Selective Monitoring Without Delay for Probabilistic Systems

Julie Parreaux[1]
supervised by Stefan Kiefer[2]

[1]ENS Rennes

[2]Oxford University

August 30, 2018

Introduction

Challenge: fault detection and control of large complex systems in runtime

Formal verification methods to proof the correctness in runtime

- Test: feasible but not proof
- Model Checking: proof for all runs but not feasible
- Monitoring: proof for the current run

Introduction

Challenge: fault detection and control of large complex systems in runtime

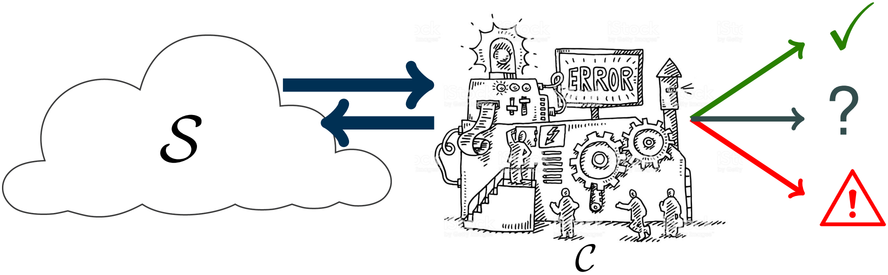
Formal verification methods to proof the correctness in runtime

- Test: feasible but not proof
- Model Checking: proof for all runs but not feasible
- **Monitoring: proof for the current run**

Monitoring and probabilities

Monitoring problem

Input: a system $\mathcal{S}$, a monitor $\mathcal{C}$ and a run r of $\mathcal{S}$
Output: r is a correct run on $\mathcal{S}$



Credit image: www.Dory.fr and www.istockphoto.com

Monitoring and probabilities

Monitoring introduces overhead
Problem: Minimize overhead

Monitoring with probability

- Runtime Verification with State Estimation problem [1](RVSE): model with probabilistic system

[1] Runtime Verification with State Estimation, Stoller et al., 2012

Monitoring and probabilities

Monitoring introduces overhead
Minimize overhead : use probability
Problem: Uncertainty vs Overhead

Monitoring with probability

- Runtime Verification with State Estimation problem (RVSE): model with probabilistic system
- Runtime Verification with Particle Filtering problem [2](RVPF): trade-off between uncertainty and overhead

[2] Runtime Verification with Particle Filtering, Kalajdzic et al., 2013

Monitoring and probabilities

Monitoring introduces overhead
Minimize overhead : use probability
Uncertainty vs Overhead : trade-off

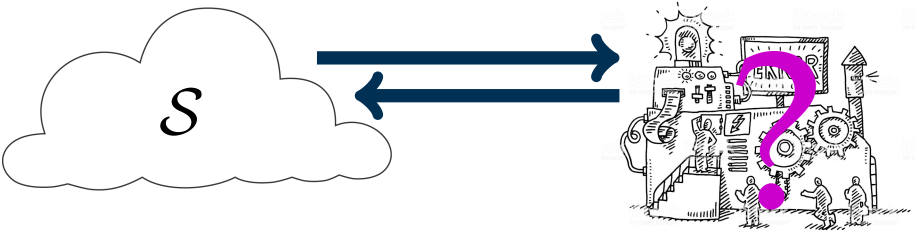
Our approach

The smallest overhead achievable without compromising precision at all.

Diagnosability

Diagnosability problem

Input: a system $\mathcal{S}$
Output: Exists a monitor for $\mathcal{S}$



Credit image: www.Dory.fr and www.istockphoto.com

Diagnosability

Diagnosability problem

Input: a system $\mathcal{S}$
Output: Exists a monitor for $\mathcal{S}$

Discrete event system [3]: P to check and EXPTIME to build the monitor [4]

[3] Diagnosability of discrete event systems, Sampath et al., 1995
[4] A polynomial algorithm for testing diagnosability of discrete-event systems, Jiang et al., 2001

