Timed games

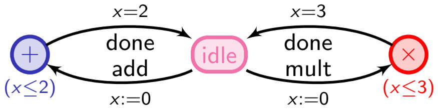
Patricia Bouyer-Decitre

LSV, CNRS & ENS Cachan, France

Why (timed) games?

- to model uncertainty

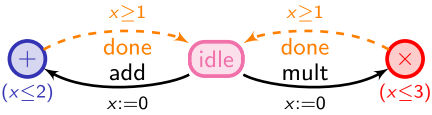
Example of a processor in the taskgraph example



Why (timed) games?

- to model uncertainty

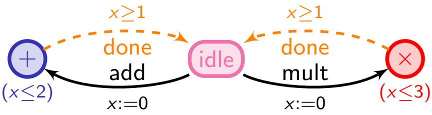
Example of a processor in the taskgraph example



Why (timed) games?

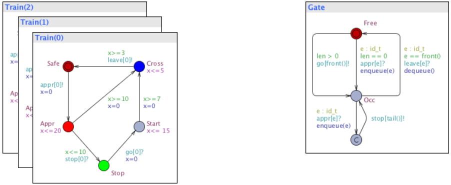
- to model uncertainty

Example of a processor in the taskgraph example



- to model an interaction with an environment

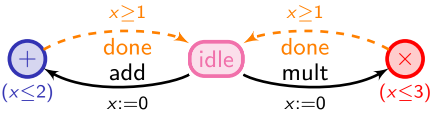
Example of the gate in the train/gate example



Why (timed) games?

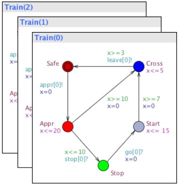
- to model uncertainty

Example of a processor in the taskgraph example



- to model an interaction with an environment

Example of the gate in the train/gate example

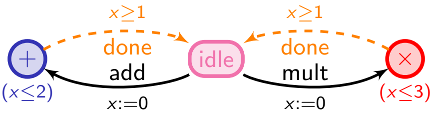


?

Why (timed) games?

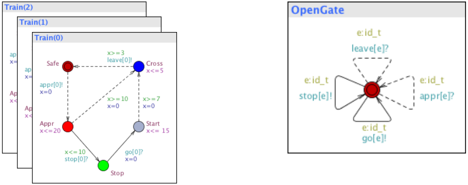
- to model uncertainty

Example of a processor in the taskgraph example

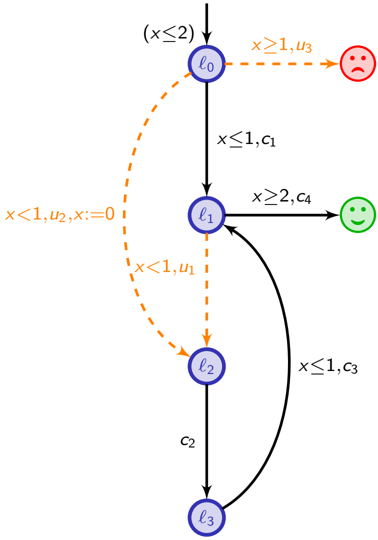


- to model an interaction with an environment

Example of the gate in the train/gate example



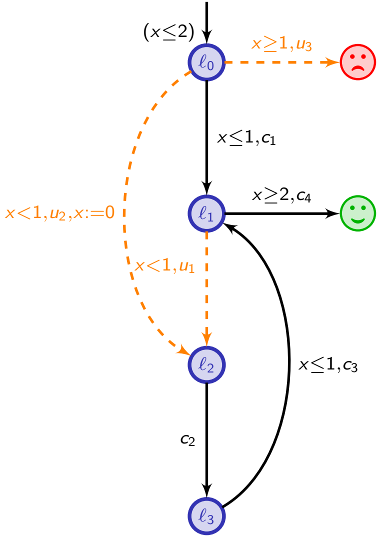
An example of a timed game



Rule of the game

- Aim: avoid ☹ and reach ☺

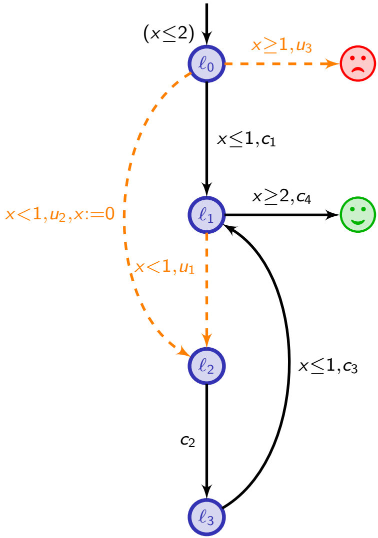
An example of a timed game



Rule of the game

- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

An example of a timed game

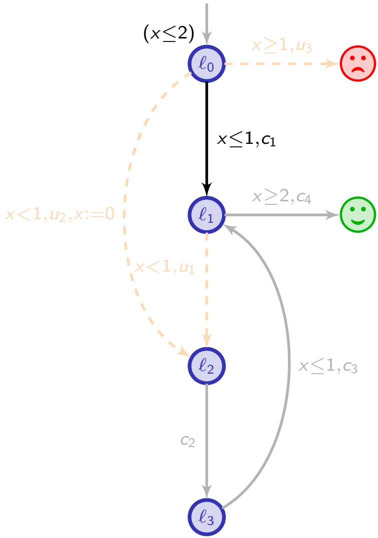


Rule of the game

- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

An example of a timed game



Rule of the game

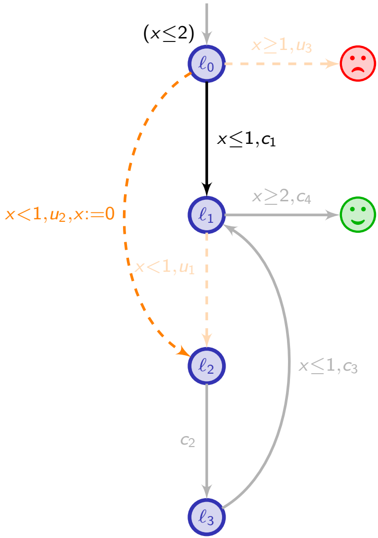
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$

An example of a timed game



Rule of the game

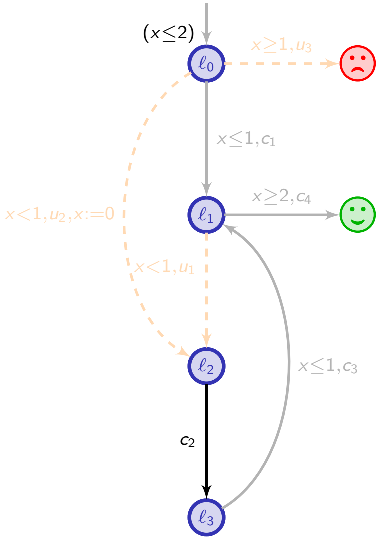
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$
  $\rightsquigarrow$ can be preempted by u_2

