Observation partielle des systèmes temporisés

Patricia Bouyer, Fabrice Chevalier
LSV – CNRS & ENS de Cachan

Moez Krichen, Stavros Tripakis
VERIMAG

CORTOS

Outline

Why partial observation?

Naturally appears in the modelling of applications:

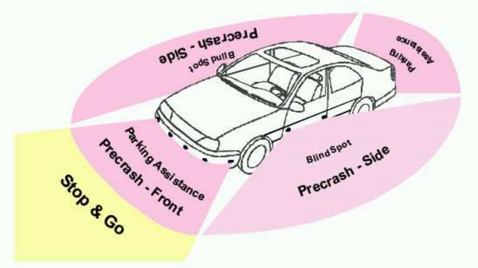
- modelling of an environment
- inherent non-determinism in applications
- partial knowledge of the system

Why partial observation?

Naturally appears in the modelling of applications:

- modelling of an environment
- inherent non-determinism in applications
- partial knowledge of the system

Example (The car periphery supervision)



- Embedded system
- Hostile environment
- Sensors
  - distances
  - speeds

Several application domains

- Control under partial observation
- Fault diagnosis
- Conformance testing
- Runtime model-checking
- Learning
- ...

Several application domains

- Control under partial observation
- **Fault diagnosis**
- Conformance testing
- Runtime model-checking
- Learning
- ...

The finite automata framework

Non-observable actions can be modelled as "ε-transitions".

The finite automata framework

Non-observable actions can be modelled as "ε-transitions".

- They can be removed from finite automata.

The finite automata framework

Non-observable actions can be modelled as "ε-transitions".

- They can be removed from finite automata.
- Partial observation is often not more difficult than global observation

The finite automata framework

Non-observable actions can be modelled as "ε-transitions".

- They can be removed from finite automata.
- Partial observation is often not more difficult than global observation:
  - control under partial observation **[Kupferman, Vardi 1997]**

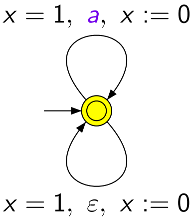
	LTL	CTL*	CTL
Partial obs.	2EXP-comp.	2EXP-comp.	EXP-comp.
Global obs.	2EXP-comp.	2EXP-comp.	EXP-comp.

  - two-player games with incomplete information **[Reif 1984]** **[Arnold, Vincent, Walukiewicz 2003]**

What's more difficult with time?

Stumbling blocks:

- ε-transitions can not be removed from timed automata

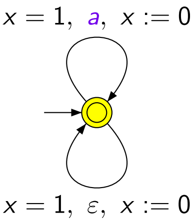


[Bérard, Diekert, Gastin, Petit 1998]

What's more difficult with time?

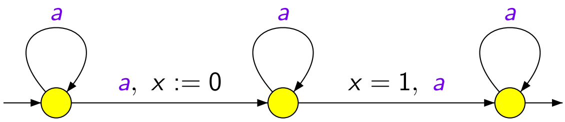
Stumbling blocks:

- ε-transitions can not be removed from timed automata



[Bérard, Diekert, Gastin, Petit 1998]

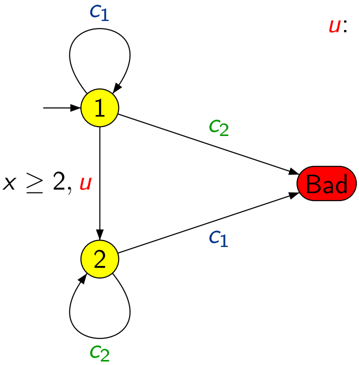
- timed automata can not be determinized nor complemented



[Alur, Dill 1990's]

Outline

A first example

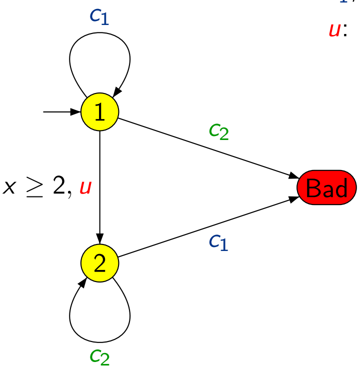


c_1, c_2: controllable actions

u: uncontrollable action

- This system is controllable under full observation...

A first example



c_1, c_2: controllable actions

u: uncontrollable and non-observable action

- This system is controllable under full observation...
- ... but not controllable under partial observation.

[Bouyer, D'Souza, Madhusudan, Petit 2003]

- On the "negative" side

Theorem

Safety and reachability timed control under partial observation is undecidable.

➜ by reduction of the universality problem for timed automata

[Bouyer, D'Souza, Madhusudan, Petit 2003]

- On the "negative" side

Theorem

Safety and reachability timed control under partial observation is undecidable.

➜ by reduction of the universality problem for timed automata

NB: *this result is robust for several modelizations...*

[Bouyer, D'Souza, Madhusudan, Petit 2003]

- On the "negative" side

Theorem

Safety and reachability timed control under partial observation is undecidable.

➜ by reduction of the universality problem for timed automata

NB: *this result is robust for several modelizations...*

- On the "positive" side

Theorem

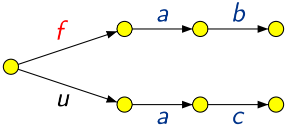
Fixing the **resources** of the controller, the control under partial observation problem becomes decidable (but 2EXPTIME-complete).

Outline

Principle of fault diagnosis [Sampath, Sengupta, Lafortune, Sinnamohideen, Teneketzis 1995]

Principle: "observe the behavior of a plant, and tell if something wrong has happened"

System:

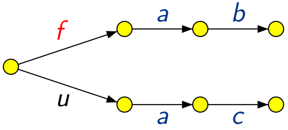


Principle of fault diagnosis

[Sampath, Sengupta, Lafortune, Sinnamohideen, Teneketzis 1995]

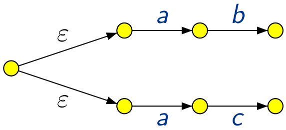
Principle: "observe the behavior of a plant, and tell if something wrong has happened"

System:



$\Sigma_o = \{a, b, c\}$ $\Sigma_u = \{f, u\}$

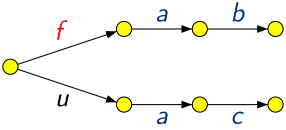
Sensors:



Principle of fault diagnosis [Sampath, Sengupta, Lafortune, Sinnamohideen, Teneketzis 1995]

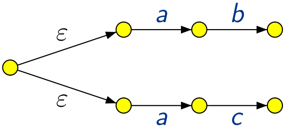
Principle: "observe the behavior of a plant, and tell if something wrong has happened"

System:



$\Sigma_o = \{a, b, c\}$ $\Sigma_u = \{f, u\}$

Sensors:

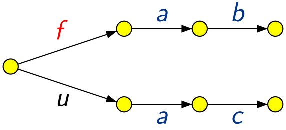


Observations: «ab» or «ac»

Principle of fault diagnosis [Sampath, Sengupta, Lafortune, Sinnamohideen, Teneketzis 1995]

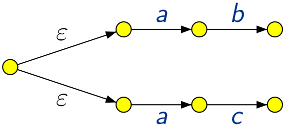
Principle: "observe the behavior of a plant, and tell if something wrong has happened"

System:



$\Sigma_o = \{a, b, c\}$ $\Sigma_u = \{f, u\}$

Sensors:



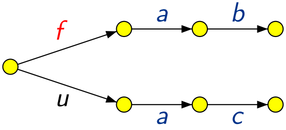
Observations: «ab» or «ac»

Did a fault occur?

Principle of fault diagnosis [Sampath, Sengupta, Lafortune, Sinnamohideen, Teneketzis 1995]

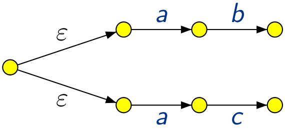
Principle: "observe the behavior of a plant, and tell if something wrong has happened"

System:



$\Sigma_o = \{a, b, c\}$ $\Sigma_u = \{f, u\}$

Sensors:



Observations: «ab» or «ac»

Did a fault occur?

The timed framework

- Plant = timed automaton
- Σ_o observable events, and Σ_u unobservable events

The timed framework

- Plant = timed automaton
- Σ_o observable events, and Σ_u unobservable events

Pb: Given an observation (timed word over Σ_o), did a fault occur?

The timed framework

- Plant = timed automaton
- Σ_o observable events, and Σ_u unobservable events

Pb: Given an observation (timed word over Σ_o), did a fault occur?
Aim: Answer within Δ units of time.

The timed framework

- Plant = timed automaton
- Σ_o observable events, and Σ_u unobservable events

Pb: Given an observation (timed word over Σ_o), did a fault occur?
Aim: Answer within Δ units of time.

Example ($\Sigma_o = \{a, b\}$ $\Sigma_u = \{f\}$)

- Execution of the plant: $w = (a, 1)(f, 3.1)(b, 4.5)$
- Observation: $\pi(w) = (a, 1)(b, 4.5)$

The timed framework

- Plant = timed automaton
- Σ_o observable events, and Σ_u unobservable events

Pb: Given an observation (timed word over Σ_o), did a fault occur?
Aim: Answer within Δ units of time.

Example ($\Sigma_o = \{a, b\}$ $\Sigma_u = \{f\}$)

- Execution of the plant: $w = (a, 1)(f, 3.1)(b, 4.5)$
- Observation: $\pi(w) = (a, 1)(b, 4.5)$

1-diagnoser:	has to announce fault on $\pi(w)$
2-diagnoser:	can announce fault on $\pi(w)$
	may announce nothing on $\pi(w)$

Δ-diagnosis

A Δ-diagnoser for $\mathcal{P}$ is a function $D : TW(\Sigma_o) \rightarrow \{0, 1\}$ such that:

Δ-diagnosis

A Δ-diagnoser for $\mathcal{P}$ is a function $D : TW(\Sigma_o) \rightarrow \{0, 1\}$ such that:

- for every non-faulty execution ρ of $\mathcal{P}$, $D(\pi_{\Sigma_o}(\rho)) = 0$

Δ-diagnosis

A Δ-diagnoser for $\mathcal{P}$ is a function $D : TW(\Sigma_o) \rightarrow \{0,1\}$ such that:

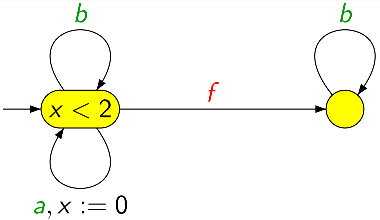
- for every non-faulty execution ρ of $\mathcal{P}$, $D(\pi_{\Sigma_o}(\rho)) = 0$
- for every Δ-faulty execution ρ of $\mathcal{P}$, $D(\pi_{\Sigma_o}(\rho)) = 1$

Δ-diagnosis

A Δ-diagnoser for $\mathcal{P}$ is a function $D : TW(\Sigma_o) \rightarrow \{0, 1\}$ such that:

- for every non-faulty execution ρ of $\mathcal{P}$, $D(\pi_{\Sigma_o}(\rho)) = 0$
- for every Δ-faulty execution ρ of $\mathcal{P}$, $D(\pi_{\Sigma_o}(\rho)) = 1$

Example



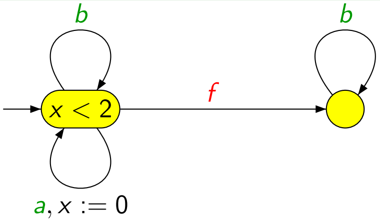
This system is 2-diagnosable...

Δ-diagnosis

A Δ-diagnoser for $\mathcal{P}$ is a function $D : TW(\Sigma_o) \rightarrow \{0, 1\}$ such that:

- for every non-faulty execution ρ of $\mathcal{P}$, $D(\pi_{\Sigma_o}(\rho)) = 0$
- for every Δ-faulty execution ρ of $\mathcal{P}$, $D(\pi_{\Sigma_o}(\rho)) = 1$

Example



This system is 2-diagnosable...

