Timed games

Patricia Bouyer-Decitre

LSV, CNRS & ENS Cachan, France
Why (timed) games?

- to model uncertainty

Example of a processor in the taskgraph example

![Diagram of processor states and transitions]

- add: $x := 0$ when $x \leq 2$
- mult: $x := 0$ when $x \leq 3$
- idle: final state
- done: final state
Why (timed) games?

- to model uncertainty

Example of a processor in the taskgraph example

![Diagram of processor states and transitions](image)
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**

![Diagram of a processor with states and transitions]

- 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

```
+  \( x \leq 2 \)
\( x \geq 1 \)
\( x := 0 \)
done
add

idle
\( x := 0 \)
done
mult
\( x \leq 3 \)

\times
```

- 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 😊

How do we play?

According to a strategy:

\[ f : \text{history} \xrightarrow{\text{texmap}} \text{char} \rightarrow (\text{delay}, \text{cont. transition}) \]
An example of a timed game

Rule of the game
- **Aim:** avoid 😞 and reach 😊
- **How do we play?** According to a strategy:

\[
\begin{align*}
\ell_0 & \xrightarrow{(x \leq 2)} (x \geq 1, u_3) \\
\ell_1 & \xrightarrow{x \leq 1, c_1} x < 1, u_2, x := 0 \\
\ell_2 & \xrightarrow{x < 1, u_1} x < 1, u_1 \\
\ell_3 & \xrightarrow{x \leq 1, c_3} x \geq 1, c_4 \\
\end{align*}
\]
An example of a timed game

Rule of the game

- **Aim:** avoid 😞 and reach 😊
- **How do we play?** According to a strategy:

  \[ f : \text{history} \mapsto (\text{delay, cont. transition}) \]
An example of a timed game

Rule of the game

- **Aim:** avoid 😞 and reach 😊
- **How do we play?** According to a strategy:
  
  \[ f : \text{history} \mapsto (\text{delay, cont. transition}) \]

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:

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

A (memoryless) winning strategy

- from \((\ell_0, 0)\), play \((0.5, c_1)\)
  \[ \sim \text{can be preempted by } u_2 \]
An example of a timed game

Rule of the game

- **Aim:** avoid 😞 and reach 😊
- **How do we play?** According to a strategy:

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

A (memoryless) winning strategy

- from \((\ell_0, 0)\), play \((0.5, c_1)\)
  - 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:
  
  \[ f : \text{history} \mapsto (\text{delay, cont. transition}) \]

A (memoryless) winning strategy

- from \((\ell_0, 0)\), play \((0.5, c_1)\)
  
  \(\leadsto\) 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:

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

A (memoryless) winning strategy

- from \((\ell_0, 0)\), play \((0.5, c_1)\)
  - 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:
  
  \[ f : \text{history} \mapsto (\text{delay, cont. transition}) \]

Problems to be considered

\[ (x \leq 2) \]

\[ x \leq 1, \ell_1 \]

\[ x < 1, u_1 \]

\[ x < 1, u_2, x := 0 \]

\[ x \geq 1, u_3 \]

\[ x \geq 2, c_4 \]

\[ x \leq 1, c_3 \]

\[ c_2 \]

\[ x \leq 1, c_1 \]
An example of a timed game

Rule of the game

- **Aim:** avoid 🙁 and reach 🙂
- **How do we play?** According to a strategy:
  \[ f : \text{history} \mapsto \text{(delay, cont. transition)} \]

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.


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.

\[ \leadsto \text{classical regions are sufficient for solving such problems} \]
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.

\[\leadsto \text{classical regions are sufficient for solving such problems}\]

**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*).
[JT07] Jurdziński, Trivedi. Reachability-time games on timed automata (*ICALP’07*).
Back to the example: computing winning states

$(x \leq 2) \quad x \geq 1, u_3$

$x \leq 1, c_1$

$x < 1, u_1$

$x < 1, u_2, x := 0$

$x \geq 2, c_4$

$x \leq 1, c_3$

$c_2$

$L_0 \quad L_1 \quad L_2 \quad L_3$
Back to the example: computing winning states

\( \ell_0 \) \( x \leq 2 \)

\( \ell_1 \) \( x \geq 1, u_3 \)

\( \ell_2 \) \( x \leq 1, c_1 \)

\( \ell_3 \) \( x \leq 1, c_3 \)

\( \ell_4 \) \( x \geq 2, c_4 \)

\( x < 1, u_2, x := 0 \)

\( x < 1, u_1 \)