Diagnosability

Diagnosability problem

Input: a system $\mathcal{S}$
Output: Exists a monitor for $\mathcal{S}$

Discrete event system : P to check and EXPTIME to build the monitor
Probabilistic system [3]: PSPACE problem [4][5]

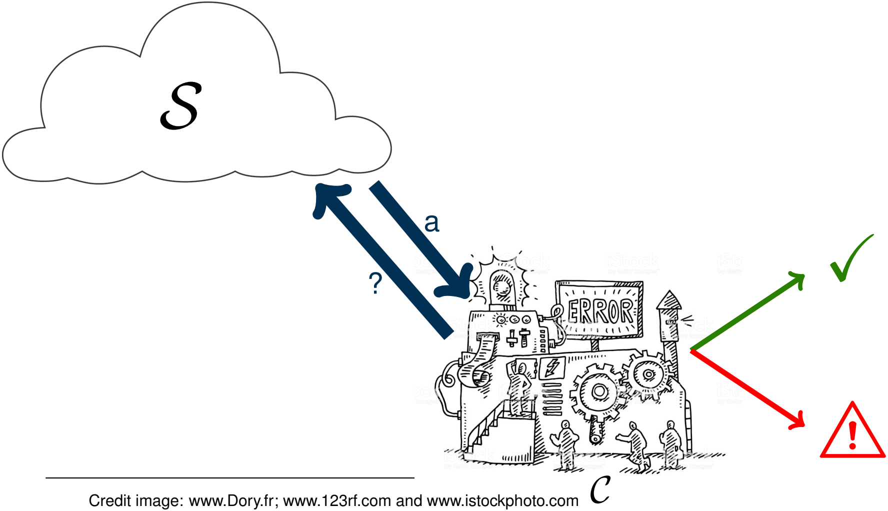
[3] Diagnosability of stochastic discrete-event systems, Thorsley et al., 2005

[4] Foundation of Diagnosis and Predictability in Probabilistic Systems, Bertrand et al., 2014

[5] Accurate Approximate Diagnosability of Stochastic Systems, Bertrand et al., 2016

