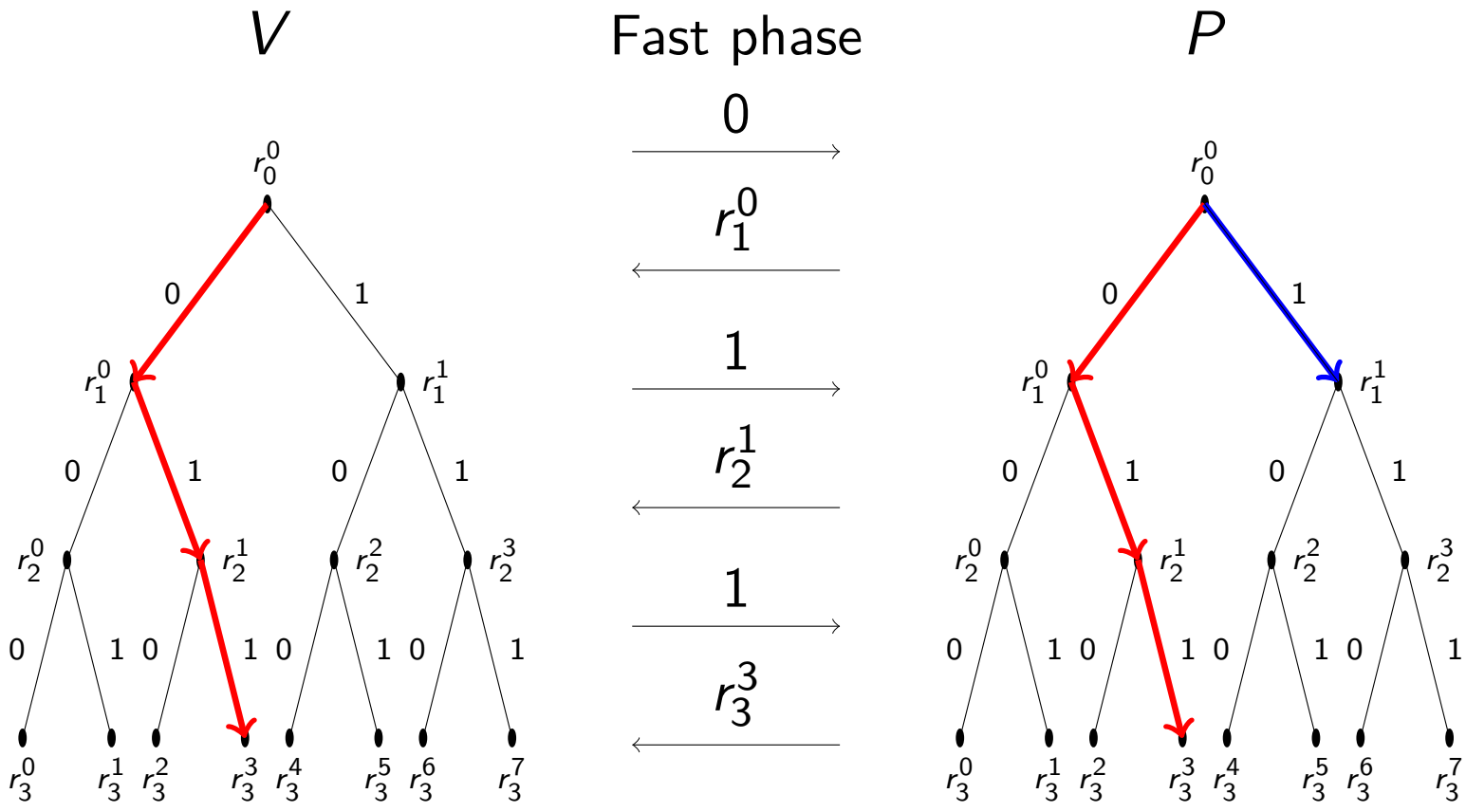
Avoine and Tchamkerten's protocol (2009)



Avoine and Tchamkerten's protocol (2009)



Avoine and Tchamkerten's protocol (2009)



Security analyis

	Mafia Fraud	Memory usage
HK protocol	$\left(\frac{3}{4}\right)^n$	linear in number of rounds
AT protocol	$\frac{1}{2^n}(1+\frac{n}{2})$	exponential in number of rounds

Research questions

1. Is there a lookup-based protocol that beats AT: $\frac{1}{2^n}(1 + \frac{n}{2})$?

Research questions

1. Is there a lookup-based protocol that beats AT: $\frac{1}{2^n}(1+\frac{n}{2})$?
   No, AT is optimal
2. Do we need an exponential memory to achieve $\frac{1}{2^n}(1+\frac{n}{2})$?

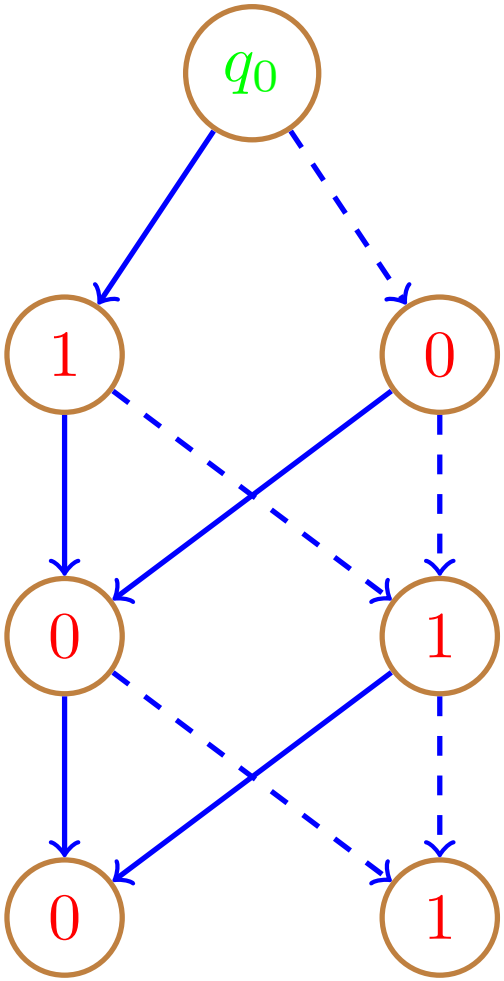
Research questions

1. Is there a lookup-based protocol that beats AT: $\frac{1}{2^n}(1+\frac{n}{2})$?
   No, AT is optimal
2. Do we need an exponential memory to achieve $\frac{1}{2^n}(1+\frac{n}{2})$?
   Yes, we can't do better than AT.
3. So, given a limit on the size of the lookup table, what's the optimal db protocol?

