Quantitative timed games

Patricia Bouyer

LSV – CNRS & ENS Cachan – France

Based on joint works with Thomas Brihaye, Ed Brinksma,
Véronique Bruyère, Uli Fahrenberg, Kim G. Larsen, Nicolas Markey,
Jean-François Raskin, Jiří Srba, and Jacob Illum Rasmussen

A (difficult) choice

When sending the title, I didn't know if I would speak about:

A (difficult) choice

When sending the title, I didn't know if I would speak about:

- timed automata with costs,

A (difficult) choice

When sending the title, I didn't know if I would speak about:

- timed automata with costs, or
- timed automata with probabilities

A (difficult) choice

When sending the title, I didn't know if I would speak about:

- timed automata with costs, or
- timed automata with probabilities

A (difficult) choice

When sending the title, I didn't know if I would speak about:

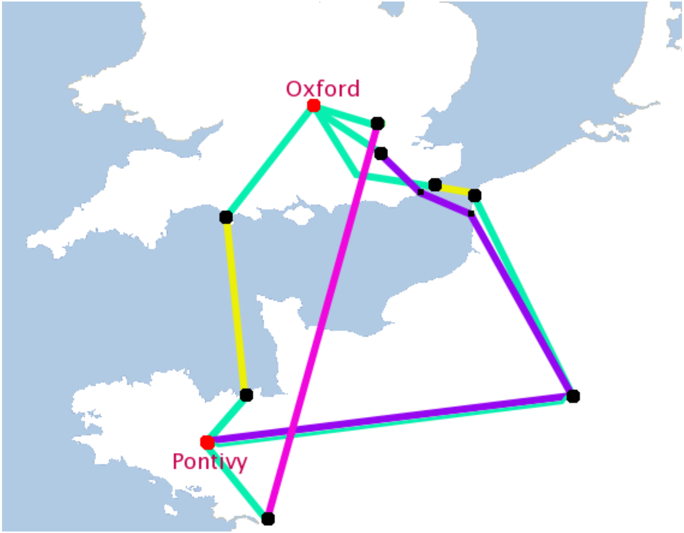
- timed automata with costs, or
- timed automata with probabilities

  $\leadsto$ talk of Vojtěch Forejt in the next session

Outline

1. Introduction
2. Weighted/priced timed automata
3. (Optimal) timed games
4. "Safe" timed games
5. Conclusion

A starting example



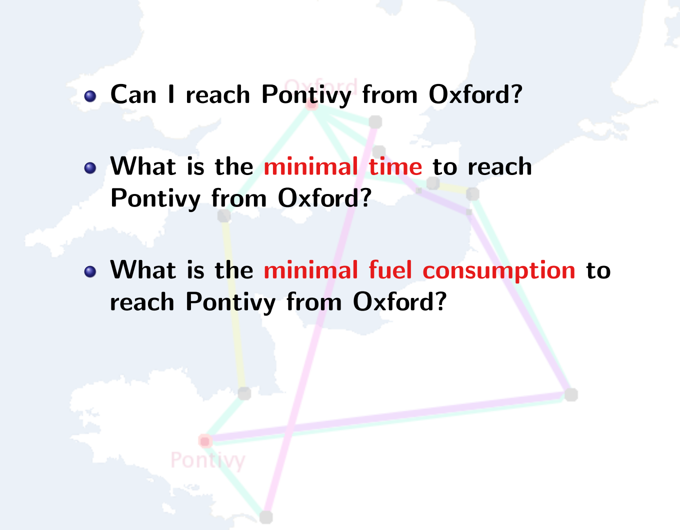
Natural questions

- **Can I reach Pontivy from Oxford?**

Natural questions

- **Can I reach Pontivy from Oxford?**
- **What is the minimal time to reach Pontivy from Oxford?**

Natural questions



- **Can I reach Pontivy from Oxford?**
- **What is the minimal time to reach Pontivy from Oxford?**
- **What is the minimal fuel consumption to reach Pontivy from Oxford?**