Selective monitoring

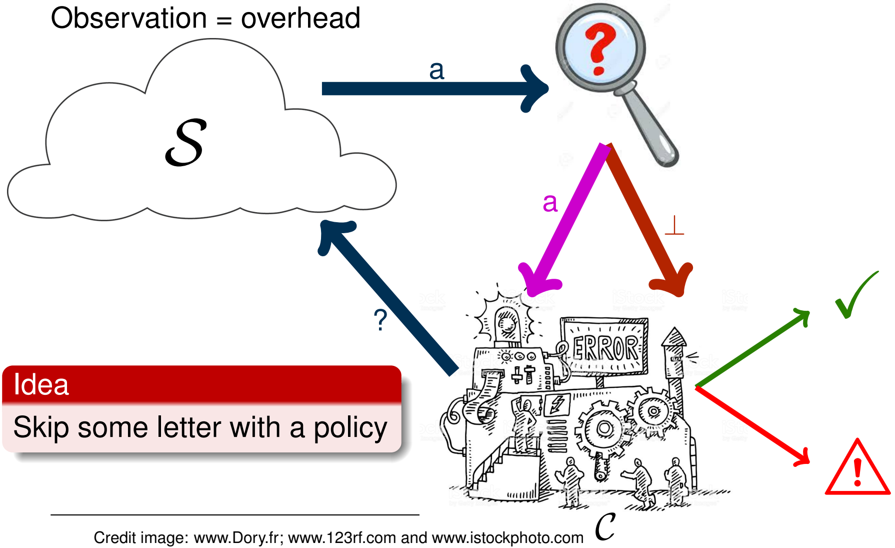
Observation = overhead



Credit image: www.Dory.fr; www.123rf.com and www.istockphoto.com

Selective monitoring

Observation = overhead



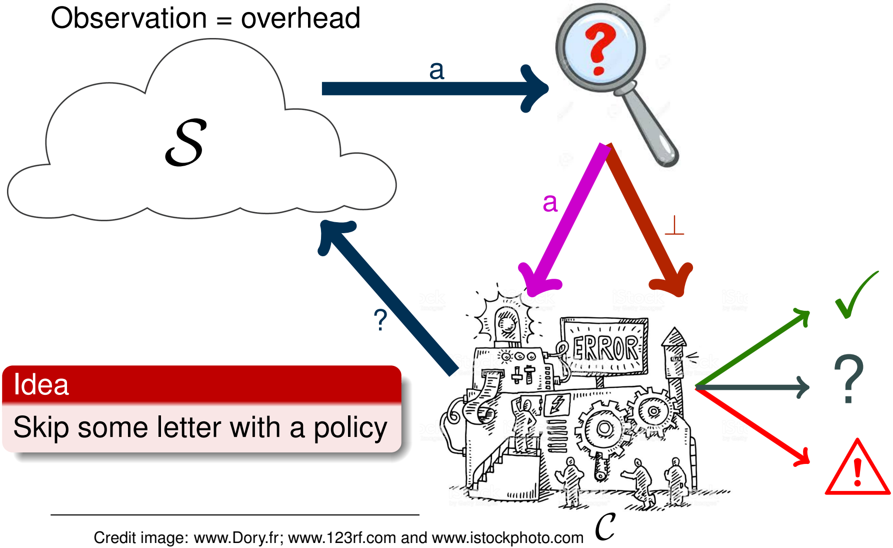
Idea

Skip some letter with a policy

Credit image: www.Dory.fr; www.123rf.com and www.istockphoto.com

Selective monitoring

Observation = overhead



Idea

Skip some letter with a policy

Credit image: www.Dory.fr; www.123rf.com and www.istockphoto.com

Selective monitoring without delay

Our goal

Design a feasible optimal policy without delay

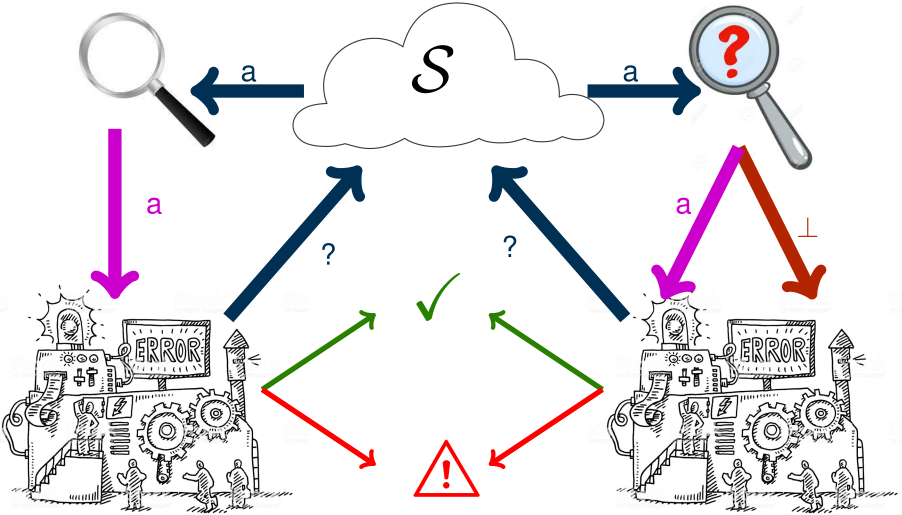
Selective monitoring without delay

Our goal

Design a feasible optimal policy without delay

Decide when it is possible

Selective monitoring without delay



Credit image: www.Dory.fr; www.123rf.com; www.pinterest.fr and www.istockphoto.com