Research questions

1. Is there a lookup-based protocol that beats AT: $\frac{1}{2^n}(1+\frac{n}{2})$?
   No, AT is optimal
2. Do we need an exponential memory to achieve $\frac{1}{2^n}(1+\frac{n}{2})$?
   Yes, we can't do better than AT.
3. So, given a limit on the size of the lookup table, what's the optimal db protocol?
   We will answer that question (partially) in this talk.

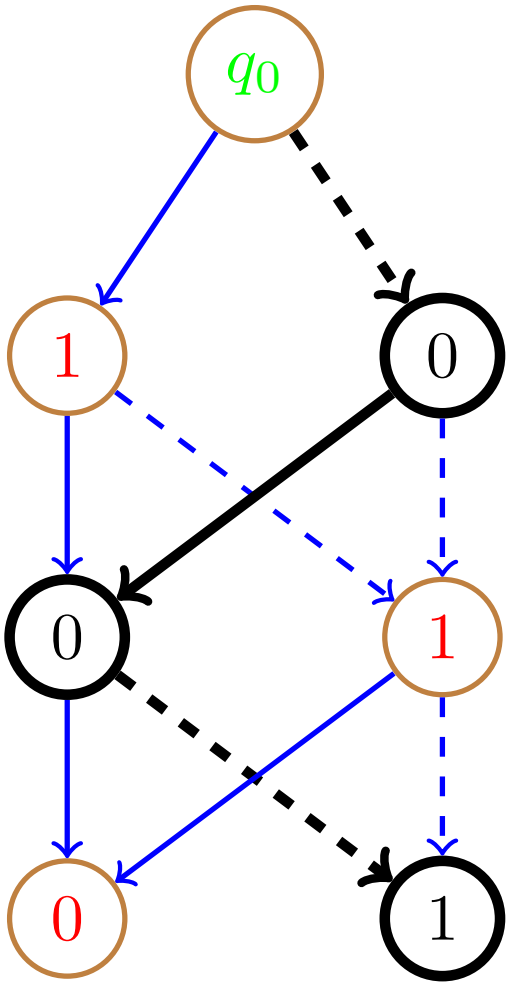
Modeling lookup-based DB protocols



$$A = (\Sigma, \Gamma, Q, q_0, \delta, \ell)$$

Σ is the set of input symbols
Γ is the set of output symbols
Q is the set of states
$q_0 \in Q$ is the initial state
$\delta: Q \times \Sigma \to Q$ is the transition function
$\ell: Q \to \Gamma$ is the state labeling function