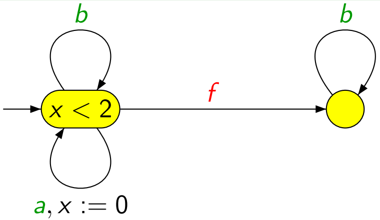
... but not 1-diagnosable

Δ-diagnosis

A Δ-diagnoser for $\mathcal{P}$ is a function $D : TW(\Sigma_o) \rightarrow \{0, 1\}$ such that:

- for every non-faulty execution ρ of $\mathcal{P}$, $D(\pi_{\Sigma_o}(\rho)) = 0$
- for every Δ-faulty execution ρ of $\mathcal{P}$, $D(\pi_{\Sigma_o}(\rho)) = 1$

Example

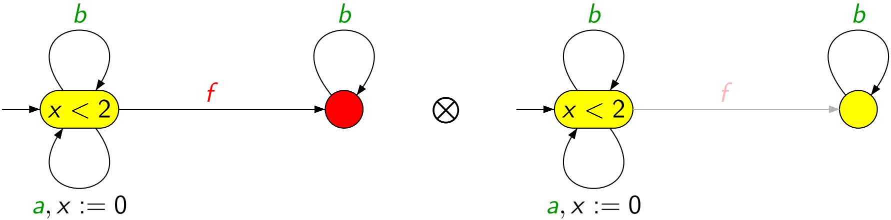


This system is 2-diagnosable...

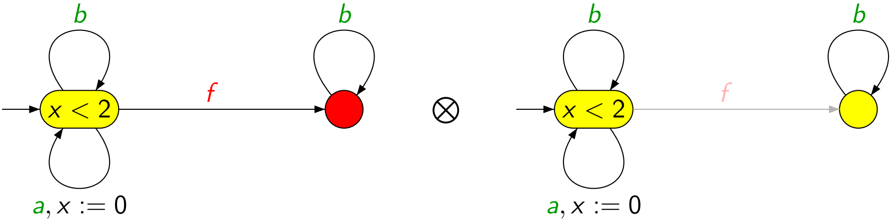
... but not 1-diagnosable

(because $(f, 0)(b, 1)$ and $(b, 1)$ raise the same observation)

Decidability of diagnosability [Tripakis 2002]

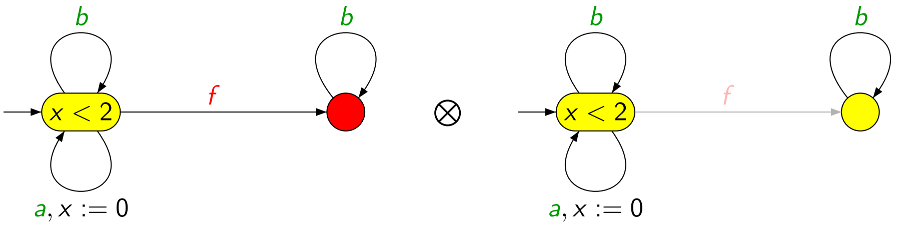


Decidability of diagnosability [Tripakis 2002]



The plant is diagnosable iff there is an infinite *non-zeno* run in the above product which goes through (●, ●)

Decidability of diagnosability [Tripakis 2002]



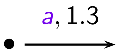
The plant is diagnosable iff there is an infinite *non-zeno* run in the above product which goes through (● , ●)

➜ The diagnosability problem is PSPACE-complete

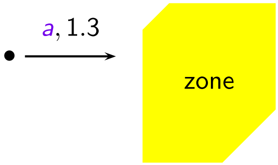
In practice: state estimation [Tripakis 2002]

•

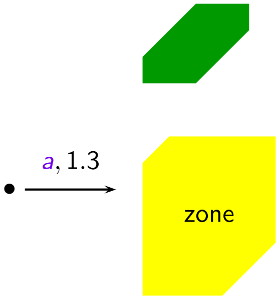
In practice: state estimation [Tripakis 2002]



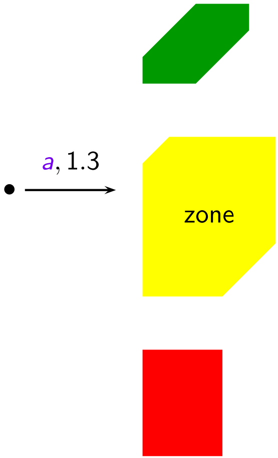
In practice: state estimation [Tripakis 2002]



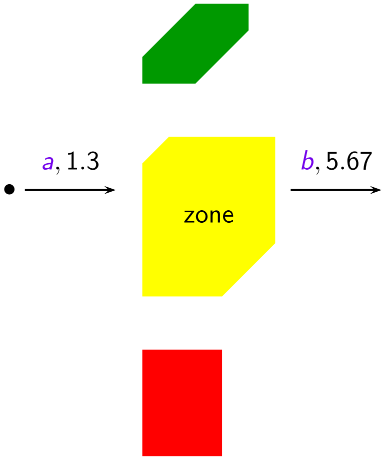
In practice: state estimation [Tripakis 2002]



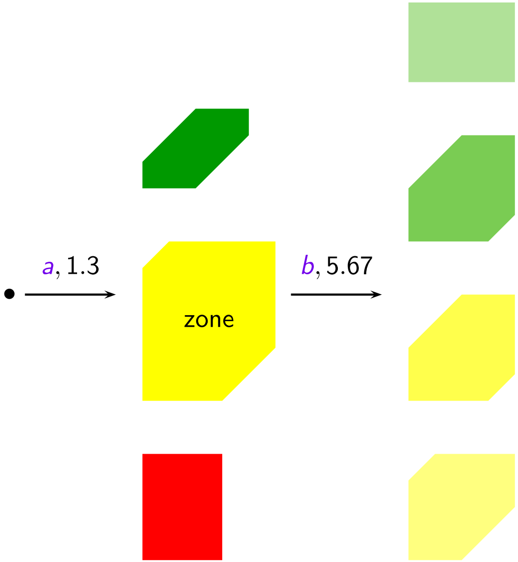
In practice: state estimation [Tripakis 2002]



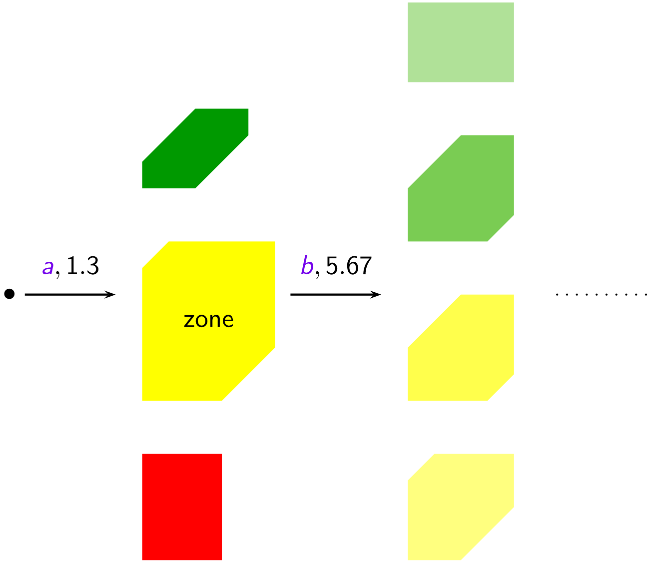
In practice: state estimation [Tripakis 2002]