Selective monitoring without delay

Our goal

Design a feasible optimal policy without delay

Decide when it is possible
Minimize the number of observations

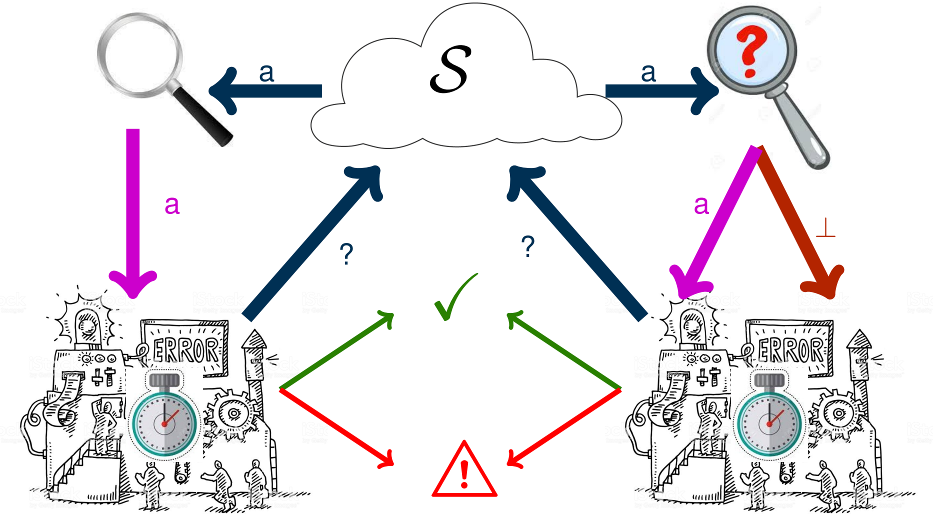
Selective monitoring without delay

Our goal

Design a feasible optimal policy without delay

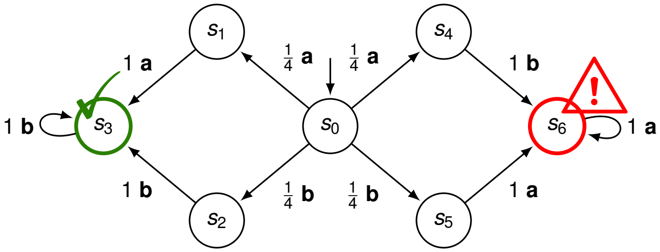
Decide when it is possible
Minimize the number of observations
Decide as soon as possible

Selective monitoring without delay

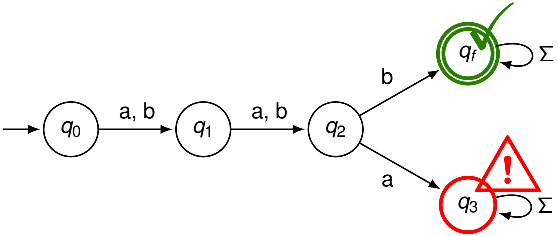


Credit image: www.Dory.fr; www.123rf.com; www.pinterest.fr and www.istockphoto.com

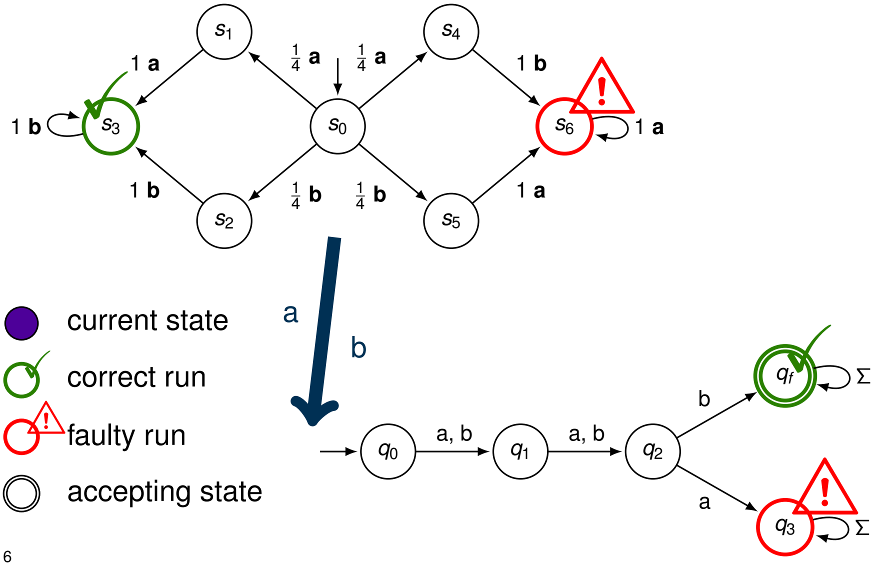
Markov Chain and Deterministic Finite Automaton



