Optimal Timed Games

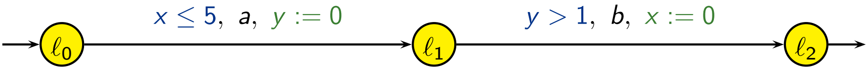
Patricia Bouyer

LSV – CNRS & ENS de Cachan – France

Timed automata

[Alur & Dill 90's]

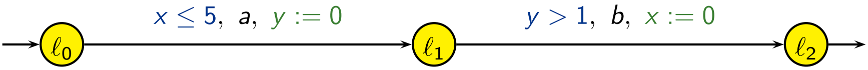
x, y : clocks



Timed automata

[Alur & Dill 90's]

x, y : clocks

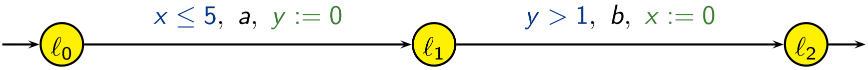


	ℓ_0	$\xrightarrow{\delta(4.1)}$	ℓ_0	$\xrightarrow{a}$	ℓ_1	$\xrightarrow{\delta(1.4)}$	ℓ_1	$\xrightarrow{b}$	ℓ_2
x	0		4.1		4.1		5.5		0
y	0		4.1		0		1.4		1.4

Timed automata

[Alur & Dill 90's]

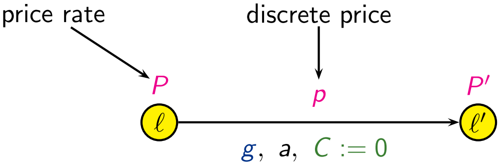
x, y : clocks



	ℓ_0	$\xrightarrow{\delta(4.1)}$	ℓ_0	$\xrightarrow{a}$	ℓ_1	$\xrightarrow{\delta(1.4)}$	ℓ_1	$\xrightarrow{b}$	ℓ_2
x	0		4.1		4.1		5.5		0
y	0		4.1		0		1.4		1.4

(clock) valuation

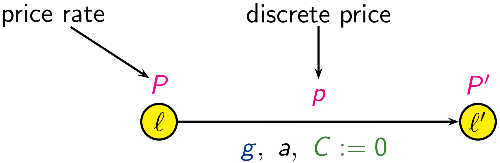
Model of priced timed automata [HSCC'01]



cost $\equiv$ price

Model of priced timed automata [HSCC'01]

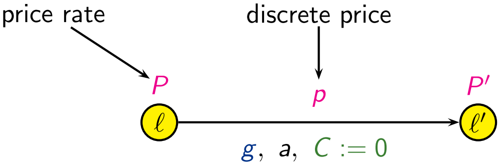
cost ≡ price



- a configuration: (ℓ, v)
- two kinds of transitions:

$$\begin{cases} (\ell, v) \xrightarrow{\delta(d)} (\ell, v+d) \\ (\ell, v) \xrightarrow{a} (\ell', v') \text{ where } \begin{cases} v \models g \\ v' = [C \leftarrow 0]v \end{cases} \text{ for some } \ell \xrightarrow{g, a, C := 0} \ell' \end{cases}$$

Model of priced timed automata [HSCC'01]



cost ≡ price

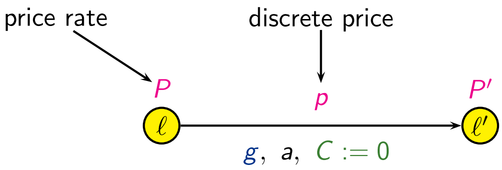
- a configuration: (ℓ, v)
- two kinds of transitions:

$$\begin{cases} (\ell, v) \xrightarrow{\delta(d)} (\ell, v+d) \\ (\ell, v) \xrightarrow{a} (\ell', v') \text{ where } \begin{cases} v \models g \\ v' = [C \leftarrow 0]v \end{cases} \text{ for some } \ell \xrightarrow{g, a, C:=0} \ell' \end{cases}$$

$$\text{Cost}\left((\ell, v) \xrightarrow{\delta(d)} (\ell, v+d)\right) = P.d \qquad \text{Cost}\left((\ell, v) \xrightarrow{a} (\ell', v')\right) = p$$

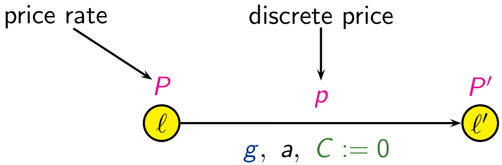
$$\text{Cost}(\rho) = \text{accumulated cost along run } \rho$$

Model of priced timed automata (cont.)



cost $\equiv$ price

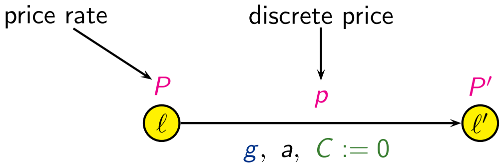
Model of priced timed automata (cont.)



cost ≡ price

- **one player problems:**
  - reachability with an optimization criterium on the price [BFH+01a,BFH+01b,LBB+01,ALTP01]
  - safety with a mean-cost optimization criterium [BBL04]

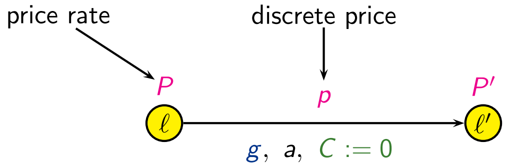
Model of priced timed automata (cont.)



cost ≡ price

- **one player problems:**
  - reachability with an optimization criterium on the price **[BFH+01a,BFH+01b,LBB+01,ALTP01]**
  - safety with a mean-cost optimization criterium **[BBL04]**
- **what if an opponent**?