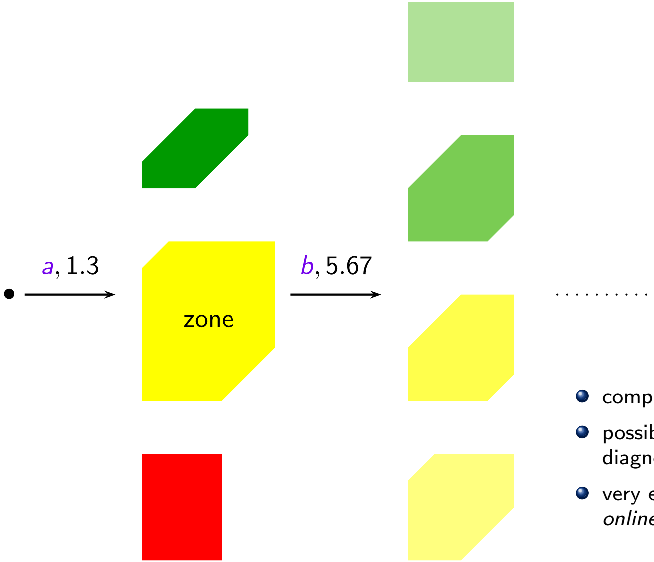
In practice: state estimation [Tripakis 2002]



In practice: state estimation [Tripakis 2002]



In practice: state estimation [Tripakis 2002]



- complete solution
- possible for *offline* diagnosis (log files)
- very expensive to run it *online*

Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

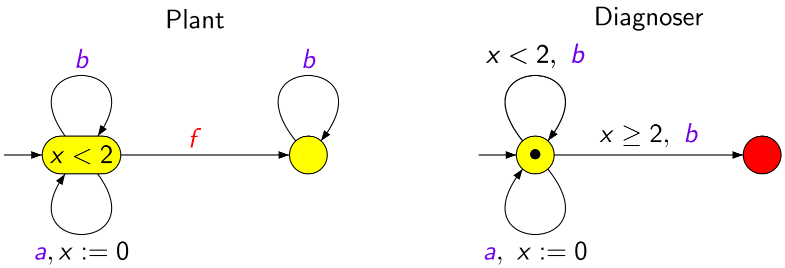
$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Example

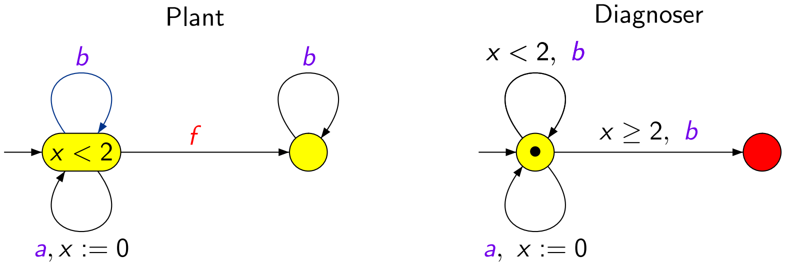


Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Example



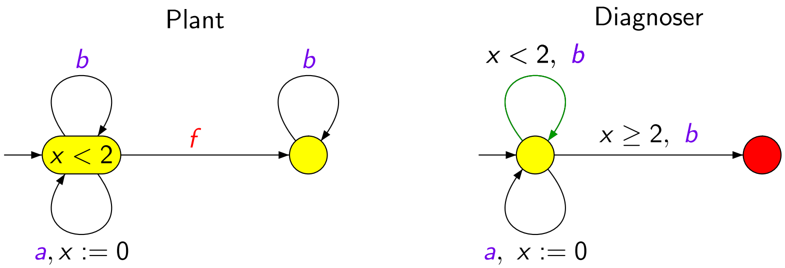
$w = (b, 0.5)$

Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Example



$w = (b, 0.5)$

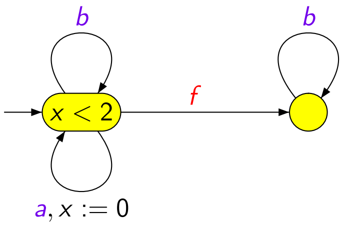
Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

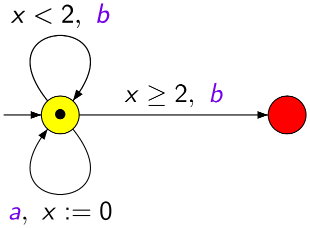
$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Example

