Parameterized verification of distributed shared-memory systems

Under the supervision of

Nathalie Bertrand   Nicolas Markey   Ocan Sankur

Nicolas Waldburger

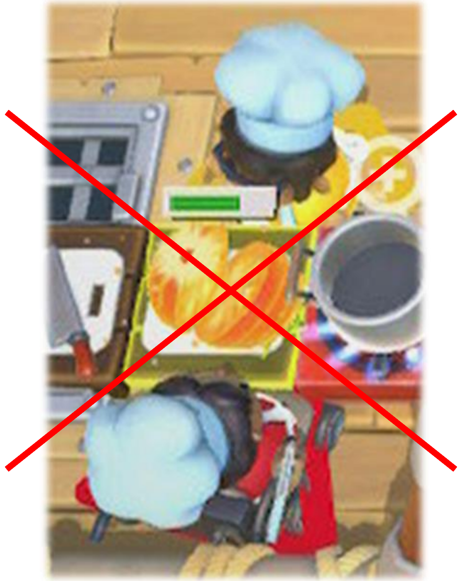
Introduction

Cooking with friends

Distributed algorithms

Distributed problems

Mutual exclusion



Consensus

Verification of distributed algorithms

Does my algorithm avoid  ?

Verification of distributed algorithms

Does my algorithm avoid  ?

Verification of distributed algorithms

Does my algorithm avoid  ?

Objective: mathematically verify distributed algorithm to give guarantees about their safety.

From algorithms to automata-based models

Peterson's mutual exclusion algorithm [Pet81]

For process $i \in \{0,1\}$:

```
while true:
        do non-critical things ;
        flag_i = true ; turn := 1 − i ;
        wait until (flag_{1−i} == false or turn == i)
        do critical things;
        flag_i = false ;
```


[Pet81] Peterson, G.: *Myths about the Mutuel Exclusion Problem.* Information Processing Letters, 1981

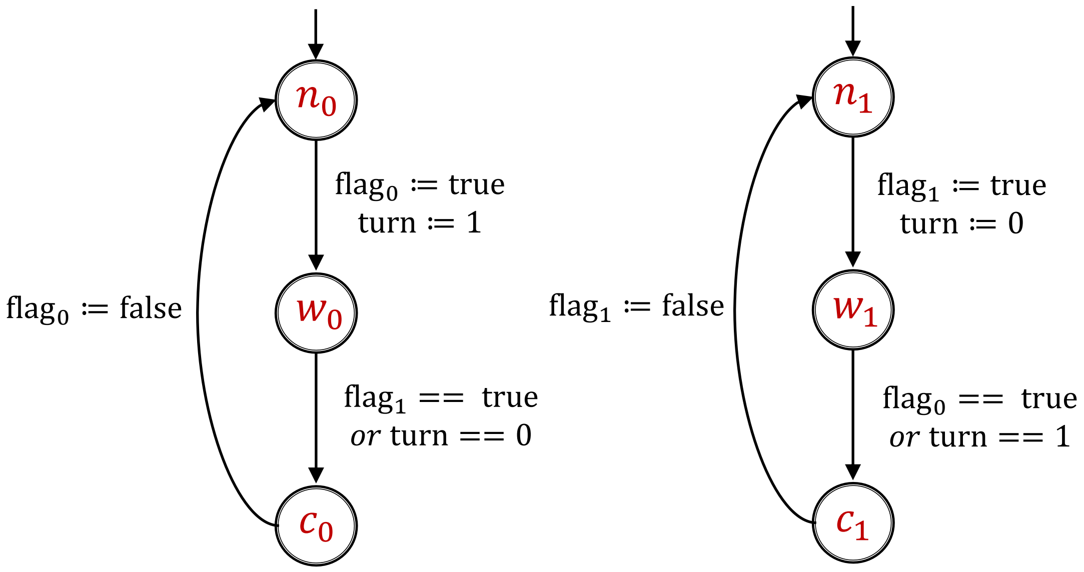
From algorithms to automata-based models

Peterson's mutual exclusion algorithm [Pet81]

For process $i \in \{0,1\}$:

while true:
    n_i do non-critical things ;
       $\text{flag}_i = \text{true}$; $\text{turn} \coloneqq 1 - i$;
    w_i **wait until** $(\text{flag}_{1-i} == \text{false}\ or\ \text{turn} == i)$
    c_i do critical things;
       $\text{flag}_\text{i} = \text{false}$;

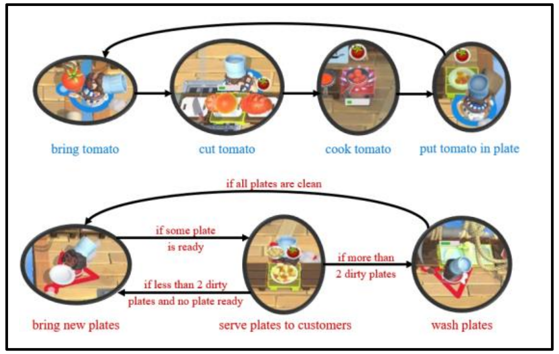
Automata-based model



Mutual exclusion = not c_0 and c_1 simultaneously

[Pet81] Peterson, G.: *Myths about the Mutuel Exclusion Problem.* Information Processing Letters, 1981

Model Checking

Does 

distributed algorithm

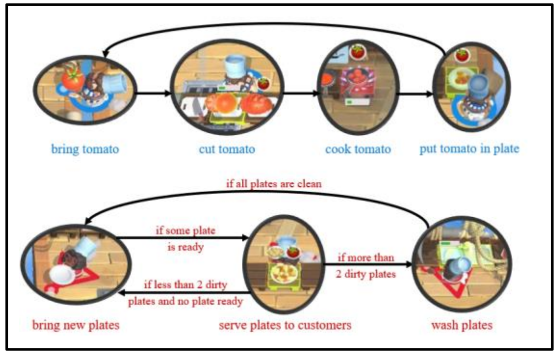
satisfy 

requirement

?

Model Checking

Does

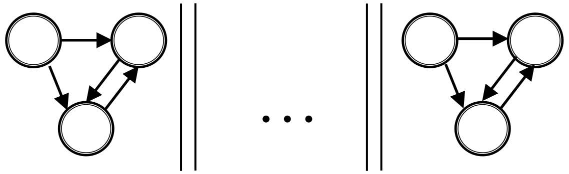


satisfy

?

distributed algorithm

requirement



$\models$

$\mathbf{G}(\neg \text{fire})$

?

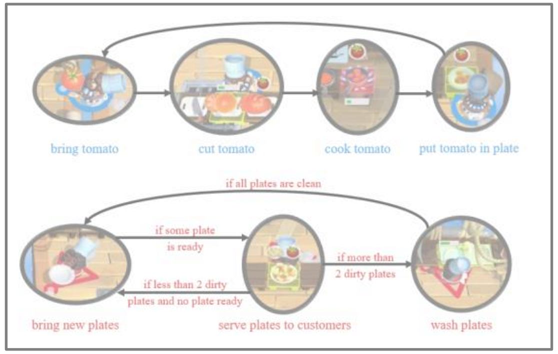
model

model-checking algorithm

property

Model Checking

Does

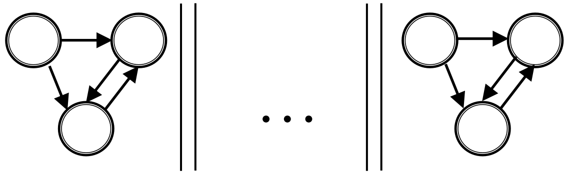


satisfy

?

distributed algorithm

requirement



$\models$

$\mathbf{G}(\neg\mathrm{fire})$

?

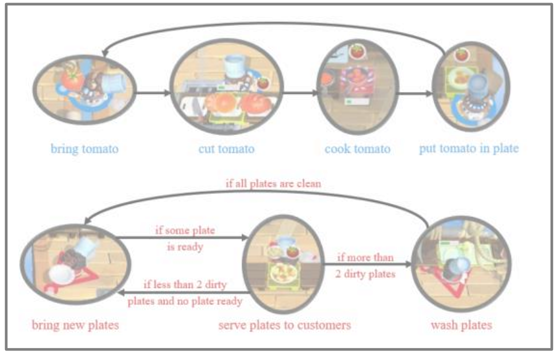
model

model-checking algorithm

property

Model Checking

Does

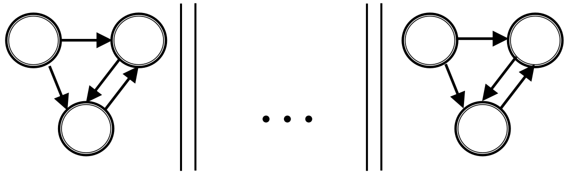


distributed algorithm

satisfy

requirement

?



$\models$

$\mathbf{G}(\neg\mathrm{fire})$

?

model

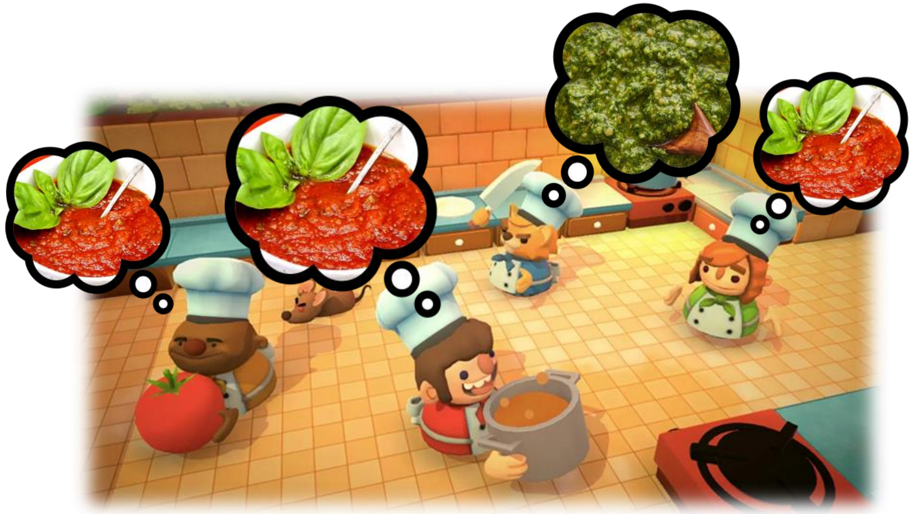
model-checking algorithm

property

Theoretical approach:
- identify relevant models and relevant properties for these models,
- study decidability and complexity questions.

The number of processes as a parameter

Consensus algorithms: designed for any number n of participants.



The number of processes as a parameter

Consensus algorithms: designed for any number n of participants.

The number of processes as a parameter

Consensus algorithms: designed for any number n of participants.

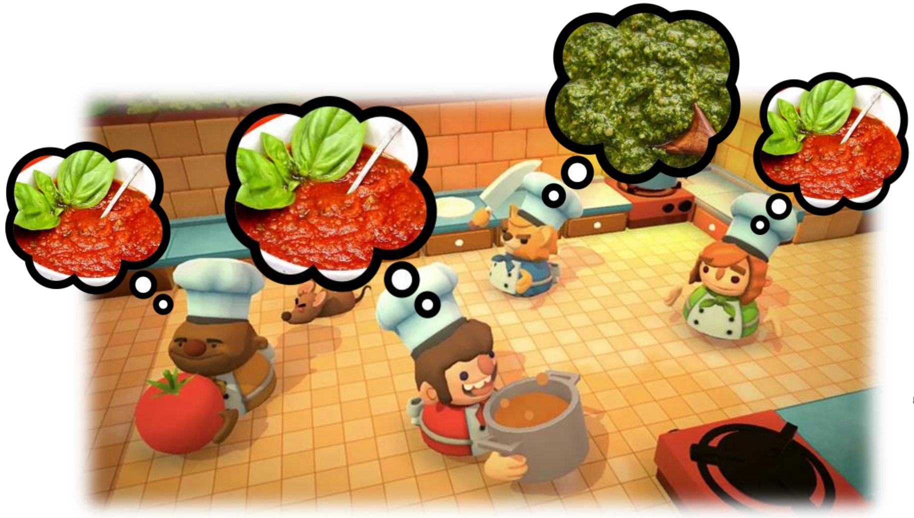
The number of processes as a parameter

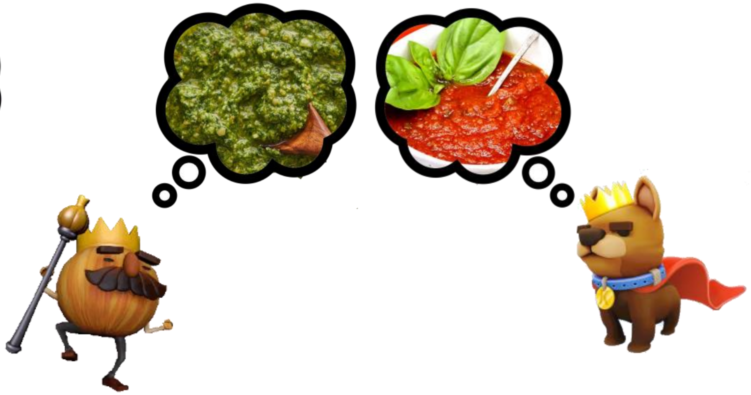
Consensus algorithms: designed for any number n of participants.

. . .

The number of processes as a parameter

Consensus algorithms: designed for any number n of participants.





. . .

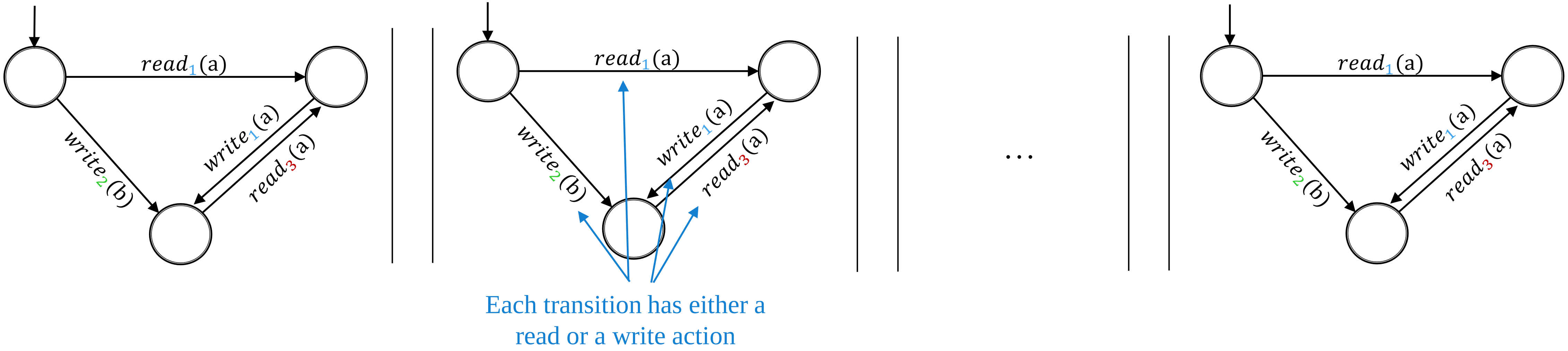
Parameterized verification: is the algorithm correct for every n?

First part
Shared-memory systems

A shared-memory model: ASMS

Asynchronous Shared-Memory Systems (ASMS) [EGM13]

Processes = identical finite-state machines, behaving asynchronously



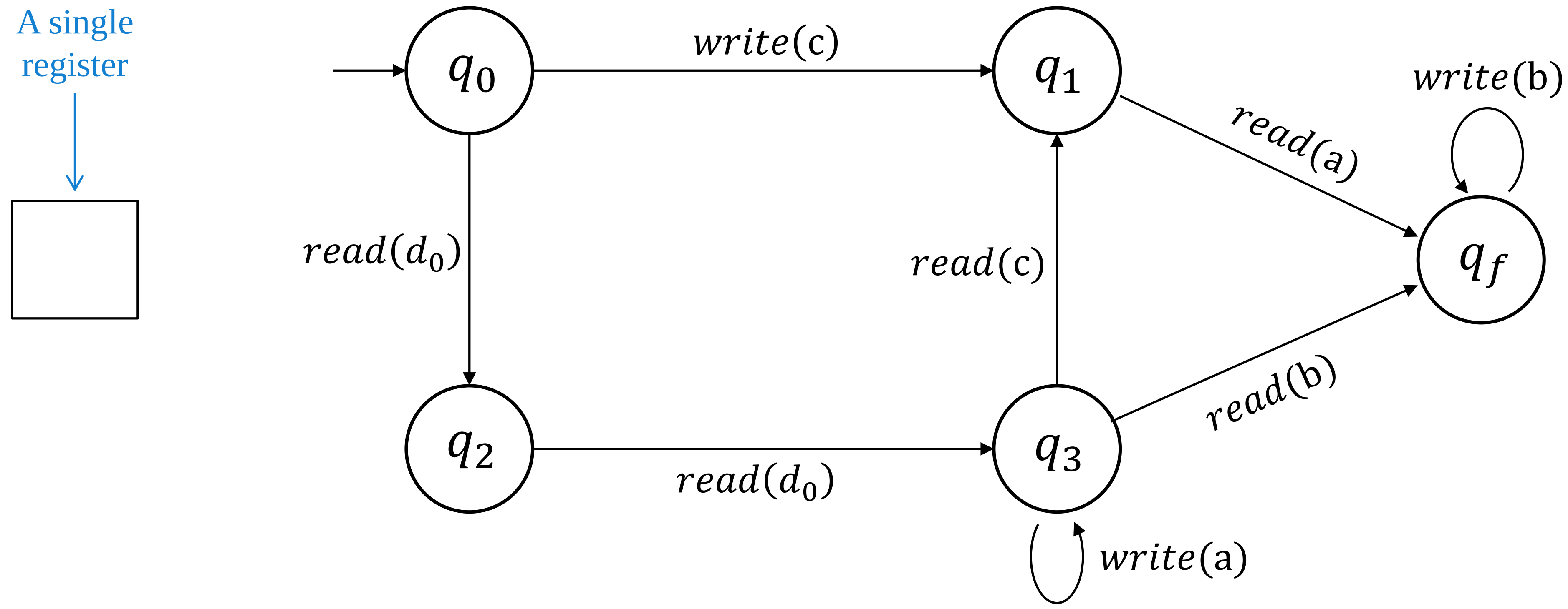
Shared memory

1	2	3
a	b	a

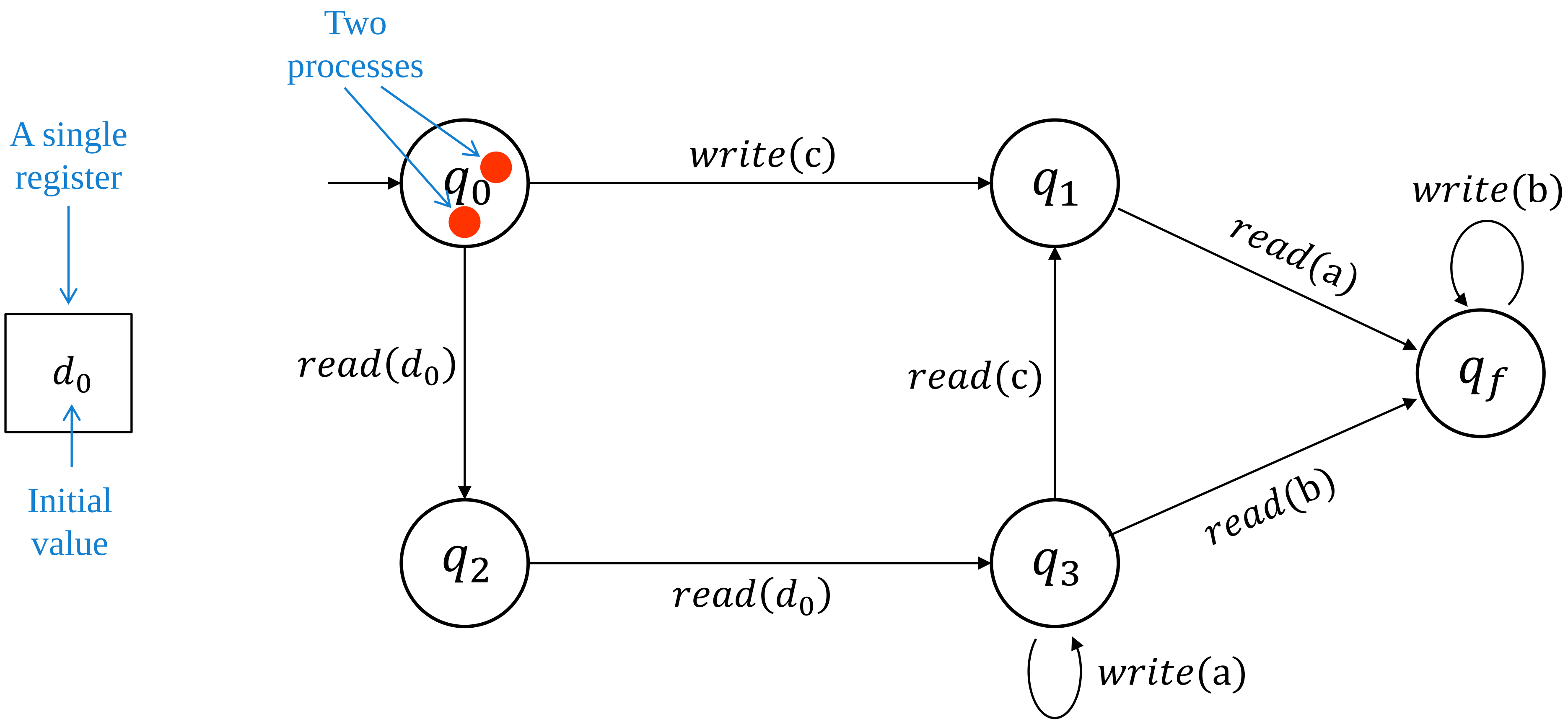
- Finite number of registers
- Finite set of values
- A special initial value

[EGM13] Esparza, J., Ganty, P., Majumdar, R.: *Parameterized Verification of Asynchronous Shared-Memory Systems*. CAV 2013.