Modeling lookup-based DB protocols

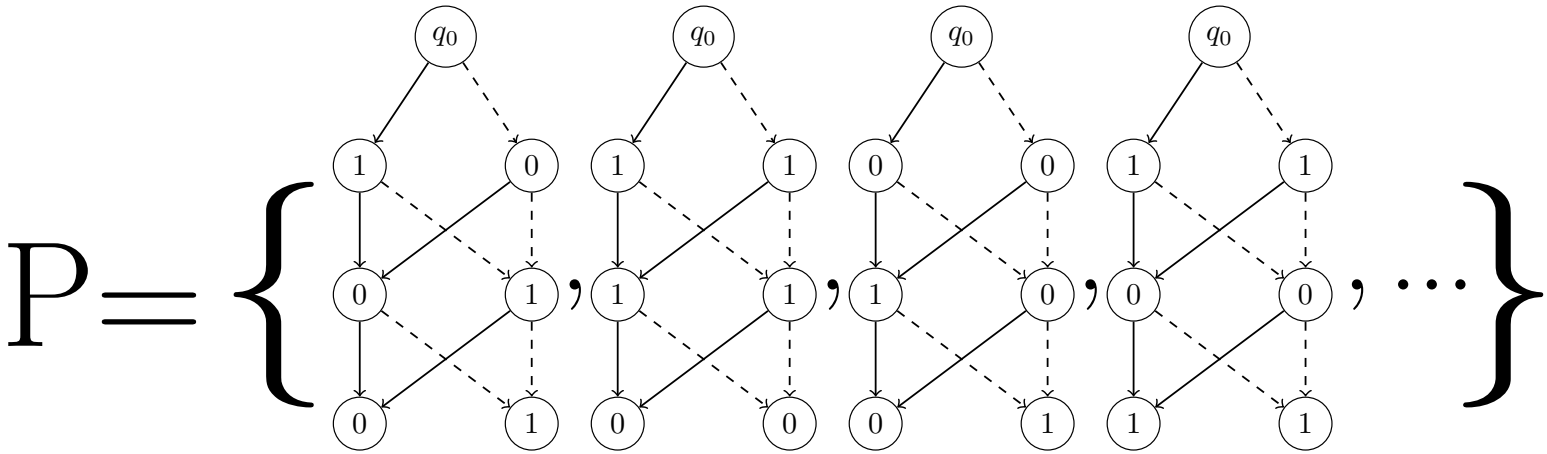


$$A = (\Sigma, \Gamma, Q, q_0, \delta, \ell)$$

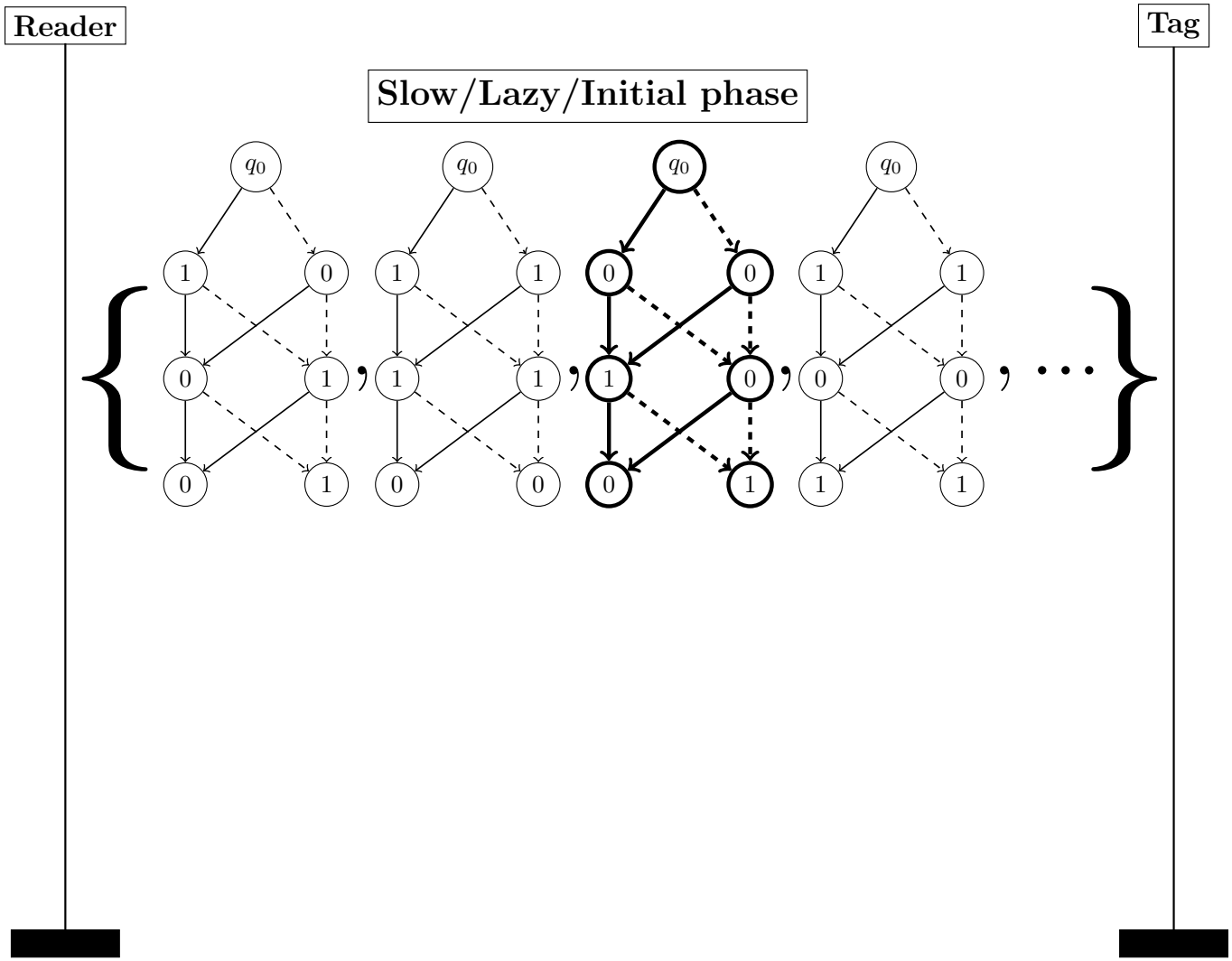
Σ is the set of input symbols
Γ is the set of output symbols
Q is the set of states
$q_0 \in Q$ is the initial state
$\delta: Q \times \Sigma \to Q$ is the transition function
$\ell: Q \to \Gamma$ is the state labeling function