Plant



Diagnoser



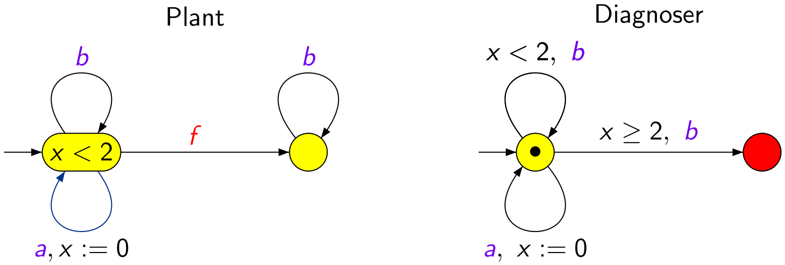
$w = (b, 0.5)$

Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Example



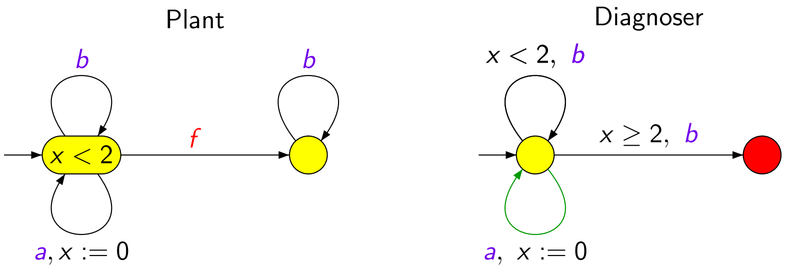
$$w = (b, 0.5)\ (a, 1)$$

Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Example



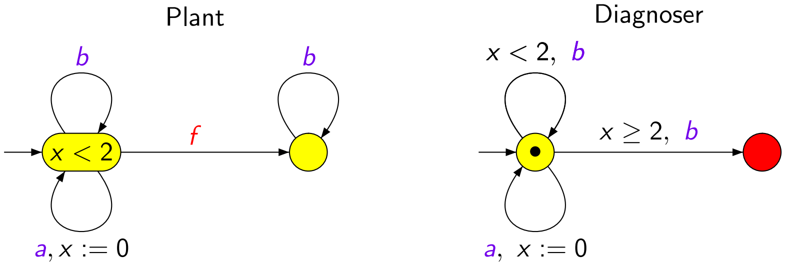
$w = (b, 0.5)\ (a, 1)$

Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Example



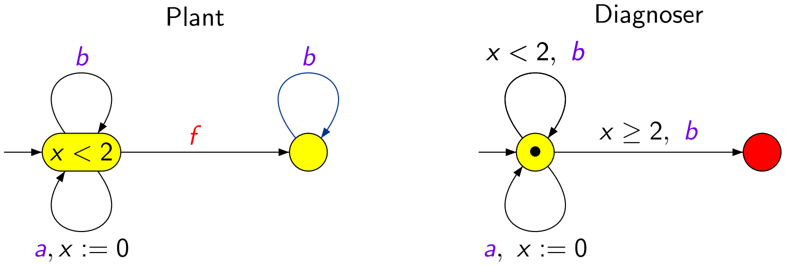
$$w = (b, 0.5)\ (a, 1)$$

Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Example



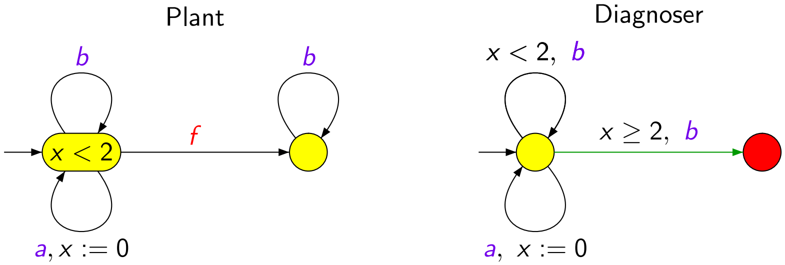
$$w = (b, 0.5)\ (a, 1)\ (b, 3)$$

Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

Example



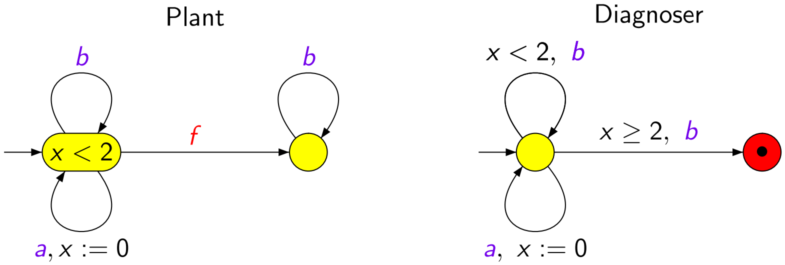
$$w = (b, 0.5)\ (a, 1)\ (b, 3)$$

Towards easier online diagnosis

➜ **Our aim:** build a deterministic diagnoser $\mathcal{O}$...

$$L_{\Delta f}(\mathcal{P}) \subseteq L(\mathcal{O}) \subseteq L_{\neg f}(\mathcal{P})^c$$

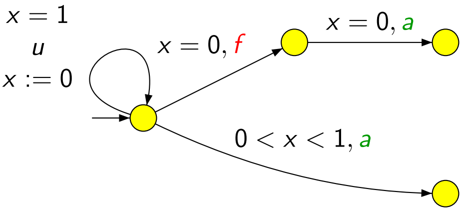
Example



$$w = (b, 0.5)\ (a, 1)\ (b, 3)$$

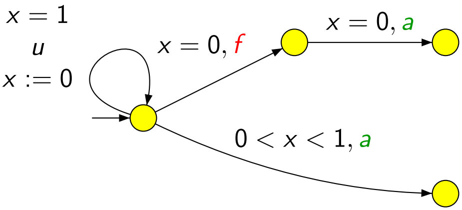
Diagnosis by deterministic timed automata

- Less general than previous diagnosis



Diagnosis by deterministic timed automata

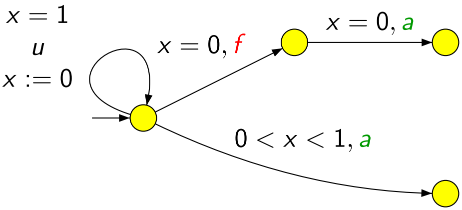
- Less general than previous diagnosis



- The diagnosis problem with DTA is not solved yet.

Diagnosis by deterministic timed automata

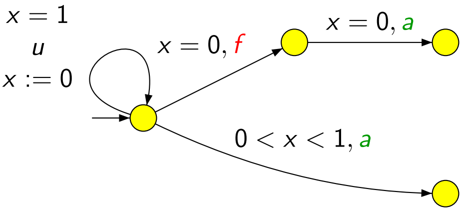
- Less general than previous diagnosis



- The diagnosis problem with DTA is not solved yet.
- The "precise" diagnosis problem and the "*asap*" diagnosis problem with DTA are undecidable. **[Chevalier 2004]**

Diagnosis by deterministic timed automata

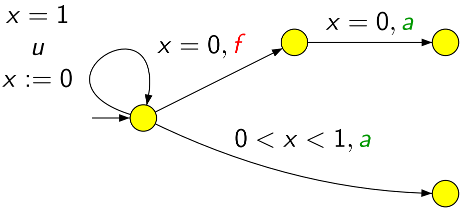
- Less general than previous diagnosis



- The diagnosis problem with DTA is not solved yet.
- The "precise" diagnosis problem and the "*asap*" diagnosis problem with DTA are undecidable. **[Chevalier 2004]**
- We restrict to bounded resources $\mu = (X, m, \max)$

Diagnosis by deterministic timed automata

- Less general than previous diagnosis



- The diagnosis problem with DTA is not solved yet.
- The "precise" diagnosis problem and the "*asap*" diagnosis problem with DTA are undecidable. **[Chevalier 2004]**
- We restrict to bounded resources $\mu = (X, m, \max)$

Theorem **[Bouyer, Chevalier, D'Souza 2005]**

Δ-diagnosis of timed systems with DTA_μ is 2EXPTIME-complete.