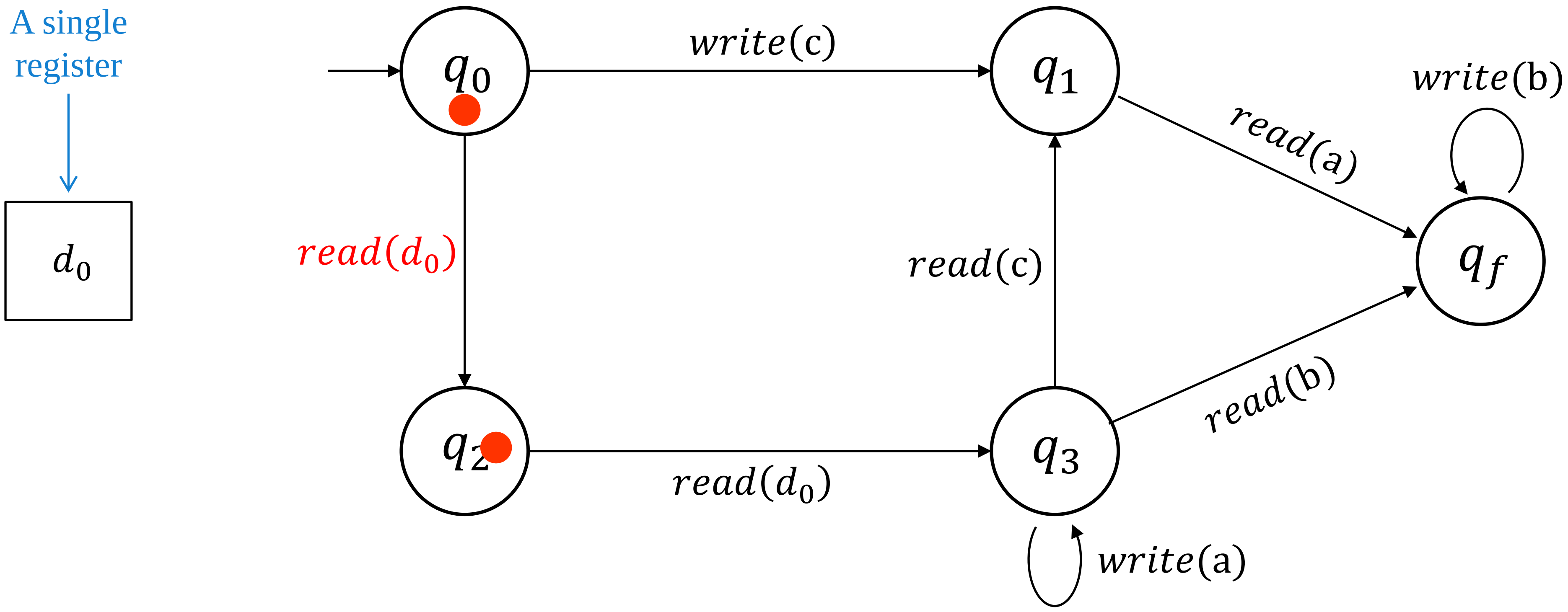
A small example



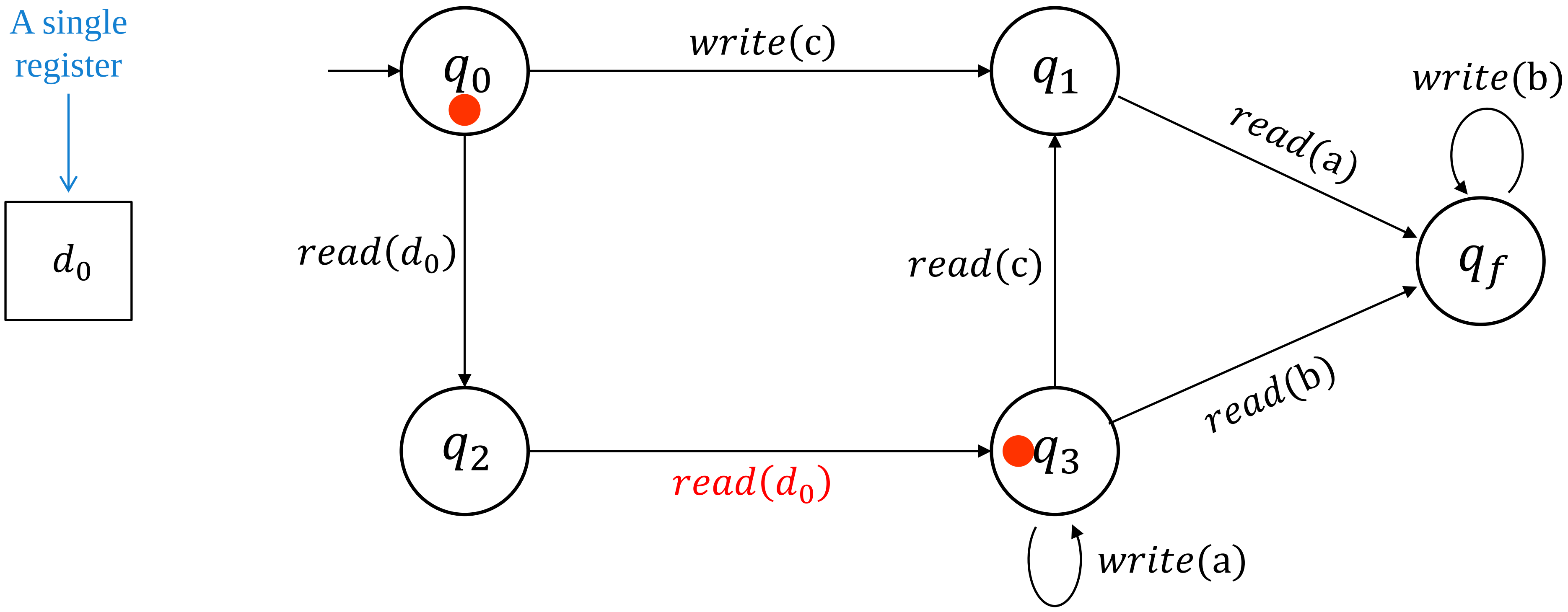
A small example



A small example

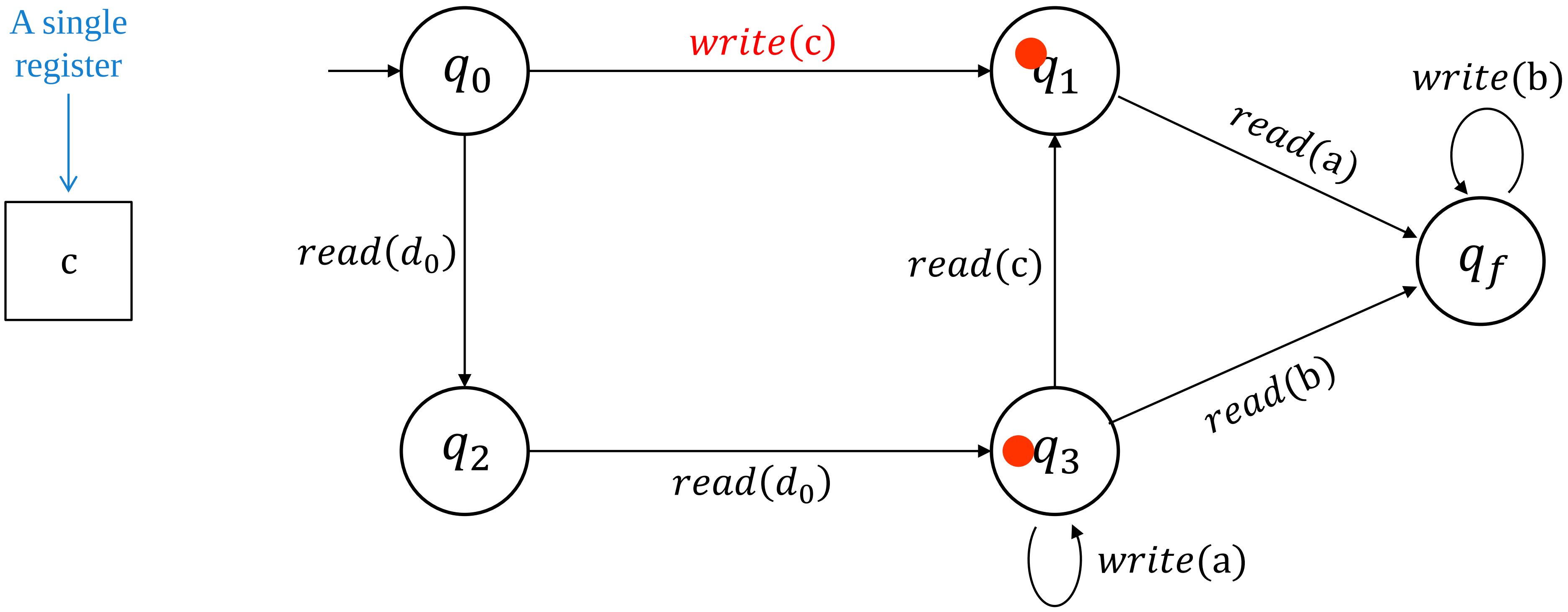


A small example

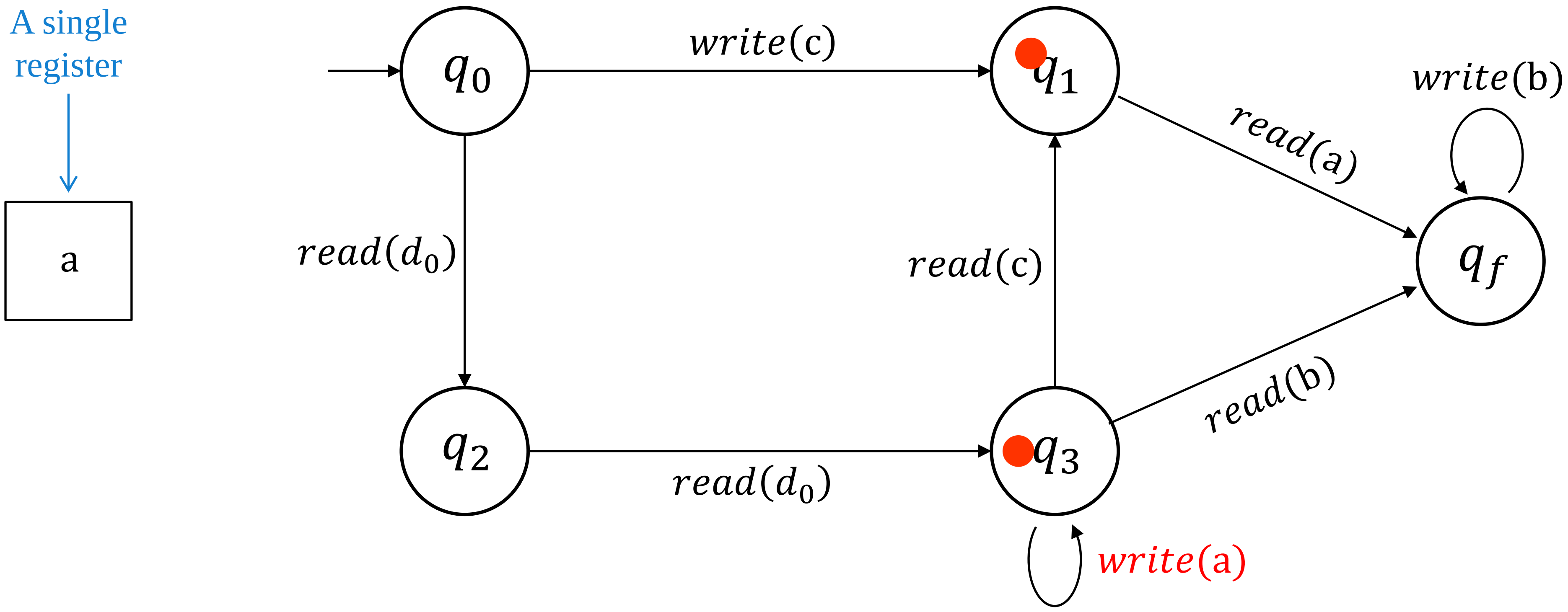


A small example

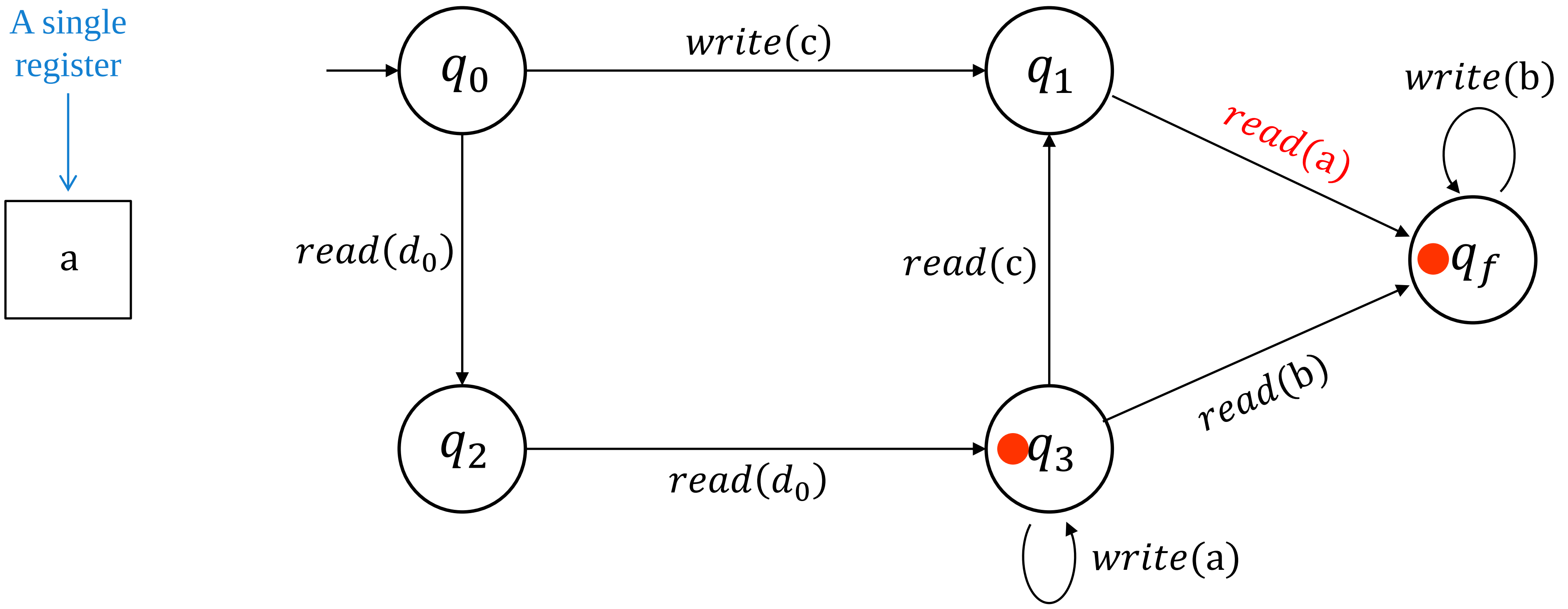
A single register



A small example

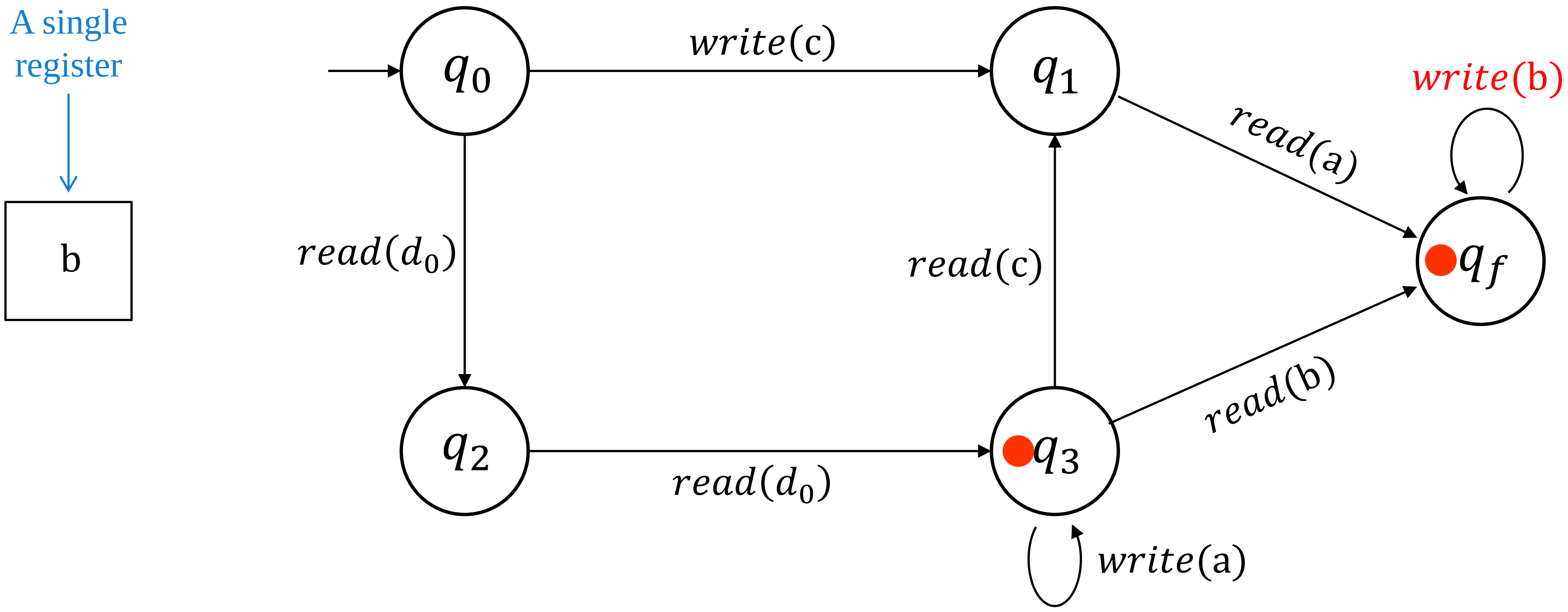


A small example

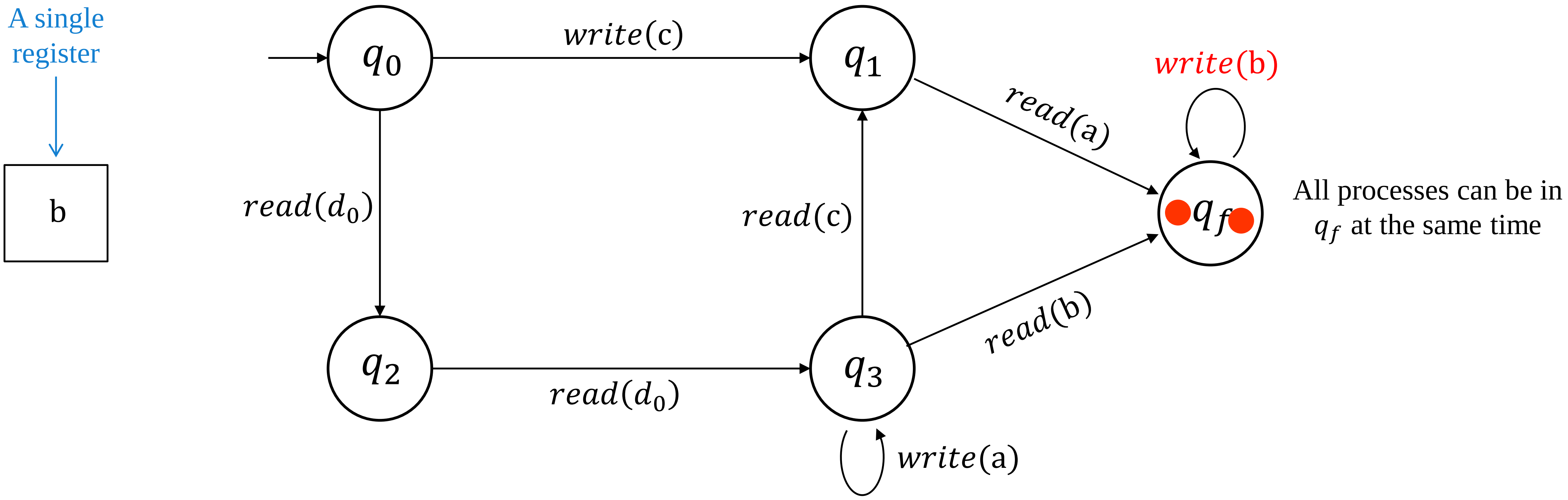


q_f covered ✓

A small example



A small example



Reachability problems

Does there exist a number of participants for which a bad event can happen?

Reachability problems

Does there exist a number of participants for which a bad event can happen?

Does there exist n such that, from init_n, there is an execution ρ that reaches a bad configuration?

n processes, all in q_0
registers initialized to d_0

Asynchronicity:
many executions for a given n

Reachability problems

Does there exist a number of participants for which a bad event can happen?

Does there exist n such that, from init_n, there is an execution ρ that reaches a bad configuration?

n processes, all in q_0
registers initialized to d_0

Asynchronicity:
many executions for a given n

COVER $\exists n, \exists \rho\colon \mathrm{init}_n \to^* \gamma\,,\ \ \gamma(q_f) \geq 1$? At least one process in *error state* q_f

Reachability problems

Does there exist a number of participants for which a bad event can happen?

Does there exist n such that, from init_n, there is an execution ρ that reaches a bad configuration?

n processes, all in q_0
registers initialized to d_0

Asynchronicity:
many executions for a given n

COVER $\exists n, \exists \rho: \mathrm{init}_n \rightarrow^* \gamma\,,\ \ \gamma(q_f) \geq 1\ ?$ At least one process in *error state* q_f

TARGET $\exists n, \exists \rho: \mathrm{init}_n \rightarrow^* \gamma\,,\ \ \forall q \neq q_f,\ \gamma(q) = 0\ ?$ All processes in q_f

Reachability problems

Does there exist a number of participants for which a bad event can happen?

Does there exist n such that, from init_n, there is an execution ρ that reaches a bad configuration?

n processes, all in q_0
registers initialized to d_0

Asynchronicity:
many executions for a given n

COVER $\exists n, \exists \rho: \mathrm{init}_n \to^* \gamma\,,\ \ \gamma(q_f) \geq 1\ ?$ At least one process in *error state* q_f

TARGET $\exists n, \exists \rho: \mathrm{init}_n \to^* \gamma\,,\ \ \forall q \neq q_f,\ \gamma(q) = 0\ ?$ All processes in q_f

Presence Reachability Problem (PRP) $\exists n, \exists \rho: \mathrm{init}_n \to^* \gamma\,,\ \ \gamma \vDash \phi\ ?$

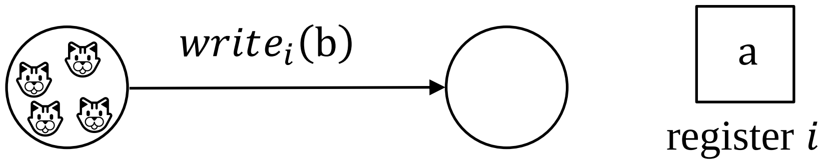
inspired by [DSTZ12]

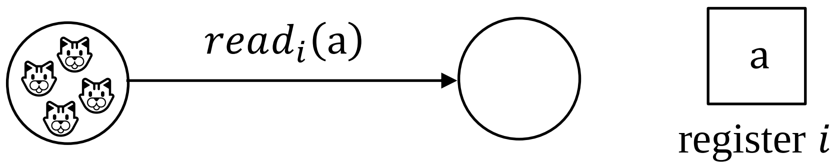
Presence constraint = Boolean combination of 'state q empty'

[DSTZ12] Giorgio Delzanno, Arnaud Sangnier, Riccardo Traverso, and Gianluigi Zavattaro. *On the Complexity of Parameterized Reachability in Reconfigurable Broadcast Networks*. FSTTCS 2012.

An abstraction for PRP

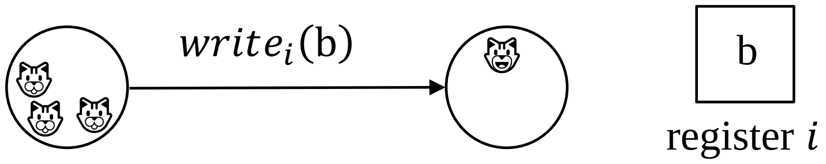
Copycat: a process can copy another one in same state.

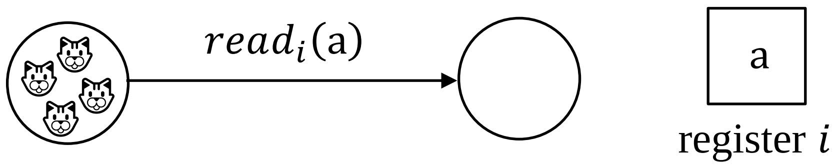




An abstraction for PRP

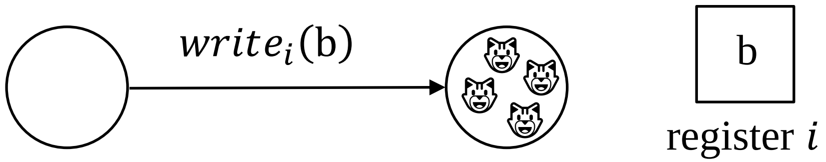
Copycat: a process can copy another one in same state.

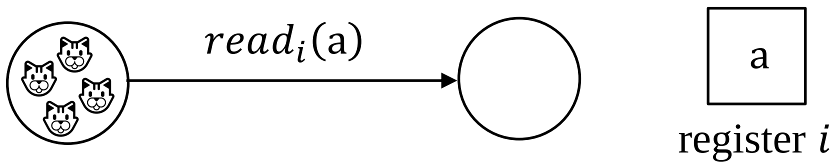




An abstraction for PRP

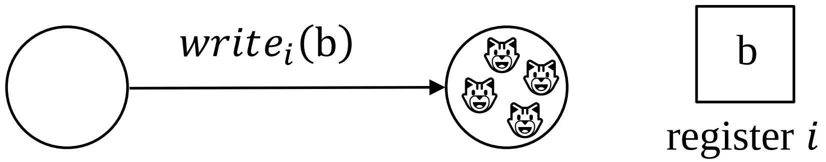
Copycat: a process can copy another one in same state.

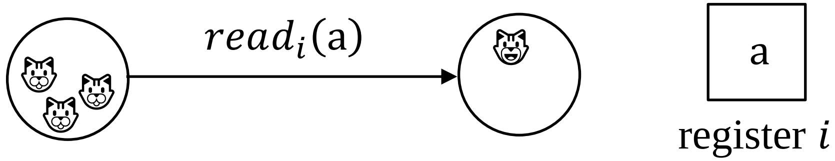




An abstraction for PRP

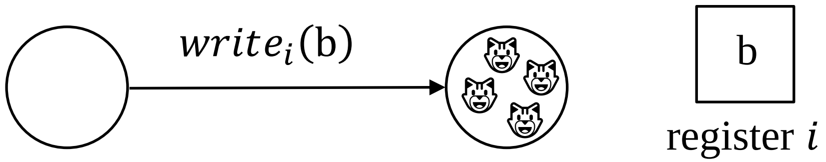
Copycat: a process can copy another one in same state.

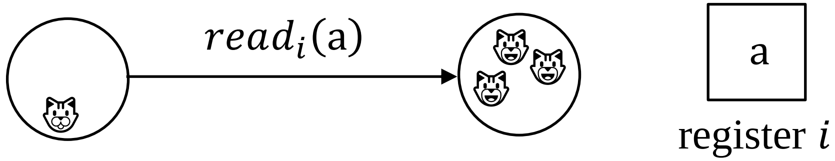




An abstraction for PRP

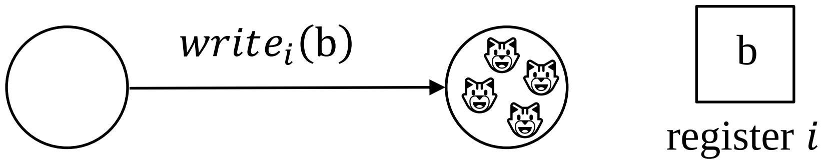
Copycat: a process can copy another one in same state.

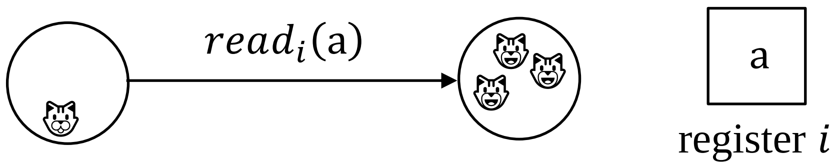




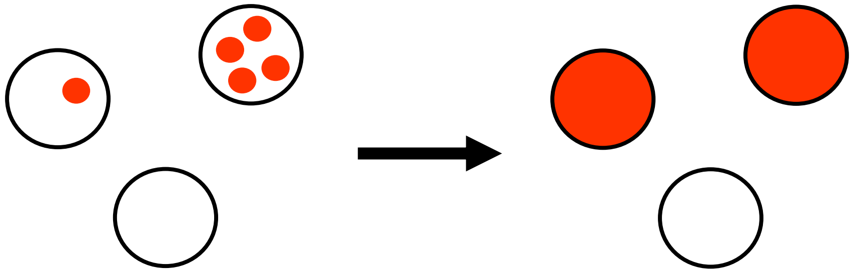
An abstraction for PRP

Copycat: a process can copy another one in same state.



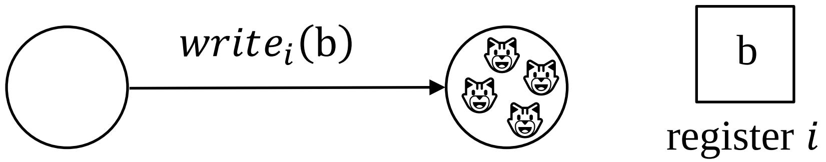


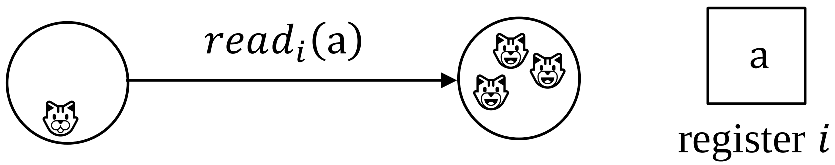
$0 - 1$ abstraction: store whether 0 process or at least 1



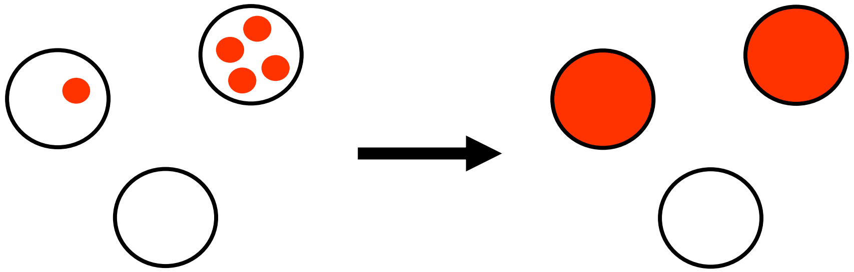
An abstraction for PRP

Copycat: a process can copy another one in same state.