Model of priced timed automata (cont.)



cost ≡ price

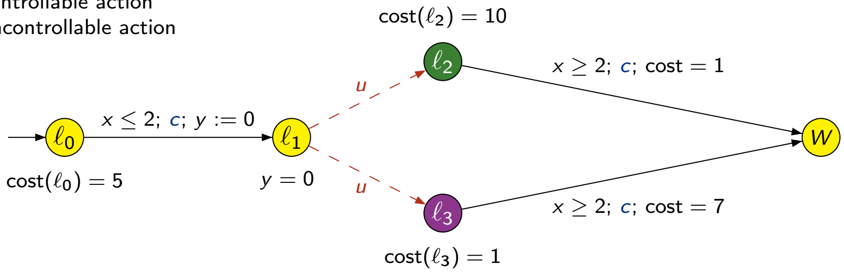
- **one player problems:**
  - reachability with an optimization criterium on the price **[BFH+01a,BFH+01b,LBB+01,ALTP01]**
  - safety with a mean-cost optimization criterium **[BBL04]**

- **what if an opponent**?

➜ **optimal reachability timed game**

An example

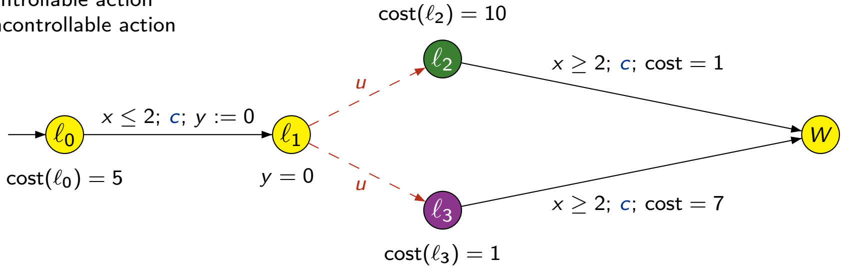
c: controllable action
u: uncontrollable action



Question: what is the optimal price we can ensure in state ℓ_0?

An example

c: controllable action
u: uncontrollable action

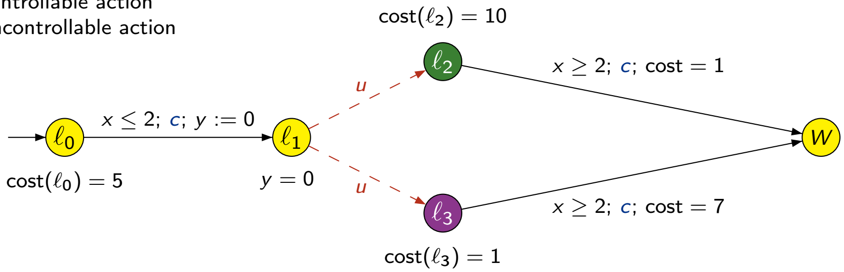


Question: what is the optimal price we can ensure in state ℓ_0?

$$5t + 10(2 - t) + 1$$

An example

c: controllable action
u: uncontrollable action

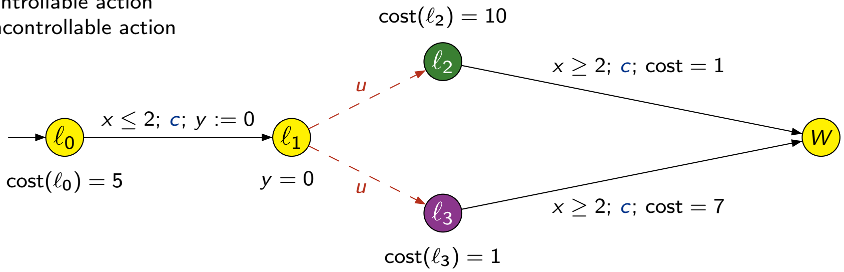


Question: what is the optimal price we can ensure in state ℓ_0?

$$5t + 10(2 - t) + 1 \ , \ 5t + (2 - t) + 7$$

An example

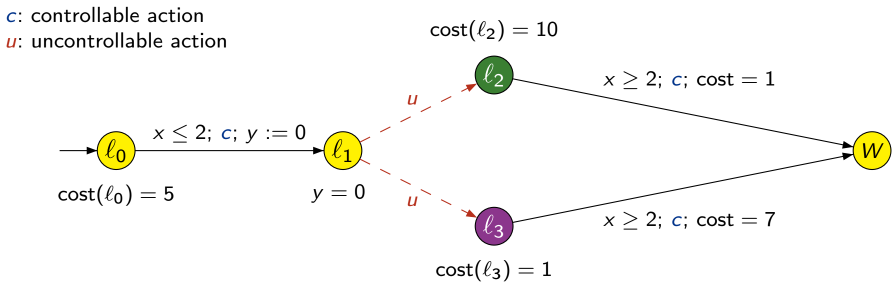
c: controllable action
u: uncontrollable action



Question: what is the optimal price we can ensure in state ℓ_0?

$$\max\ (\ 5t + 10(2-t) + 1\ ,\ 5t + (2-t) + 7\ )$$

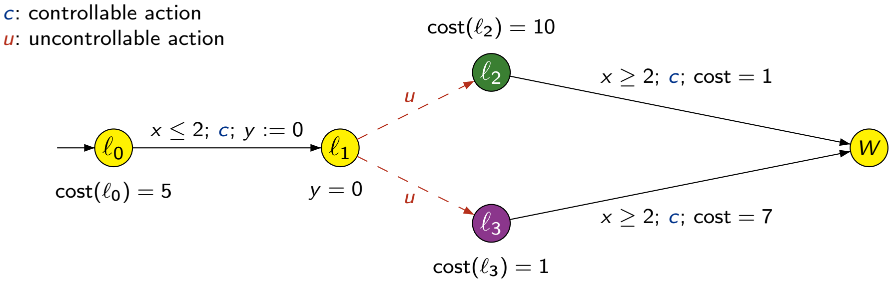
An example



Question: what is the optimal price we can ensure in state ℓ_0?