An example of a timed game



Rule of the game

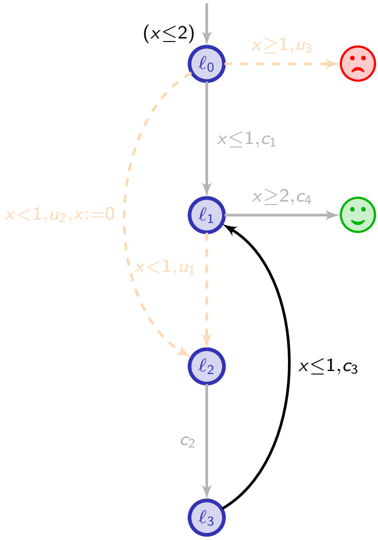
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$
  $\rightsquigarrow$ can be preempted by u_2
- from $(\ell_2, \star)$, play $(1 - \star, c_2)$

An example of a timed game



Rule of the game

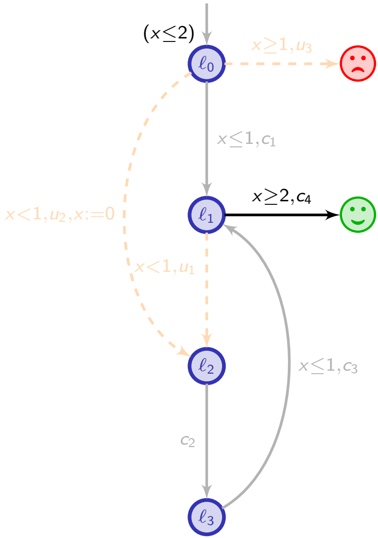
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$
  $\rightsquigarrow$ can be preempted by u_2
- from $(\ell_2, \star)$, play $(1 - \star, c_2)$
- from $(\ell_3, 1)$, play $(0, c_3)$

An example of a timed game



Rule of the game

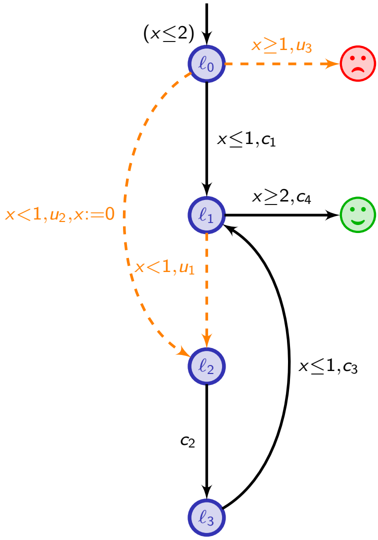
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$
  $\rightsquigarrow$ can be preempted by u_2
- from $(\ell_2, \star)$, play $(1 - \star, c_2)$
- from $(\ell_3, 1)$, play $(0, c_3)$
- from $(\ell_1, 1)$, play $(1, c_4)$

An example of a timed game



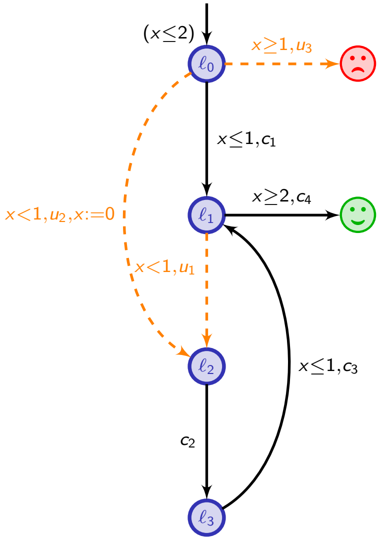
Rule of the game

- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

Problems to be considered

An example of a timed game



Rule of the game

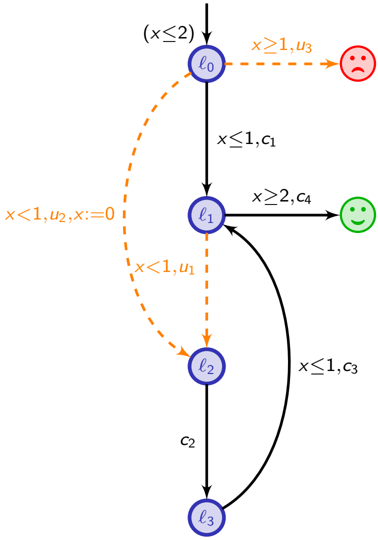
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

Problems to be considered

- Does there exist a winning strategy?

An example of a timed game



Rule of the game

- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

Problems to be considered

- Does there exist a winning strategy?
- If yes, compute one (as simple as possible).

Decidability of timed games

Theorem [AMPS98,HK99]

Reachability and safety timed games are decidable and EXPTIME-complete. Furthermore memoryless and "region-based" strategies are sufficient.

[AMPS98] Asarin, Maler, Pnueli, Sifakis. Controller synthesis for timed automata *(SSC'98)*.
[HK99] Henzinger, Kopke. Discrete-time control for rectangular hybrid automata *(Theoretical Computer Science)*.

Decidability of timed games

Theorem [AMPS98,HK99]

Reachability and safety timed games are decidable and EXPTIME-complete. Furthermore memoryless and "region-based" strategies are sufficient.

$\rightsquigarrow$ classical regions are sufficient for solving such problems

[AMPS98] Asarin, Maler, Pnueli, Sifakis. Controller synthesis for timed automata *(SSC'98)*.
[HK99] Henzinger, Kopke. Discrete-time control for rectangular hybrid automata *(Theoretical Computer Science)*.

Decidability of timed games

Theorem [AMPS98,HK99]

Reachability and safety timed games are decidable and EXPTIME-complete. Furthermore memoryless and "region-based" strategies are sufficient.