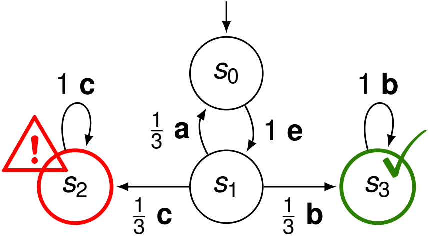
- current state
- correct run
- faulty run
- accepting state



Markov Chain and Deterministic Finite Automaton

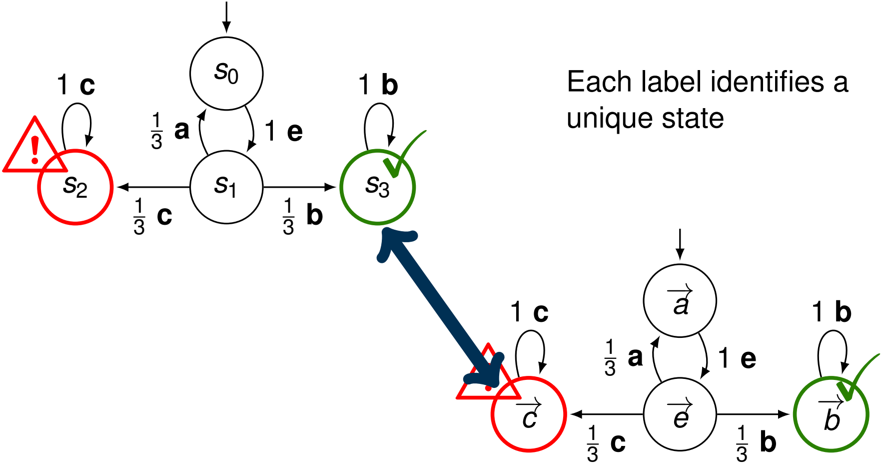


Non-Hidden Markov Chain



Each label identifies a unique state

Non-Hidden Markov Chain



Each label identifies a unique state

Observation policy

Observation policy: choice between observing and skipping

Observation policy

$aaaaaaa \Longrightarrow a\bot\bot aaa\bot$

Assumption: the policy decides with a finite prefix

A feasible optimal policy

Input: a Markov Chain $\mathcal{M}$, a Deterministic Finite Automaton $\mathcal{A}$ and a threshold r
Output: feasible policy whose expected number of observations is smaller than r

A feasible optimal policy

Input: a Markov Chain $\mathcal{M}$, a Deterministic Finite Automaton $\mathcal{A}$ and a threshold r
Output: feasible policy whose expected number of observations is smaller than r

Result in the general case [6]

Problem is undecidable.

[6] Selective Monitoring, Grigore et al., 2018

A feasible optimal policy

Input: a Markov Chain $\mathcal{M}$, a Deterministic Finite Automaton $\mathcal{A}$ and a threshold r
Output: feasible policy whose expected number of observations is smaller than r

Result in the general case

Problem is undecidable.

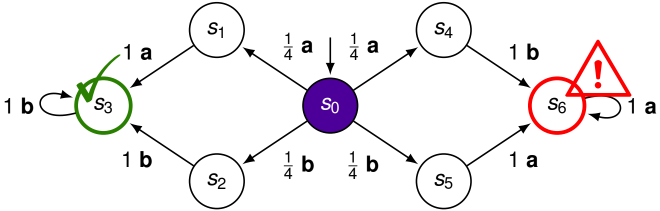
Result in the non-Hidden Markov Chain case [6]

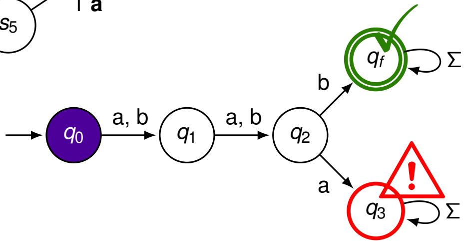
Problem is decidable in P
There exists a feasible optimal policy can be compute in P.