$$\inf_{0 \leq t \leq 2} \max\left(5t + 10(2-t) + 1 \,,\, 5t + (2-t) + 7 \right) = 14 + \frac{1}{3}$$

An example



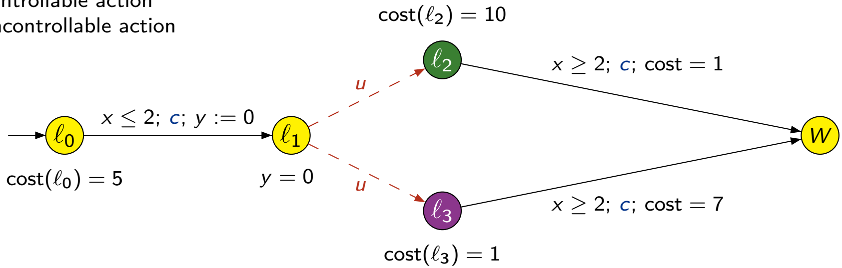
Question: what is the optimal price we can ensure in state ℓ_0?

$$\inf_{0 \le t \le 2} \max\left(5t + 10(2-t) + 1 \,,\; 5t + (2-t) + 7 \right) = 14 + \frac{1}{3}$$

➜ **strategy:** wait in ℓ_0, and when $t = \frac{4}{3}$, go to ℓ_1

An example

c: controllable action
u: uncontrollable action



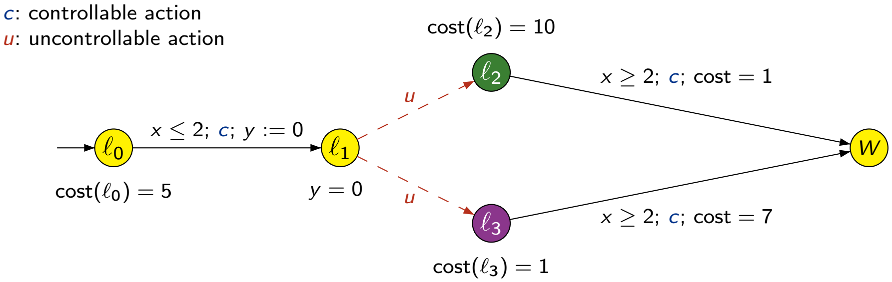
Question: what is the optimal price we can ensure in state ℓ_0?

$$\inf_{0 \leq t \leq 2} \max \left(5t + 10(2-t) + 1 \, , \, 5t + (2-t) + 7 \right) = 14 + \frac{1}{3}$$

➜ **strategy:** wait in ℓ_0, and when $t = \frac{4}{3}$, go to ℓ_1

- How to automatically compute such optimal prices?

An example



Question: what is the optimal price we can ensure in state ℓ_0?

$$\inf_{0 \leq t \leq 2} \max \left(5t + 10(2-t) + 1 \;,\; 5t + (2-t) + 7 \right) = 14 + \frac{1}{3}$$

➜ **strategy:** wait in ℓ_0, and when $t = \frac{4}{3}$, go to ℓ_1

- How to automatically compute such optimal prices?
- How to synthesize optimal strategies (if one exists)?

A hot topic!

- [La Torre, Mukhopadhyay, Murano – TCS 2002]:
    - case of acyclic games

A hot topic!

- **[La Torre, Mukhopadhyay, Murano – TCS 2002]**:
  - case of acyclic games
- **[Alur, Bernadsky, Madhusudan – ICALP 2004]**:
  - k-step games can be solved in exponential time and may need an exponential number of splittings
  - under a strongly non-*zeno* assumption, optimal cost is computable

A hot topic!

- **[La Torre, Mukhopadhyay, Murano – TCS 2002]**:
  - case of acyclic games
- **[Alur, Bernadsky, Madhusudan – ICALP 2004]**:
  - k-step games can be solved in exponential time and may need an exponential number of splittings
  - under a strongly non-*zeno* assumption, optimal cost is computable
- **[Bouyer, Cassez, Fleury, Larsen – FSTTCS 2004]**:
  - structural properties of strategies: may need memory, state-based strategies for a subclass of games
  - under a strongly non-*zeno* assumption, optimal cost is computable

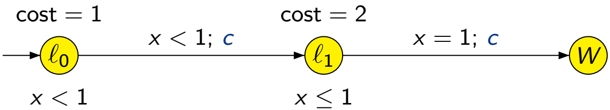
A hot topic!

- [La Torre, Mukhopadhyay, Murano – TCS 2002]:
  - case of acyclic games
- [Alur, Bernadsky, Madhusudan – ICALP 2004]:
  - k-step games can be solved in exponential time and may need an exponential number of splittings
  - under a strongly non-*zeno* assumption, optimal cost is computable
- [Bouyer, Cassez, Fleury, Larsen – FSTTCS 2004]:
  - structural properties of strategies: may need memory, state-based strategies for a subclass of games
  - under a strongly non-*zeno* assumption, optimal cost is computable
- [Brihaye, Bruyère, Raskin – FORMATS 2005]:
  - with five clocks, optimal cost is not computable!
  - with one clock and one stopwatch cost, optimal cost is computable

A hot topic!