$$\Omega_A\,(101) = 001$$

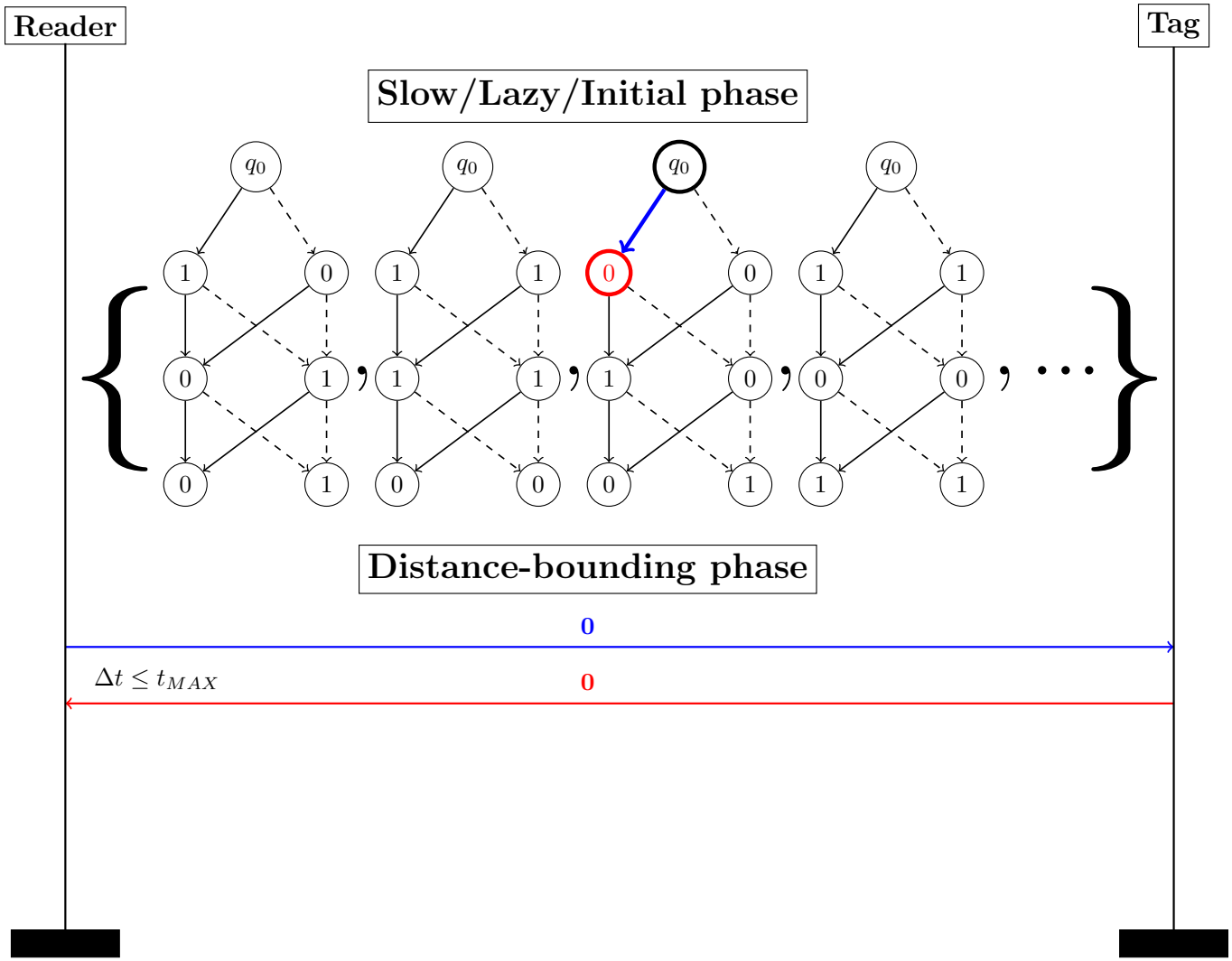
Modeling lookup-based DB protocols



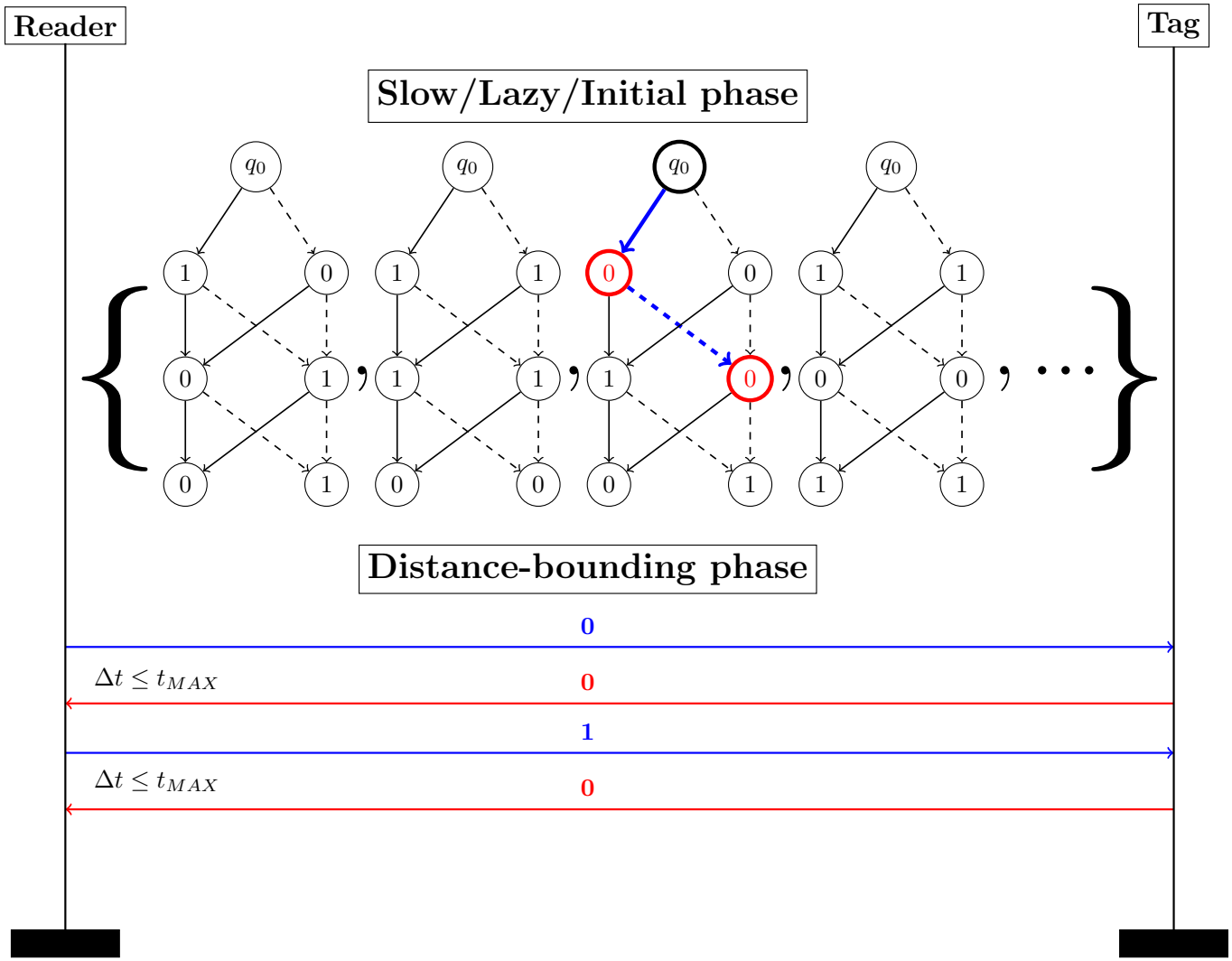
Protocol execution



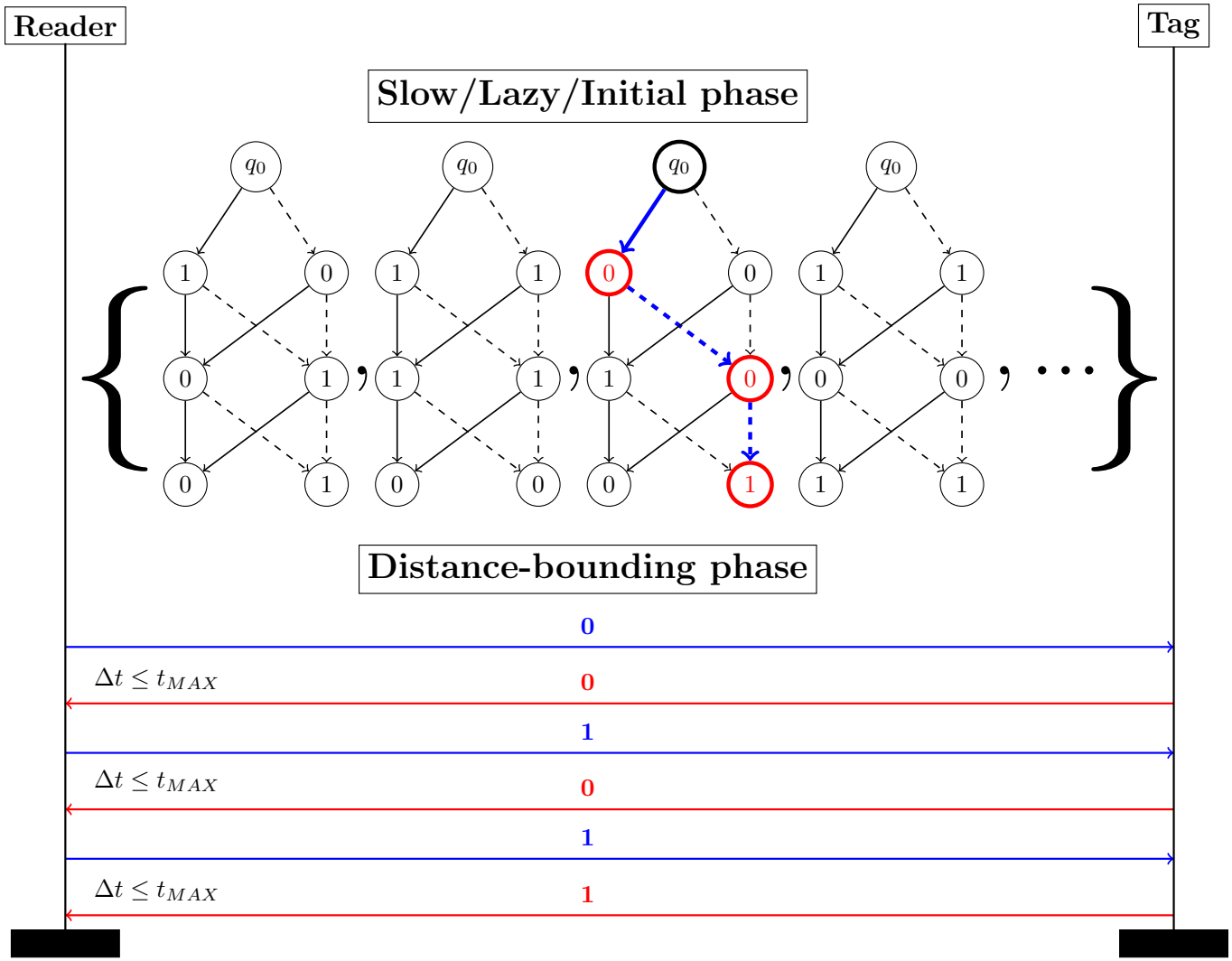
Protocol execution



Protocol execution



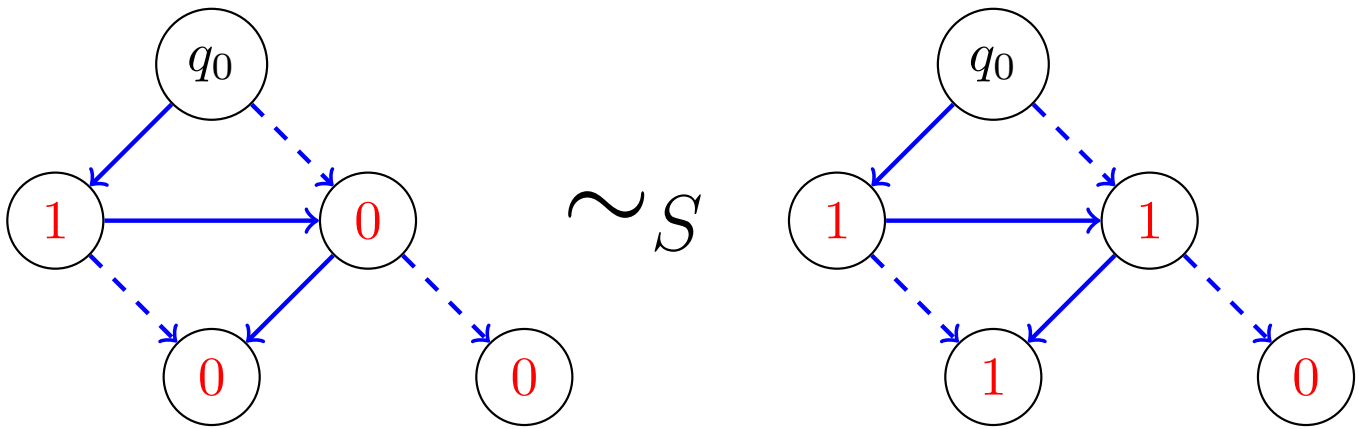
Protocol execution



Automata equivalence relations