[6] Selective Monitoring, Grigore et al., 2018

A feasible optimal policy with delay

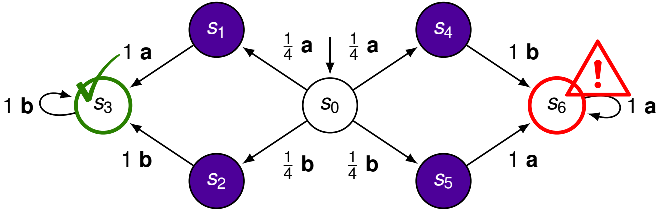
The feasible policy decides with delay.



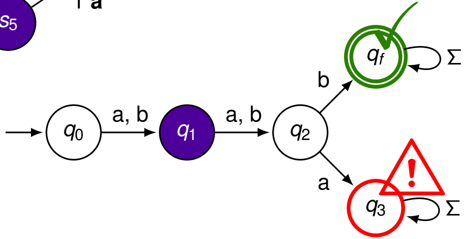


A feasible optimal policy with delay

The feasible policy decides with delay.

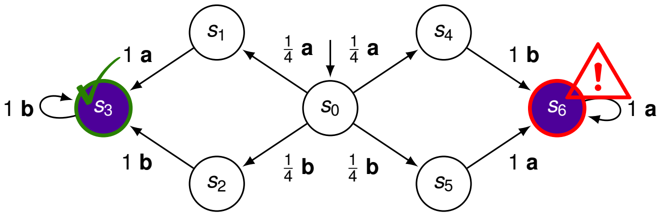


Policy: $\perp$

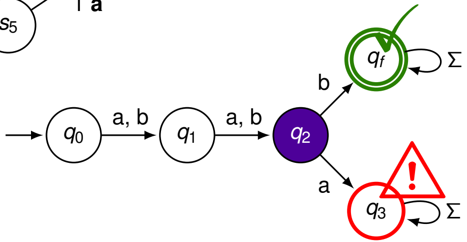


A feasible optimal policy with delay

The feasible policy decides with delay.

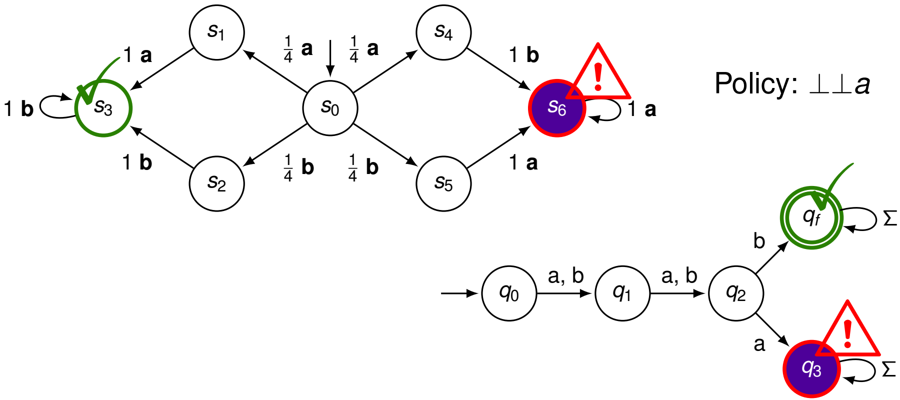


Policy: $\perp\perp$



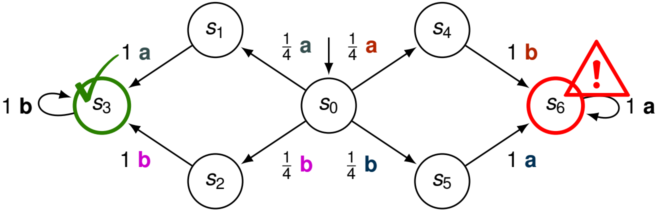
A feasible optimal policy with delay

The feasible policy decides with delay.