$0-1$ abstraction: store whether 0 process or at least 1



Sound and complete for PRP:

- copycat property
- number n of processes is arbitrary
- presence constraints do not count processes

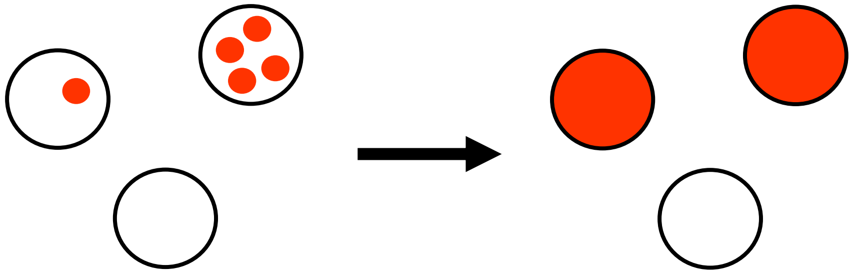
PRP $\quad \exists n, \exists \rho : \mathrm{init}_n \rightarrow^* \gamma \,, \quad \gamma \vDash \phi$?

Boolean combination of 'state q empty'

An abstraction for PRP

Copycat: a process can copy another one in same state.

$0-1$ abstraction: store whether 0 process or at least 1



Sound and complete for PRP:
- copycat property
- number n of processes is arbitrary
- presence constraints do not count processes

PRP $\quad \exists n, \exists \rho: \mathrm{init}_n \rightarrow^* \gamma\,, \quad \gamma \vDash \phi\,?$

Boolean combination of 'state q empty'

Theorem **[Wal23]**: PRP is in NP.

[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.

Contributions on reachability problems [Wal23]

	COVER	TARGET	DNF-PRP	PRP
General case	NP-complete	NP-complete	NP-complete	NP-complete
Not initialized	PTIME	NP-complete	NP-complete	NP-complete
One register	PTIME	PTIME	PTIME	NP-complete

[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.

Contributions on reachability problems [Wal23]

	COVER	TARGET	DNF-PRP	PRP
General case	NP-complete	NP-complete	NP-complete	NP-complete
Not initialized	PTIME	NP-complete	NP-complete	NP-complete
One register	PTIME	PTIME	PTIME	NP-complete

PTIME [EGM13]

previously known result

[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.
[EGM13] Javier Esparza, Pierre Ganty, and Rupak Majumdar. *Parameterized Verification of Asynchronous Shared-Memory Systems.* CAV 2013.

A more general result

Structural Theorem **[Wal24]**: In ASMS, the *diameter* is doubly-exponentially bounded.

If $\gamma \to^* \gamma'$ then there is

$$\rho\colon\ \gamma = \gamma_0 \xrightarrow{t_1} \xrightarrow{t_1} \xrightarrow{t_1} \dots \xrightarrow{t_1} \gamma_1 \xrightarrow{t_2} \xrightarrow{t_2} \xrightarrow{t_2} \dots \xrightarrow{t_2} \gamma_2 \xrightarrow{t_3} \qquad \dots \qquad \xrightarrow{t_\ell} \xrightarrow{t_\ell} \xrightarrow{t_\ell} \xrightarrow{t_\ell} \dots \xrightarrow{t_\ell} \gamma_\ell = \gamma' \qquad \ell \leq B \text{ phases, } B \text{ 2-exp}$$

[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

A more general result

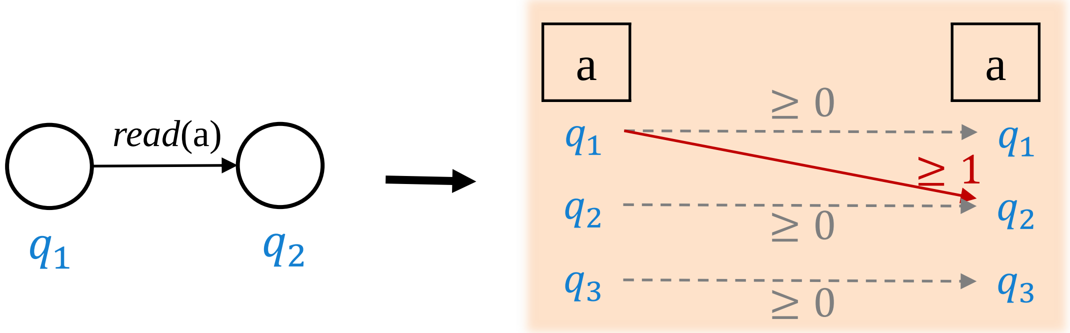
Structural Theorem **[Wal24]**: In ASMS, the *diameter* is doubly-exponentially bounded.

If $\gamma \rightarrow^* \gamma'$ then there is

$$\rho\colon\ \gamma = \gamma_0 \xrightarrow{t_1}\xrightarrow{t_1}\xrightarrow{t_1}\ldots\xrightarrow{t_1}\gamma_1\xrightarrow{t_2}\xrightarrow{t_2}\xrightarrow{t_2}\ldots\xrightarrow{t_2}\gamma_2\xrightarrow{t_3}\quad\ldots\quad\xrightarrow{t_\ell}\xrightarrow{t_\ell}\xrightarrow{t_\ell}\xrightarrow{t_\ell}\ldots\xrightarrow{t_\ell}\gamma_\ell = \gamma'$$

$\ell \leq B$ phases, B 2-exp

Proof using an abstraction called transfer flows.
Relies on a bound from well-quasi-order theory [SS24].



[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.
[SS24] Sylvain Schmitz, Lia Schütze. *On the Length of Strongly Monotone Descending Chains over* $\mathbb{N}^d$. ICALP 2024.

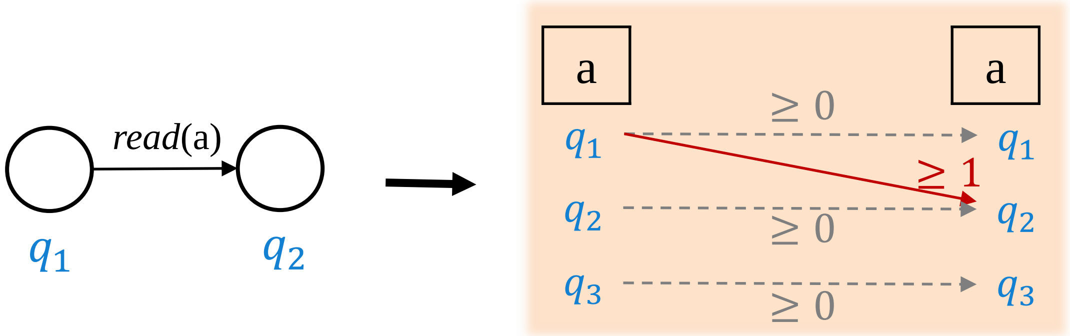
A more general result

Structural Theorem **[Wal24]**: In ASMS, the *diameter* is doubly-exponentially bounded.

If $\gamma \rightarrow^* \gamma'$ then there is

$$\rho\colon\ \gamma = \gamma_0 \xrightarrow{t_1}\xrightarrow{t_1}\xrightarrow{t_1}\ldots\xrightarrow{t_1}\gamma_1 \xrightarrow{t_2}\xrightarrow{t_2}\xrightarrow{t_2}\ldots\xrightarrow{t_2}\gamma_2 \xrightarrow{t_3} \quad \ldots \quad \xrightarrow{t_\ell}\xrightarrow{t_\ell}\xrightarrow{t_\ell}\xrightarrow{t_\ell}\ldots\xrightarrow{t_\ell}\gamma_\ell = \gamma' \qquad \ell \leq B \text{ phases, } B \text{ 2-exp}$$

Proof using an abstraction called transfer flows.
Relies on a bound from well-quasi-order theory [SS24].



Corollary **[Wal24]**: The following problems for ASMS are decidable:

- emptiness of expressions obtained using presence constraints, Boolean operators and Post* and Pre* operators;

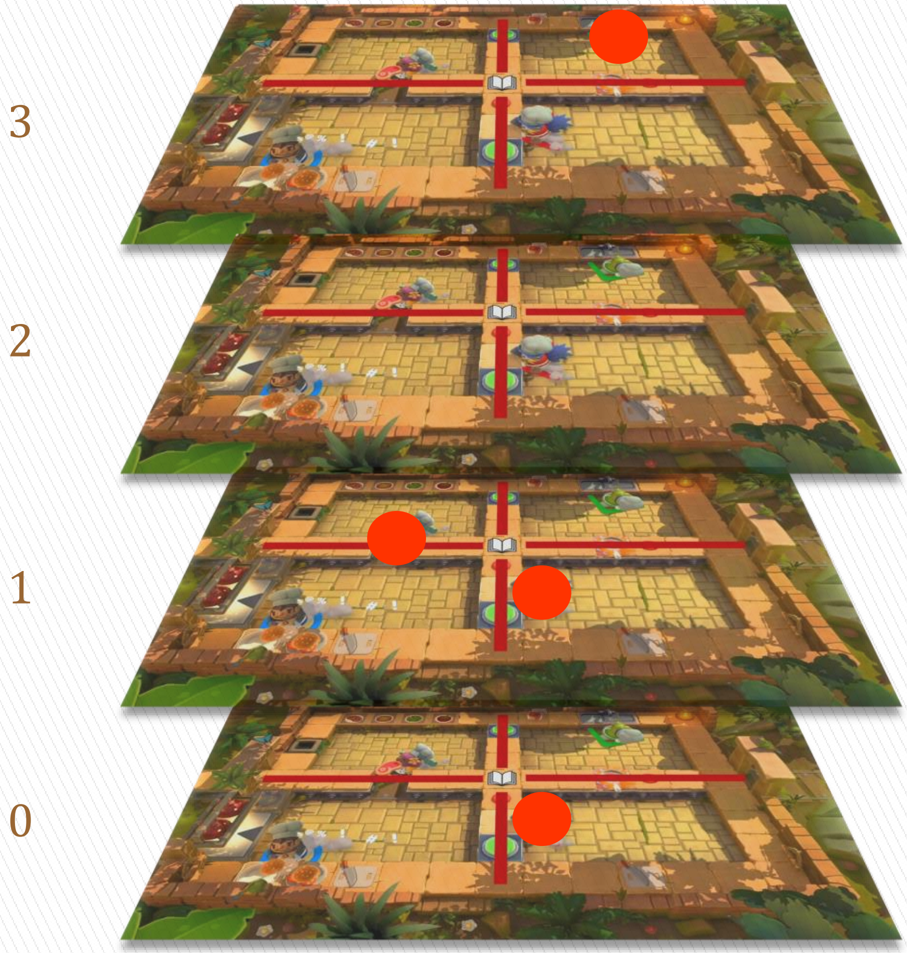
  $$\mathrm{Post}^*(\mathrm{Init}) \subseteq \mathrm{Pre}^*(\#q_f \geq 1)$$

- verification of LTL formulas over transitions, without the next operator.

  $$\exists n\ \exists \rho \text{ infinite from } \mathrm{init}_n, \rho \vDash \mathrm{t}_1\ \mathrm{U}\ (\mathrm{G}\ \mathrm{t}_2)$$

[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.
[SS24] Sylvain Schmitz, Lia Schütze. *On the Length of Strongly Monotone Descending Chains over* $\mathbb{N}^d$. ICALP 2024.

Second part
Round-based shared-memory systems



A round-based consensus algorithm

Aspnes' consensus algorithm [Asp02]

Shared registers: $(\mathrm{rg}_b[r])_{b\in\{0,1\},r\in\mathbb{N}}$ all initialized to ⊥;

Variable k is private
= asynchronous rounds

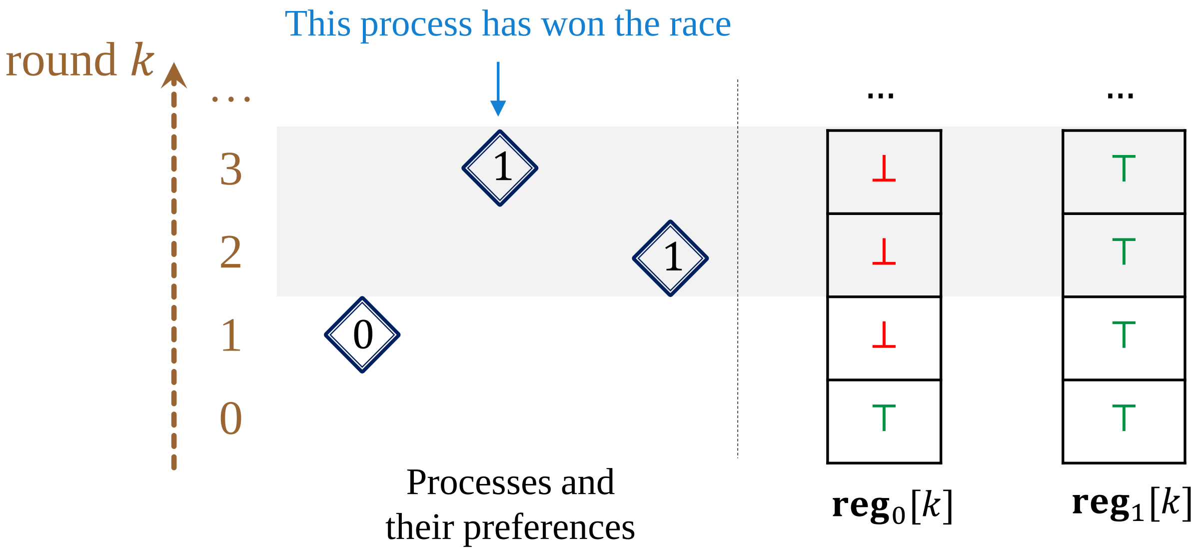
Unboundedly many shared
registers: 2 per round

Read and write to registers
of nearby rounds

```
bool p ∈ {0,1} %preference of the process
for k = 0 to ∞:
        read from rg_0[k] and rg_1[k];
        if rg_0[k] = ⊤ and rg_1[k] = ⊥ then p ≔ 0;
        else if rg_0[k] =⊥ and rg_1[k] = ⊤ then p ≔ 1;
        write ⊤ to rg_p[k];
        if k > 0:
                read from rg_{1−p}[k − 1];
                if rg_{1−p}[k − 1] = ⊥:
                        return p;
```


[Asp02] James Aspnes. *Fast deterministic consensus in a noisy environment.* Journal of Algorithms 2002.

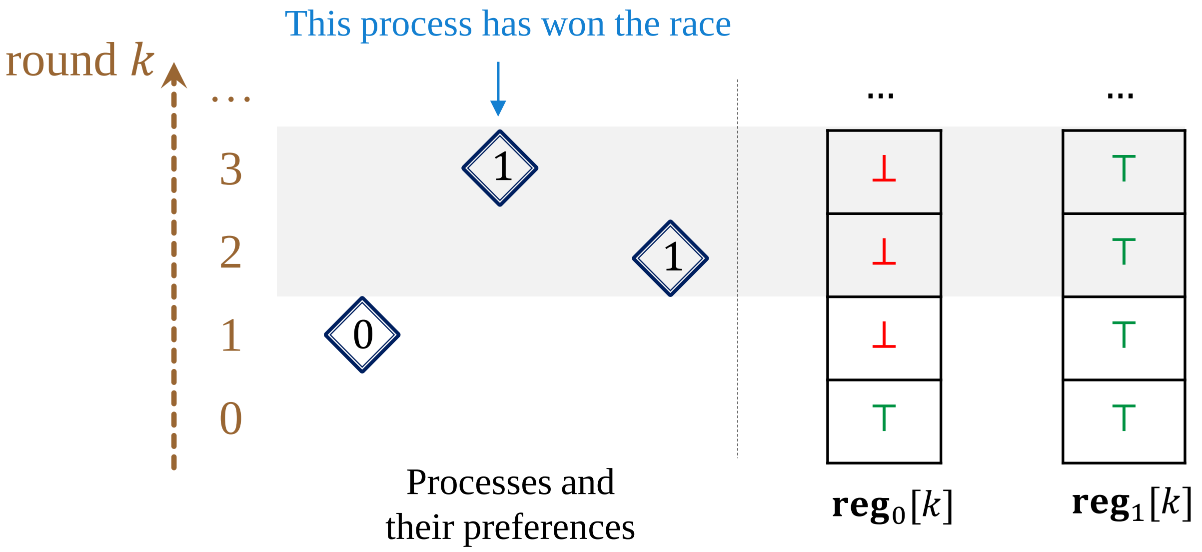
More about Aspnes' algorithm



- Race between the processes.
- Stochastic scheduler that models a noisy environment.
- Almost-surely terminates but does not always terminate (workaround for an impossibility result [FLP85]).
- Unboundedly many rounds are needed.

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Mike Paterson. *Impossibility of Distributed Consensus with One Faulty Process*. JACM 1985.

More about Aspnes' algorithm



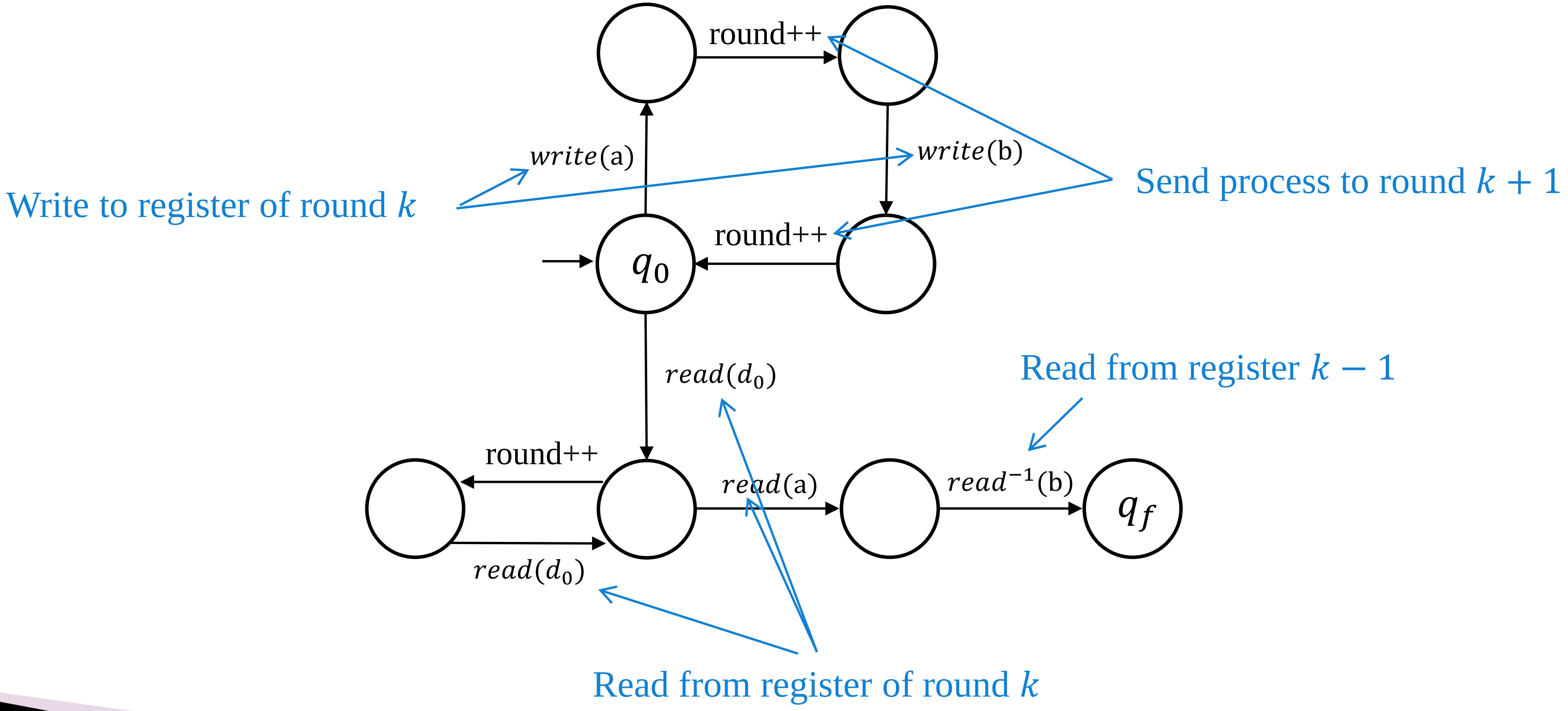
- Race between the processes.
- Stochastic scheduler that models a noisy environment.
- Almost-surely terminates but does not always terminate (workaround for an impossibility result [FLP85]).
- Unboundedly many rounds are needed.

(not modelled for now, topic of third part)

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Mike Paterson. *Impossibility of Distributed Consensus with One Faulty Process*. JACM 1985.

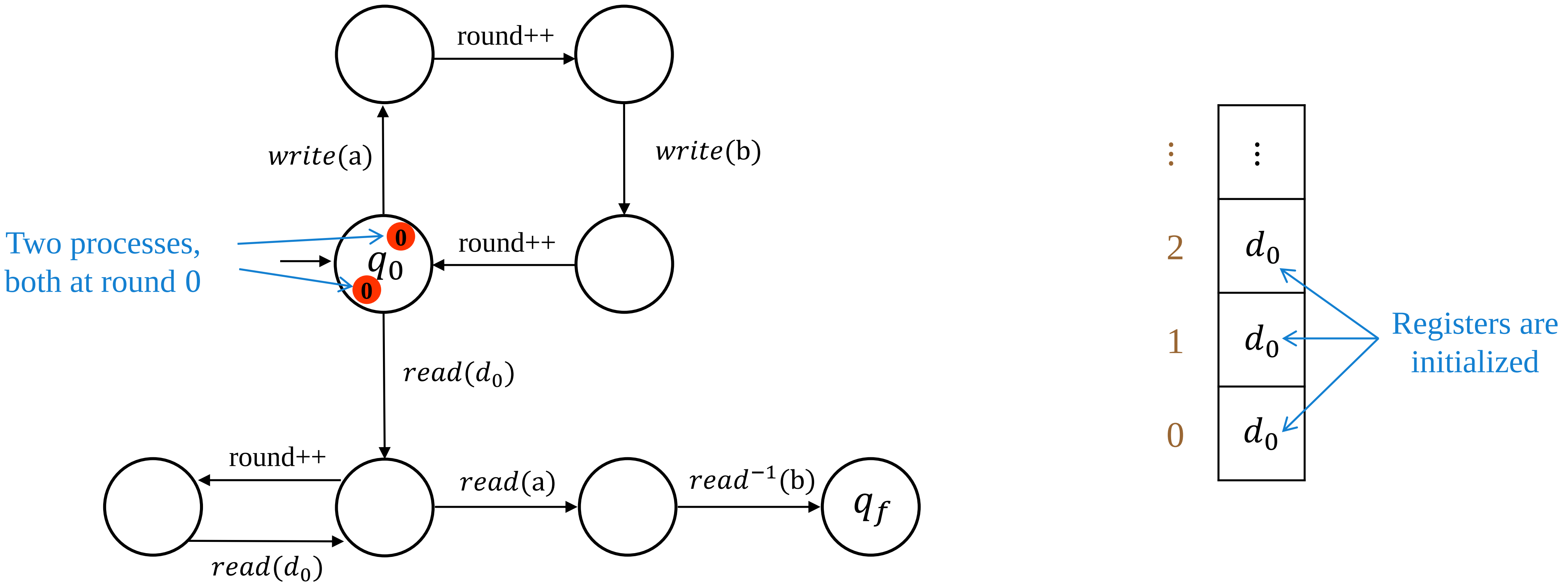
Round-based ASMS

New model for round-based algorithms **[BMSW22]**

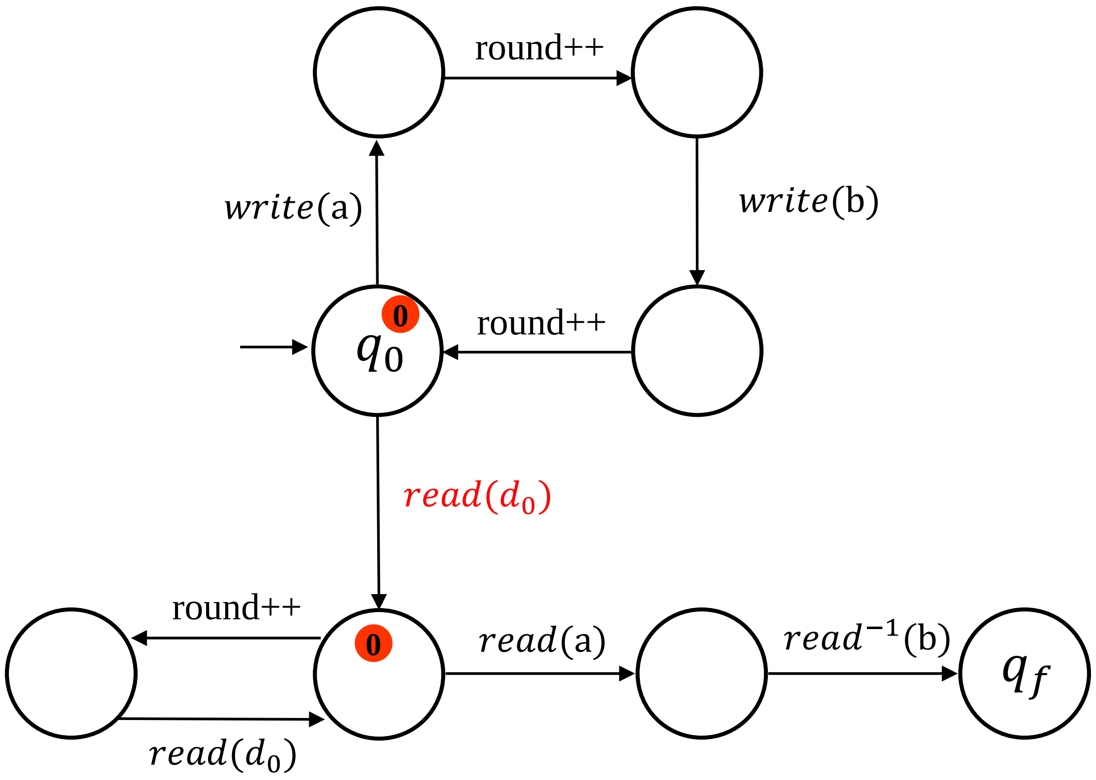


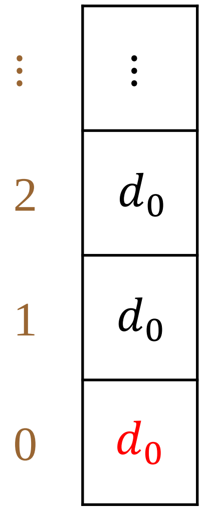
[BMSW22] Nathalie Bertrand, Nicolas Markey, Ocan Sankur, **Nicolas Waldburger**: *Parameterized safety verification of round-based shared-memory systems*. ICALP 2022.