Diagnosis as a game

We will transform the diagnosis problem into a two-player safety game:

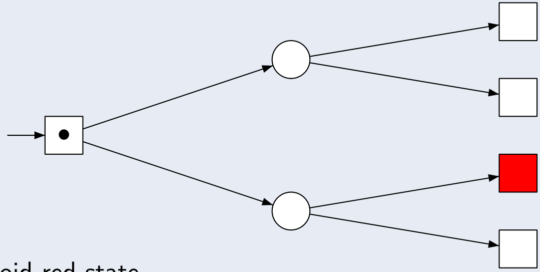
- one player is the diagnoser $\square$
- the other player is the environment $\bigcirc$

Diagnosis as a game

We will transform the diagnosis problem into a two-player safety game:

- one player is the diagnoser □
- the other player is the environment ○

Reminder (two-player safety game)



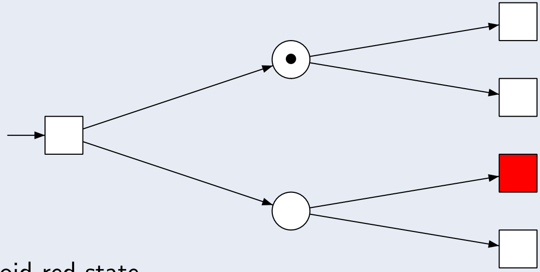
□ : avoid red state

Diagnosis as a game

We will transform the diagnosis problem into a two-player safety game:

- one player is the diagnoser □
- the other player is the environment ○

Reminder (two-player safety game)



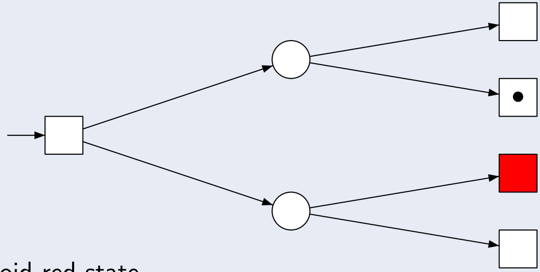
□ : avoid red state

Diagnosis as a game

We will transform the diagnosis problem into a two-player safety game:

- one player is the diagnoser □
- the other player is the environment ○

Reminder (two-player safety game)



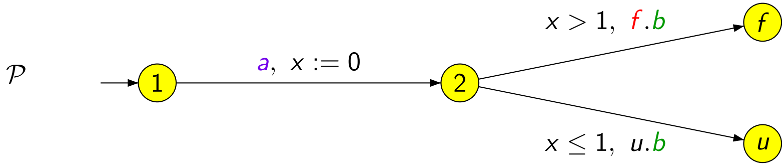
□ : avoid red state

Diagnosis as a game

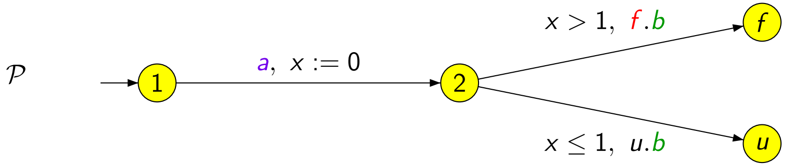
We will transform the diagnosis problem into a two-player safety game:

- one player is the diagnoser □
- the other player is the environment ○

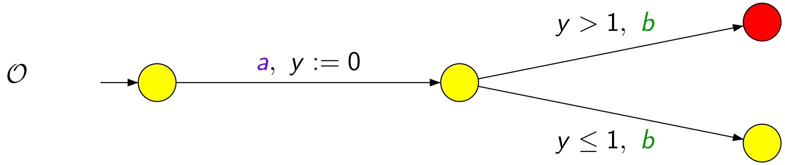
The plant is Δ-DTA$_\mu$-diagnosable iff □ has a winning strategy.

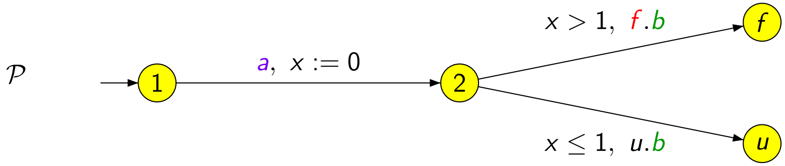


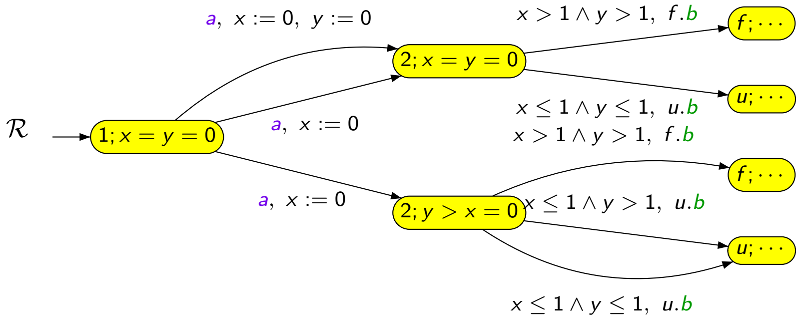
Is there a diagnoser for the plant with one clock and constants 0 and 1?