- State-label-insensitive relation ($\sim_S$)

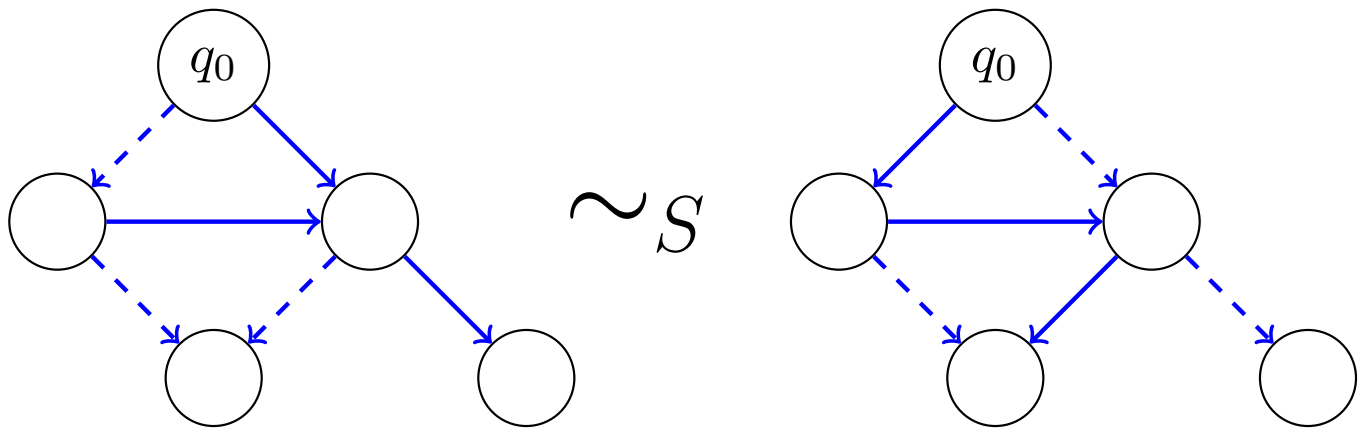
$$(\Sigma, \Gamma, Q, q_0, \delta, \ell) \sim_S (\Sigma, \Gamma, Q, q_0, \delta, \ell')$$



Automata equivalence relations

- State-label-insensitive relation ($\sim_S$)

$$(\Sigma, \Gamma, Q, q_0, \delta, \ell) \sim_S (\Sigma, \Gamma, Q, q_0, \delta, \ell')$$