\( c_2 \)
Back to the example: computing winning states

\[ (x \leq 2) \]

\[ x \geq 1, u_3 \]

\[ x \leq 1, c_1 \]

\[ x < 1, u_1 \]

\[ x < 1, u_2, x := 0 \]

\[ x \geq 2, c_4 \]

\[ x \leq 1, c_3 \]

\[ c_2 \]

\[ \ell_0 \]

\[ \ell_1 \]

\[ \ell_2 \]

\[ \ell_3 \]
Back to the example: computing winning states

\( \ell_0 (x \leq 2) \)

\( x \geq 1, u_3 \)

\( x \leq 1, c_1 \)

\( x < 1, u_2, x := 0 \)

\( x < 1, u_1 \)

\( x \leq 1, c_3 \)

\( c_2 \)

\( x \geq 2, c_4 \)

\( x \leq 1, c_1 \)

\( x < 1, u_1 \)

\( x < 1, u_2, x := 0 \)
Back to the example: computing winning states

\[ \begin{align*}
\ell_0 & \quad (x \leq 2) \\
\ell_1 & \quad x \geq 1, u_3 \\
\ell_2 & \quad x \leq 1, c_1 \\
\ell_3 & \quad x < 1, u_1 \\
\ell_0 & \quad x < 1, u_2, x := 0 \\
\ell_1 & \quad x \geq 2, c_4 \\
\ell_2 & \quad x \leq 1, c_3 \\
\ell_3 & \quad c_2
\end{align*} \]
Back to the example: computing winning states

\[ x \leq 2 \]

\[ x \geq 1, u_3 \]

\[ x \leq 1, c_1 \]

\[ x < 1, u_1 \]

\[ x < 1, u_2, x := 0 \]

\[ x \geq 2, c_4 \]

\[ x \leq 1, c_3 \]

\[ c_2 \]
Back to the example: computing winning states

\( x \leq 2 \)

\( x \geq 1, u_3 \)

\( x \leq 1, c_1 \)

\( x < 1, u_1 \)

\( x < 1, u_2, x := 0 \)

\( x \geq 2, c_4 \)

\( x \leq 1, c_3 \)

\( x \leq 1, c_2 \)
Back to the example: computing winning states

\[ (x \leq 2) \]

\[ \ell_0 \]

\[ x \geq 1, u_3 \]

\[ x \leq 1, c_1 \]

\[ x < 1, u_2, x := 0 \]

\[ \ell_1 \]

\[ x \geq 2, c_4 \]

\[ x < 1, u_1 \]

\[ \ell_2 \]

\[ c_2 \]

\[ x \leq 1, c_3 \]

\[ \ell_3 \]
Decidability *via* attractors
Decidability via attractors

\[ \text{Pred}^a(X) = \{ \bullet | \bullet \xrightarrow{a} \bullet \in X \} \]
Decidability via attractors

\[ \text{Pred}^a(X) = \{ \bullet \mid \bullet \overset{a}{\rightarrow} \bullet \in X \} \]

controllable and uncontrollable discrete predecessors:

\[
\text{cPred}(X) = \bigcup_{a \text{ cont.}} \text{Pred}^a(X) \quad \text{uPred}(X) = \bigcup_{a \text{ uncont.}} \text{Pred}^a(X)
\]
Decidability via attractors

- $\text{Pred}^a(X) = \{ \bullet | \bullet \xrightarrow{a} \bullet \in X \}$

- controllable and uncontrollable discrete predecessors:

$$c\text{Pred}(X) = \bigcup_{a \text{ cont.}} \text{Pred}^a(X)$$

$$u\text{Pred}(X) = \bigcup_{a \text{ uncont.}} \text{Pred}^a(X)$$

- time controllable predecessors:

- delay $t$ t.u.

$$\begin{array}{c}
\text{should be safe}
\end{array}$$
Decidability \textit{via} attractors

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

- controllable and uncontrollable discrete predecessors:

\[ \text{cPred}(X) = \bigcup_{a \text{ cont.}} \text{Pred}^{a}(X) \quad \text{uPred}(X) = \bigcup_{a \text{ uncont.}} \text{Pred}^{a}(X) \]

- time controllable predecessors:

\[ \text{Pred}_{\delta}(X, \text{Safe}) = \{ \bullet \mid \exists t \geq 0, \bullet \xrightarrow{\delta(t)} \bullet \]

\[ \text{and } \forall 0 \leq t' \leq t, \bullet \xrightarrow{\delta(t')} \bullet \in \text{Safe} \} \]
Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \text{Pred}_\delta(c\text{Pred}(X), \neg u\text{Pred}(\neg X))$$
Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \text{Pred}_\delta(c\text{Pred}(X), \neg u\text{Pred}(\neg X))$$

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

$$\text{Attr}_1(😊) = \pi(😊)$$
Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \text{Pred}_\delta(c\text{Pred}(X), \neg u\text{Pred}(\neg X))$$