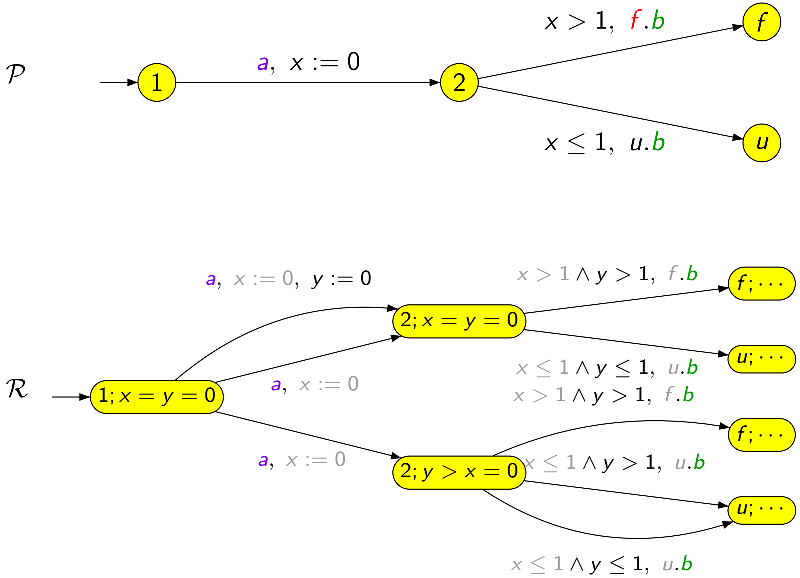


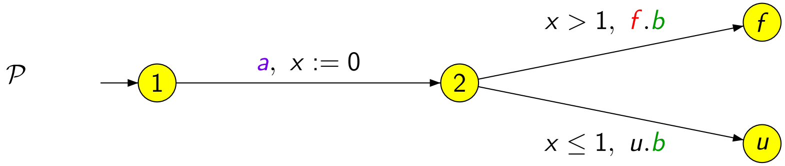
Is there a diagnoser for the plant with one clock and constants 0 and 1?

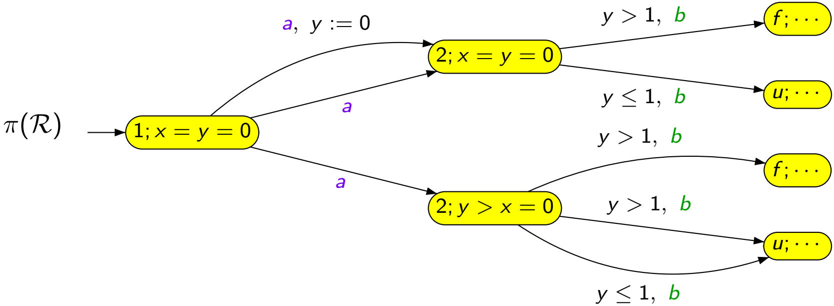


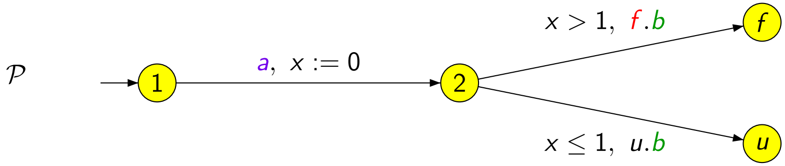

$\mathcal{P}$
$a,\ x := 0$
$x > 1,\ f.b$
$x \leq 1,\ u.b$
1
2
f
u

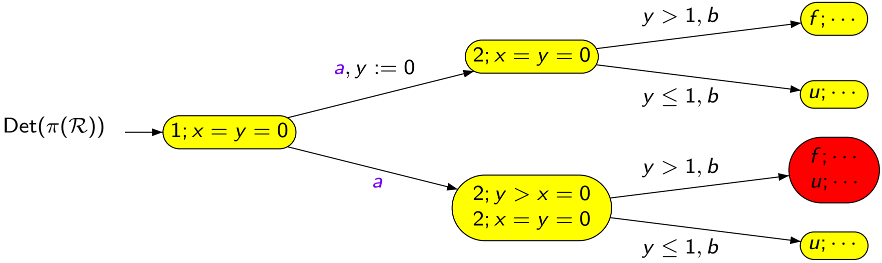

$\mathcal{R}$
$a,\ x := 0,\ y := 0$
$a,\ x := 0$
$a,\ x := 0$
$1; x = y = 0$
$2; x = y = 0$
$2; y > x = 0$
$x > 1 \land y > 1,\ f.b$
$x \leq 1 \land y \leq 1,\ u.b$
$x > 1 \land y > 1,\ f.b$
$x \leq 1 \land y > 1,\ u.b$
$x \leq 1 \land y \leq 1,\ u.b$
$f; \cdots$
$u; \cdots$
$f; \cdots$
$u; \cdots$

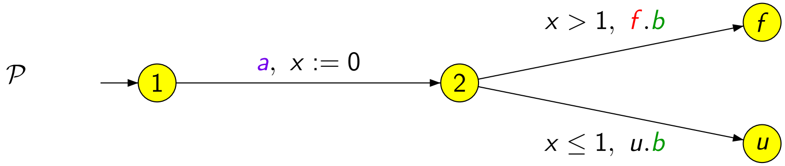

$\mathcal{P}$
1
$a, \; x := 0$
2
$x > 1, \; f.b$
f
$x \leq 1, \; u.b$
u
$\mathcal{R}$
$1; x = y = 0$
$a, \; x := 0, \; y := 0$
$a, \; x := 0$
$2; x = y = 0$
$2; y > x = 0$
$x > 1 \wedge y > 1, \; f.b$
$f; \cdots$
$x \leq 1 \wedge y \leq 1, \; u.b$
$u; \cdots$
$x > 1 \wedge y > 1, \; f.b$
$f; \cdots$
$x \leq 1 \wedge y > 1, \; u.b$
$u; \cdots$
$x \leq 1 \wedge y \leq 1, \; u.b$


$\mathcal{P}$
1
$a,\ x := 0$
2
$x > 1,\ f.b$
$x \leq 1,\ u.b$
f
u

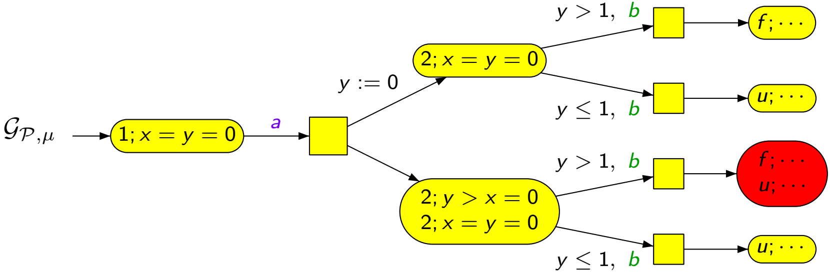

$\pi(\mathcal{R})$
$1; x = y = 0$
$a,\ y := 0$
a
a
$2; x = y = 0$
$2; y > x = 0$
$y > 1,\ b$
$y \leq 1,\ b$
$y > 1,\ b$
$y > 1,\ b$
$y \leq 1,\ b$
$f; \cdots$
$u; \cdots$
$f; \cdots$
$u; \cdots$