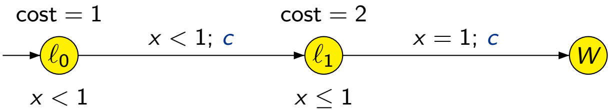
- [La Torre, Mukhopadhyay, Murano – TCS 2002]:
  - case of acyclic games
- [Alur, Bernadsky, Madhusudan – ICALP 2004]:
  - *k*-step games can be solved in exponential time and may need an exponential number of splittings
  - under a strongly non-*zeno* assumption, optimal cost is computable
- [Bouyer, Cassez, Fleury, Larsen – FSTTCS 2004]:
  - structural properties of strategies: may need memory, state-based strategies for a subclass of games
  - under a strongly non-*zeno* assumption, optimal cost is computable
- [Brihaye, Bruyère, Raskin – FORMATS 2005]:
  - with five clocks, optimal cost is not computable!
  - with one clock and one stopwatch cost, optimal cost is computable
- [Bouyer, Brihaye, Markey – Submitted to IPL, 2005]:
  - with three clocks, optimal cost is not computable

Do optimal strategies always exist?



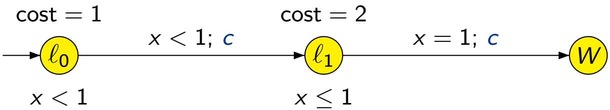
$$\begin{cases} f(\ell_0, x < 1) = \lambda \\ \\ f(\ell_1, x < 1) = \lambda \\ f(\ell_1, x = 1) = c \end{cases}$$

Do optimal strategies always exist?



$$\begin{cases} f(\ell_0, x<1) = \lambda \\ \\ f(\ell_1, x<1) = \lambda \\ f(\ell_1, x=1) = c \end{cases} \quad \rightsquigarrow \quad \begin{cases} f_\varepsilon(\ell_0, x<1-\varepsilon) = \lambda \\ f_\varepsilon(\ell_0, 1-\varepsilon \leq x < 1) = c \\ f_\varepsilon(\ell_1, x<1) = \lambda \\ f_\varepsilon(\ell_1, x=1) = c \end{cases}$$

Do optimal strategies always exist?

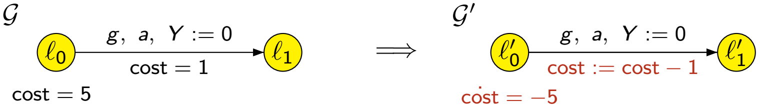


$$\begin{cases} f(\ell_0, x<1) = \lambda \\ \\ f(\ell_1, x<1) = \lambda \\ f(\ell_1, x=1) = c \end{cases} \quad \rightsquigarrow \quad \begin{cases} f_\varepsilon(\ell_0, x<1-\varepsilon) = \lambda \\ f_\varepsilon(\ell_0, 1-\varepsilon \leq x < 1) = c \\ f_\varepsilon(\ell_1, x<1) = \lambda \\ f_\varepsilon(\ell_1, x=1) = c \end{cases}$$

➜ **no optimal strategy exists**, but rather a family $(f_\varepsilon)_{\varepsilon>0}$ of ε-approximating strategies ($\text{cost}(f_\varepsilon) = 1+\varepsilon$)

An encoding

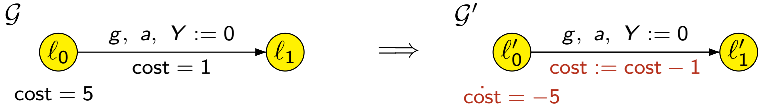
Idea: tranform the cost into a decreasing linear hybrid variable



Winning: W Winning: $W \wedge \text{cost} \geq 0$

An encoding

Idea: tranform the cost into a decreasing linear hybrid variable



Winning: W

Winning: $W \wedge \text{cost} \geq 0$

Theorem

For priced timed games (under some hypotheses),

$$\left.\begin{array}{r} \exists f \text{ winning strategy in } \mathcal{G} \\ \text{s.t. } \text{cost}(f,(\ell,v)) \leq \gamma \end{array}\right\} \iff (\ell, v, \text{cost} = \gamma) \text{ winning in } \mathcal{G}'$$

+ constructive proof

An encoding (2)

The set of winning states in $\mathcal{G}'$ is upward-closed for the cost, *i.e.* of the form

$$\bigcup_{i \in I} (P_i \wedge cost \succ_i k_i) \qquad \text{(with } \succ_i \text{ either } > \text{ or } \geq\text{)}$$

An encoding (2)

The set of winning states in $\mathcal{G}'$ is upward-closed for the cost, *i.e.* of the form

$$\bigcup_{i \in I} (P_i \wedge cost \succ_i k_i) \qquad \text{(with } \succ_i \text{ either } > \text{ or } \geq\text{)}$$

Corollary

For priced timed games (under some hypotheses),

- *"reachable" optimal cost, or not* (cost $\geq \gamma$ *or* cost $> \gamma$*)*
- *existence of an optimal strategy decidable*

+ constructive proof

An encoding (2)

The set of winning states in $\mathcal{G}'$ is upward-closed for the cost, *i.e.* of the form

$$\bigcup_{i \in I} (P_i \wedge cost \succ_i k_i) \qquad \text{(with } \succ_i \text{ either } > \text{ or } \geq\text{)}$$

Corollary

For priced timed games (under some hypotheses),

- *"reachable" optimal cost, or not (cost $\geq \gamma$ or cost $> \gamma$)*
- *existence of an optimal strategy decidable*