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

  $$\text{Attr}_1(😃) = \pi(😃)$$

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

  $$\text{Attr}_2(😃) = \pi(\text{Attr}_1(😃))$$
Timed games with a reachability objective

We write:

\[ \pi(X) = X \cup \text{Pred}_\delta(\text{cPred}(X), \neg\text{uPred}(\neg X)) \]

- The states from which one can ensure \( \blacksquare \) in no more than 1 step is:

\[ \text{Attr}_1(\blacksquare) = \pi(\blacksquare) \]

- The states from which one can ensure \( \blacksquare \) in no more than 2 steps is:

\[ \text{Attr}_2(\blacksquare) = \pi(\text{Attr}_1(\blacksquare)) \]

- \( \ldots \)
Timed games with a reachability objective

We write:

\[ \pi(X) = X \cup \text{Pred}_\delta(c\text{Pred}(X), \neg u\text{Pred}(\neg X)) \]

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

  \[ \text{Attr}_1(😊) = \pi(😊) \]

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

  \[ \text{Attr}_2(😊) = \pi(\text{Attr}_1(😊)) \]

- The states from which one can ensure 😊 in no more than \( n \) steps is:

  \[ \text{Attr}_n(😊) = \pi(\text{Attr}_{n-1}(😊)) \]
Timed games with a reachability objective

We write:

$$\pi(X) = X \cup \text{Pred}_\delta(c\text{Pred}(X), \neg u\text{Pred}(\neg X))$$

- The states from which one can ensure 😊 in no more than 1 step is:
  $$\text{Attr}_1(😊) = \pi(😊)$$

- The states from which one can ensure 😊 in no more than 2 steps is:
  $$\text{Attr}_2(😊) = \pi(\text{Attr}_1(😊))$$

- ...  
- The states from which one can ensure 😊 in no more than $n$ steps is:
  $$\text{Attr}_n(😊) = \pi(\text{Attr}_{n-1}(😊))$$
  $$= \pi^n(😊)$$
Stability w.r.t. regions

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

- if $X$ is a union of regions, then:
  - $\text{Pred}_a(X)$ is a union of regions,
  - and so are $\text{cPred}(X)$ and $\text{uPred}(X)$.
- Does $\pi$ also preserve unions of regions?
Stability w.r.t. regions

- if $X$ is a union of regions, then:
  - $\text{Pred}_a(X)$ is a union of regions,
  - and so are $\text{cPred}(X)$ and $\text{uPred}(X)$.
- Does $\pi$ also preserve unions of regions?
Stability w.r.t. regions

- if $X$ is a union of regions, then:
  - $\text{Pred}_a(X)$ is a union of regions,
  - and so are $c\text{Pred}(X)$ and $u\text{Pred}(X)$.
- Does $\pi$ also preserve unions of regions?
Stability w.r.t. regions

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

- Does $\pi$ also preserve unions of regions?

![Diagram showing the relationships between cPred(X), uPred(¬X), and Predδ(cPred(X), ¬uPred(¬X))].
Stability w.r.t. regions

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

\[\text{cPred}(X)\]

\[\text{uPred}(\neg X)\]

\[\text{Pred}_\delta(\text{cPred}(X), \neg \text{uPred}(\neg X))\]
Stability w.r.t. regions

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

(but it does not preserve zones...)

\[
\begin{align*}
\text{cPred}(X) \\
\text{uPred}(\neg X) \\
\text{Pred}_\delta(\text{cPred}(X), \neg \text{uPred}(\neg X))
\end{align*}
\]
Stability w.r.t. regions

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

(cPred($X$), uPred($\neg X$))

(but it does not preserve zones...)

$\leadsto$ the computation of $\pi^*(\text{smiley})$ terminates!
Stability w.r.t. regions

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

![Diagram showing regions and their relationships]

(cPred($X$), ¬uPred(¬$X$))

(but it does not preserve zones...)

$\leadsto$ the computation of $\pi^*$ terminates!

... and is correct
Timed games with a safety objective

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

$$\tilde{\pi}(X) = \text{Pred}_\delta(X \cap c\text{Pred}(X), \neg u\text{Pred}(\neg X))$$

instead of $\pi$, and compute $\tilde{\pi}^*(\neg ☹)$
Timed games with a safety objective

- We can use operator \( \tilde{\pi} \) defined by

\[
\tilde{\pi}(X) = \text{Pred}_\delta(X \cap c\text{Pred}(X), \neg u\text{Pred}(\neg X))
\]

instead of \( \pi \), and compute \( \tilde{\pi}^*(\neg \frownie) \)

- 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

\[ \rightsquigarrow \text{can be done adding a Büchi winning condition} \]
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
  \( \leadsto \) 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
  \[\leadsto\text{can be done adding a Büchi winning condition}\]