An example of execution

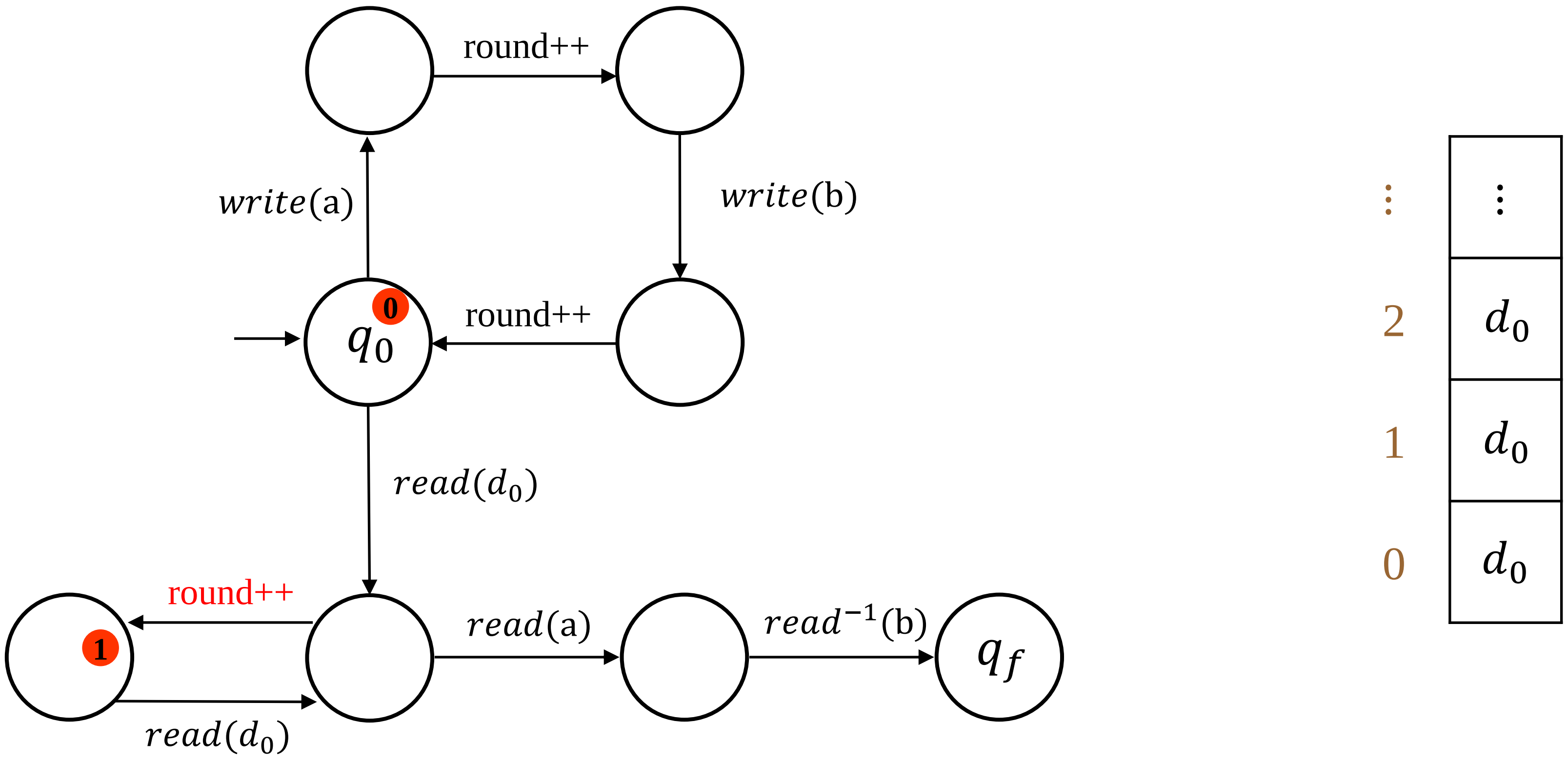


An example of execution

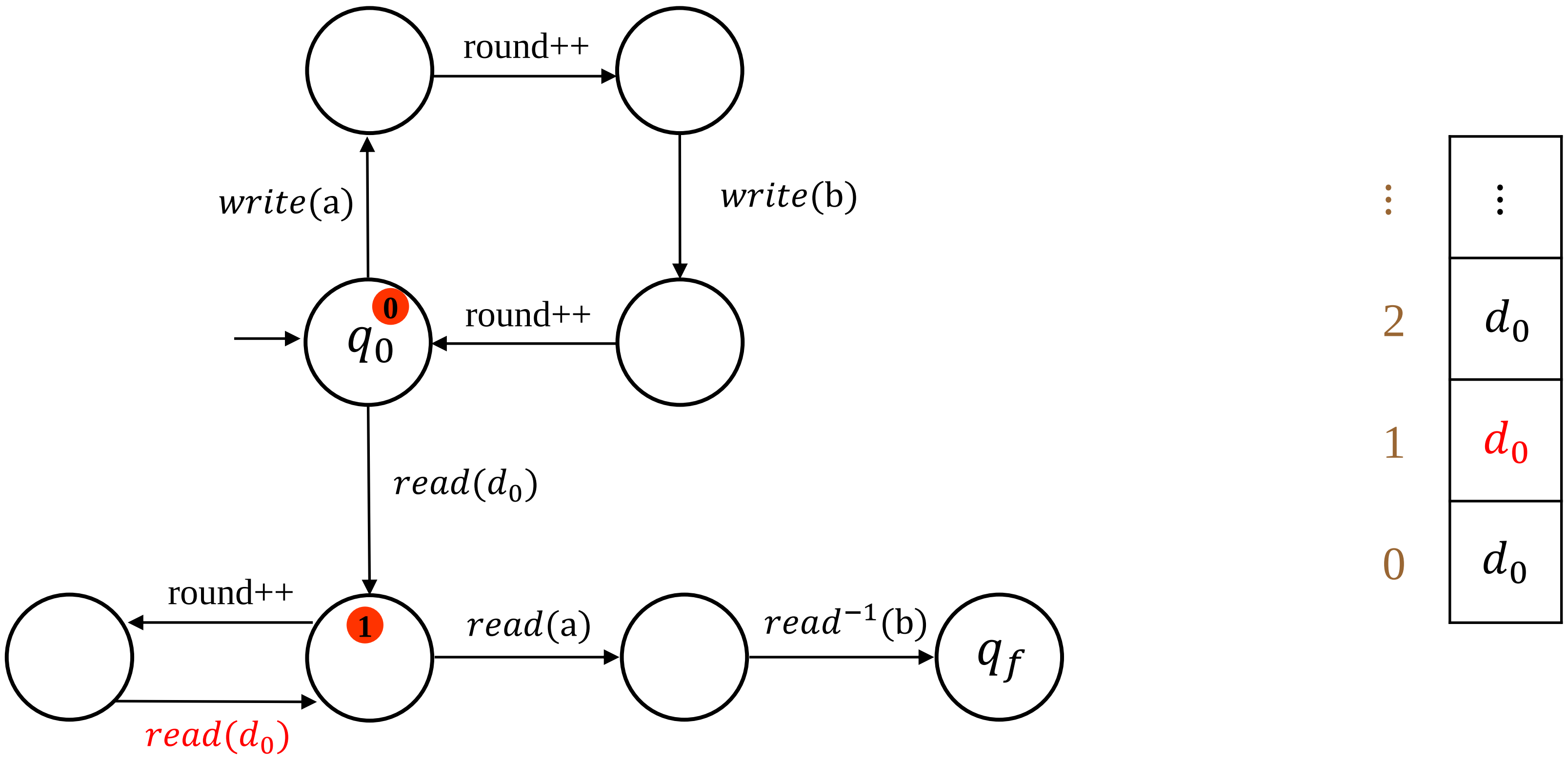




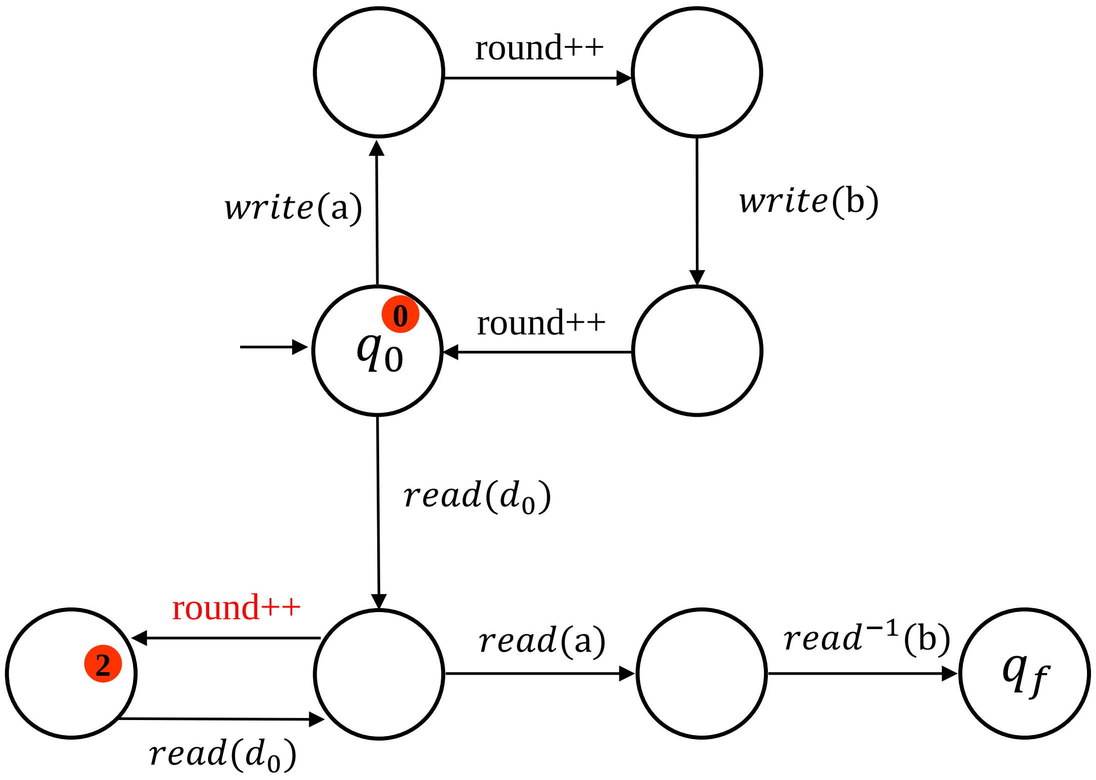
An example of execution

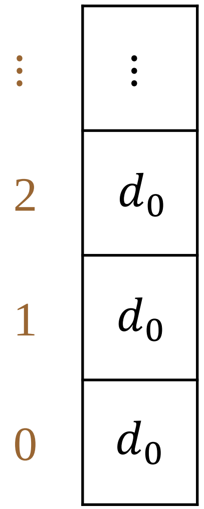


An example of execution

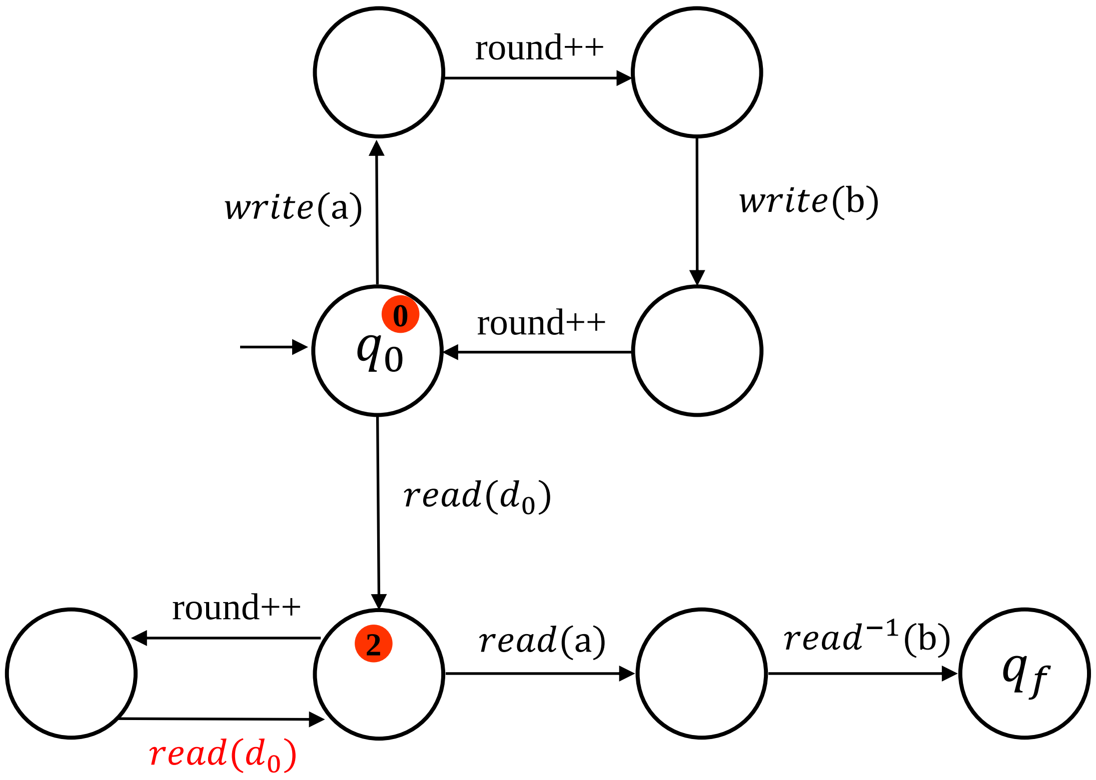


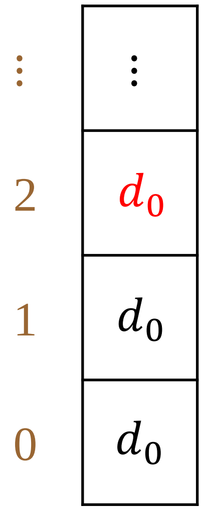
An example of execution



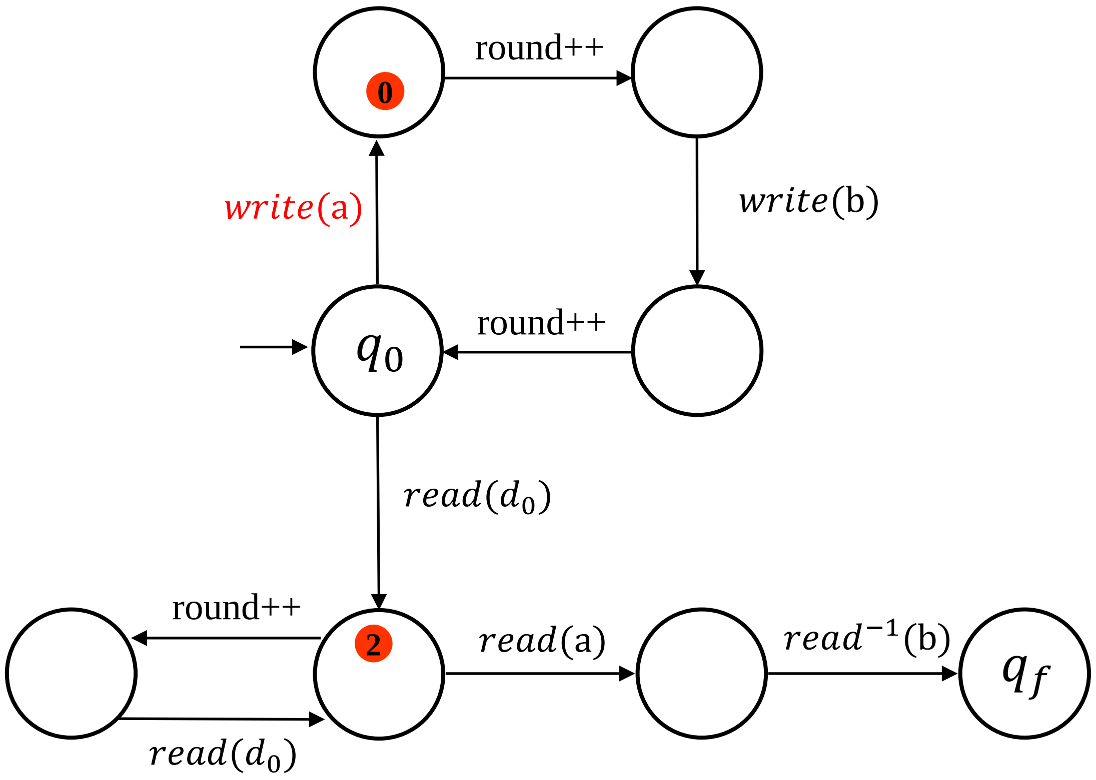


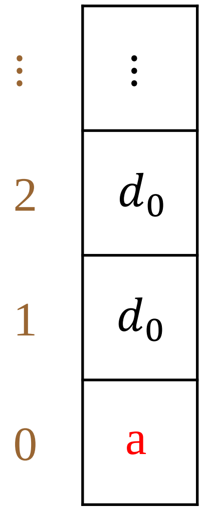
An example of execution



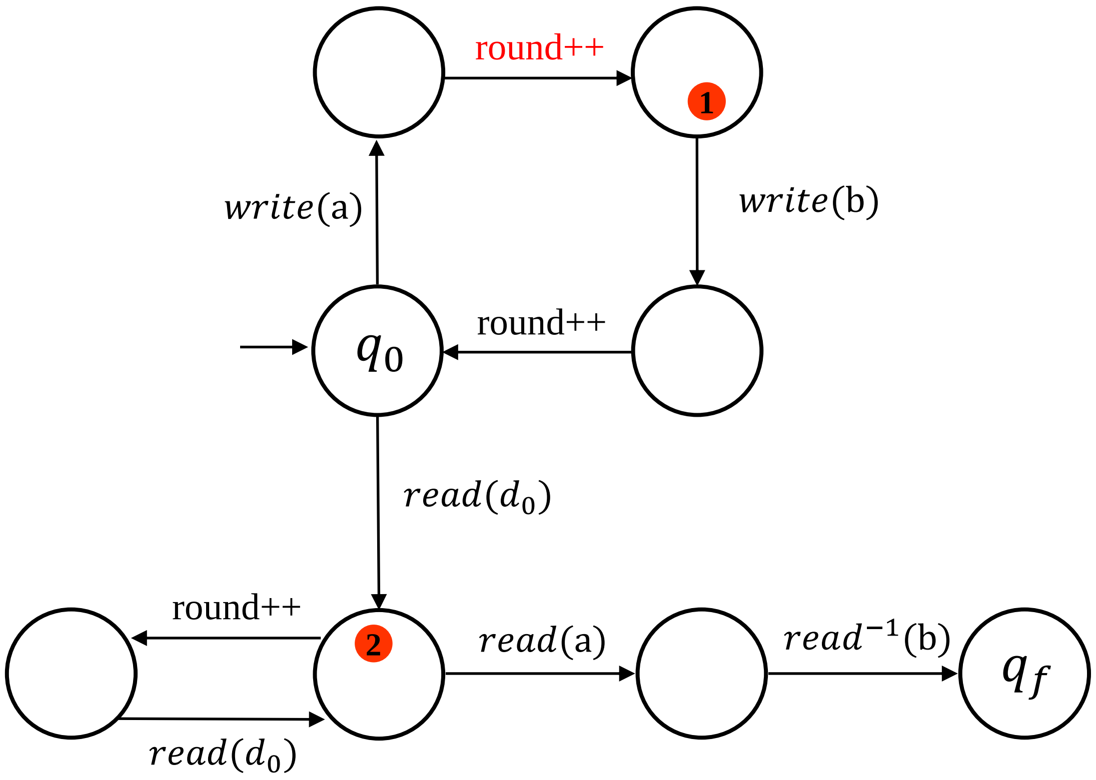


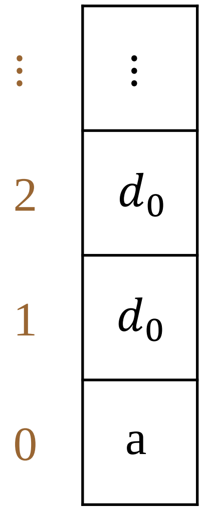
An example of execution



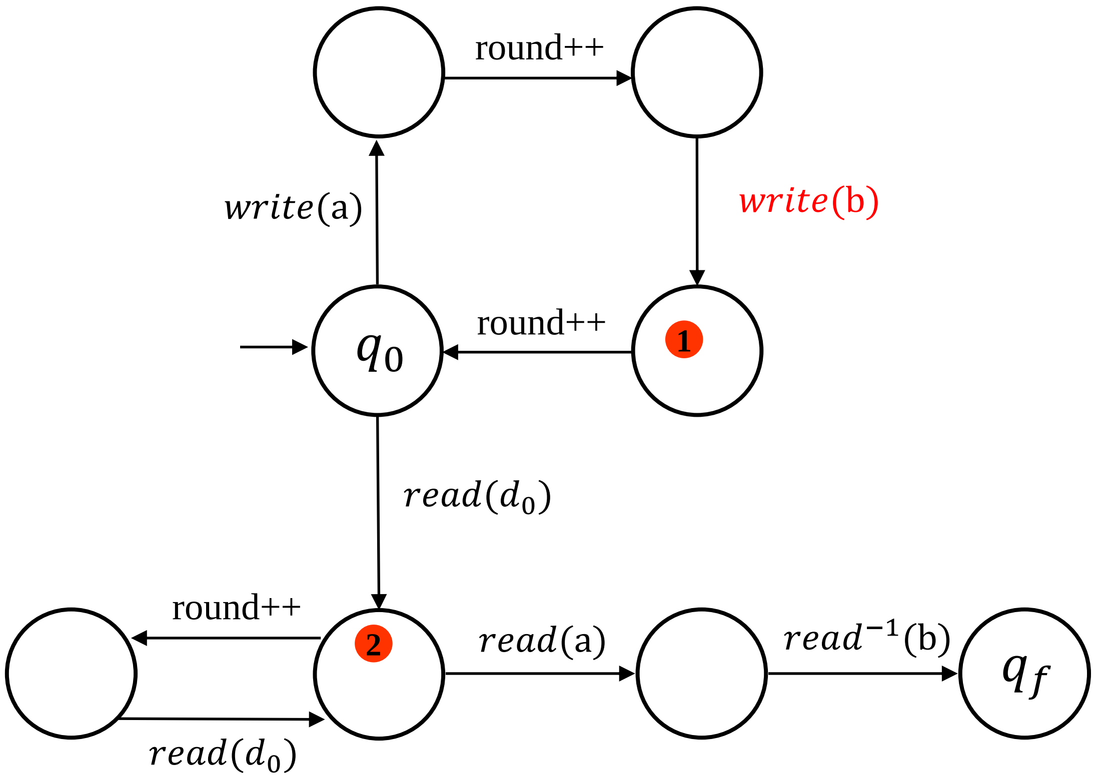


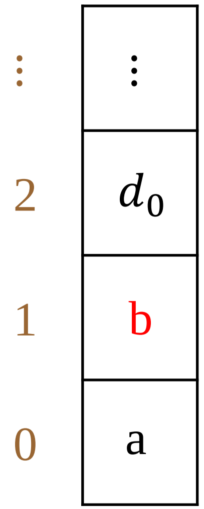
An example of execution



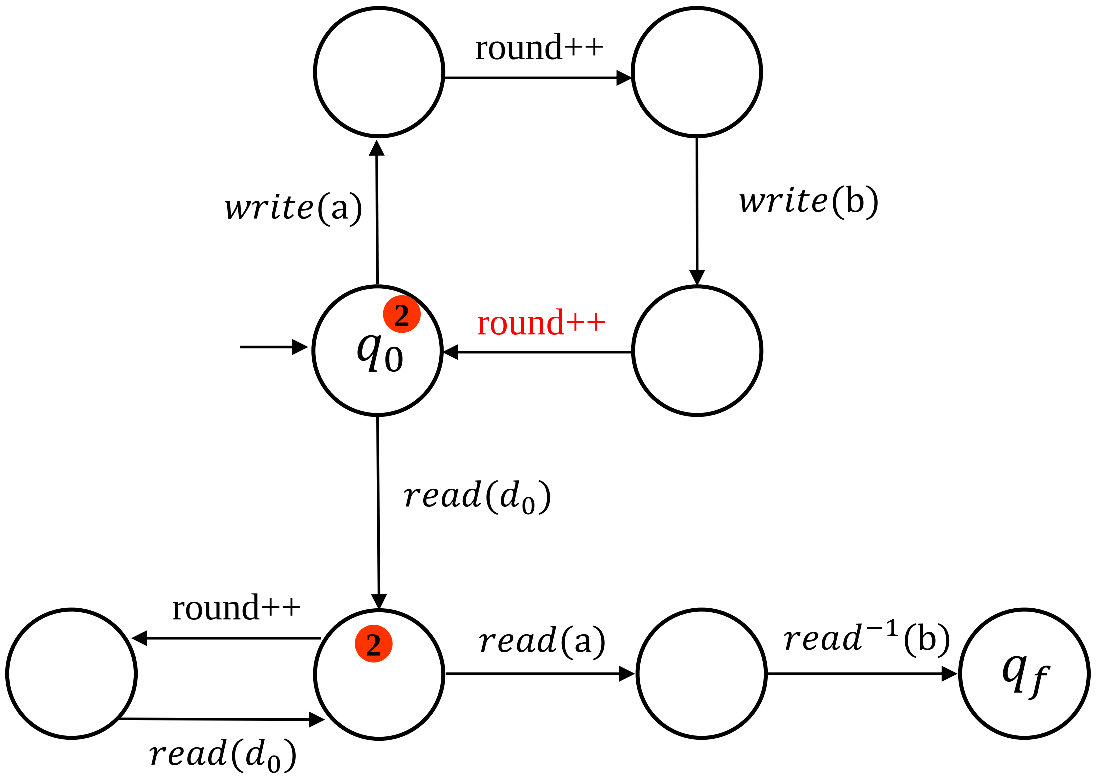


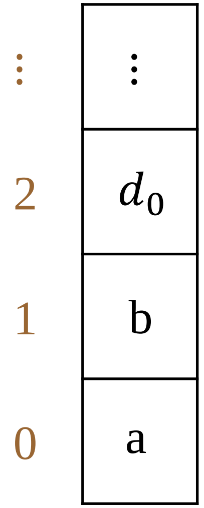
An example of execution



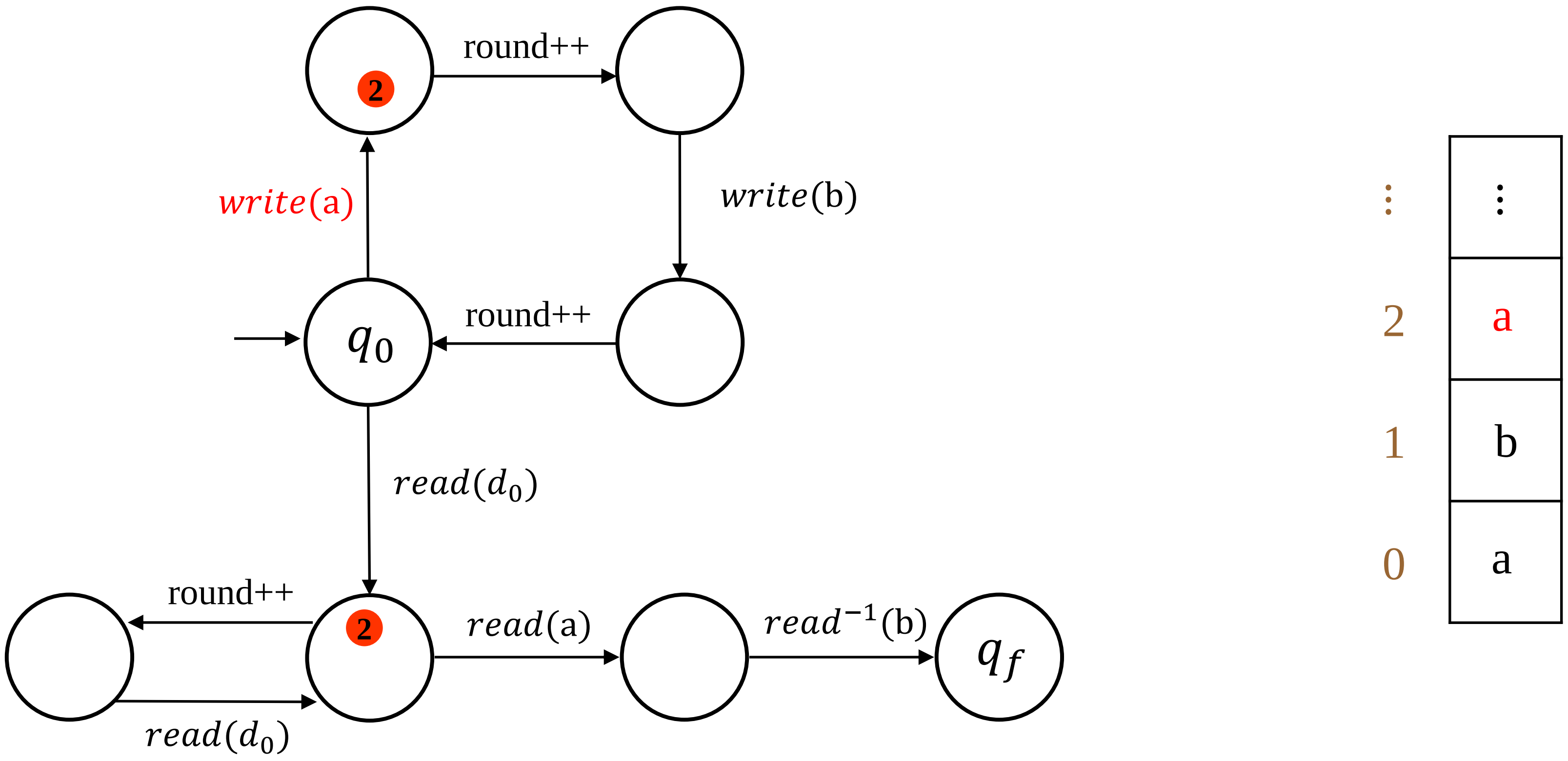


An example of execution

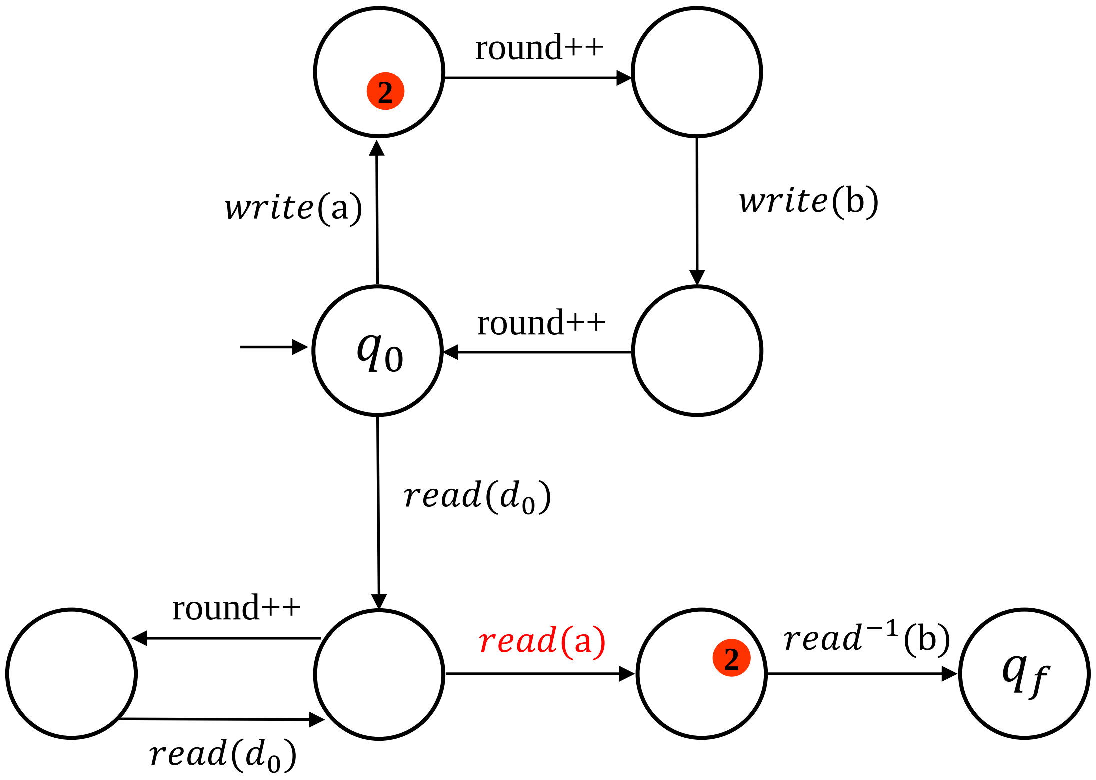


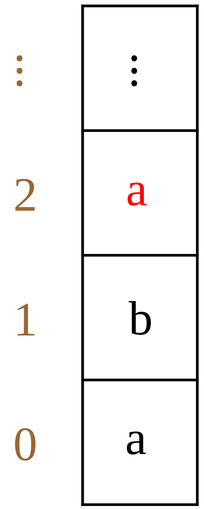


An example of execution

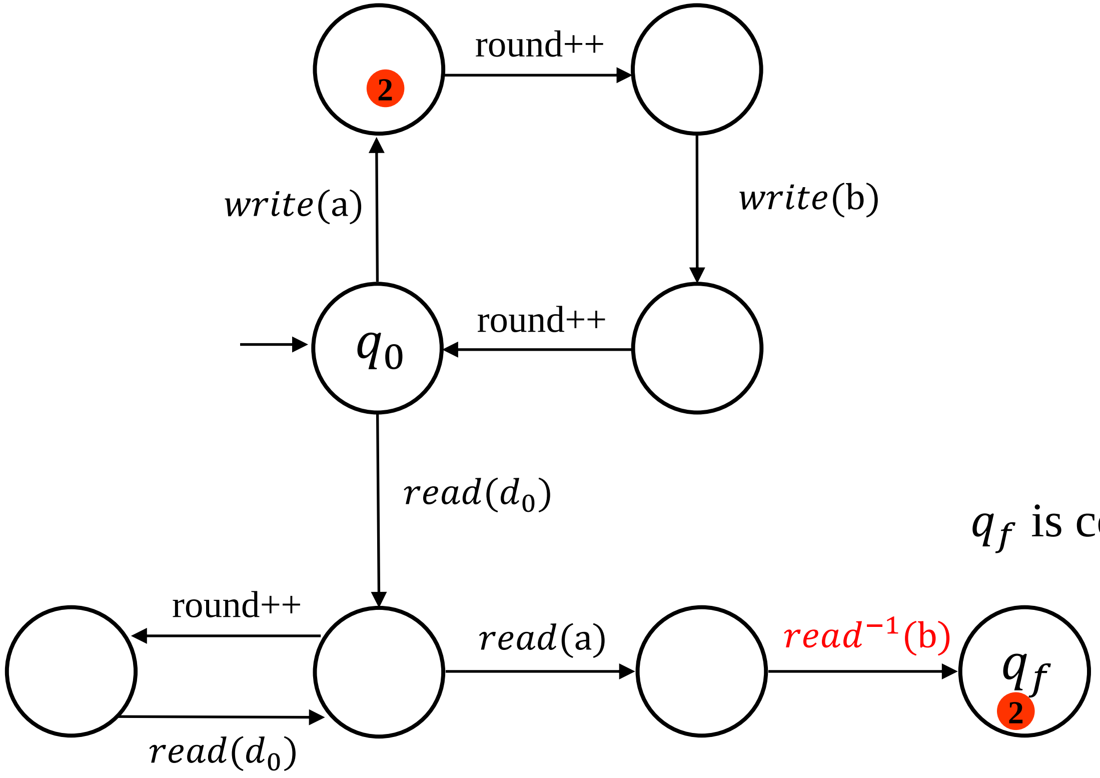


An example of execution





An example of execution



q_f is covered ✓

⋮	⋮
2	a
1	b
0	a

Reachability problems in round-based ASMS

COVER $\exists n, \exists \rho\colon \mathrm{init}_n \to^* \gamma\ ,\ \exists k,\ \gamma(q_f, k) \geq 1$?

for some round k, some process in state q_f and at round k

Reachability problems in round-based ASMS

COVER $\exists n, \exists \rho : \mathrm{init}_n \to^* \gamma\ ,\ \exists k,\ \gamma(q_f, k) \geq 1$?

for some round k, some process in state q_f and at round k

TARGET $\exists n, \exists \rho : \mathrm{init}_n \to^* \gamma\ ,\ \forall k, \forall q \neq q_f,\ \gamma(q, k) = 0$?

Reachability problems in round-based ASMS

COVER $\exists n, \exists \rho: \mathrm{init}_n \to^* \gamma\ ,\ \exists k,\ \gamma(q_f, k) \geq 1$?

for some round k, some process in state q_f and at round k

TARGET $\exists n, \exists \rho: \mathrm{init}_n \to^* \gamma\ ,\ \forall k, \forall q \neq q_f,\ \gamma(q, k) = 0$?

Presence Reachability Problem (PRP) $\exists n, \exists \rho: \mathrm{init}_n \to^* \gamma\ ,\ \gamma \vDash \psi$?

presence constraint = first-order formula on rounds with no nested quantifiers

Example: $\underbrace{\exists k\ (\gamma(q_1, k+1) \geq 1 \wedge \gamma(q_1, k) = 0)}_{\text{For some } k,\ (q_1, k+1) \text{ not empty but } (q_1, k) \text{ empty}} \vee \underbrace{\forall k\ \gamma(q_0, k) = 0}_{\text{no process is in } q_0}$

Complexity results

Theorem **[BMSW22]**: In round-based ASMS, COVER is PSPACE-complete.

[BMSW22] Nathalie Bertrand, Nicolas Markey, Ocan Sankur, **Nicolas Waldburger**: *Parameterized safety verification of round-based shared-memory systems*. ICALP 2022.