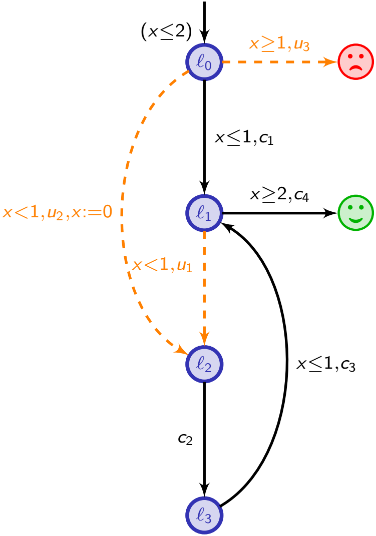
$\rightsquigarrow$ classical regions are sufficient for solving such problems

Theorem [AM99,BHPR07,JT07]

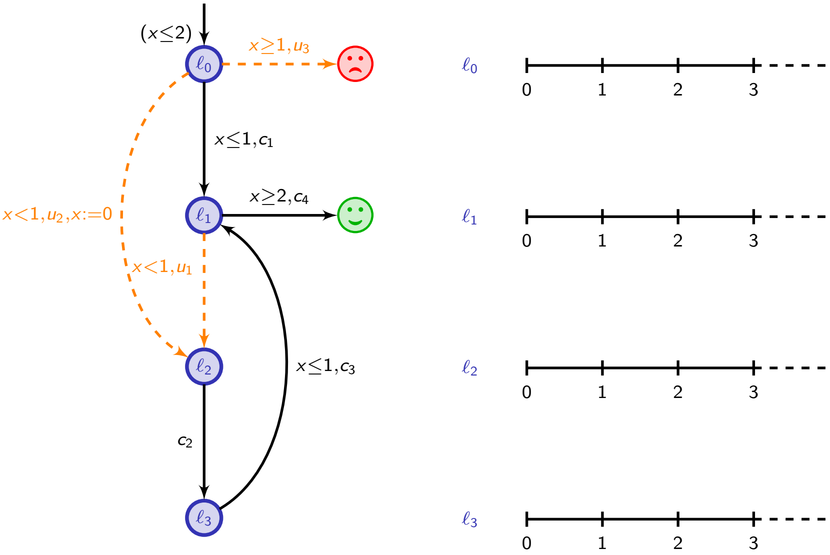
Optimal-time reachability timed games are decidable and EXPTIME-complete.

[AM99] Asarin, Maler. As soon as possible: time optimal control for timed automata *(HSCC'99)*.
[BHPR07] Brihaye, Henzinger, Prabhu, Raskin. Minimum-time reachability in timed games *(ICALP'07)*.
[JT07] Jurdziński, Trivedi. Reachability-time games on timed automata *(ICALP'07)*.

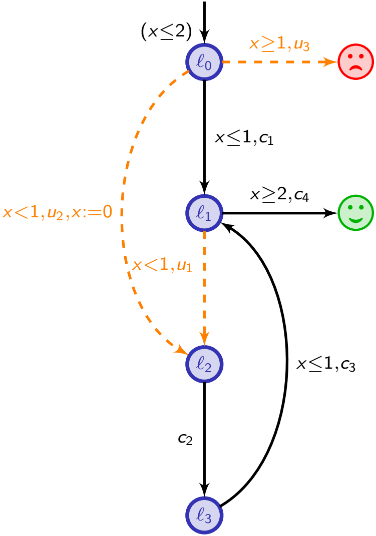
Back to the example: computing winning states