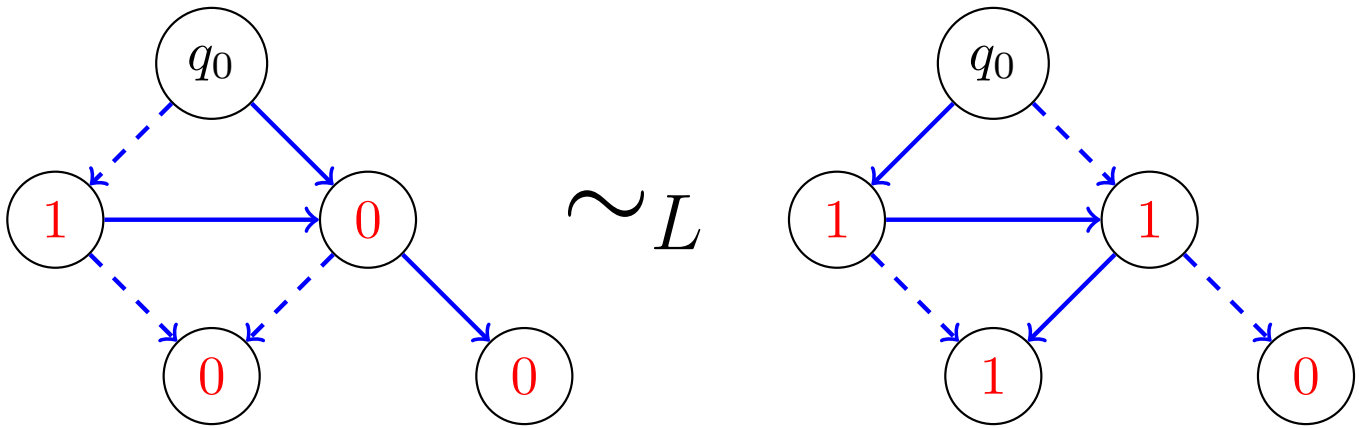


Automata equivalence relations

- Label-insensitive relation ($\sim_L$)

$$(\Sigma, \Gamma, Q, q_0, \delta, \ell) \sim_L (\Sigma, \Gamma, Q, q_0, \delta', \ell')$$

such that $\forall q \in Q : \ \{\delta(q, c) \mid c \in \Sigma\} = \{\delta'(q, c) \mid c \in \Sigma\}$.

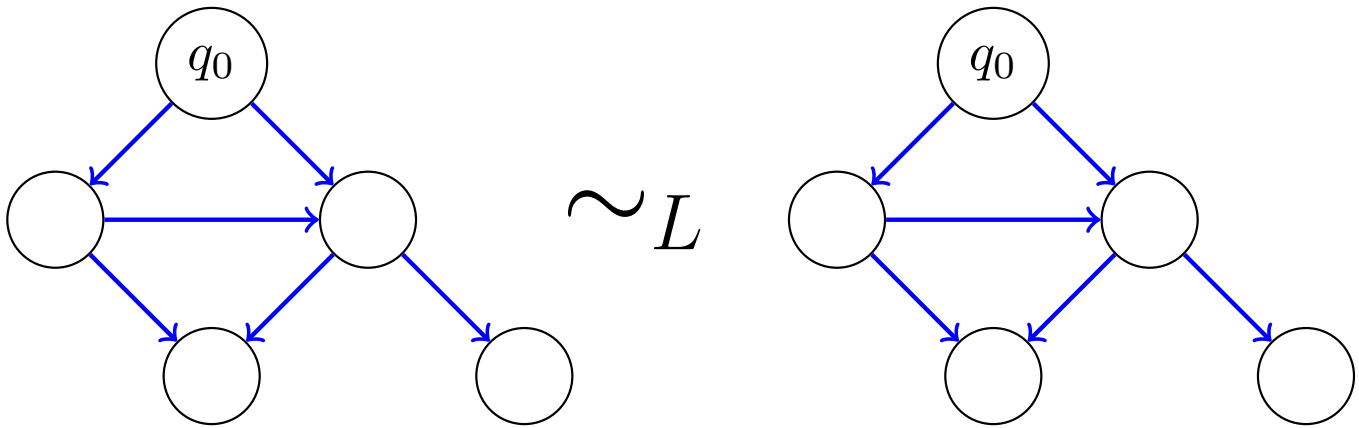


Automata equivalence relations

- Label-insensitive relation ($\sim_L$)

$$(\Sigma, \Gamma, Q, q_0, \delta, \ell) \sim_L (\Sigma, \Gamma, Q, q_0, \delta', \ell')$$

such that $\forall q \in Q : \ \{\delta(q, c) \mid c \in \Sigma\} = \{\delta'(q, c) \mid c \in \Sigma\}$.



Consistency and closeness

All lookup-based protocols are consistent and closed with respect to $\sim_S$. Except for Poulidor (Trujillo et al. 2010) which is consistent and closed with respect to $\sim_L$.

Consistency and closeness

All lookup-based protocols are consistent and closed with respect to $\sim_S$. Except for Poulidor (Trujillo et al. 2010) which is consistent and closed with respect to $\sim_L$.

- A protocol P is consistent w.r.t $\sim_R$ iff

$$A, A' \in P : A \sim_R A'$$

- A protocol P is closed under $\sim_R$ iff

$$\forall (A, A') \in \sim_R : A \in P \implies A' \in P$$

A transformation towards optimality

- The closure of P w.r.t $\sim_R$, denoted by $\overline{P}^{\mathrm{R}}$, is the minimal superset of P that is closed under $\sim_R$.

Theorem

For any layered lookup-based protocol P the following holds:

$$\mathcal{M}(P) \geq \mathcal{M}\left(\overline{P}^{\mathrm{S}}\right) \geq \mathcal{M}\left(\overline{\{A\}}^{\mathrm{L}}\right),$$

for some $A \in P$.
Moreover, the size of $\overline{\{A\}}^{\mathrm{L}}$ is at most the size of P.

A transformation towards optimality

- The closure of P w.r.t $\sim_R$, denoted by $\overline{P}^{\mathrm{R}}$, is the minimal superset of P that is closed under $\sim_R$.

Theorem

For any layered lookup-based protocol P the following holds:

$$\mathcal{M}(P) \geq \mathcal{M}\left(\overline{P}^{\mathrm{S}}\right) \geq \mathcal{M}\left(\overline{\{A\}}^{\mathrm{L}}\right),$$

for some $A \in P$.
Moreover, the size of $\overline{\{A\}}^{\mathrm{L}}$ is at most the size of P.

- Protocols with random state labels and transition labels are better.

A transformation towards optimality

- The closure of P w.r.t $\sim_R$, denoted by $\overline{P}^{\mathrm{R}}$, is the minimal superset of P that is closed under $\sim_R$.

Theorem

For any layered lookup-based protocol P the following holds:

$$\mathcal{M}(P) \geq \mathcal{M}\left(\overline{P}^{\mathrm{S}}\right) \geq \mathcal{M}\left(\overline{\{A\}}^{\mathrm{L}}\right),$$

for some $A \in P$.
Moreover, the size of $\overline{\{A\}}^{\mathrm{L}}$ is at most the size of P.

- Protocols with random state labels and transition labels are better.
- The transformation $\overline{\{A\}}^{\mathrm{L}}$ of P is an improvement.