Complexity results

Theorem **[BMSW22]**: In round-based ASMS, COVER is PSPACE-complete.

Theorem **[Wal23]**: In round-based ASMS, PRP is PSPACE-complete.

[BMSW22] Nathalie Bertrand, Nicolas Markey, Ocan Sankur, **Nicolas Waldburger**: *Parameterized safety verification of round-based shared-memory systems.* ICALP 2022.

[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.

A representation for executions

Witness execution: $\rho: \gamma_0 \xrightarrow{\theta_1} \gamma_1 \xrightarrow{\theta_2} \gamma_2 \xrightarrow{\theta_3} \gamma_3 \xrightarrow{\theta_4} \gamma_4 \xrightarrow{\theta_5} \gamma_5 \xrightarrow{\theta_6} \gamma_6 \xrightarrow{\theta_7} \gamma_7 \xrightarrow{\theta_8} \gamma_8 \xrightarrow{\theta_9} \gamma_9 \xrightarrow{\theta_{10}} \gamma_{10} \vDash \psi$

A representation for executions

Witness execution: $\rho: \gamma_0 \xrightarrow{\theta_1} \gamma_1 \xrightarrow{\theta_2} \gamma_2 \xrightarrow{\theta_3} \gamma_3 \xrightarrow{\theta_4} \gamma_4 \xrightarrow{\theta_5} \gamma_5 \xrightarrow{\theta_6} \gamma_6 \xrightarrow{\theta_7} \gamma_7 \xrightarrow{\theta_8} \gamma_8 \xrightarrow{\theta_9} \gamma_9 \xrightarrow{\theta_{10}} \gamma_{10} \vDash \psi$

$\downarrow$

Actions:	$\boxed{\theta_1}$	$\boxed{\theta_2}$	$\boxed{\theta_3}$	$\boxed{\theta_4}$	$\boxed{\theta_5}$	$\boxed{\theta_6}$	$\boxed{\theta_7}$	$\boxed{\theta_8}$	$\boxed{\theta_9}$	$\boxed{\theta_{10}}$
Rounds:	0	1	1	0	0	2	3	4	1	2

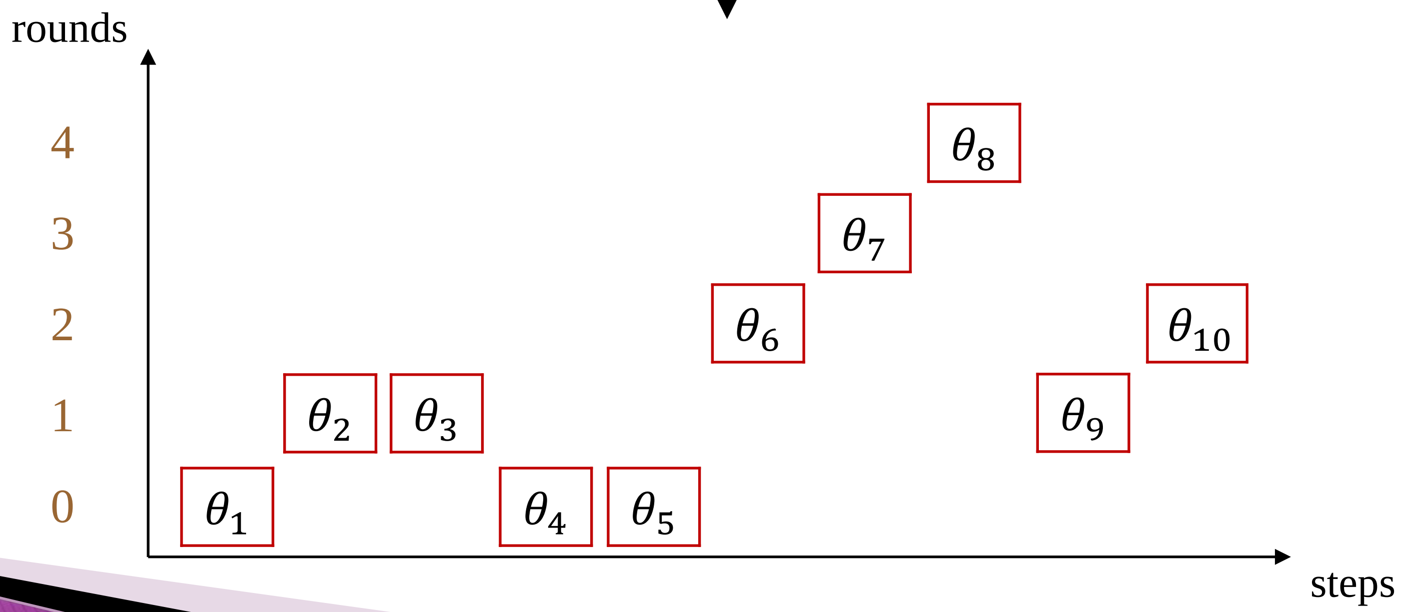
$\theta_i \in \Delta \times \mathbb{N}$:
transition and round

A representation for executions

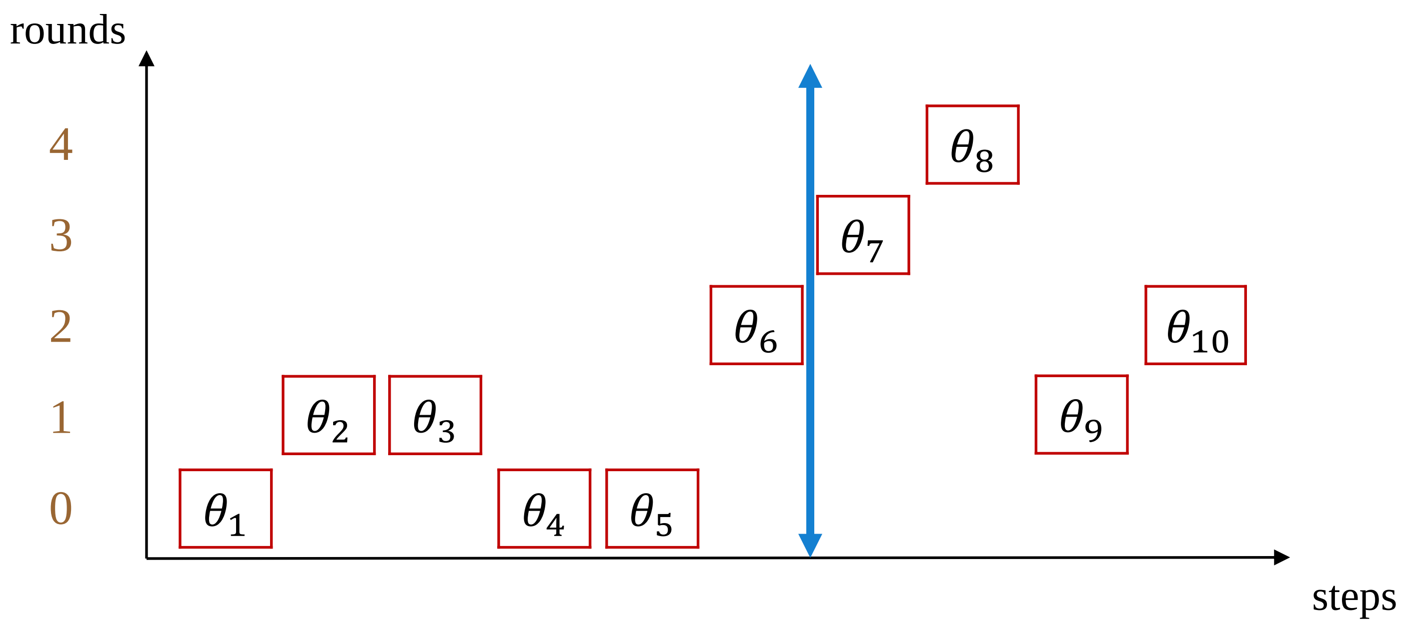
Witness execution: $\rho: \gamma_0 \xrightarrow{\theta_1} \gamma_1 \xrightarrow{\theta_2} \gamma_2 \xrightarrow{\theta_3} \gamma_3 \xrightarrow{\theta_4} \gamma_4 \xrightarrow{\theta_5} \gamma_5 \xrightarrow{\theta_6} \gamma_6 \xrightarrow{\theta_7} \gamma_7 \xrightarrow{\theta_8} \gamma_8 \xrightarrow{\theta_9} \gamma_9 \xrightarrow{\theta_{10}} \gamma_{10} \vDash \psi$

Actions:	$\boxed{\theta_1}$	$\boxed{\theta_2}$	$\boxed{\theta_3}$	$\boxed{\theta_4}$	$\boxed{\theta_5}$	$\boxed{\theta_6}$	$\boxed{\theta_7}$	$\boxed{\theta_8}$	$\boxed{\theta_9}$	$\boxed{\theta_{10}}$
Rounds:	0	1	1	0	0	2	3	4	1	2

$\theta_i \in \Delta \times \mathbb{N}$:
transition and round

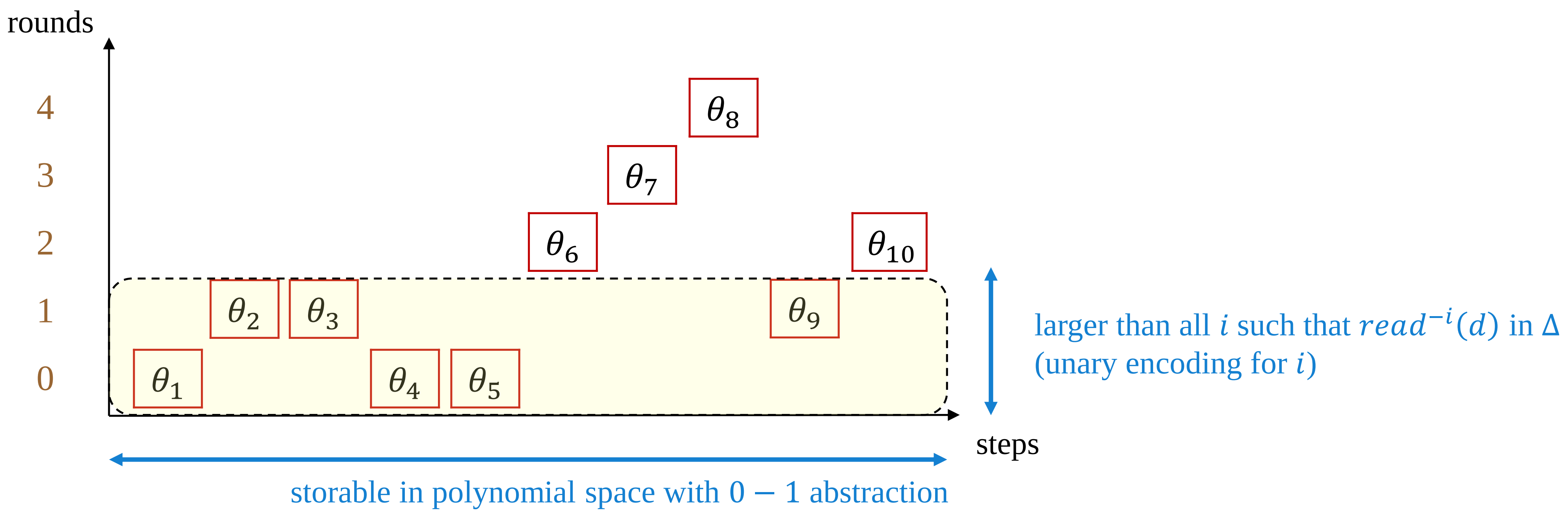


A non-deterministic polynomial-space algorithm

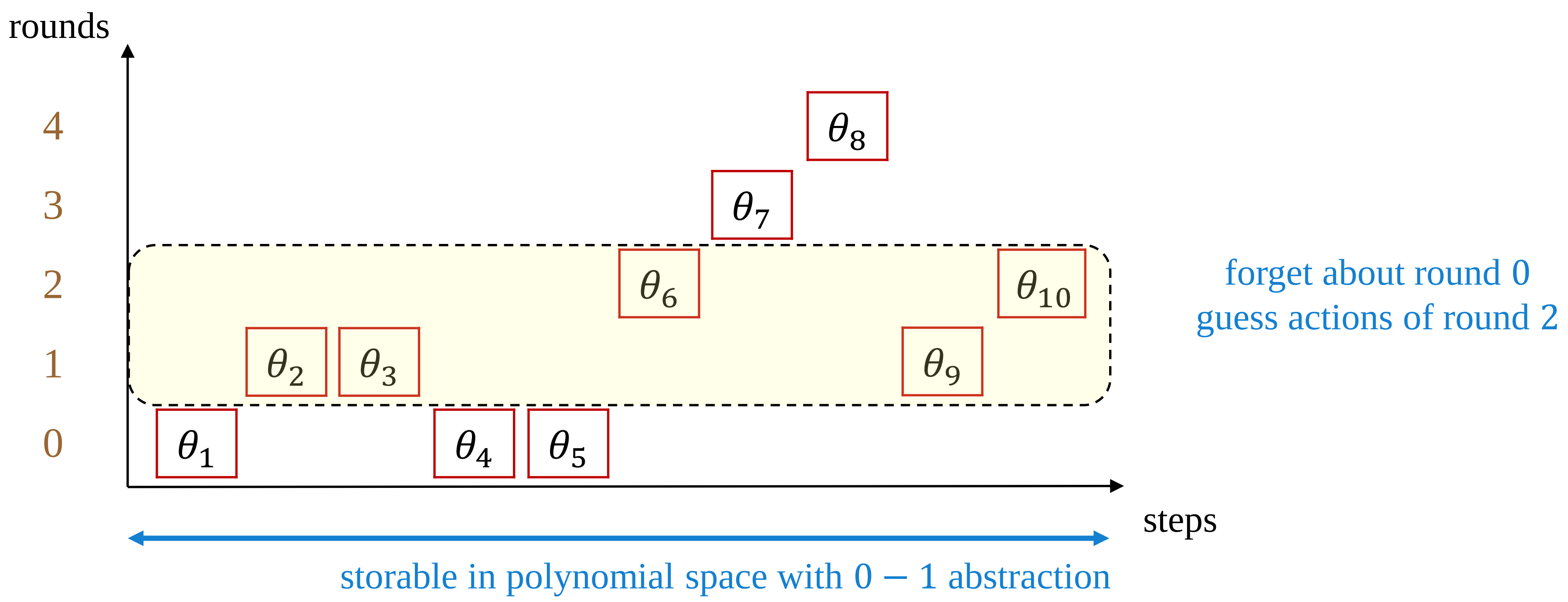


configurations may stretch across
exponentially many rounds

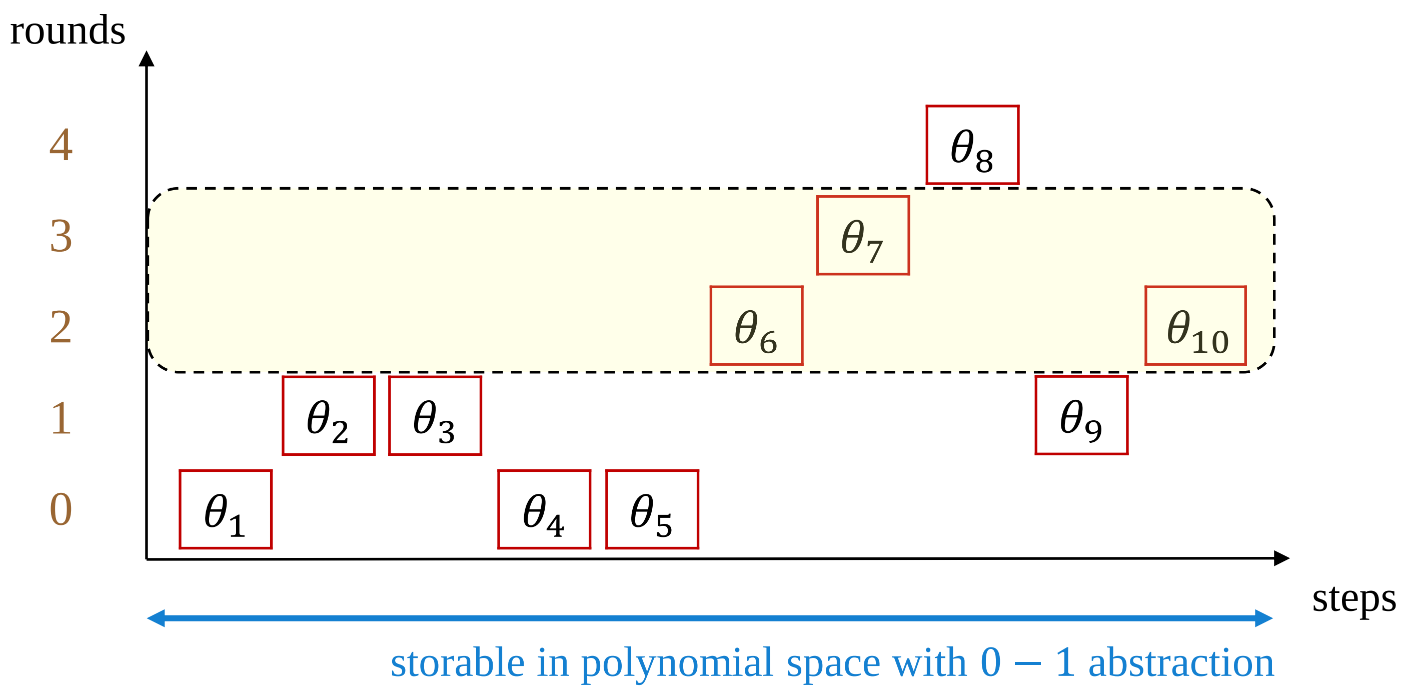
A non-deterministic polynomial-space algorithm