Alternative models [AFH+03,BLMO07]
- concurrent and symmetric games
- some incorporate non-Zenoness in the winning condition
  \[\leadsto\text{those games are not determined} \frownie\]

[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
  \[\leadsto\] can be done adding a Büchi winning condition

Alternative models \([\text{AFH}+03,\text{BLMO07}]\)

- concurrent and symmetric games
- some incorporate non-Zenoness in the winning condition
  \[\leadsto\] those games are not determined

... and they may not represent a proper interaction with an environment
Application of timed games to strong timed bisimulation

This is a relation between • and ◦ such that:

\[
\forall \exists a / u_{1D6FF}(d) \quad \forall d > 0 \exists / u_{1D6FF}(d)
\]

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

Theorem

Strong timed (bi)simulation between timed automata is decidable and EXPTIME-complete.
Application of timed games to strong timed bisimulation

This is a relation between $\bullet$ and $\bullet$ such that:

$$\forall (\exists d > 0 (\exists)) \quad \text{and vice-versa (swap $\bullet$ and $\bullet$)}$$

Theorem

Strong timed (bi)simulation between timed automata is decidable and EXPTIME-complete.
Application of timed games to strong timed bisimulation

This is a relation between \( \bullet \) and \( \bullet \) such that:

\[
\forall u \in \mathbb{D} \quad \exists \exists u \in \mathbb{D} \\
\exists \exists u \in \mathbb{D} \quad \forall u \in \mathbb{D}
\]

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

Theorem

Strong timed (bi)simulation between timed automata is decidable and EXPTIME-complete.
Application of timed games to strong timed bisimulation

This is a relation between ● and ○ such that:

∀ \exists a \quad \forall d > 0 \quad \delta(d)

Theorem

Strong timed (bi)simulation between timed automata is decidable and EXPTIME-complete.
Application of timed games to strong timed bisimulation

This is a relation between $\bullet$ and $\bullet$ such that:

\begin{align*}
\forall & \quad \bullet \xrightarrow{a} \bullet \\
\exists & \quad \bullet \xrightarrow{a} \bullet
\end{align*}

\begin{align*}
\forall d > 0 & \quad \bullet \xrightarrow{\delta(d)} \bullet \\
\exists & \quad \bullet \xrightarrow{\delta(d)} \bullet
\end{align*}

Theorem
Strong timed (bi)simulation between timed automata is decidable and EXPTIME-complete.
Application of timed games to strong timed bisimulation

This is a relation between \( \bullet \) and \( \bullet \) such that:

\[
\forall \quad \exists \\
\exists \quad \forall
\]

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

Theorem
Strong timed (bi)simulation between timed automata is decidable and EXPTIME-complete.
Application of timed games to strong timed bisimulation

This is a relation between \( \bullet \) and \( \bullet \) such that:

\[
\forall a \quad \exists \quad \forall d > 0 \quad \exists \quad \delta(d)
\]

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

**Theorem**

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

$g_1, a, Y_1 := 0$

$g_2, a, Y_2 := 0$

$g_1, a, Y_1 := 0$

$g_2, a, Y_2 := 0$

$g, a, Y := 0$

$g_1, a, Y_1 := 0$

$g_2, a, Y_2 := 0$

$g, a, Y := 0$

$g_1, a, Y_1 := 0$

$g_2, a, Y_2 := 0$

$g, a, Y := 0$
timed automaton $A$

$p \xrightarrow{g_1, a, Y_1 := 0} p_1$

$p \xrightarrow{g_2, a, Y_2 := 0} p_2$

$\ldots$

$q \xrightarrow{g, a, Y := 0} q'$

$\ldots$

$p, q$

$\cdots$

$\cdots$

tester
timed automaton $A$

$\quad p \xrightarrow{g_1, a, Y_1:=0} p_1$
$\quad p \xrightarrow{g_2, a, Y_2:=0} p_2'$

$\quad q \xrightarrow{g, a, Y:=0} q'$

$\quad g_1, a, Y_1:=0, z:=0 \xrightarrow{g_1, a, Y_1:=0, z:=0} (z=0)$
$\quad g_2, a, Y_2:=0, z:=0 \xrightarrow{g_2, a, Y_2:=0, z:=0} (z=0)$

$\quad (z=0)$

$\quad (z=0)$

$\quad (z=0)$

$\quad (z=0)$

prover
tester
\( \mathcal{A} \) and \( \mathcal{B} \) are strongly timed bisimilar iff the prover \( \bigcirc \) has a winning strategy to avoid \( \frownie \)
What else?

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

---

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]

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 \[CDL+07\]
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!

---

[BDMP03] Bouyer, D’Souza, Madhusudan, Petit. Timed control with partial observability (CAV’03).