A transformation towards optimality

- The closure of P w.r.t $\sim_R$, denoted by $\overline{P}^{\mathrm{R}}$, is the minimal superset of P that is closed under $\sim_R$.

Theorem

For any layered lookup-based protocol P the following holds:

$$\mathcal{M}(P) \geq \mathcal{M}\left(\overline{P}^{\mathrm{S}}\right) \geq \mathcal{M}\left(\overline{\{A\}}^{\mathrm{L}}\right),$$

for some $A \in P$.
Moreover, the size of $\overline{\{A\}}^{\mathrm{L}}$ is at most the size of P.

- Protocols with random state labels and transition labels are better.
- The transformation $\overline{\{A\}}^{\mathrm{L}}$ of P is an improvement.
- Let $A \in \mathit{Tree}$, then $HK \bigcup \mathit{Tree}$ is not better than $\overline{\{A\}}^{\mathrm{L}}$.

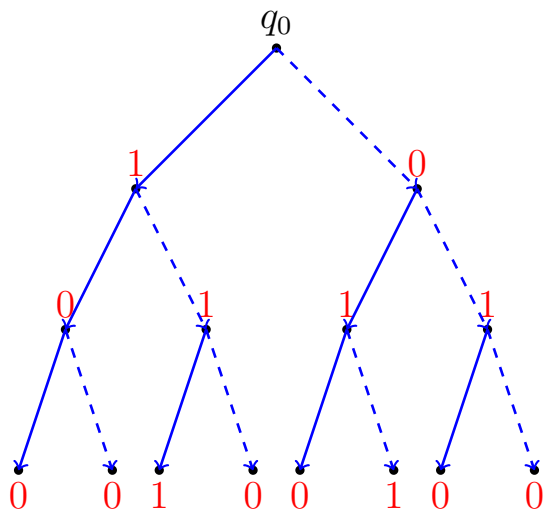
Layered Protocols

Definition

A protocol P is layered if and only if in any automaton two different input sequences reach different states, i.e.,

$$\forall A \in P, \forall x, y \in \Sigma^* : |x| \neq |y| \implies \hat{\delta}(x) \neq \hat{\delta}(y).$$

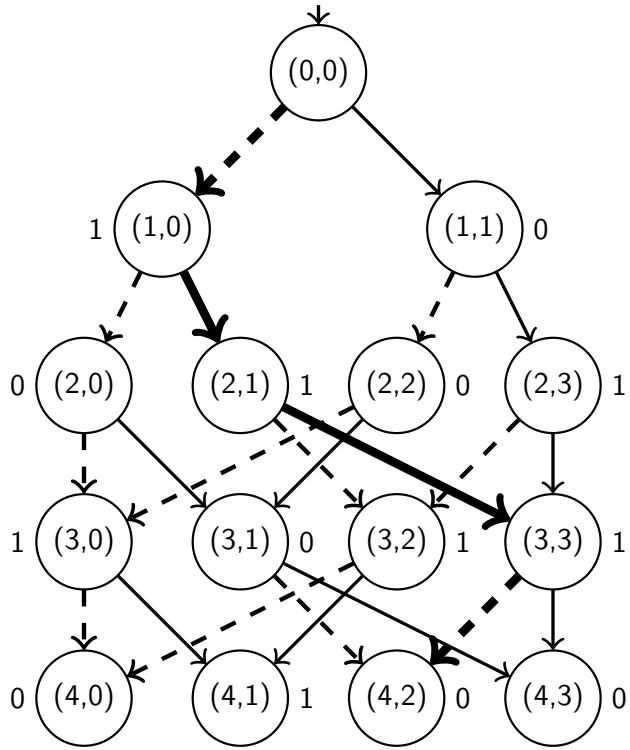
Example: Tree-based (Avoine et al. 2009).



Optimal protocol

Theorem

A layered protocol with maximum girth, given a bound on the number of states, is either optimal or can be made optimal via application of the $\sim_L$-closure.