A feasible optimal policy with delay

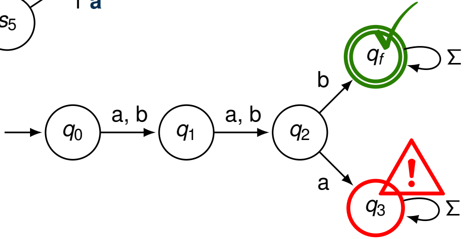
The feasible policy decides with delay.



Policy: $\bot\bot a$

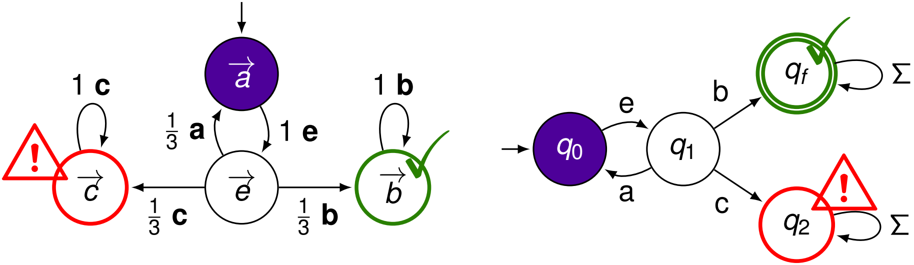
aa
bb
positively deciding

ab
ba
negatively deciding



A feasible policy without delay $\rho_{ZeroDelay}$

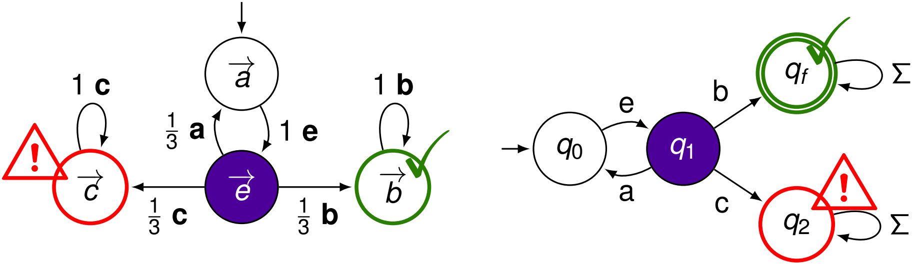
Apply the policy on $w = eb^{\omega}$



The policy skips the next letter

A feasible policy without delay $\rho_{ZeroDelay}$

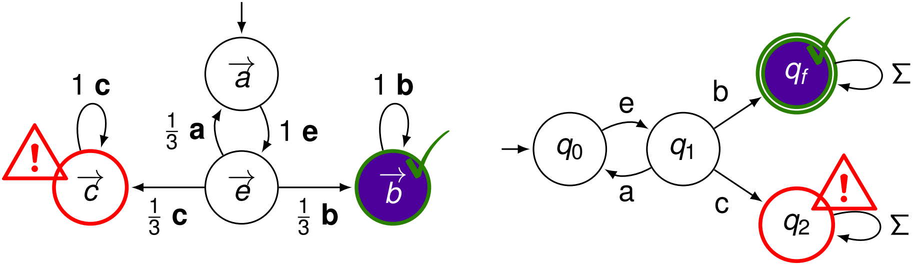
Apply the policy on $w = e\mathbf{b}b^{\omega}$



The policy observes the next letter

A feasible policy without delay $\rho_{ZeroDelay}$

Apply the policy on $w = eb^{\omega}$



The policy return *yes*

Results in general case

Feasible policy without delay

If v is a deciding prefix then the observation of v by $\rho_{ZeroDelay}$ is also deciding.

Non-optimal policy

The policy $\rho_{ZeroDelay}$ is not optimal.

Complexity

We can compute $\rho_{ZeroDelay}$ in PSPACE.

Results in non-Hidden Markov Chain case

Optimal policy

The policy $\rho_{ZeroDelay}$ is optimal.

Complexity

We can compute $\rho_{ZeroDelay}$ in P.