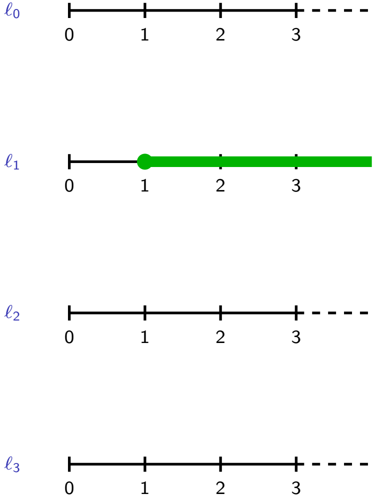


Back to the example: computing winning states

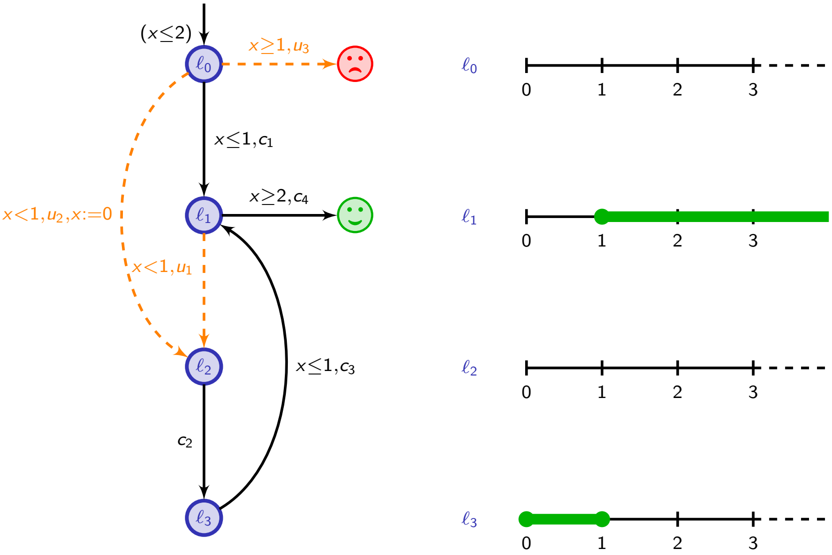


Back to the example: computing winning states

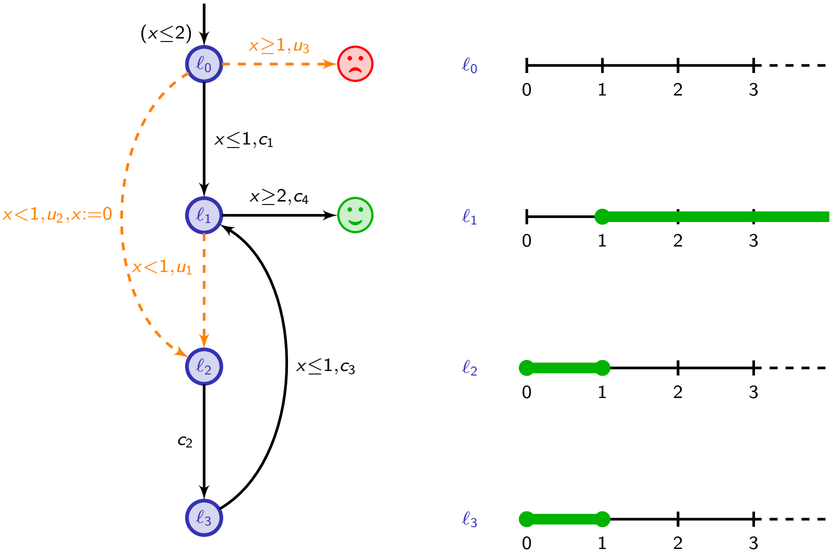




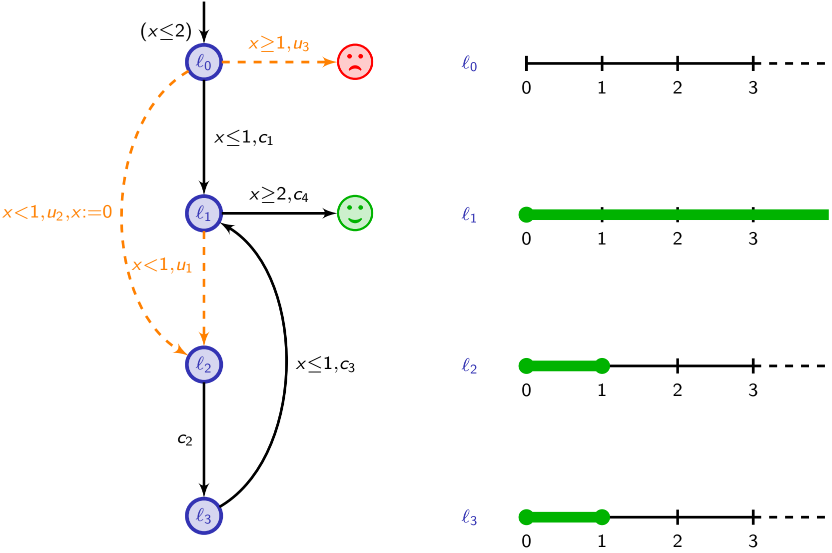
Back to the example: computing winning states



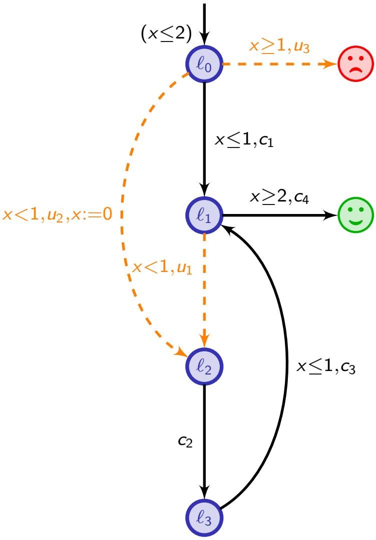
Back to the example: computing winning states

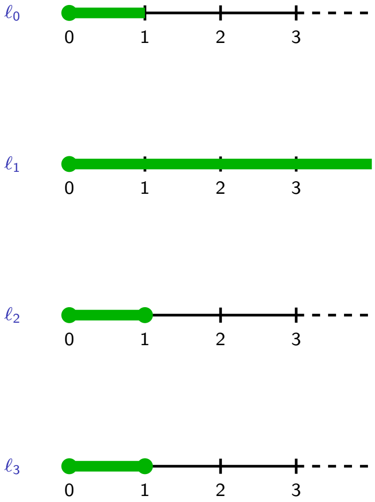


Back to the example: computing winning states

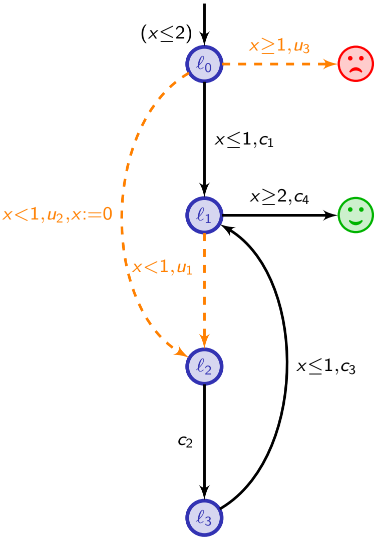


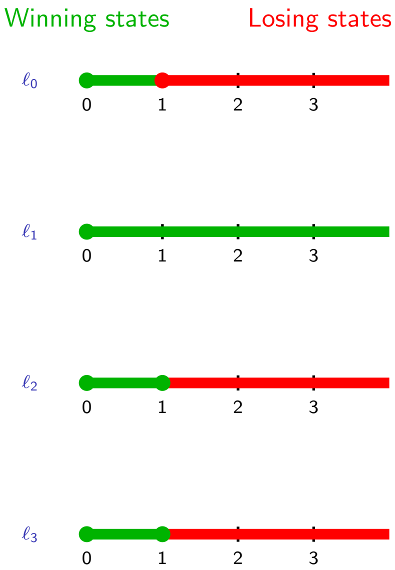
Back to the example: computing winning states





Back to the example: computing winning states





Decidability *via* attractors

Decidability *via* attractors

- $\mathrm{Pred}^{a}(X) = \{\bullet \mid \bullet \xrightarrow{a} \bullet \in X\}$

Decidability *via* attractors

- $\mathrm{Pred}^a(X) = \{\bullet \mid \bullet \xrightarrow{a} \bullet \in X\}$

- controllable and uncontrollable discrete predecessors:

$$\mathrm{cPred}(X) = \bigcup_{a \text{ cont.}} \mathrm{Pred}^a(X) \qquad \mathrm{uPred}(X) = \bigcup_{a \text{ uncont.}} \mathrm{Pred}^a(X)$$

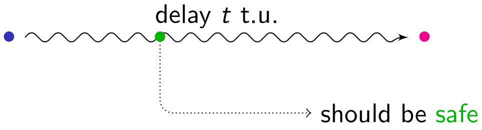
Decidability *via* attractors

- $\mathsf{Pred}^a(X) = \{\bullet \mid \bullet \xrightarrow{a} \bullet \in X\}$

- controllable and uncontrollable discrete predecessors:

$$\mathsf{cPred}(X) = \bigcup_{a \text{ cont.}} \mathsf{Pred}^a(X) \qquad\qquad \mathsf{uPred}(X) = \bigcup_{a \text{ uncont.}} \mathsf{Pred}^a(X)$$

- time controllable predecessors:



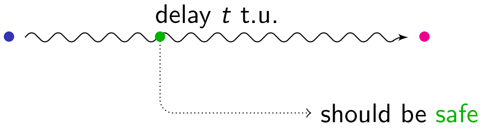
Decidability *via* attractors

- $\mathsf{Pred}^a(X) = \{\bullet \mid \bullet \xrightarrow{a} \bullet \in X\}$

- controllable and uncontrollable discrete predecessors:

$$\mathsf{cPred}(X) = \bigcup_{a \text{ cont.}} \mathsf{Pred}^a(X) \qquad\qquad \mathsf{uPred}(X) = \bigcup_{a \text{ uncont.}} \mathsf{Pred}^a(X)$$