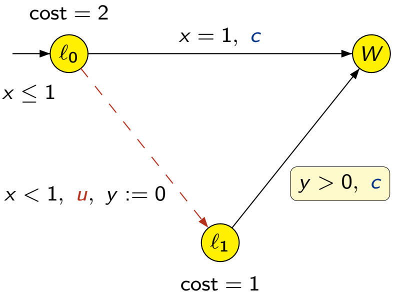
+ constructive proof

Nature of the strategy:

- state-based for the hybrid game, thus **cost-dependent** for the timed game
- cost-dependence is unavoidable in general!
- cost-independent strategies for syntactical restrictions of the games

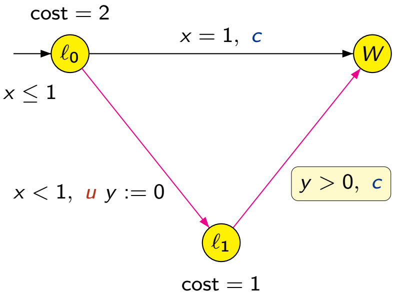
  c: large constraints, u: strict constraints

Cost-dependence is unavoidable



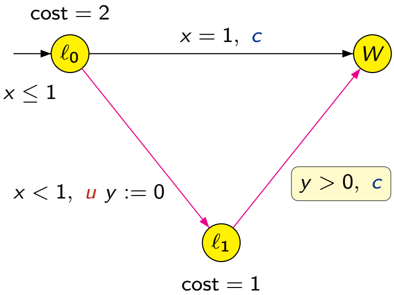
- **optimal cost:** 2
- **optimal strategy:**

Cost-dependence is unavoidable



- **optimal cost:** 2
- **optimal strategy:** if d is the time before a u occurs, and d' is the time waited in ℓ_1, the cost of the run is $2.d + d'$.

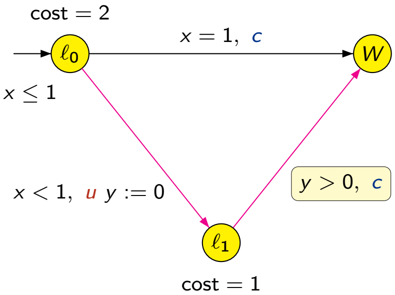
Cost-dependence is unavoidable



- **optimal cost:** 2
- **optimal strategy:** if d is the time before a u occurs, and d' is the time waited in ℓ_1, the cost of the run is $2.d + d'$.

$$2.d + d' \leq 2$$

Cost-dependence is unavoidable



- **optimal cost:** 2
- **optimal strategy:** if d is the time before a u occurs, and d' is the time waited in ℓ_1, the cost of the run is $2.d + d'$.

$$2.d + d' \leq 2$$

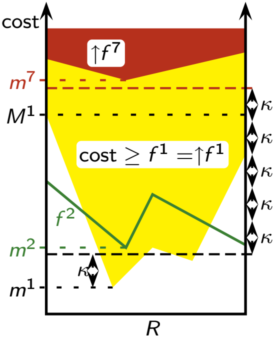
$$(\text{accumulated cost}) + d' \leq 2$$

Hypotheses for termination

- all clocks are bounded (not restrictive)
- the cost function is ***strictly non-zeno***
  → This condition is restrictive, but is decidable

Hypotheses for termination

- all clocks are bounded (not restrictive)
- the cost function is ***strictly non-zeno***
  → This condition is restrictive, but is decidable



Complexity bounds

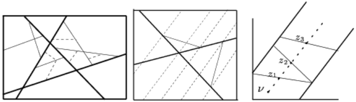
[Alur, Bernadsky, Madhusudan 2004]

➜ consider the k-step optimal game problem

Complexity bounds

[Alur, Bernadsky, Madhusudan 2004]
➜ consider the k-step optimal game problem

- can be solved in **exponential time**!

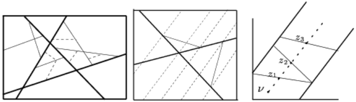


(a) A nested partition (b) A tube partition (c) An atomic tube

Complexity bounds

[Alur, Bernadsky, Madhusudan 2004]
➜ consider the k-step optimal game problem

- can be solved in **exponential time**!



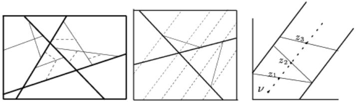
(a) A nested partition (b) A tube partition (c) An atomic tube

- within a region, an exponential number of splittings is sometimes necessary

Complexity bounds

[Alur, Bernadsky, Madhusudan 2004]
➜ consider the k-step optimal game problem

- can be solved in **exponential time**!



(a) A nested partition (b) A tube partition (c) An atomic tube

- within a region, an exponential number of splittings is sometimes necessary

Idea:

- environment chooses $r \in [0, 1]$,
- controller has to produce its binary encoding up to k digits

➜ Controller must have 2^k different strategies

Shape of the reduction

Original reduction: [Brihaye, Bruyère, Raskin 2005]
This reduction: [Bouyer, Brihaye, Markey 2005]

Shape of the reduction

Original reduction: [Brihaye, Bruyère, Raskin 2005]
This reduction: [Bouyer, Brihaye, Markey 2005]

Simulation of a two-counter machine:

- player 1 simulates the two-counter machine
- player 2 checks that player 1 does not cheat

Shape of the reduction

Original reduction: [Brihaye, Bruyère, Raskin 2005]
This reduction: [Bouyer, Brihaye, Markey 2005]

Simulation of a two-counter machine:

- player 1 simulates the two-counter machine
- player 2 checks that player 1 does not cheat

Encoding of the counters:

- counter c_1 is encoded by a clock x_1 s.t. $x_1 = \frac{1}{2^{c_1}}$
- counter c_2 is encoded by a clock x_2 s.t. $x_2 = \frac{1}{3^{c_2}}$
- x_1 and x_2 will be alternatively x, y or z

Shape of the reduction

Original reduction: [Brihaye, Bruyère, Raskin 2005]
This reduction: [Bouyer, Brihaye, Markey 2005]

Simulation of a two-counter machine:

- player 1 simulates the two-counter machine
- player 2 checks that player 1 does not cheat

Encoding of the counters:

- counter c_1 is encoded by a clock x_1 s.t. $x_1 = \frac{1}{2^{c_1}}$
- counter c_2 is encoded by a clock x_2 s.t. $x_2 = \frac{1}{3^{c_2}}$
- x_1 and x_2 will be alternatively x, y or z

The aim of player 1 is to win (reach a W-state) with cost ≤ 3,

Shape of the reduction

Original reduction: [Brihaye, Bruyère, Raskin 2005]
This reduction: [Bouyer, Brihaye, Markey 2005]

Simulation of a two-counter machine:

- player 1 simulates the two-counter machine
- player 2 checks that player 1 does not cheat

Encoding of the counters:

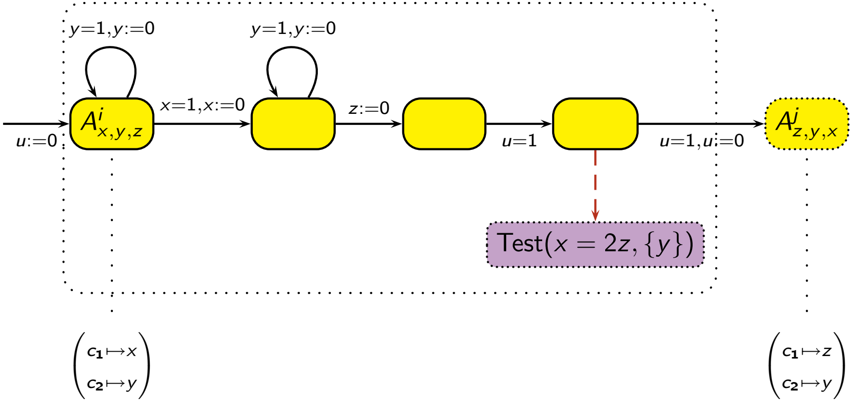
- counter c_1 is encoded by a clock x_1 s.t. $x_1 = \frac{1}{2^{c_1}}$
- counter c_2 is encoded by a clock x_2 s.t. $x_2 = \frac{1}{3^{c_2}}$
- x_1 and x_2 will be alternatively x, y or z

The aim of player 1 is to win (reach a W-state) with cost ≤ 3, and

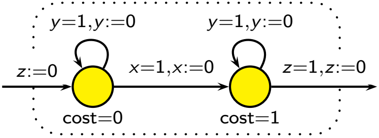
Player 1 has a winning strategy with cost ≤ 3 iff the two-counter machine halts

Simulation of an incrementation

Instruction i: $c_1 + +$; goto instruction j

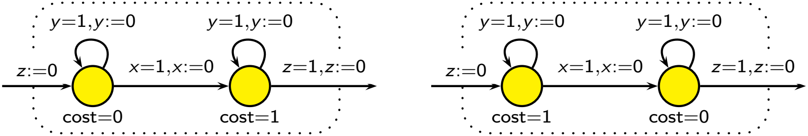


Adding x or $1 - x$ to the cost variable



The cost is increased by x_0

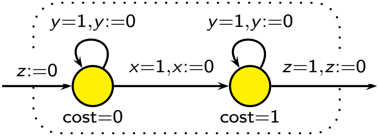
Adding x or $1 - x$ to the cost variable



The cost is increased by x_0 The cost is increased by $1 - x_0$

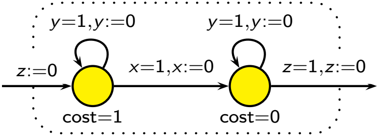
Adding x or $1 - x$ to the cost variable

$\mathrm{Add}^{+}(x, \{z\})$



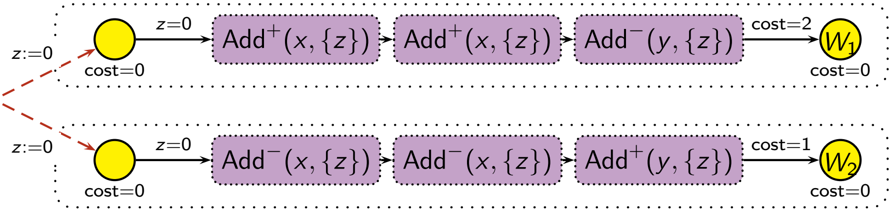
The cost is increased by x_0

$\mathrm{Add}^{-}(x, \{z\})$



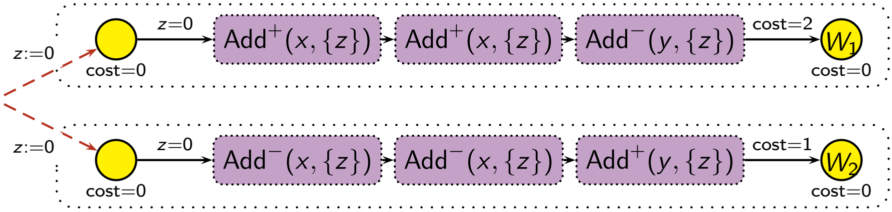
The cost is increased by $1 - x_0$

Checking $y = 2x$



In W_1, cost $= 2x_0 + (1 - y_0) + 2$.
In W_2, cost $= 2(1 - x_0) + y_0 + 1$.