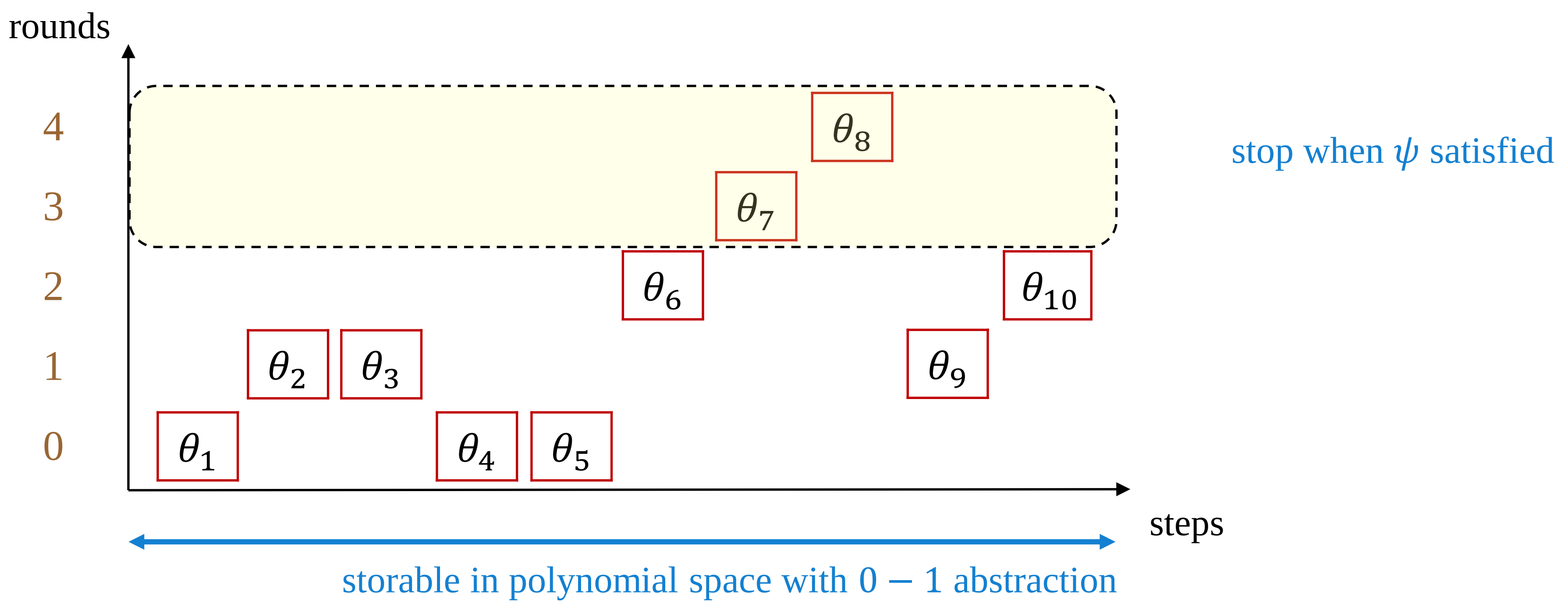
A non-deterministic polynomial-space algorithm



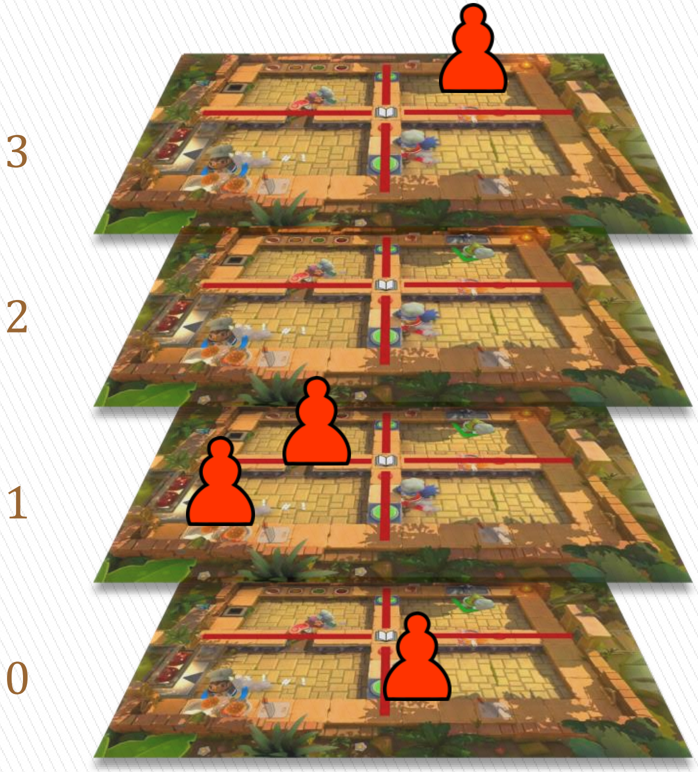
A non-deterministic polynomial-space algorithm



A non-deterministic polynomial-space algorithm



Third part
Round-based ASMS under stochastic schedulers



Stochastic scheduler

Needed for almost-sure termination in Aspnes' consensus algorithm.

Stochastic scheduler

Needed for almost-sure termination in Aspnes' consensus algorithm.

Qualitative probabilistic model-checking: does ρ satisfy the property with probability 1?

Stochastic scheduler

Needed for almost-sure termination in Aspnes' consensus algorithm.

Qualitative probabilistic model-checking: does ρ satisfy the property with probability 1?

almost-sure coverability: one process in q_f
almost-sure termination: all processes in q_f

Stochastic scheduler

Needed for almost-sure termination in Aspnes' consensus algorithm.

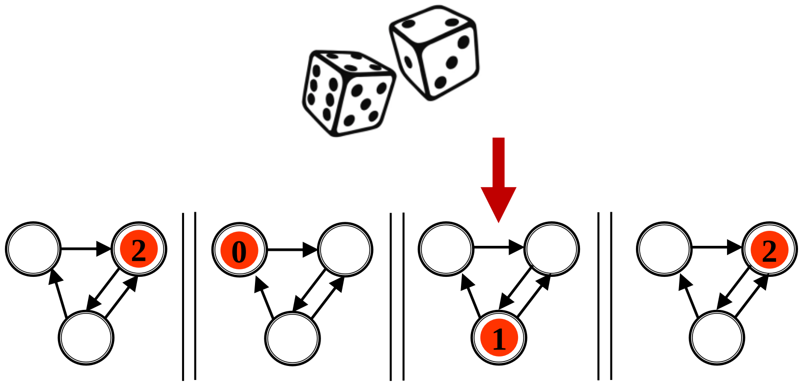
Qualitative probabilistic model-checking: does ρ satisfy the property with probability 1?

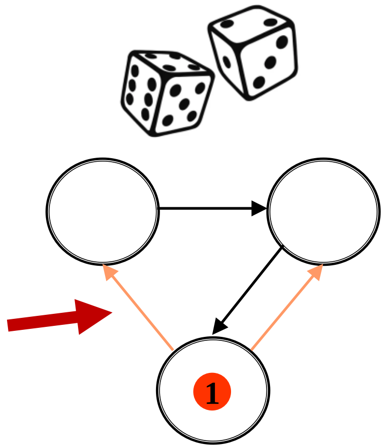
almost-sure coverability: one process in q_f

almost-sure termination: all processes in q_f

Memoryless and uniform stochastic scheduler:

select which process plays uniformly at random then select the transition uniformly at random among possible transitions





Is almost-sure COVER really probabilistic?

Proposition [BMRSS16]: In ASMS without rounds, for all $n \geq 1$:

$$\mathbb{P}_n\big(\mathbf{F}(\#q_f \geq 1)\big) = 1 \iff \mathrm{Post}^*(\mathrm{init}_n) \subseteq \mathrm{Pre}^*(\#q_f \geq 1).$$

q_f covered with probability 1 $\iff$ from every reachable configuration, q_f can be covered

[BMRSS16] Patricia Bouyer, Nicolas Markey, Mickael Randour, Arnaud Sangnier, and Daniel Stan. *Reachability in Networks of Shared-memory Protocols under Stochastic Schedulers*. ICALP 2016.

[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

Is almost-sure COVER really probabilistic?

Proposition [BMRSS16]: In ASMS without rounds, for all $n \geq 1$:

$$\mathbb{P}_n\big(\mathbf{F}(\#q_f \geq 1)\big) = 1 \iff \mathrm{Post}^*(\mathrm{init}_n) \subseteq \mathrm{Pre}^*(\#q_f \geq 1).$$

q_f covered with probability 1 $\Leftrightarrow$ from every reachable configuration, q_f can be covered

Proposition **[Wal24]**: There are round-based ASMS where, for all n large enough:

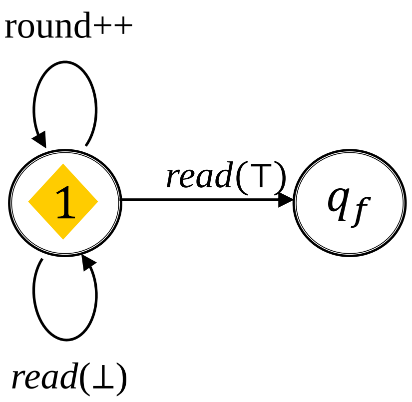
- $\mathrm{Post}^*(\mathrm{init}_n) \subseteq \mathrm{Pre}^*\big(\#q_f \geq 1\big)$ but
- $\mathbb{P}_n\big(\mathbf{F}(\#q_f \geq 1)\big) < 1$.

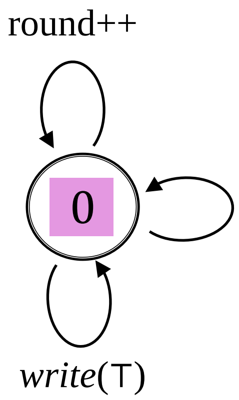
Because of random-walk behaviors (next slide).

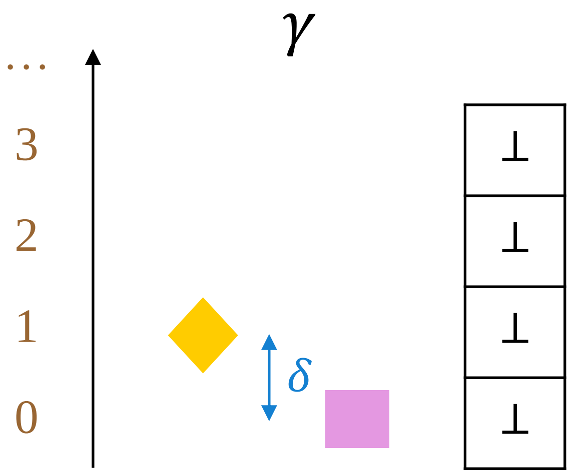
[BMRSS16] Patricia Bouyer, Nicolas Markey, Mickael Randour, Arnaud Sangnier, and Daniel Stan. *Reachability in Networks of Shared-memory Protocols under Stochastic Schedulers*. ICALP 2016.

[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems*. PhD thesis (submitted), 2024.

Random walk behaviors



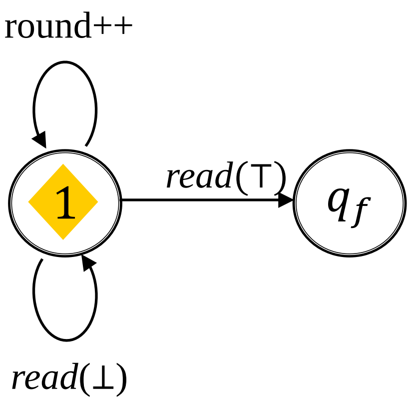


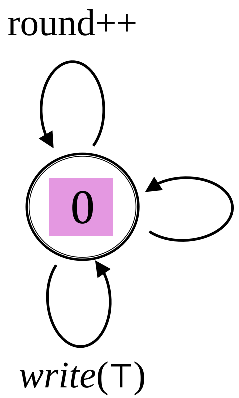


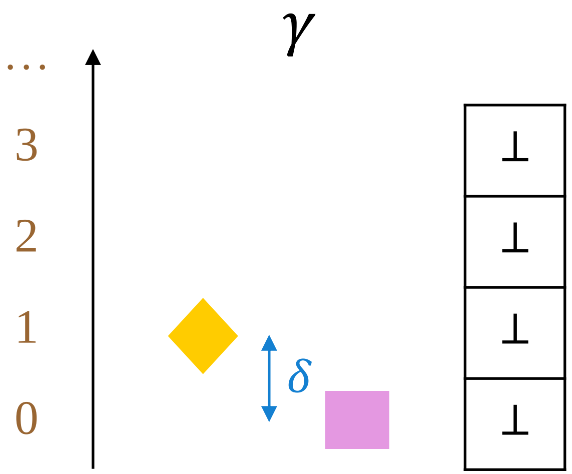
must read $\top$ to get to q_f

only can write $\top$

Random walk behaviors







must read $\top$ to get to q_f

only can write $\top$

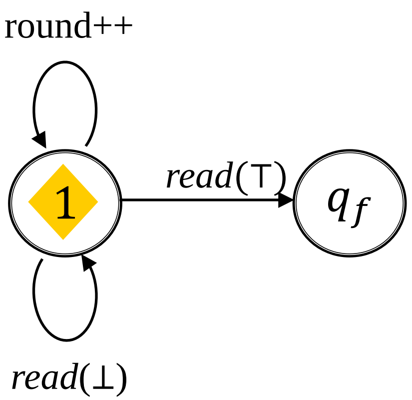
has probability $\frac{1}{2}$ to increment round

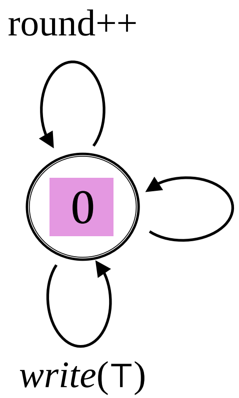
always 2 possible transitions, selected uniformly at random

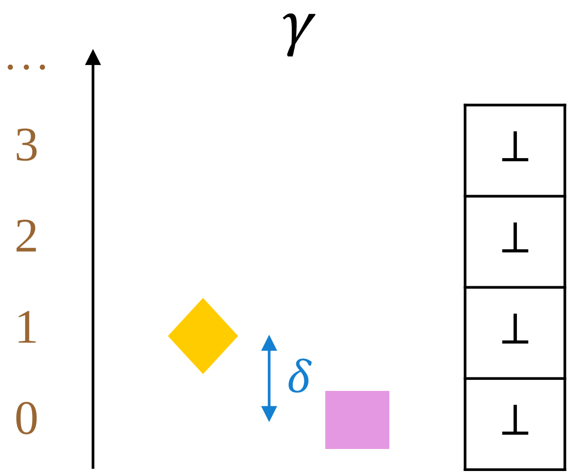
has probability $\frac{1}{3}$ to increment round

always 3 possible transitions

Random walk behaviors







must read ⊤ to get to q_f

has probability $\frac{1}{2}$ to increment round

always 2 possible transitions, selected uniformly at random

only can write ⊤

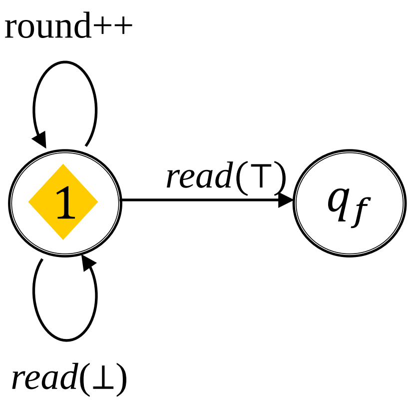
has probability $\frac{1}{3}$ to increment round

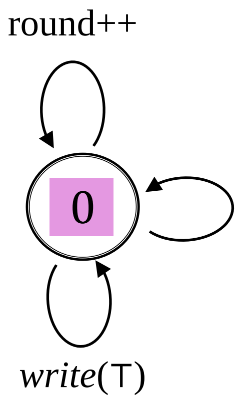
always 3 possible transitions

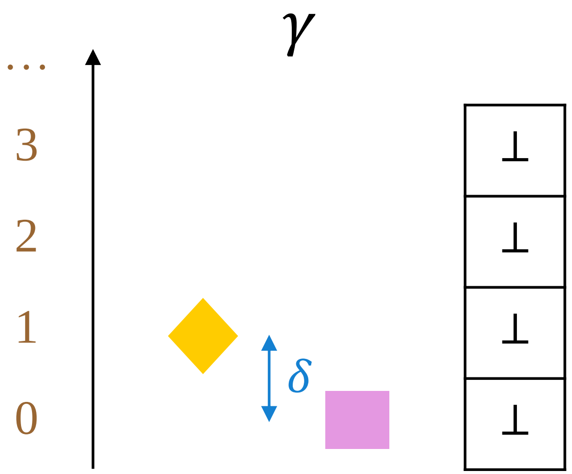
one-dimensional biased random walk

⇒ probability > 0 that always above

Random walk behaviors







must read ⊤ to get to q_f

has probability $\frac{1}{2}$ to increment round

always 2 possible transitions, selected uniformly at random

only can write ⊤

has probability $\frac{1}{3}$ to increment round

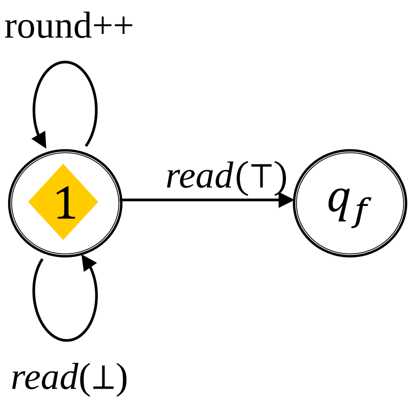
always 3 possible transitions

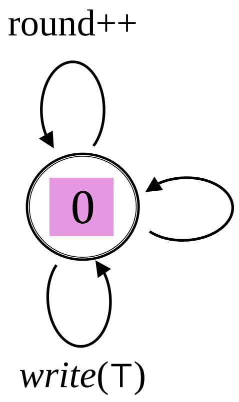
one-dimensional biased random walk

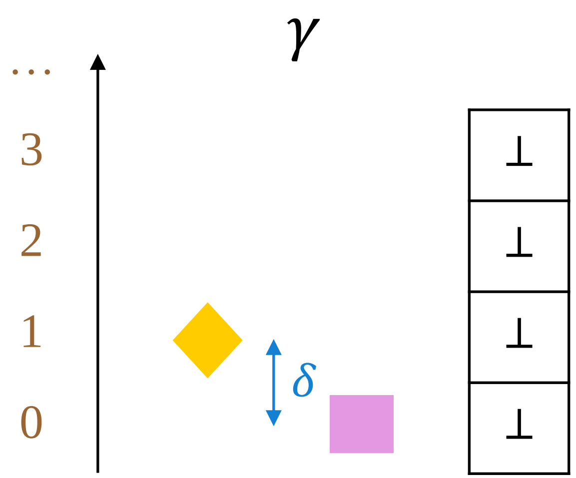
$\Rightarrow$ probability > 0 that always above

non-zero probability that q_f never covered from γ, but $\text{Post}^*(\gamma) \subseteq \text{Pre}^*(\#q_f \geq 1)$

Random walk behaviors







must read ⊤ to get to q_f

only can write ⊤

has probability $\frac{1}{2}$ to increment round

always 2 possible transitions, selected uniformly at random

has probability $\frac{1}{3}$ to increment round

always 3 possible transitions

one-dimensional biased random walk

⇒ probability > 0 that ◆ always above ■

non-zero probability that q_f never covered from γ, but $\text{Post}^*(\gamma) \subseteq \text{Pre}^*(\#q_f \geq 1)$

Some examples involve multi-dimensional random walk behaviors ☹

Almost-sure Obstruction Freedom

In fault-tolerant consensus algorithms, one process left in isolation must terminate.

Definition: A round-based ASMS is ASOF (almost-surely obstruction-free) when, for all n, for every reachable configuration, if all processes crash except one then this process ends in q_f with probability one.

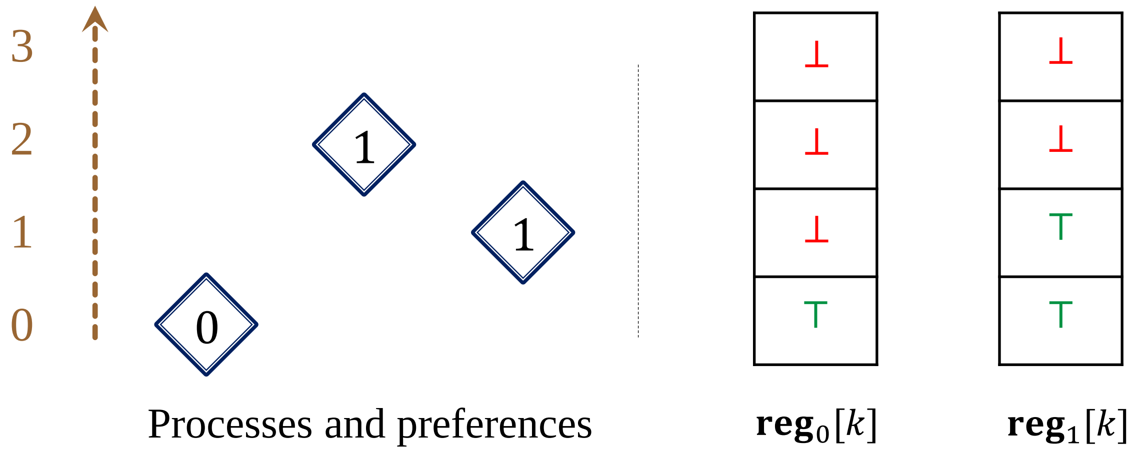
[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

Almost-sure Obstruction Freedom

In fault-tolerant consensus algorithms, one process left in isolation must terminate.

Definition: A round-based ASMS is ASOF (almost-surely obstruction-free) when, for all n, for every reachable configuration, if all processes crash except one then this process ends in q_f with probability one.

In Aspnes' consensus algorithm:



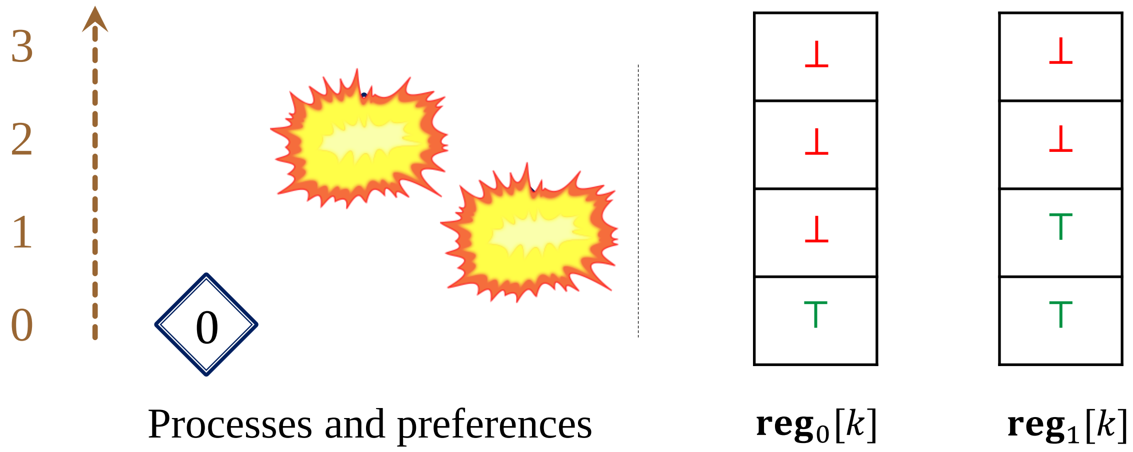
[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

Almost-sure Obstruction Freedom

In fault-tolerant consensus algorithms, one process left in isolation must terminate.

Definition: A round-based ASMS is ASOF (almost-surely obstruction-free) when, for all n, for every reachable configuration, if all processes crash except one then this process ends in q_f with probability one.