- time controllable predecessors:



$$\mathsf{Pred}_\delta(X, \mathsf{Safe}) = \{\bullet \mid \exists t \geq 0,\ \bullet \xrightarrow{\delta(t)} \bullet$$

$$\text{and } \forall 0 \leq t' \leq t,\ \bullet \xrightarrow{\delta(t')} \bullet \in \mathsf{Safe}\}$$

Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \mathsf{Pred}_\delta(\mathsf{cPred}(X), \neg\mathsf{uPred}(\neg X))$$

Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \mathsf{Pred}_\delta(\mathsf{cPred}(X), \neg\mathsf{uPred}(\neg X))$$

- The states from which one can ensure ☺ in no more than 1 step is:

$$\mathsf{Attr}_1(☺) = \pi(☺)$$

Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \mathsf{Pred}_\delta(\mathsf{cPred}(X), \neg\mathsf{uPred}(\neg X))$$

- The states from which one can ensure ☺ in no more than 1 step is:

$$\mathsf{Attr}_1(☺) = \pi(☺)$$

- The states from which one can ensure ☺ in no more than 2 steps is:

$$\mathsf{Attr}_2(☺) = \pi(\mathsf{Attr}_1(☺))$$

Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \mathsf{Pred}_\delta(\mathsf{cPred}(X), \neg\mathsf{uPred}(\neg X))$$

- The states from which one can ensure $\smiley$ in no more than 1 step is:

$$\mathsf{Attr}_1(\smiley) = \pi(\smiley)$$

- The states from which one can ensure $\smiley$ in no more than 2 steps is:

$$\mathsf{Attr}_2(\smiley) = \pi(\mathsf{Attr}_1(\smiley))$$

- ...

Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \mathsf{Pred}_\delta(\mathsf{cPred}(X), \neg\mathsf{uPred}(\neg X))$$

- The states from which one can ensure ☺ in no more than 1 step is:

$$\mathsf{Attr}_1(☺) = \pi(☺)$$

- The states from which one can ensure ☺ in no more than 2 steps is:

$$\mathsf{Attr}_2(☺) = \pi(\mathsf{Attr}_1(☺))$$

- ...
- The states from which one can ensure ☺ in no more than n steps is:

$$\mathsf{Attr}_n(☺) \quad = \quad \pi(\mathsf{Attr}_{n-1}(☺))$$

Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \mathsf{Pred}_\delta(\mathsf{cPred}(X), \neg\mathsf{uPred}(\neg X))$$

- The states from which one can ensure ☺ in no more than 1 step is:

$$\mathsf{Attr}_1(\smiley) = \pi(\smiley)$$

- The states from which one can ensure ☺ in no more than 2 steps is:

$$\mathsf{Attr}_2(\smiley) = \pi(\mathsf{Attr}_1(\smiley))$$

- ...
- The states from which one can ensure ☺ in no more than n steps is:

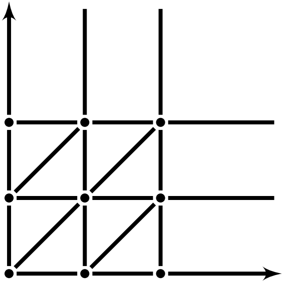
$$\begin{aligned} \mathsf{Attr}_n(\smiley) &= \pi(\mathsf{Attr}_{n-1}(\smiley)) \\ &= \pi^n(\smiley) \end{aligned}$$

Stability w.r.t. regions

- if X is a union of regions, then:
  - $\mathrm{Pred}_a(X)$ is a union of regions,
  - and so are $\mathrm{cPred}(X)$ and $\mathrm{uPred}(X)$.

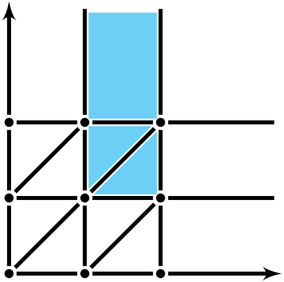
Stability w.r.t. regions

- if X is a union of regions, then:
  - $\mathsf{Pred}_a(X)$ is a union of regions,
  - and so are $\mathsf{cPred}(X)$ and $\mathsf{uPred}(X)$.
- Does π also preserve unions of regions?



Stability w.r.t. regions

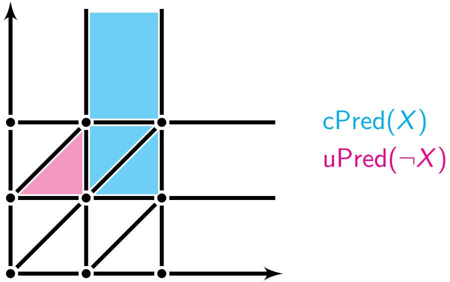
- if X is a union of regions, then:
  - $\mathsf{Pred}_a(X)$ is a union of regions,
  - and so are $\mathsf{cPred}(X)$ and $\mathsf{uPred}(X)$.
- Does π also preserve unions of regions?



cPred(X)

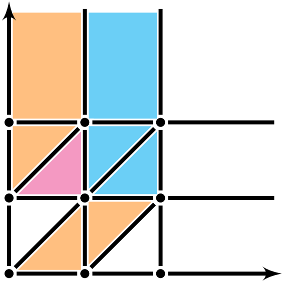
Stability w.r.t. regions

- if X is a union of regions, then:
  - $\mathsf{Pred}_a(X)$ is a union of regions,
  - and so are $\mathsf{cPred}(X)$ and $\mathsf{uPred}(X)$.
- Does π also preserve unions of regions?



Stability w.r.t. regions

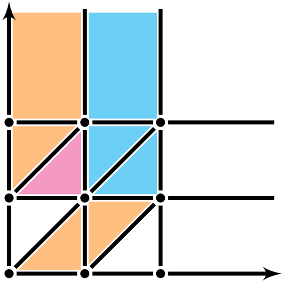
- if X is a union of regions, then:
  - $\mathsf{Pred}_a(X)$ is a union of regions,
  - and so are $\mathsf{cPred}(X)$ and $\mathsf{uPred}(X)$.
- Does π also preserve unions of regions?



$\mathsf{cPred}(X)$
$\mathsf{uPred}(\neg X)$
$\mathsf{Pred}_\delta(\mathsf{cPred}(X), \neg \mathsf{uPred}(\neg X))$

Stability w.r.t. regions

- if X is a union of regions, then:
  - $\mathrm{Pred}_a(X)$ is a union of regions,
  - and so are $\mathrm{cPred}(X)$ and $\mathrm{uPred}(X)$.
- Does π also preserve unions of regions? Yes!