P
a, x := 0
1
2
x > 1, f.b
f
x ≤ 1, u.b
u


Det(π(R))
1; x = y = 0
a, y := 0
2; x = y = 0
y > 1, b
f; ···
y ≤ 1, b
u; ···
a
2; y > x = 0
2; x = y = 0
y > 1, b
f; ···
u; ···
y ≤ 1, b
u; ···



$\mathcal{P}$
1
$a, \; x := 0$
2
$x > 1, \; f.b$
f
$x \leq 1, \; u.b$
u



$\mathcal{G}_{\mathcal{P},\mu}$
$1; x = y = 0$
a
$y := 0$
$2; x = y = 0$
$y > 1, \; b$
$f; \cdots$
$y \leq 1, \; b$
$u; \cdots$
$2; y > x = 0$
$2; x = y = 0$
$y > 1, \; b$
$f; \cdots$
$u; \cdots$
$y \leq 1, \; b$
$u; \cdots$

Diagnosis by DTA_μ

Proposition

☐ has a winning strategy in $\mathcal{G}_{\mathcal{P},\mu}$ iff there is a diagnoser for $\mathcal{P}$ in DTA_μ.

➜ Δ-DTA_μ-diagnosability is in 2EXPTIME.

Diagnosis by DTA_μ

Proposition

$\square$ has a winning strategy in $\mathcal{G}_{\mathcal{P},\mu}$ iff there is a diagnoser for $\mathcal{P}$ in DTA_μ.

➜ Δ-DTA_μ-diagnosability is in 2EXPTIME.

Moreover, we can simulate an Alternating Turing Machine using exponential space with a diagnosis problem...

➜ Δ-DTA_μ-diagnosability is 2EXPTIME-hard.

Diagnosis by event-recording timed automata

[Alur, Fix, Henzinger 1994]

- one clock x_a per event a
- clock x_a is reset when a occurs

Diagnosis by event-recording timed automata

[Alur, Fix, Henzinger 1994]

- one clock x_a per event a
- clock x_a is reset when a occurs

Property

- Event-recording timed automata are determinizable **[Alur, Fix, Henzinger 1994]**
- Event-recording timed automata are *input-determined* **[D'Souza, Tabareau 2004]**

Diagnosis by event-recording timed automata

[Alur, Fix, Henzinger 1994]

- one clock x_a per event a
- clock x_a is reset when a occurs

Property

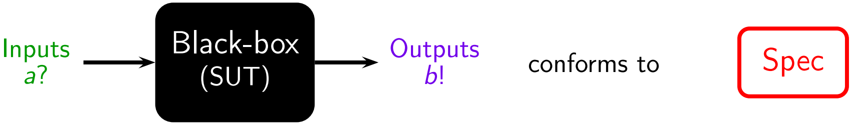
- Event-recording timed automata are determinizable
  [Alur, Fix, Henzinger 1994]
- Event-recording timed automata are *input-determined*
  [D'Souza, Tabareau 2004]

➜ Diagnosis (with bounded resources) becomes PSPACE-complete
[Bouyer, Chevalier, D'Souza 2005]

Outline

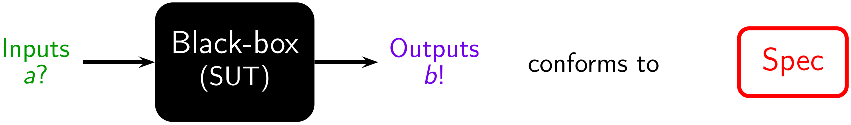
Black-box conformance testing

[Krichen, Tripakis 2004,2005]



Black-box conformance testing

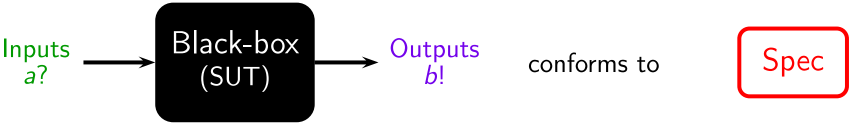
[Krichen, Tripakis 2004,2005]



- The specification is given as an I/O-timed automaton with ε-transitions.

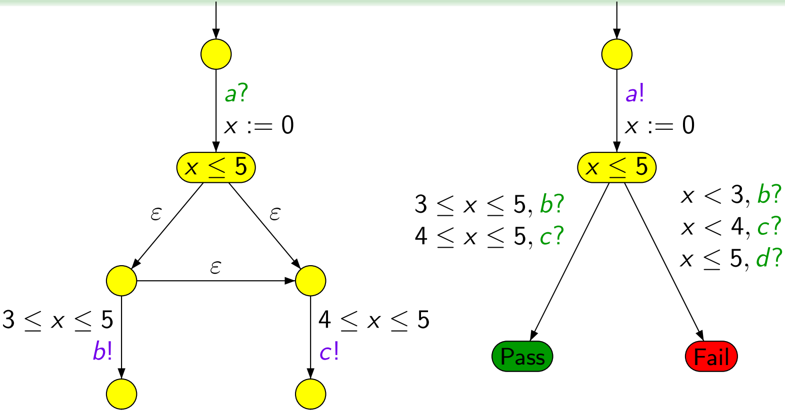
Black-box conformance testing

[Krichen, Tripakis 2004,2005]



- The specification is given as an I/O-timed automaton with ε-transitions.
- A test is a strategy of interaction between the tester and the SUT. The tester tries to demonstrate that the SUT does not conform to the specification, while the SUT tries to prevent doing so.

Example



A specification | A possible test

Techniques used for building tests

Very similar to those for fault diagnosis:

- state estimation in the specification
- fixing the resources, synthesis of strategies in the game between the SUT and the tester

Outline

Conclusion & further developments

Conclusion

- Partial observation adds much complexity to many problems
- **Main reason:** there is no real nice theory of regular timed languages

Conclusion & further developments

Conclusion

- Partial observation adds much complexity to many problems
- **Main reason:** there is no real nice theory of regular timed languages

Further developments

- Algorithms for control under partial observation
- Further notions of partial observation: time is no more completely observable, but only through a *tick* action
- Fault diagnosis with DTA/ERA
- Get rid of some resources or the Δ parameter
- Control under partial observation for other classes of systems (*e.g.* o-minimal hybrid games)