Conclusion: our policy

	General case	Non-Hidden Markov Chain case
Feasible	✓	✓
Without delay	✓	✓
Optimality	$\times$	✓
Complexity	PSPACE	P

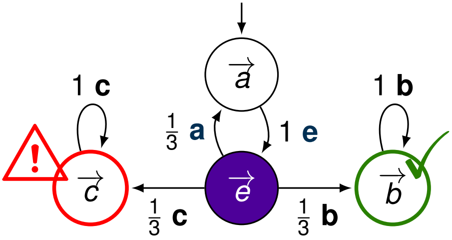
Futur works

- Complexity hardness result on the optimal policy choice
- Undecidability result
- Feasible optimal policy without delay in other classes of Markov Chain
- Policy implementation and study in practice

Prefix

Class of prefix

- *enabled*: emit by the MC

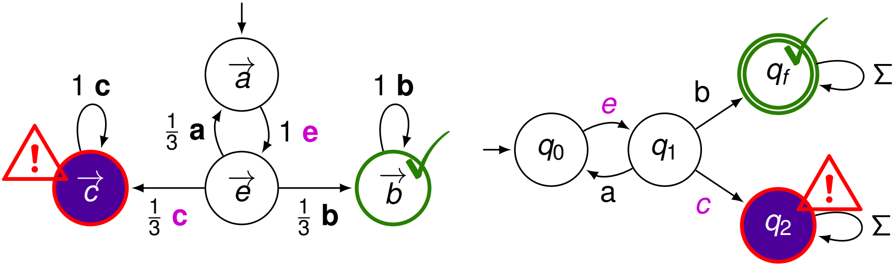


eae is enabled but eaa is not enabled

Prefix

Class of prefix

- *enabled*: emit by the MC
- *negatively deciding*: all infinite word describe a faulty run

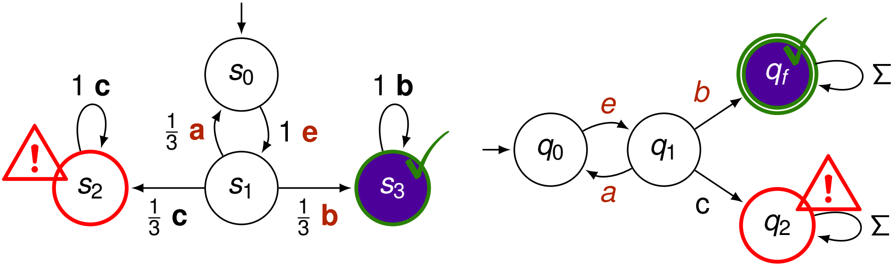


ec is negatively deciding

Prefix

Class of prefix

- *enabled*: emit by the MC
- *negatively deciding*: all infinite word describe a faulty run
- *positively deciding*: all infinite word describe a correct run



eaeb is positively deciding

Observation policy

Deciding policy

A policy *decides* w when the observation of w has a deciding prefix.

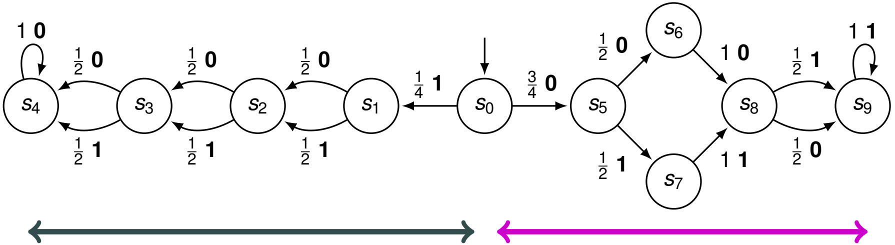
Feasible policy

A *feasible* policy decides w if and only if w has a deciding prefix.

Non-optimality

Non-optimal policy

The policy $\rho_{ZeroDelay}$ is not optimal.



ρ observes 4 letters
$\rho_{ZeroDelay}$ observes 3 letters

ρ observes 2 letters
$\rho_{ZeroDelay}$ observes 3 letters

$$3 = Ex(\rho_{ZeroDelay}) > Ex(\rho) = 2.5$$