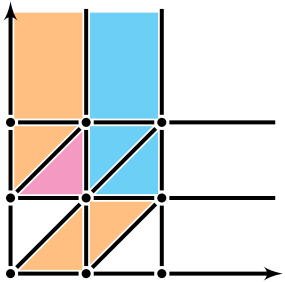
$\mathrm{cPred}(X)$

$\mathrm{uPred}(\neg X)$

$\mathrm{Pred}_\delta(\mathrm{cPred}(X), \neg\mathrm{uPred}(\neg X))$

Stability w.r.t. regions

- if X is a union of regions, then:
  - $\mathrm{Pred}_a(X)$ is a union of regions,
  - and so are $\mathrm{cPred}(X)$ and $\mathrm{uPred}(X)$.
- Does π also preserve unions of regions? Yes!



$\mathrm{cPred}(X)$

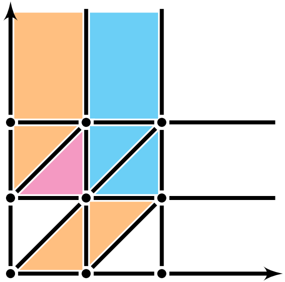
$\mathrm{uPred}(\neg X)$

$\mathrm{Pred}_\delta(\mathrm{cPred}(X), \neg\mathrm{uPred}(\neg X))$

(but it does not preserve zones...)

Stability w.r.t. regions

- if X is a union of regions, then:
  - $\mathrm{Pred}_a(X)$ is a union of regions,
  - and so are $\mathrm{cPred}(X)$ and $\mathrm{uPred}(X)$.
- Does π also preserve unions of regions? Yes!

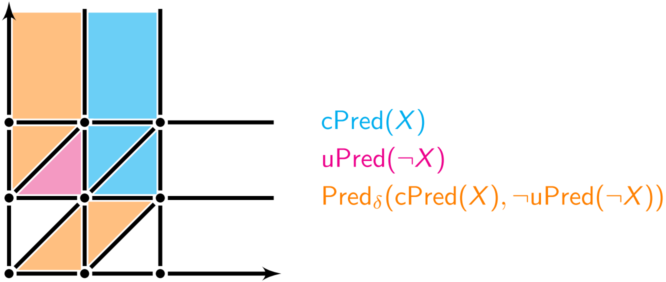


(but it does not preserve zones...)

$\rightsquigarrow$ the computation of π^*(☺) terminates!

Stability w.r.t. regions

- if X is a union of regions, then:
  - $\mathsf{Pred}_a(X)$ is a union of regions,
  - and so are $\mathsf{cPred}(X)$ and $\mathsf{uPred}(X)$.
- Does π also preserve unions of regions? Yes!



(but it does not preserve zones...)

$\rightsquigarrow$ the computation of $\pi^*(☺)$ terminates!
... and is correct

Timed games with a safety objective

- We can use operator $\widetilde{\pi}$ defined by

$$\widetilde{\pi}(X) = \mathsf{Pred}_\delta(X \cap \mathsf{cPred}(X), \neg \mathsf{uPred}(\neg X))$$

instead of π, and compute $\widetilde{\pi}^*(\neg \text{☹})$

Timed games with a safety objective

- We can use operator $\widetilde{\pi}$ defined by

$$\widetilde{\pi}(X) = \mathsf{Pred}_\delta(X \cap \mathsf{cPred}(X), \neg \mathsf{uPred}(\neg X))$$

  instead of π, and compute $\widetilde{\pi}^*(\neg \frownie)$

- It is also stable w.r.t. regions.

Some remarks

Control games

Our games are control games,

Some remarks

Control games

Our games are control games, and in particular they:

- are asymmetric
  - the environment can preempt any decision of the controller
  - we take the point-of-view of the controller

Some remarks

Control games

Our games are control games, and in particular they:

- are asymmetric
  - the environment can preempt any decision of the controller
  - we take the point-of-view of the controller
- are neither concurrent nor turn-based

Some remarks

Control games

Our games are control games, and in particular they:

- are asymmetric
  - the environment can preempt any decision of the controller
  - we take the point-of-view of the controller
- are neither concurrent nor turn-based
- do not take into account Zenoness considerations

Some remarks

Control games

Our games are control games, and in particular they:

- are asymmetric
  - the environment can preempt any decision of the controller
  - we take the point-of-view of the controller
- are neither concurrent nor turn-based
- do not take into account Zenoness considerations

  $\rightsquigarrow$ can be done adding a Büchi winning condition

Some remarks

Control games

Our games are control games, and in particular they:

- are asymmetric
  - the environment can preempt any decision of the controller
  - we take the point-of-view of the controller
- are neither concurrent nor turn-based
- do not take into account Zenoness considerations

$\rightsquigarrow$ can be done adding a Büchi winning condition

Alternative models [AFH+03,BLMO07]

- concurrent and symmetric games
- some incorporate non-Zenoness in the winning condition

[AFH+03] de Alfaro, Faella, Henzinger, Majumdar, Stoelinga.
[BLMO07] Brihaye, Laroussinie, Markey, Oreiby. Timed Concurrent Game Structures *(CONCUR'07)*.

Some remarks

Control games

Our games are control games, and in particular they:

- are asymmetric
  - the environment can preempt any decision of the controller
  - we take the point-of-view of the controller
- are neither concurrent nor turn-based
- do not take into account Zenoness considerations

$\rightsquigarrow$ can be done adding a Büchi winning condition

Alternative models [AFH+03,BLMO07]

- concurrent and symmetric games
- some incorporate non-Zenoness in the winning condition

$\rightsquigarrow$ those games are not determined ☹

[AFH+03] de Alfaro, Faella, Henzinger, Majumdar, Stoelinga.
[BLMO07] Brihaye, Laroussinie, Markey, Oreiby. Timed Concurrent Game Structures *(CONCUR'07)*.

Some remarks

Control games

Our games are control games, and in particular they:

- are asymmetric
  - the environment can preempt any decision of the controller
  - we take the point-of-view of the controller
- are neither concurrent nor turn-based
- do not take into account Zenoness considerations

$\rightsquigarrow$ can be done adding a Büchi winning condition

Alternative models [AFH+03,BLMO07]

- concurrent and symmetric games
- some incorporate non-Zenoness in the winning condition

$\rightsquigarrow$ those games are not determined ☹

... and they may not represent a proper interaction with an environment ☹

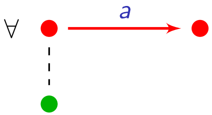
[AFH+03] de Alfaro, Faella, Henzinger, Majumdar, Stoelinga.
[BLMO07] Brihaye, Laroussinie, Markey, Oreiby. Timed Concurrent Game Structures *(CONCUR'07)*.