In Aspnes' consensus algorithm:



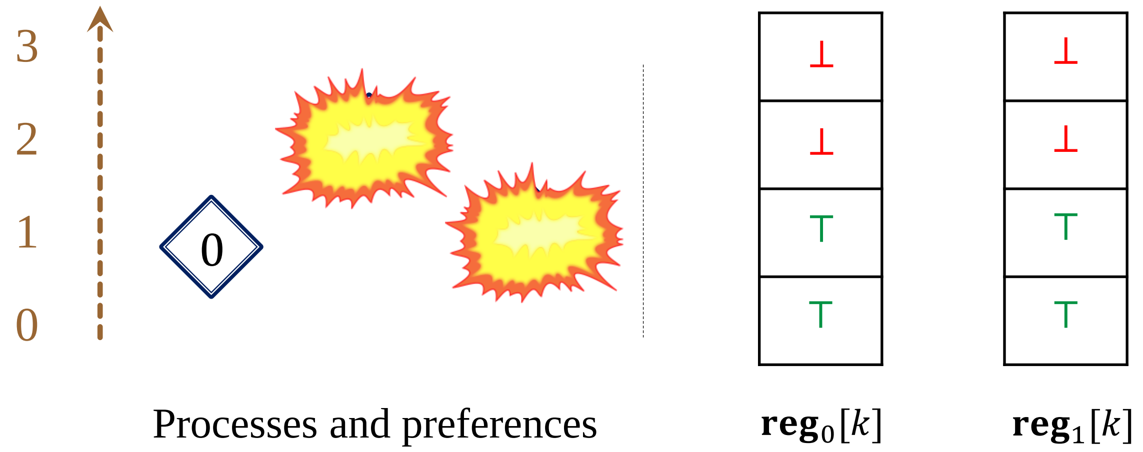
[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

Almost-sure Obstruction Freedom

In fault-tolerant consensus algorithms, one process left in isolation must terminate.

Definition: A round-based ASMS is ASOF (almost-surely obstruction-free) when, for all n, for every reachable configuration, if all processes crash except one then this process ends in q_f with probability one.

In Aspnes' consensus algorithm:



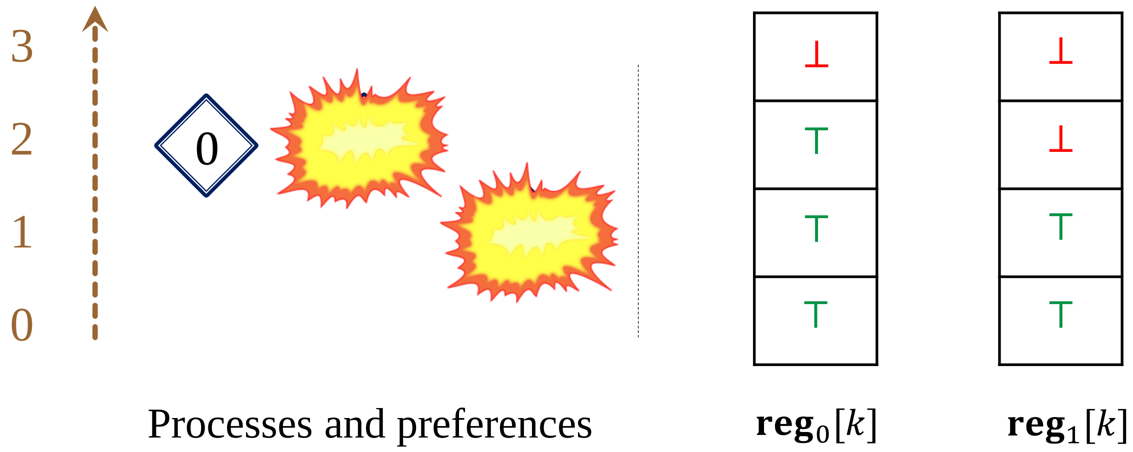
[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

Almost-sure Obstruction Freedom

In fault-tolerant consensus algorithms, one process left in isolation must terminate.

Definition: A round-based ASMS is ASOF (almost-surely obstruction-free) when, for all n, for every reachable configuration, if all processes crash except one then this process ends in q_f with probability one.

In Aspnes' consensus algorithm:



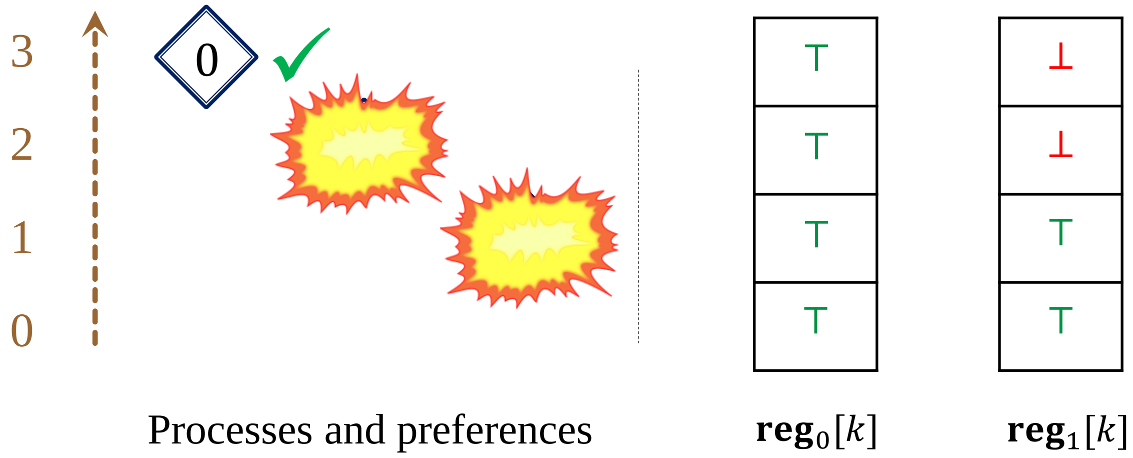
[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

Almost-sure Obstruction Freedom

In fault-tolerant consensus algorithms, one process left in isolation must terminate.

Definition: A round-based ASMS is ASOF (almost-surely obstruction-free) when, for all n, for every reachable configuration, if all processes crash except one then this process ends in q_f with probability one.

In Aspnes' consensus algorithm:



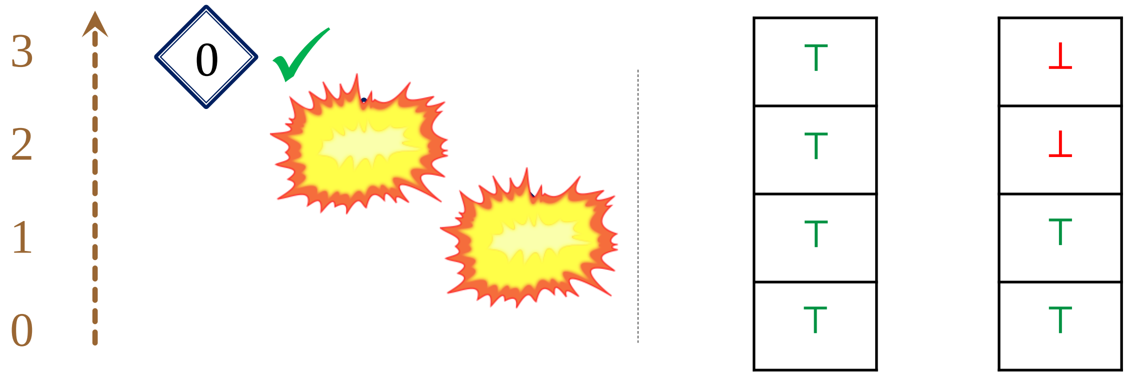
[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

Almost-sure Obstruction Freedom

In fault-tolerant consensus algorithms, one process left in isolation must terminate.

Definition: A round-based ASMS is ASOF (almost-surely obstruction-free) when, for all n, for every reachable configuration, if all processes crash except one then this process ends in q_f with probability one.

In Aspnes' consensus algorithm:



Processes and preferences $\mathbf{reg}_0[k]$ $\mathbf{reg}_1[k]$

Proposition **[Wal24]***:* ASOF implies that, for all n, all processes almost-surely terminate.

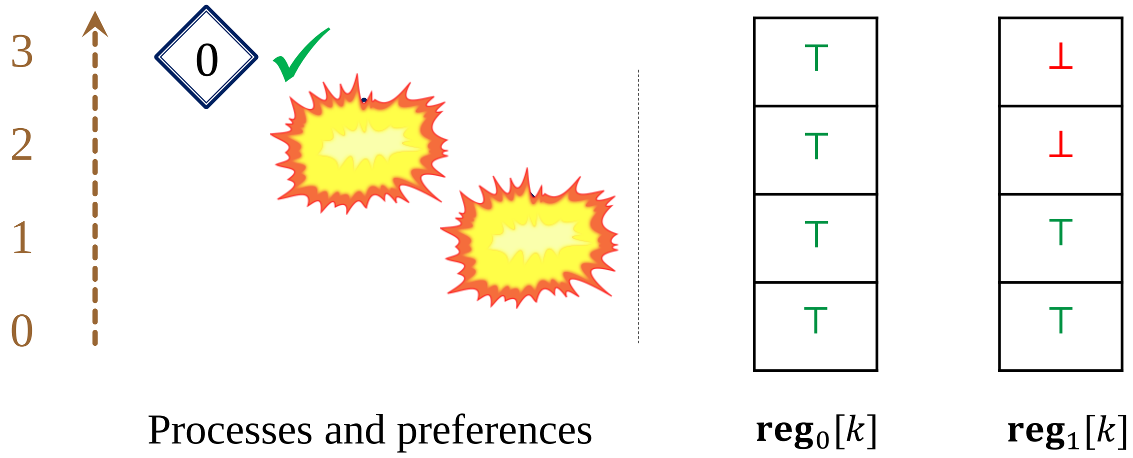
[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

Almost-sure Obstruction Freedom

In fault-tolerant consensus algorithms, one process left in isolation must terminate.

Definition: A round-based ASMS is ASOF (almost-surely obstruction-free) when, for all n, for every reachable configuration, if all processes crash except one then this process ends in q_f with probability one.

In Aspnes' consensus algorithm:



Proposition **[Wal24]***:* ASOF implies that, for all n, all processes almost-surely terminate.

Theorem **[Wal24]***:* Deciding whether a round-based ASMS is ASOF is PSPACE-complete.

[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024.

Conclusion

Summary

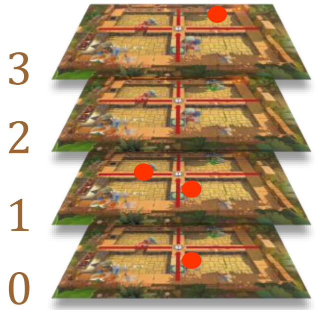
First part: Shared-memory systems

- A parameterized model: asynchronous shared-memory systems (ASMS)
- Problems: reachability problems, presence reachability problem (PRP)
- Between PTIME and NP-complete
- A doubly-exponential bound on the diameter.

Summary

First part: Shared-memory systems

- A parameterized model: asynchronous shared-memory systems (ASMS)
- Problems: reachability problems, presence reachability problem (PRP)
- Between PTIME and NP-complete
- A doubly-exponential bound on the diameter.



Second part: Round-based shared-memory systems

- Captures round-based algorithms such as Aspnes' consensus algorithm
- PRP with quantification over round values
- PRP is decidable and PSPACE-complete

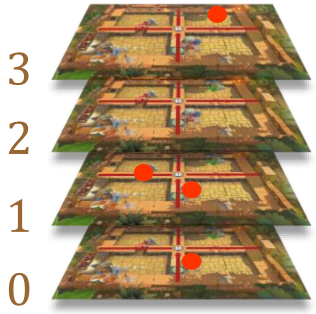
Summary

First part: Shared-memory systems

- A parameterized model: asynchronous shared-memory systems (ASMS)
- Problems: reachability problems, presence reachability problem (PRP)
- Between PTIME and NP-complete
- A doubly-exponential bound on the diameter.

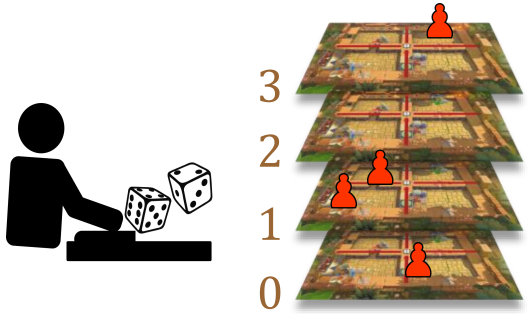
Second part: Round-based shared-memory systems

- Captures round-based algorithms such as Aspnes' consensus algorithm
- PRP with quantification over round values
- PRP is decidable and PSPACE-complete



Third part: Round-based ASMS under a stochastic scheduler

- Random walk behaviors
- ASOF property, related to fault-tolerant algorithms
- ASOF implies almost-sure termination and can be decided in PSPACE



Other contributions

- On broadcast networks where processes have identifiers and private registers:

Lucie Guillou, Corto Mascle and **Nicolas Waldburger.** *Parameterized Broadcast Networks with Registers: from NP to the Frontiers of Decidability.* FoSSaCS 2024.

- On population protocols where processes have unchangeable data that can be tested for equality:

Steffen van Bergerem, Roland Guttenberg, Sandra Kiefer, Corto Mascle, **Nicolas Waldburger** and Chana Weil-Kennedy. *Verification of Population Protocols with Unordered Data*. ICALP 2024.

- On one-counter automata with disequality test on the counter:

Dmitry Chistikov, Jérôme Leroux, Henry Sinclair-Banks and **Nicolas Waldburger**. *Invariants for One-Counter Automata with Disequality Tests*. CONCUR 2024.

- On hyperLTL for population protocols (uses transfer flow techniques):

Nicolas Waldburger, Chana Weil-Kennedy, Pierre Ganty and César Sánchez. *Temporal Hyperproperties for Population Protocols*. In preparation (unpublished).

Work performed during a stay at IMDEA Madrid (funded by Rennes Métropole)

Perspectives

Open problem: Can the bound of the diameter from the structural theorem be improved to simply-exponential?

The structural theorem can be phrased in a more general fashion and for more general systems.
This makes it a generalized phrasing of preexisting open problems [BMRSS16][BGW22].

Open problem: Complexity and decidability of almost-sure reachability problems for round-based ASMS?

Mathematically very challenging.

Future work: Find models that retain decidability of (at least) COVER while capturing more round-based algorithms.

[BMRSS16] Patricia Bouyer, Nicolas Markey, Mickael Randour, Arnaud Sangnier, and Daniel Stan. *Reachability in Networks of Shared-memory Protocols under Stochastic Schedulers*. ICALP 2016.

[BGW22] A. R. Balasubramanian, Lucie Guillou, and Chana Weil-Kennedy. *Parameterized Analysis of Reconfigurable Broadcast Networks*, FoSSaCS 2022.

Perspectives

Open problem: Can the bound of the diameter from the structural theorem be improved to simply-exponential?

The structural theorem can be phrased in a more general fashion and for more general systems.
This makes it a generalized phrasing of preexisting open problems [BMRSS16][BGW22].

Open problem: Complexity and decidability of almost-sure reachability problems for round-based ASMS?

Mathematically very challenging.

Future work: Find models that retain decidability of (at least) COVER while capturing more round-based algorithms.

Thank you for your attention!

[BMRSS16] Patricia Bouyer, Nicolas Markey, Mickael Randour, Arnaud Sangnier, and Daniel Stan. *Reachability in Networks of Shared-memory Protocols under Stochastic Schedulers*. ICALP 2016.

[BGW22] A. R. Balasubramanian, Lucie Guillou, and Chana Weil-Kennedy. *Parameterized Analysis of Reconfigurable Broadcast Networks*, FoSSaCS 2022.

Complexity of COVER in ASMS

COVER $\quad \exists n, \exists \rho: \text{init}_n \to^* \gamma\,,\ \ \gamma(q_f) \geq 1$?

Theorem **[Wal23]**: COVER is NP-complete.

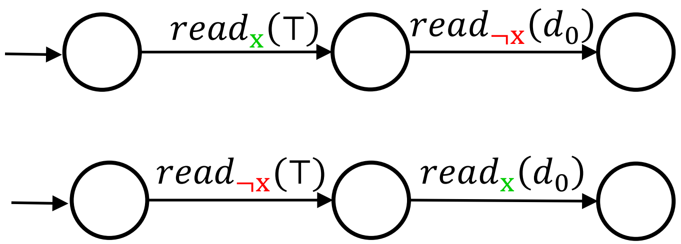
Reduction from 3-SAT

For each variable x in the SAT formula:

x	d_0
¬x	$\top$

Check x:

Check $\neg x$:



x true when x written *before* ¬x

x false when ¬x written *before* x

Directly relies on initialization of the registers !

Proposition **[Wal23]**: COVER is PTIME when registers are not initialized, or when the number of registers is fixed.

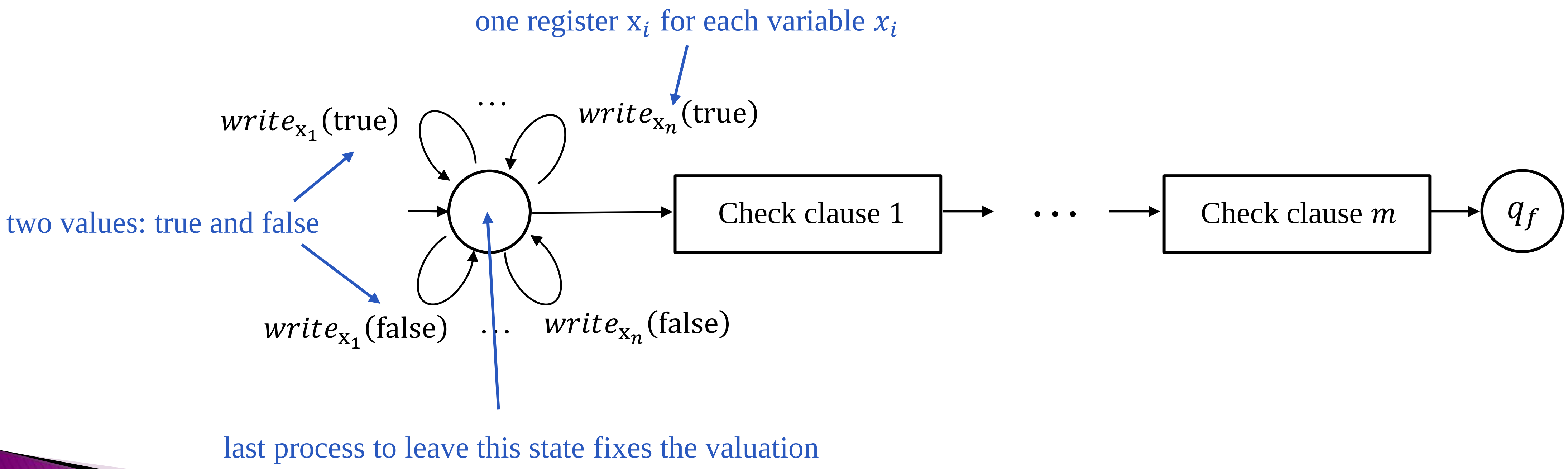
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems*. MFCS 2023.

Complexity of TARGET in ASMS

TARGET $\exists n, \exists \rho\colon \mathrm{init}_n \rightarrow^* \gamma, \forall q \neq q_f,\ \gamma(q) = 0?$

Theorem **[Wal23]**: TARGET is NP-complete, and NP-hard already with uninitialized registers.

Again, reduction from 3-SAT.



[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.

TARGET in ASMS with a single register

TARGET $\exists n, \exists \rho \colon \mathrm{init}_n \to^* \gamma \ , \forall q \neq q_f, \ \gamma(q) = 0?$

Theorem **[Wal23]**: TARGET is in PTIME when one register only.

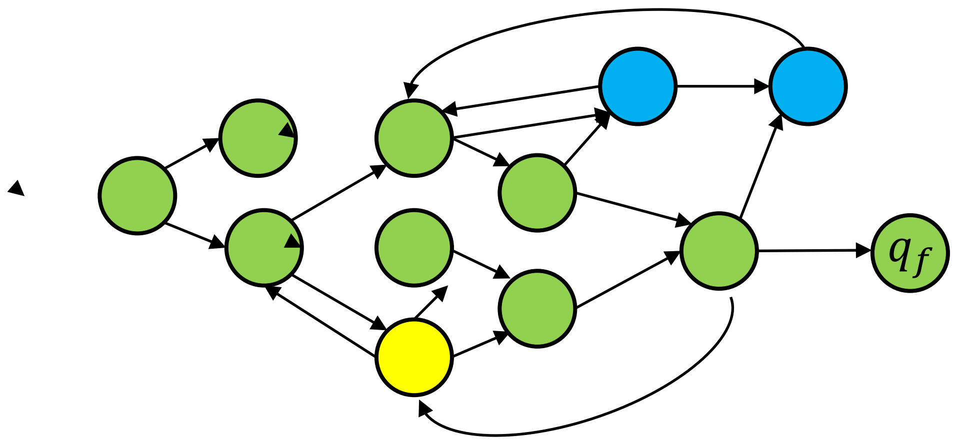
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.

TARGET in ASMS with a single register

TARGET $\exists n, \exists \rho: \text{init}_n \rightarrow^* \gamma, \forall q \neq q_f, \ \gamma(q) = 0?$

Theorem **[Wal23]**: TARGET is in PTIME when one register only.

Algorithm inspired by broadcast protocols [Fou15].



= state q is *coverable*
= there are n, γ and $\rho: \text{init}_n \rightarrow^* \gamma$ such that $\gamma(q)$>0

= state q is *backwards coverable*
= there are n, γ, γ' and $\rho: \gamma \rightarrow^* \gamma'$ (starting with a write action) such that $\gamma(q)$>0 and $\gamma'(q') = 0$ for all $q' \neq q_f$

= both coverable and backwards coverable

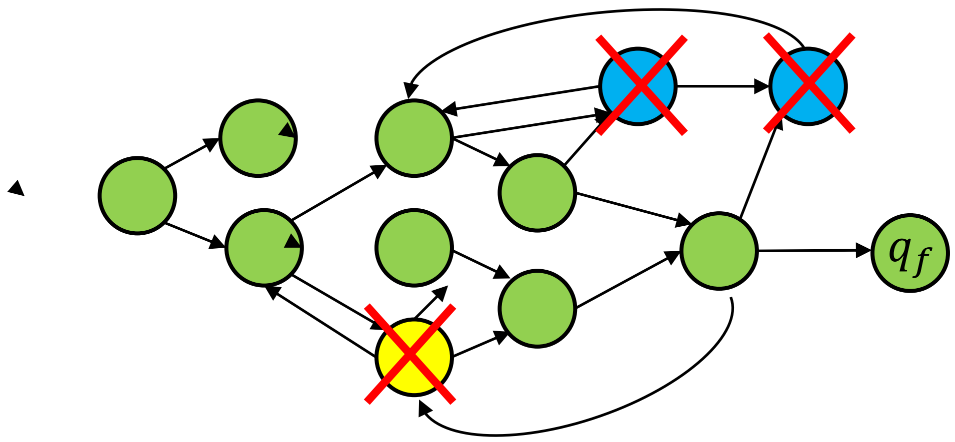
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.
[Fou15] Paulin Fournier. *Parameterized verification of networks of many identical processes.* PhD thesis, 2015

TARGET in ASMS with a single register

TARGET $\quad \exists n, \exists \rho\colon \mathrm{init}_n \to^* \gamma, \forall q \neq q_f,\ \gamma(q) = 0?$

Theorem **[Wal23]**: TARGET is in PTIME when one register only.

Algorithm inspired by broadcast protocols [Fou15].



= state q is *coverable*
= there are n, γ and $\rho\colon \mathrm{init}_n \to^* \gamma$ such that $\gamma(q)>0$

= state q is *backwards coverable*
= there are n, γ, γ' and $\rho\colon \gamma \to^* \gamma'$ (starting with a write action) such that $\gamma(q)>0$ and $\gamma'(q') = 0$ for all $q' \neq q_f$

= both coverable and backwards coverable

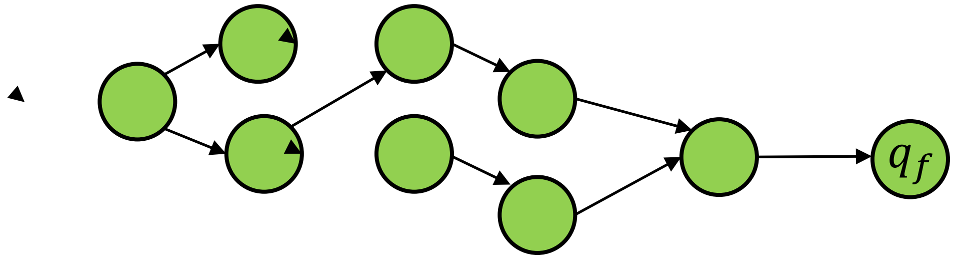
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.
[Fou15] Paulin Fournier. *Parameterized verification of networks of many identical processes.* PhD thesis, 2015

TARGET in ASMS with a single register

TARGET $\exists n, \exists \rho: \mathrm{init}_n \rightarrow^* \gamma, \forall q \neq q_f,\ \gamma(q) = 0?$

Theorem **[Wal23]**: TARGET is in PTIME when one register only.

Algorithm inspired by broadcast protocols [Fou15].



= state q is *coverable*
= there are n, γ and $\rho: \mathrm{init}_n \rightarrow^* \gamma$ such that $\gamma(q)>0$

= state q is *backwards coverable*
= there are n, γ, γ' and $\rho: \gamma \rightarrow^* \gamma'$ (starting with a write action) such that $\gamma(q)>0$ and $\gamma'(q') = 0$ for all $q' \neq q_f$

= both coverable and backwards coverable

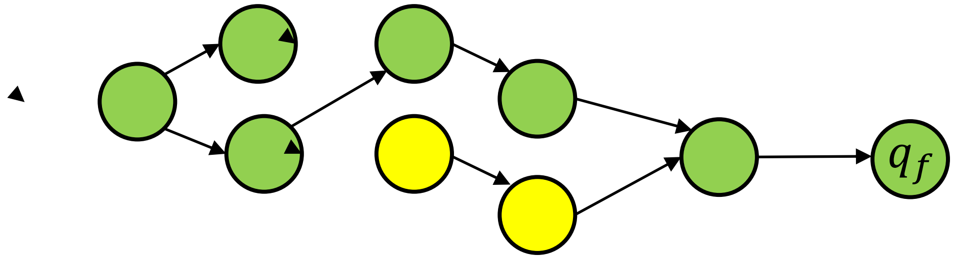
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.
[Fou15] Paulin Fournier. *Parameterized verification of networks of many identical processes.* PhD thesis, 2015

TARGET in ASMS with a single register

TARGET $\exists n, \exists \rho \colon \mathrm{init}_n \to^* \gamma, \forall q \neq q_f,\ \gamma(q) = 0?$

Theorem **[Wal23]**: TARGET is in PTIME when one register only.