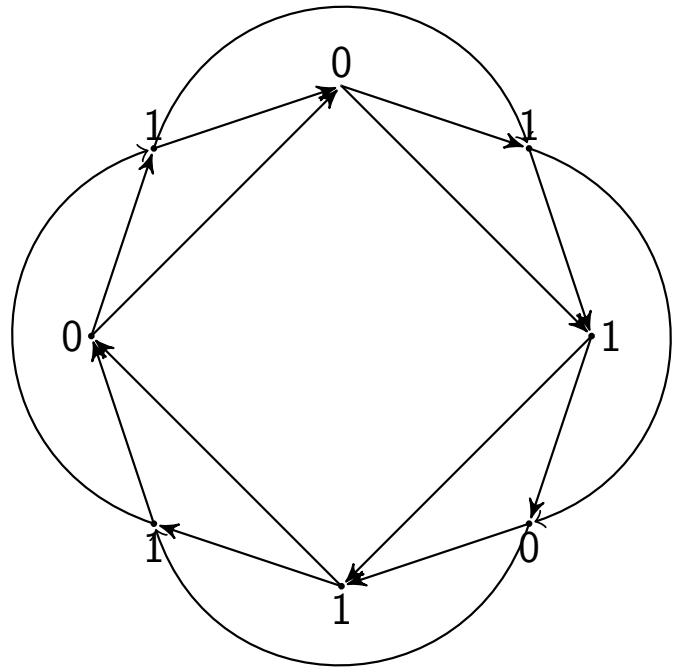


Comparative analysis

y	Nondominated instances in I_y	**Attribute values** n	p_m	p_d	p_t	b	c	m	f	total
2^{-32}	KA-{37, 0.85}	37	$2^{-32.0}$	$2^{-2.0}$	$2^{0.0}$	1	1	0Kb	*false*	2
	BC-{32}	32	$2^{-32.0}$	$2^{-32.0}$	$2^{0.0}$	1	2	0Kb	*true*	97
	Tree-{48, 6}	48	$2^{-32.0}$	$2^{-21.0}$	$2^{0.0}$	1	1	1Kb	*false*	156
	TMA-{53}	53	$2^{-32.0}$	$2^{-32.0}$	$2^{0.0}$	1	1	0Kb	*false*	1
	SwissKnife-{32}	32	$2^{-32.0}$	$2^{-13.0}$	$2^{-13.0}$	1	2	1Kb	*true*	97
	Modular-{39, 32}	39	$2^{-32.0}$	$2^{-16.0}$	$2^{0.0}$	1	1	2Kb	*false*	3
	SKI-{78, 2}	78	$2^{-32.0}$	$2^{-32.0}$	$2^{-78.0}$	2	1	1Kb	*false*	51
2^{-48}	Poulidor-{61}	61	$2^{-48.0}$	$2^{-25.0}$	$2^{0.0}$	1	1	0Kb	*false*	1
	KA-{53, 0.95}	53	$2^{-48.0}$	$2^{-1.0}$	$2^{0.0}$	1	1	0Kb	*false*	4
	BC-{48}	48	$2^{-48.0}$	$2^{-48.0}$	$2^{0.0}$	1	2	0Kb	*true*	81
	Tree-{72, 6}	72	$2^{-48.0}$	$2^{-32.0}$	$2^{0.0}$	1	1	2Kb	*false*	120
	TMA-{80}	80	$2^{-48.0}$	$2^{-48.0}$	$2^{0.0}$	1	1	0Kb	*false*	1
	SwissKnife-{48}	48	$2^{-48.0}$	$2^{-19.0}$	$2^{-19.0}$	1	2	1Kb	*true*	81
	Modular-{58, 32}	58	$2^{-48.0}$	$2^{-24.0}$	$2^{0.0}$	1	1	2Kb	*false*	4
	SKI-{116, 2}	116	$2^{-48.0}$	$2^{-48.0}$	$2^{-116.0}$	2	1	1Kb	*false*	13

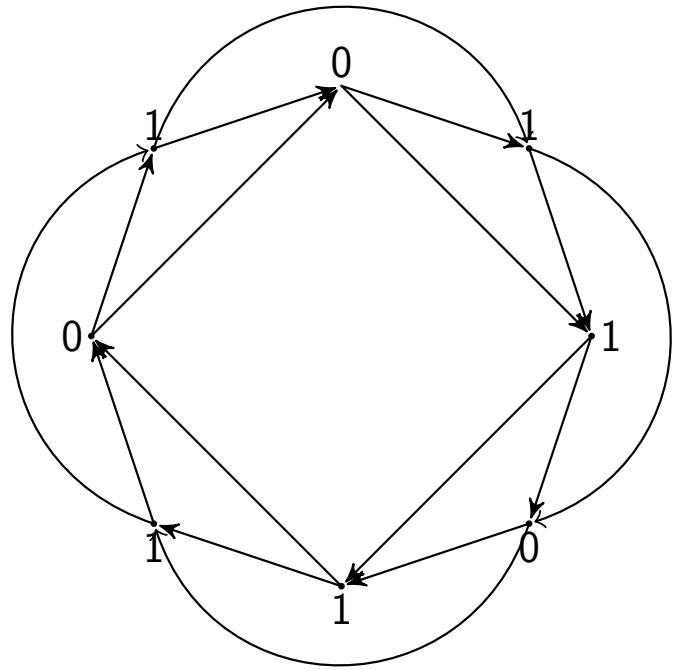
Poulidor and Cayley graphs

- Poulidor is a Cayley graph



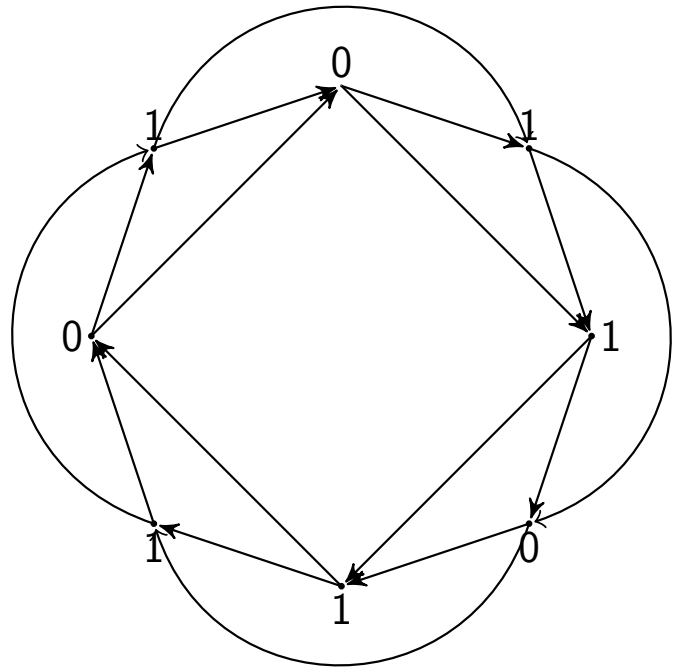
Poulidor and Cayley graphs

- Poulidor is a Cayley graph
- Cayley graphs tend to have large girth



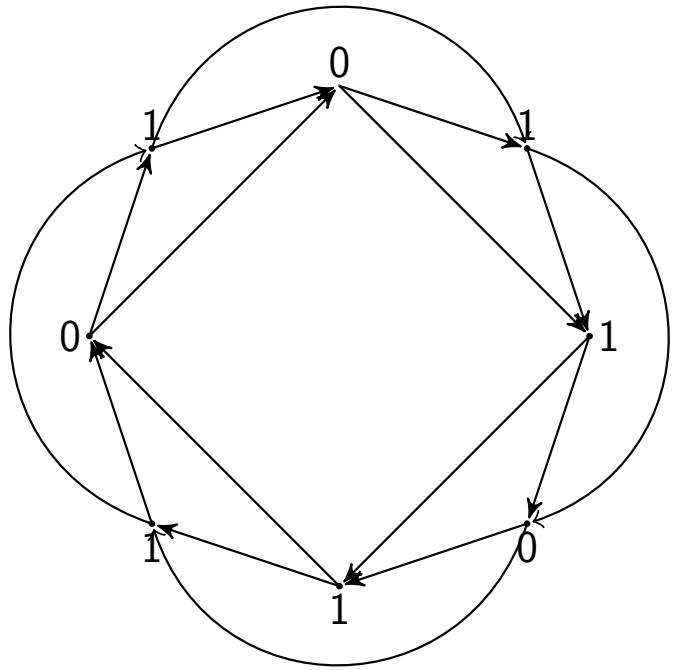
Poulidor and Cayley graphs

- Poulidor is a Cayley graph
- Cayley graphs tend to have large girth
- Large-girth graphs with expander properties have been used to design hash functions



Poulidor and Cayley graphs

- Poulidor is a Cayley graph
- Cayley graphs tend to have large girth
- Large-girth graphs with expander properties have been used to design hash functions
- So, is there a connection between distance-bounding and graph-based hash functions?



Conclusions

- Better understanding and generic treatment of lookup-based distance-bounding protocols.
- Fundamental results on security and memory usage.
- First lookup-based protocol that can be proven optimal
- Connection with graph-based hash functions