Application of timed games to strong timed bisimulation

This is a relation between $\textcolor{red}{\bullet}$ and $\textcolor{green}{\bullet}$ such that:

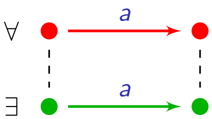
Application of timed games to strong timed bisimulation

This is a relation between • and • such that:



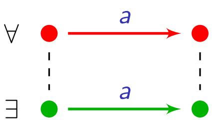
Application of timed games to strong timed bisimulation

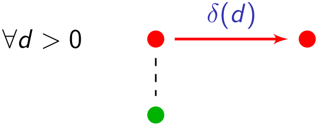
This is a relation between • and • such that:



Application of timed games to strong timed bisimulation

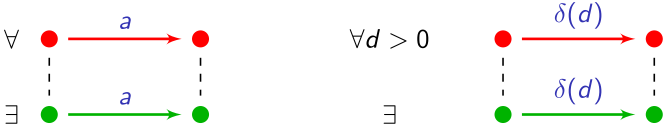
This is a relation between • and • such that:





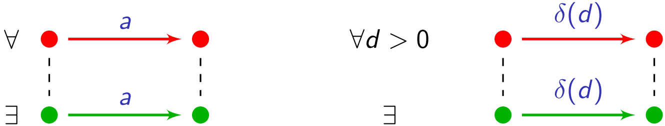
Application of timed games to strong timed bisimulation

This is a relation between • and • such that:



Application of timed games to strong timed bisimulation

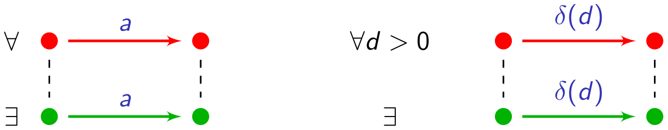
This is a relation between • and • such that:



... and vice-versa (swap • and •) for the bisimulation relation.

Application of timed games to strong timed bisimulation

This is a relation between • and • such that:

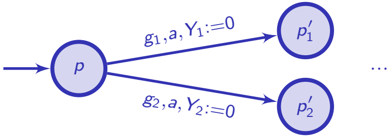


... and vice-versa (swap • and •) for the bisimulation relation.

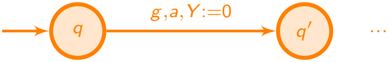
Theorem

Strong timed (bi)simulation between timed automata is decidable and EXPTIME-complete.