Algorithm inspired by broadcast protocols [Fou15].



= state q is *coverable*
= there are n, γ and $\rho\colon \mathrm{init}_n \to^* \gamma$ such that $\gamma(q)$>0

= state q is *backwards coverable*
= there are n, γ, γ' and $\rho\colon \gamma \to^* \gamma'$ (starting with a write action) such that $\gamma(q)$>0 and $\gamma'(q') = 0$ for all $q' \neq q_f$

= both coverable and backwards coverable

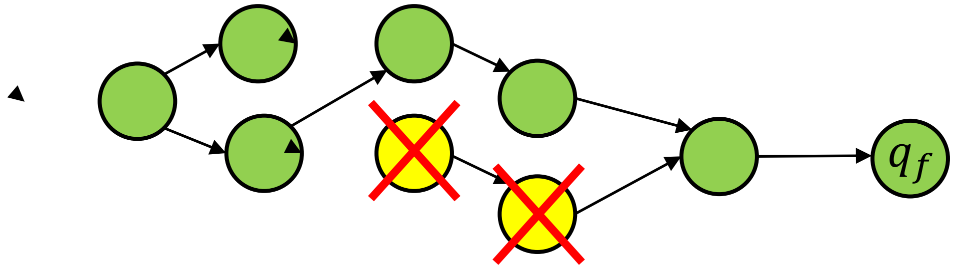
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.
[Fou15] Paulin Fournier. *Parameterized verification of networks of many identical processes.* PhD thesis, 2015

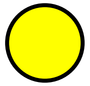
TARGET in ASMS with a single register

TARGET $\exists n, \exists \rho: \mathrm{init}_n \to^* \gamma, \forall q \neq q_f,\ \gamma(q) = 0?$

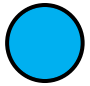
Theorem **[Wal23]**: TARGET is in PTIME when one register only.

Algorithm inspired by broadcast protocols [Fou15].

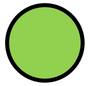




= state q is *coverable*
= there are n, γ and $\rho: \mathrm{init}_n \to^* \gamma$ such that $\gamma(q)>0$



= state q is *backwards coverable*
= there are n, γ, γ' and $\rho: \gamma \to^* \gamma'$ (starting with a write action) such that $\gamma(q)>0$ and $\gamma'(q') = 0$ for all $q' \neq q_f$



= both coverable and backwards coverable

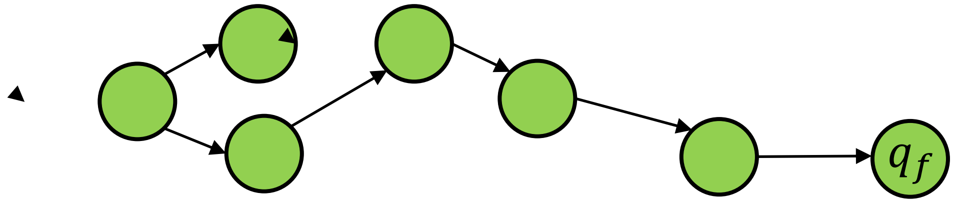
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.
[Fou15] Paulin Fournier. *Parameterized verification of networks of many identical processes.* PhD thesis, 2015

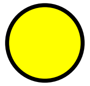
TARGET in ASMS with a single register

TARGET $\exists n, \exists \rho : \mathrm{init}_n \rightarrow^* \gamma, \forall q \neq q_f, \ \gamma(q) = 0?$

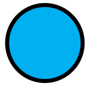
Theorem **[Wal23]**: TARGET is in PTIME when one register only.

Algorithm inspired by broadcast protocols [Fou15].

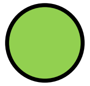




= state q is *coverable*
= there are n, γ and ρ: $\mathrm{init}_n \rightarrow^* \gamma$ such that $\gamma(q)$>0



= state q is *backwards coverable*
= there are n, γ, γ' and ρ: $\gamma \rightarrow^* \gamma'$ (starting with a write action) such that $\gamma(q)$>0 and $\gamma'(q') = 0$ for all $q' \neq q_f$



= both coverable and backwards coverable

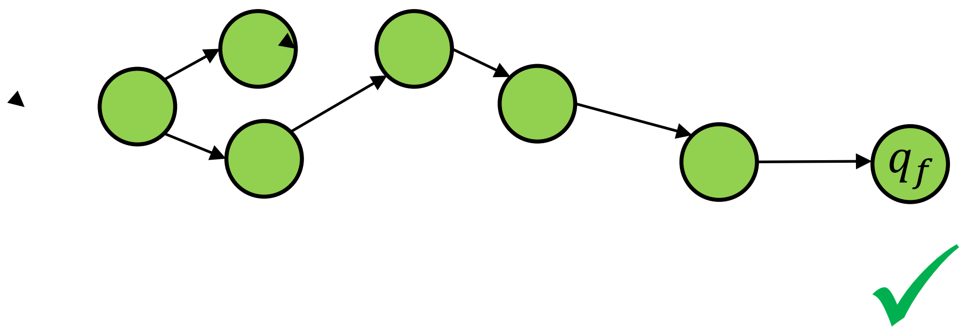
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.
[Fou15] Paulin Fournier. *Parameterized verification of networks of many identical processes.* PhD thesis, 2015

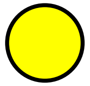
TARGET in ASMS with a single register

TARGET $\exists n, \exists \rho\colon \mathrm{init}_n \to^* \gamma, \forall q \neq q_f,\ \gamma(q) = 0?$

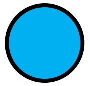
Theorem **[Wal23]**: TARGET is in PTIME when one register only.

Algorithm inspired by broadcast protocols [Fou15].

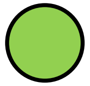




= state q is *coverable*
= there are n, γ and $\rho\colon \mathrm{init}_n \to^* \gamma$ such that $\gamma(q)>0$



= state q is *backwards coverable*
= there are n, γ, γ' and $\rho\colon \gamma \to^* \gamma'$ (starting with a write action) such that $\gamma(q)>0$ and $\gamma'(q') = 0$ for all $q' \neq q_f$



= both coverable and backwards coverable

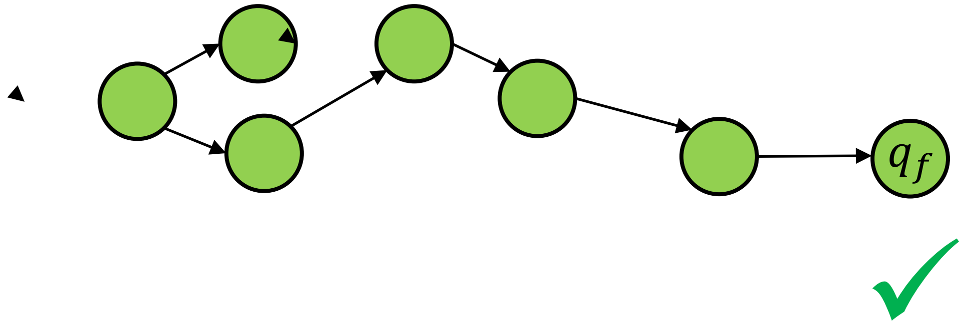
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.
[Fou15] Paulin Fournier. *Parameterized verification of networks of many identical processes.* PhD thesis, 2015

TARGET in ASMS with a single register

TARGET $\exists n, \exists \rho\colon \mathrm{init}_n \to^* \gamma, \forall q \neq q_f,\ \gamma(q) = 0?$

Theorem **[Wal23]**: TARGET is in PTIME when one register only.

Algorithm inspired by broadcast protocols [Fou15].



= state q is *coverable*
= there are n, γ and $\rho\colon \mathrm{init}_n \to^* \gamma$ such that $\gamma(q)>0$

= state q is *backwards coverable*
= there are n, γ, γ' and $\rho\colon \gamma \to^* \gamma'$ (starting with a write action) such that $\gamma(q)>0$ and $\gamma'(q') = 0$ for all $q' \neq q_f$

= both coverable and backwards coverable

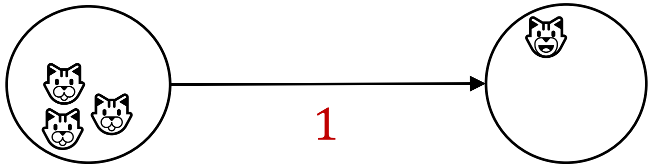
Corollary **[Wal23]**: PRP is in PTIME when one register only and the formula is in DNF.

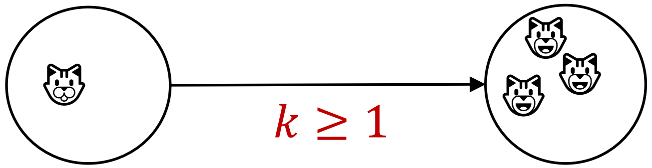
[Wal23] Nicolas Waldburger. *Checking Presence Reachability Properties on Parameterized Shared-Memory Systems.* MFCS 2023.
[Fou15] Paulin Fournier. *Parameterized verification of networks of many identical processes.* PhD thesis, 2015

Complexity results for ASMS

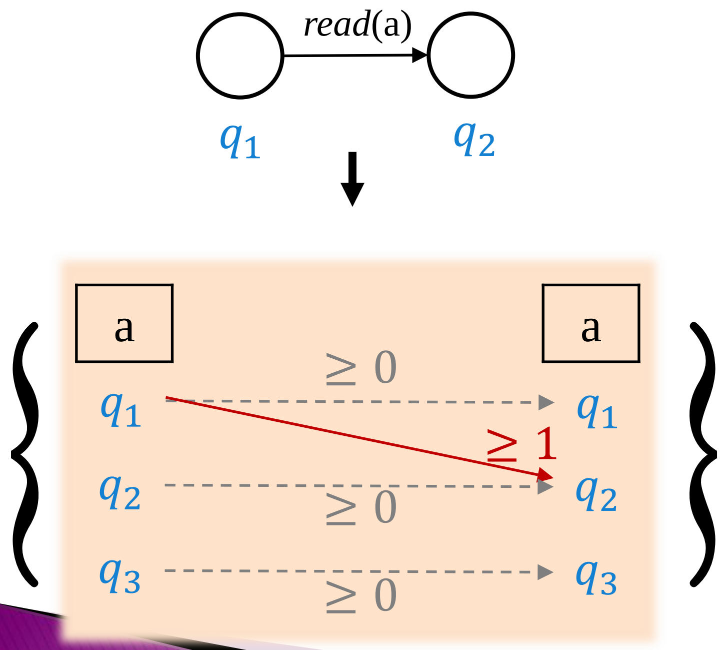
	COVER	TARGET	DNF-PRP	PRP
General case	NP-complete	NP-complete	NP-complete	NP-complete
Not initialized	PTIME-complete	NP-complete	NP-complete	NP-complete
One register	PTIME-complete	PTIME-complete	PTIME-complete	NP-complete
Number of registers as parameter	FPT	W[2]-hard	W[2]-hard	W[2]-hard

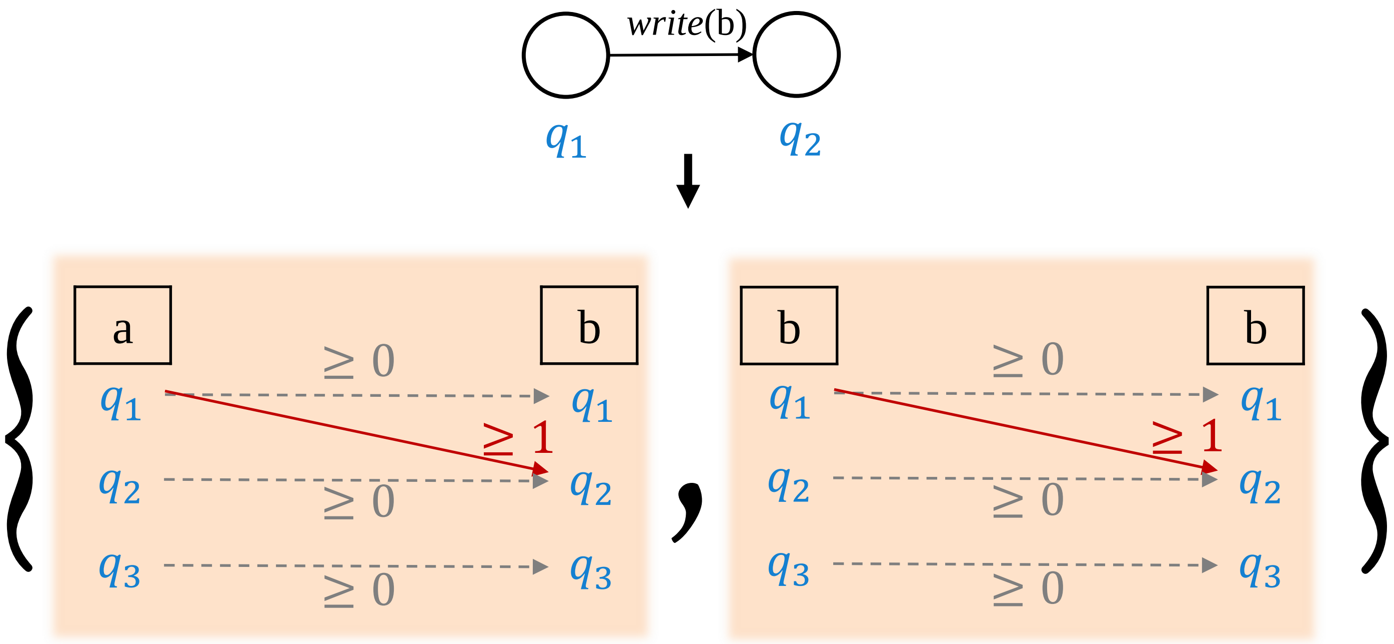
Transfer flows

Copycat property: if 

then 

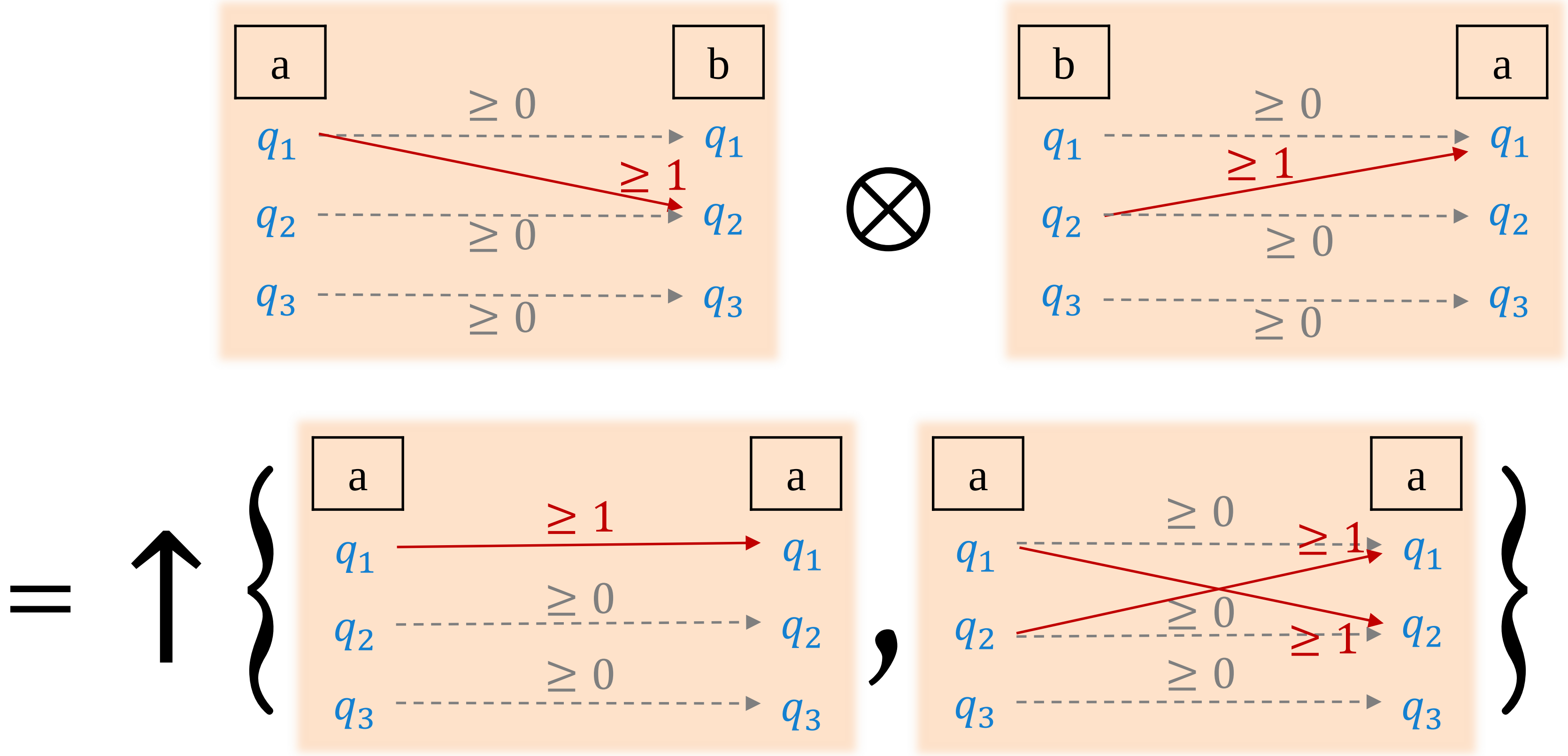
Each transition is represented by a set of *transfer flows*.



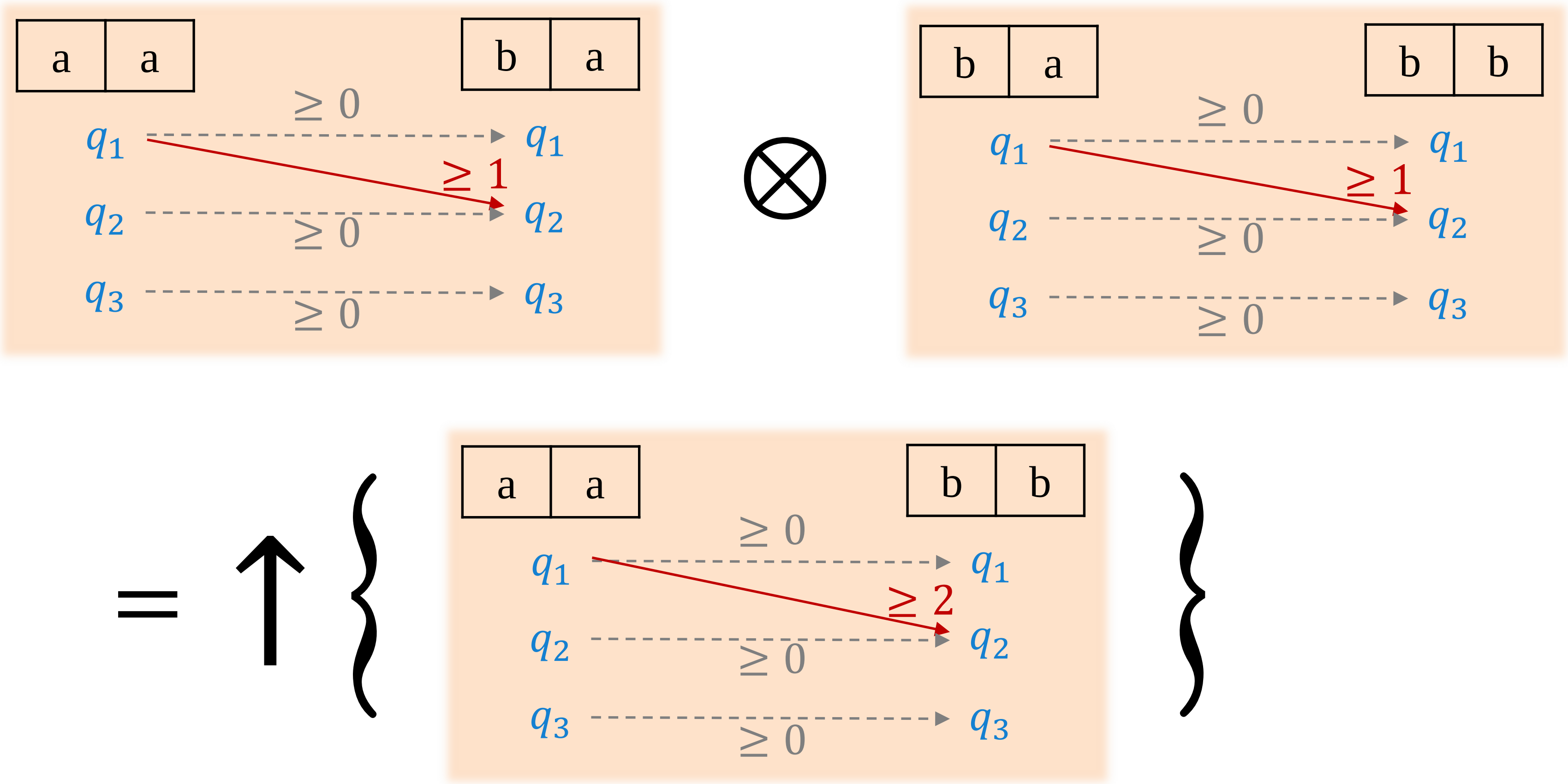


Composition of transfer flows

Given transfer flows $\mathrm{tf}_1, \mathrm{tf}_2$, set $\mathrm{tf}_1 \otimes \mathrm{tf}_2$ = transfer flows possible with tf_1 then tf_2.



Composition of transfer flows 2



The structural theorem on transfer flows

$\mathcal{T}_1$ $:=$ transfer flows of single transitions
$\mathcal{T}_k$ $:= \mathcal{T}_{k-1} \otimes \mathcal{T}_1$ $\sim$ executions of less than k transitions
$\mathcal{T}^*$ $:= \bigcup_{k\in\mathbb{N}} \mathcal{T}_k$ $\sim$ all possible executions

Structural Theorem **[Wal24]**: $\mathcal{T}^* = \mathcal{T}_B$ for B doubly-exponential in the size of the system.

Proof with:

- well-quasi-order theory,
- a transformation of transfer flows into vectors,
- a bound on the length of descending chains of $\mathbb{N}^d$ that generalizes Rackoff's bound [SS24].

[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis, 2024.
[SS24] Sylvain Schmitz, Lia Schütze. *On the Length of Strongly Monotone Descending Chains over* $\mathbb{N}^d$. ICALP 2024.

The consensus problem

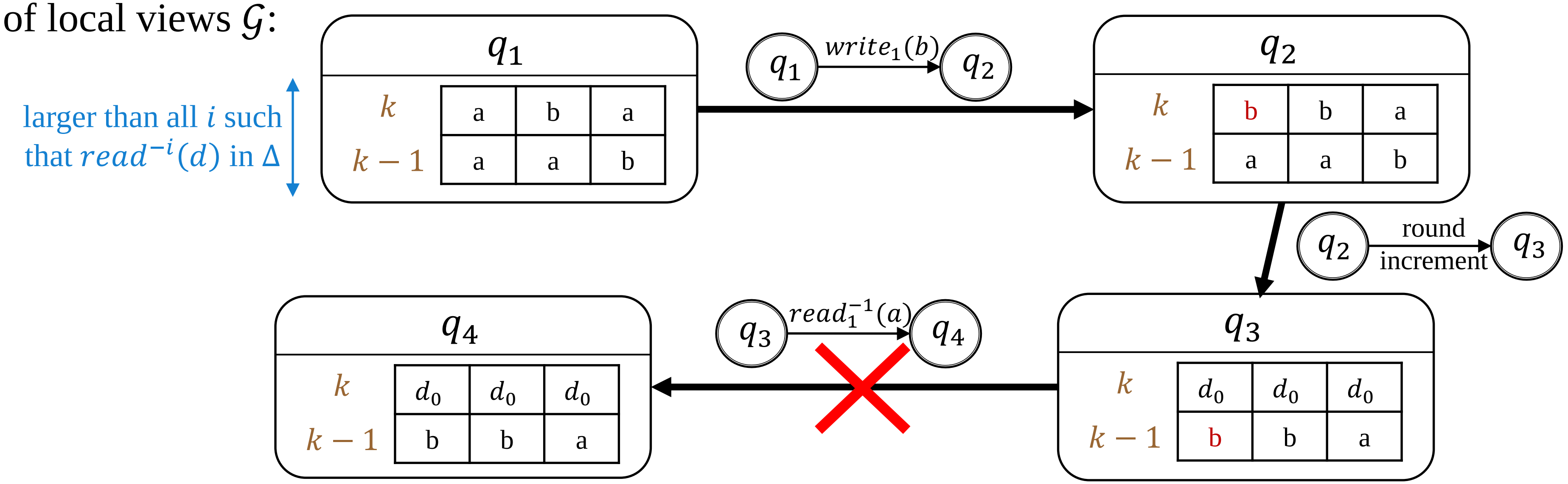
Binary consensus problem:
Each process starts with an initial preference $p \in \{0,1\}$.
Validity: If a process decided value p, some process started with preference p.
Agreement: Two processes that decide decide of the same value.
Termination: All processes eventually decide of a value.

[FLP85] Michael J. Fischer, Nancy A. Lynch, and Mike Paterson.
Impossibility of Distributed Consensus with One Faulty Process. JACM 1985.

Deciding ASOF

Theorem **[Wal24]**: Deciding whether a round-based ASMS is ASOF is a PSPACE-complete problem.

Graph of local views $\mathcal{G}$:



Lemma **[Wal24]**: not ASOF $\Leftrightarrow$ there is $\ell \in \mathcal{G}$ reachable whose SCC is bottom and does not have vertices with q_f.

reachable =there is $\gamma \in \text{Post}^*(\Gamma_0)$ such that some process in γ has local view ℓ
can be reduced to PRP

$\blacksquare$

[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis (submitted), 2024

Deciding ASOF with the local view graph

Lemma **[Wal24]**: The round-based ASMS is not ASOF if and only if there is a local view ℓ such that:

- there are n, γ reachable from $\gamma_0(n)$ such that some process in γ has local view ℓ,
- the strongly connected component S of ℓ in $\mathcal{G}$ is bottom,
- S does not have any vertex with state q_f.

Non-deterministic polynomial-space algorithm to decide whether a protocol is not ASOF:

- Guess ℓ,
- Check the existence of γ (reduces to round-based PRP),
- explore $\mathcal{G}$ to check the conditions on S.

$\blacksquare$

[Wal24] Nicolas Waldburger. *Parameterized verification of distributed shared-memory systems.* PhD thesis, 2024.