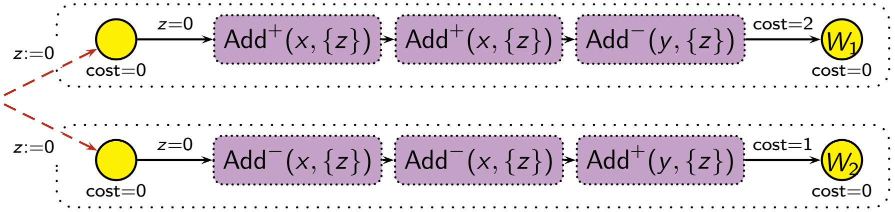
Checking $y = 2x$



In W_1, cost $= 2x_0 + (1 - y_0) + 2$.
In W_2, cost $= 2(1 - x_0) + y_0 + 1$.

- if $y_0 < 2x_0$, player 2 chooses the first branch: in W_1, cost > 3

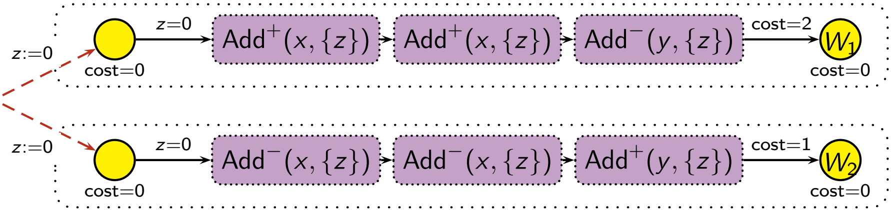
Checking $y = 2x$



In W_1, cost $= 2x_0 + (1 - y_0) + 2$.
In W_2, cost $= 2(1 - x_0) + y_0 + 1$.

- if $y_0 < 2x_0$, player 2 chooses the first branch: in W_1, cost > 3
- if $y_0 > 2x_0$, player 2 chooses the second branch: in W_2, cost > 3

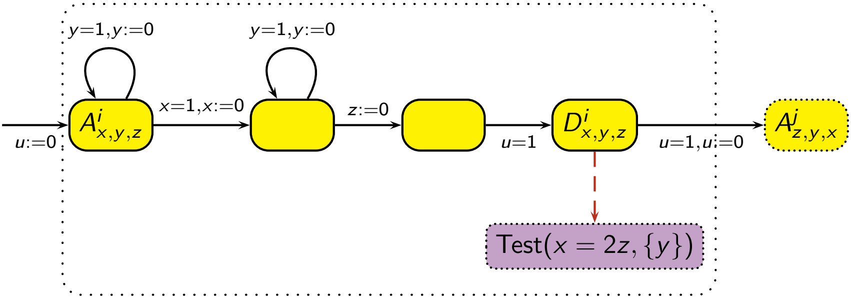
Checking $y = 2x$



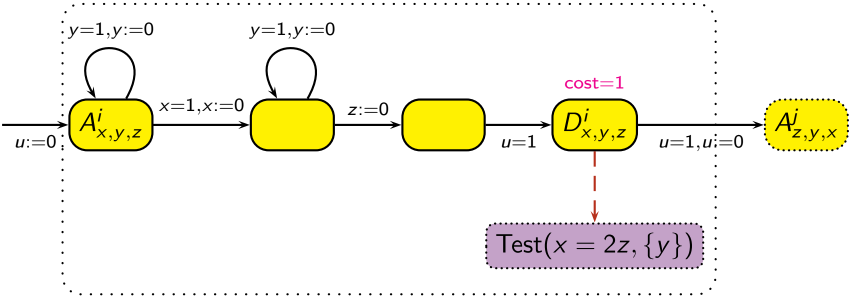
In W_1, cost $= 2x_0 + (1 - y_0) + 2$.
In W_2, cost $= 2(1 - x_0) + y_0 + 1$.

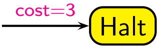
- if $y_0 < 2x_0$, player 2 chooses the first branch: in W_1, cost > 3
- if $y_0 > 2x_0$, player 2 chooses the second branch: in W_2, cost > 3
- if $y_0 = 2x_0$, in W_1 or in W_2, cost $= 3$.

How to get rid of tick clock u?



How to get rid of tick clock u?

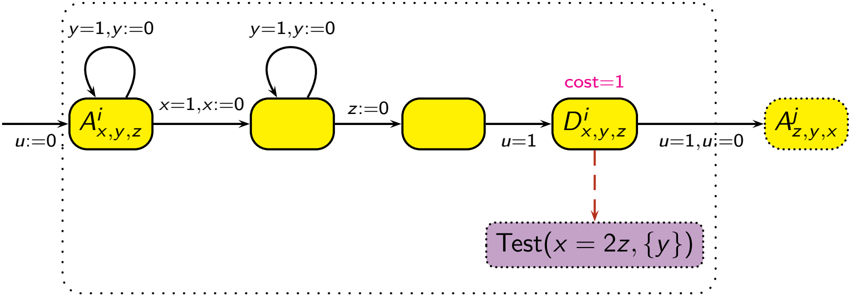




We will ensure that:

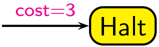
- no cost is accumulated in D-states

How to get rid of tick clock u?