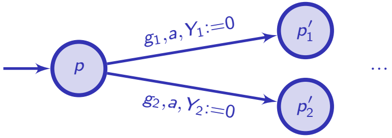
timed automaton $\mathcal{A}$



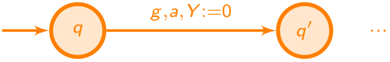
timed automaton $\mathcal{B}$

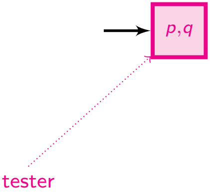


timed automaton $\mathcal{A}$

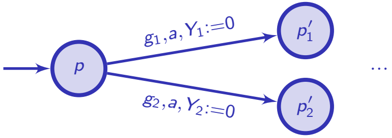


timed automaton $\mathcal{B}$

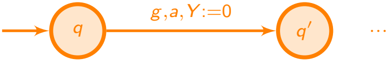


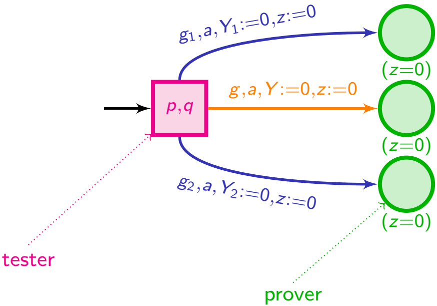


timed automaton $\mathcal{A}$

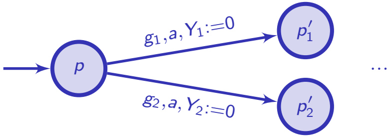


timed automaton $\mathcal{B}$

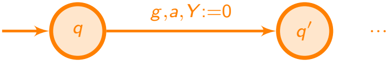


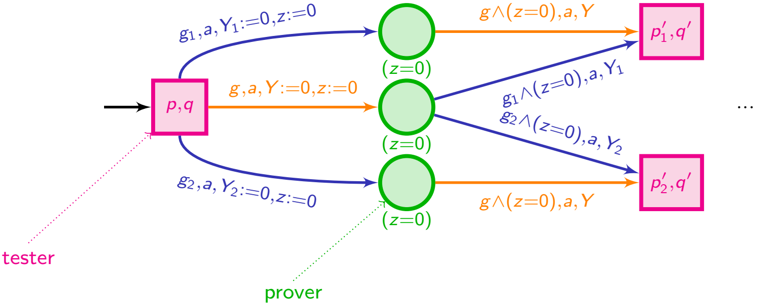


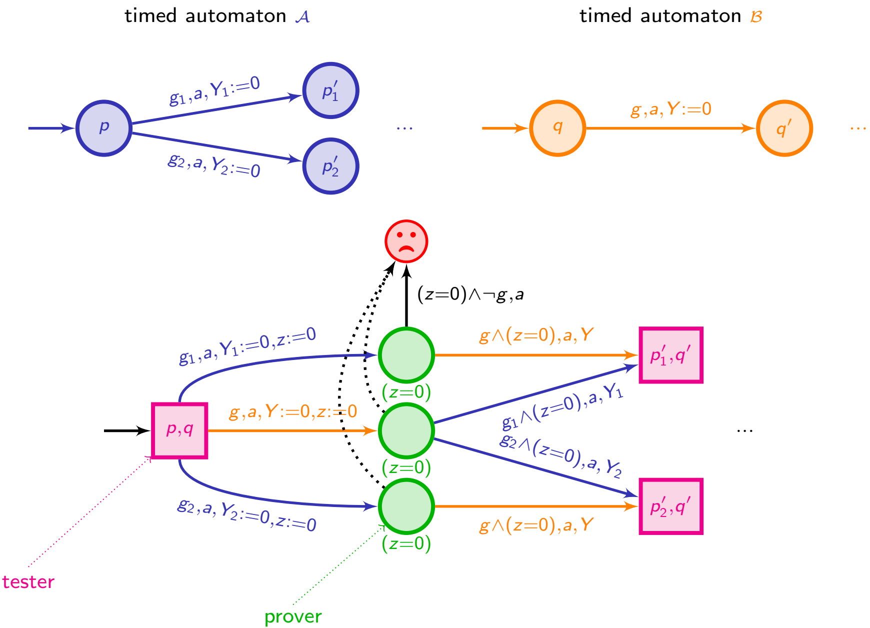
timed automaton $\mathcal{A}$

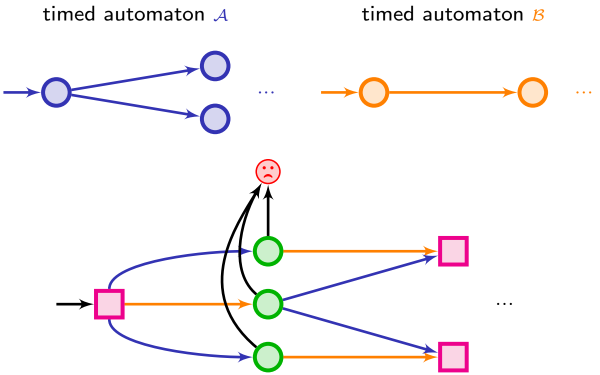


timed automaton $\mathcal{B}$






timed automaton $\mathcal{A}$
$g_1, a, Y_1{:=}0$
$g_2, a, Y_2{:=}0$
p
p_1'
p_2'
...
timed automaton $\mathcal{B}$
$g, a, Y{:=}0$
q
q'
...
$(z{=}0) \wedge \neg g, a$
$g_1, a, Y_1{:=}0, z{:=}0$
$g, a, Y{:=}0, z{:=}0$
$g_2, a, Y_2{:=}0, z{:=}0$
p, q
$(z{=}0)$
$(z{=}0)$
$(z{=}0)$
$g \wedge (z{=}0), a, Y$
$g_1 \wedge (z{=}0), a, Y_1$
$g_2 \wedge (z{=}0), a, Y_2$
$g \wedge (z{=}0), a, Y$
p_1', q'
p_2', q'
...
tester
prover



$\mathcal{A}$ and $\mathcal{B}$ are strongly timed bisimilar

iff

the prover has a winning strategy to avoid

What else?

- Implementation: Uppaal-Tiga implements a forward algorithm to compute winning states and winning strategies [CDF+05,BCD+07]

[CDF+05] Cassez, David, Fleury, Larsen, Lime. Efficient on-the-fly algorithms for the analysis of timed games *(CONCUR'05)*.
[BCD+07] Berhmann, Cougnard, David, Fleury, Larsen, Lime. Uppaal-Tiga: Time for playing games! *(CAV'07)*.

What else?

- Implementation: Uppaal-Tiga implements a forward algorithm to compute winning states and winning strategies [CDF+05,BCD+07]
  - A climate controller in a pig stable (Skov A/S) [JRLD07]

[CDF+05] Cassez, David, Fleury, Larsen, Lime. Efficient on-the-fly algorithms for the analysis of timed games *(CONCUR'05)*.
[BCD+07] Berhmann, Cougnard, David, Fleury, Larsen, Lime. Uppaal-Tiga: Time for playing games! *(CAV'07)*.
[JRLD07] Jessen, Rasmussen, Larsen, David. Guided Controller Synthesis for Climate Controller Using Uppaal Tiga *(FORMATS'07)*.

What else?

- Implementation: Uppaal-Tiga implements a forward algorithm to compute winning states and winning strategies [CDF+05,BCD+07]
  - A climate controller in a pig stable (Skov A/S) [JRLD07]
  - A pump controller (Hydac Gmbh) [CJL+09]

[CDF+05] Cassez, David, Fleury, Larsen, Lime. Efficient on-the-fly algorithms for the analysis of timed games *(CONCUR'05)*.
[BCD+07] Berhmann, Cougnard, David, Fleury, Larsen, Lime. Uppaal-Tiga: Time for playing games! *(CAV'07)*.
[JRLD07] Jessen, Rasmussen, Larsen, David. Guided Controller Synthesis for Climate Controller Using Uppaal Tiga *(FORMATS'07)*.
[CJL+09] Cassez, Jessen, Larsen, Raskin, Reynier.Automatic Synthesis of Robust and Optimal Controllers – An Industrial Case Study *(HSCC'09)*.

What else?

- Implementation: Uppaal-Tiga implements a forward algorithm to compute winning states and winning strategies [CDF+05,BCD+07]
  - A climate controller in a pig stable (Skov A/S) [JRLD07]
  - A pump controller (Hydac Gmbh) [CJL+09]
- Partial observation/Incomplete information:
  - action-based observation: undecidable [BDMP03]
  - finite-observation of states: decidable [CDL+07]

[CDF+05] Cassez, David, Fleury, Larsen, Lime. Efficient on-the-fly algorithms for the analysis of timed games *(CONCUR'05)*.
[BCD+07] Berhmann, Cougnard, David, Fleury, Larsen, Lime. Uppaal-Tiga: Time for playing games! *(CAV'07)*.
[JRLD07] Jessen, Rasmussen, Larsen, David. Guided Controller Synthesis for Climate Controller Using Uppaal Tiga *(FORMATS'07)*.
[CJL+09] Cassez, Jessen, Larsen, Raskin, Reynier.Automatic Synthesis of Robust and Optimal Controllers – An Industrial Case Study *(HSCC'09)*.
[BDMP03] Bouyer, D'Souza, Madhusudan, Petit. Timed control with partial observability *(CAV'03)*.
[CDL+07] Cassez, David, Larsen, Lime, Raskin. Timed control with observation based and stuttering invariant strategies *(ATVA'07)*.

What else?

- Implementation: Uppaal-Tiga implements a forward algorithm to compute winning states and winning strategies [CDF+05,BCD+07]
  - A climate controller in a pig stable (Skov A/S) [JRLD07]
  - A pump controller (Hydac Gmbh) [CJL+09]
- Partial observation/Incomplete information:
  - action-based observation: undecidable [BDMP03]
  - finite-observation of states: decidable [CDL+07]
- Quantitative constraints, see the next lecture!

[CDF+05] Cassez, David, Fleury, Larsen, Lime. Efficient on-the-fly algorithms for the analysis of timed games *(CONCUR'05)*.
[BCD+07] Berhmann, Cougnard, David, Fleury, Larsen, Lime. Uppaal-Tiga: Time for playing games! *(CAV'07)*.
[JRLD07] Jessen, Rasmussen, Larsen, David. Guided Controller Synthesis for Climate Controller Using Uppaal Tiga *(FORMATS'07)*.
[CJL+09] Cassez, Jessen, Larsen, Raskin, Reynier.Automatic Synthesis of Robust and Optimal Controllers – An Industrial Case Study *(HSCC'09)*.
[BDMP03] Bouyer, D'Souza, Madhusudan, Petit. Timed control with partial observability *(CAV'03)*.
[CDL+07] Cassez, David, Larsen, Lime, Raskin. Timed control with observation based and stuttering invariant strategies *(ATVA'07)*.