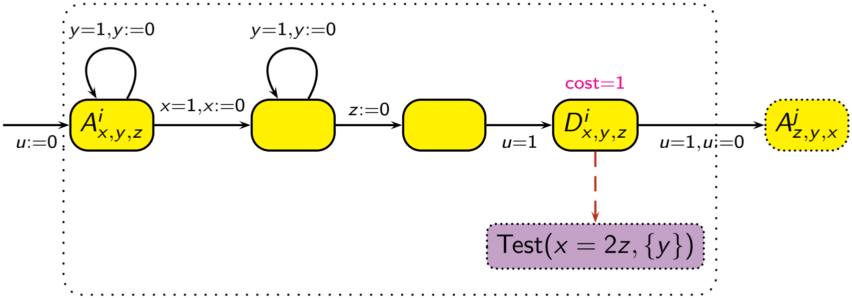


We will ensure that:

- no cost is accumulated in D-states
- the delay between the A-state and the D-state is 1 t.u.

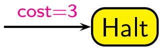


How to get rid of tick clock u?

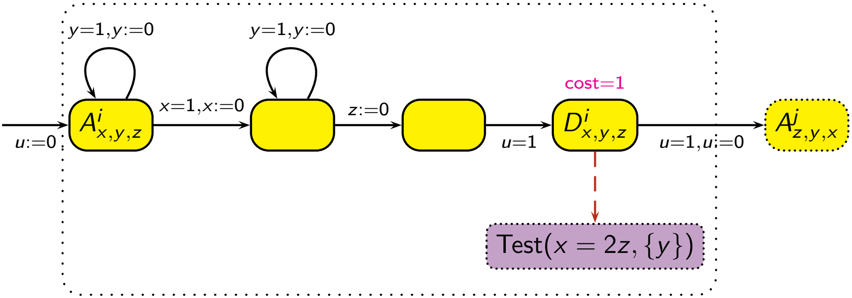


We will ensure that:

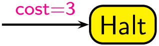
- no cost is accumulated in D-states
- the delay between the A-state and the D-state is 1 t.u.
  - the value of x in D is of the form $\frac{1}{2^n}$



How to get rid of tick clock u?

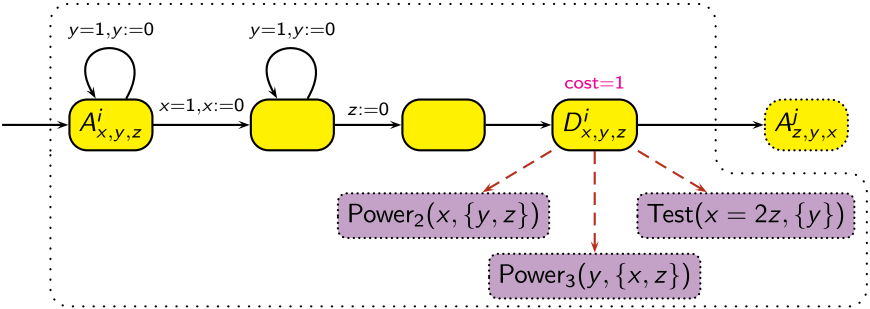


We will ensure that:

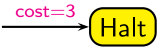


- no cost is accumulated in D-states
- the delay between the A-state and the D-state is 1 t.u.
  - the value of x in D is of the form $\frac{1}{2^n}$
  - the value of y in D is of the form $\frac{1}{3^m}$

How to get rid of tick clock u?

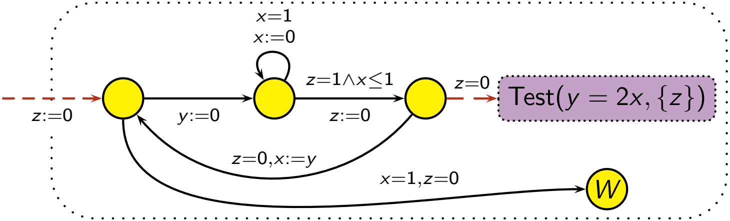


We will ensure that:



- no cost is accumulated in D-states
- the delay between the A-state and the D-state is 1 t.u.
  - the value of x in D is of the form $\frac{1}{2^n}$
  - the value of y in D is of the form $\frac{1}{3^m}$

Checking that x is of the form $\frac{1}{2^n}$



Conclusion

- Optimal cost is in general not computable in timed games.
- Under a strongly non-*zeno* hypothesis for the cost, optimal cost is computable.
- A much involved complexity bound for the number of splittings of regions.
- Properties of winning strategies.

Conclusion

- Optimal cost is in general not computable in timed games.
- Under a strongly non-*zeno* hypothesis for the cost, optimal cost is computable.
- A much involved complexity bound for the number of splittings of regions.
- Properties of winning strategies.

Further work

- Compute effectively ε-optimal winning strategies.
- Further understand this problem: provide decidable subclasses?
- And from an algorithmics point of view, what can be done? (integrate ideas from **[ABM04]** into encoding of **[BCFL04]**?)
- Adapt the forward algorithm of TiGA (game-extension of UPPAAL) **[CDFLL05]**.
- Mean-cost optimal safety timed games.

Bibliography

[ABM04] Alur, Bernadsky, Madhusudan. Optimal Reachability in Weighted Timed Games. ICALP'04 (LNCS 3142).

[BBM05] Bouyer, Brihaye, Markey. Improved Undecidability Results on Weighted Timed Automata. Submitted to IPL.

[BBR05] Brihaye, Bruyère, Raskin. On Optimal Timed Strategies. FORMATS'05 (to appear).

[BCFL04] Bouyer, Cassez, Fleury, Larsen. Optimal Strategies in Priced Timed Game Automata. FSTTCS'04 (LNCS 3328).

[BCFL05] Bouyer, Cassez, Fleury, Larsen. Synthesis of Optimal Strategies Using HyTech. GDV'04 (ENTCS 119(1)).

[LMM02] La Torre, Mukhopadhyay, Murano. Optimal-Reachability and Control for Acyclic Weighted Timed Automata. TCS'02 (IFIP 223).