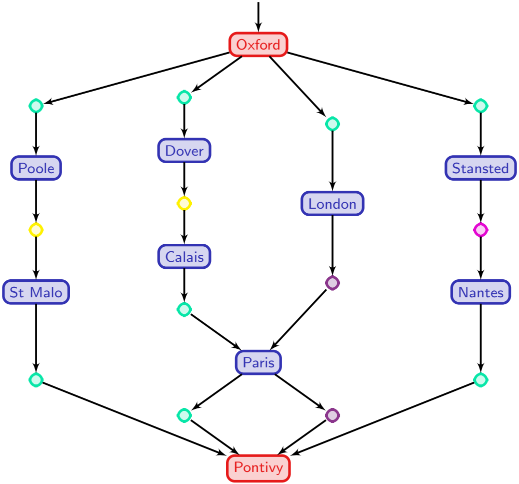
Natural questions

- **Can I reach Pontivy from Oxford?**
- **What is the minimal time to reach Pontivy from Oxford?**
- **What is the minimal fuel consumption to reach Pontivy from Oxford?**
- **What if there is an unexpected event?**

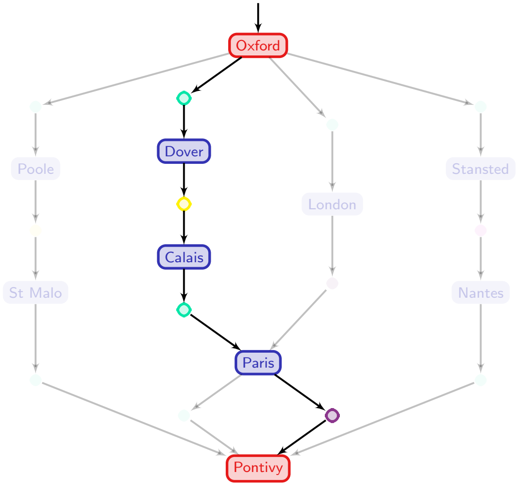
Natural questions

- **Can I reach Pontivy from Oxford?**
- **What is the minimal time to reach Pontivy from Oxford?**
- **What is the minimal fuel consumption to reach Pontivy from Oxford?**
- **What if there is an unexpected event?**
- **Can I use my computer all the way?**

A first model of the system

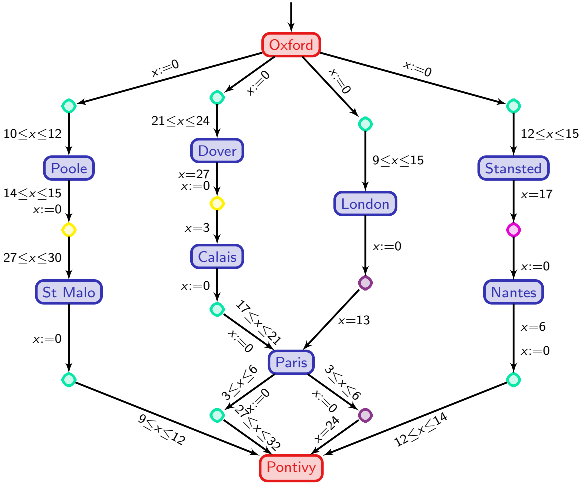


Can I reach Pontivy from Oxford?

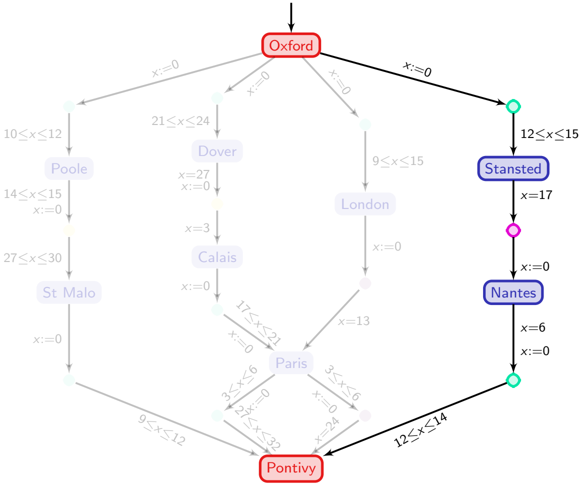


This is a reachability question in a finite graph: Yes, I can!

A second model of the system

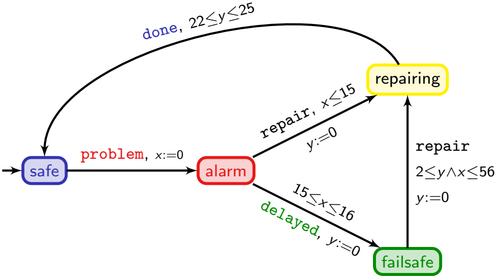


How long will that take?

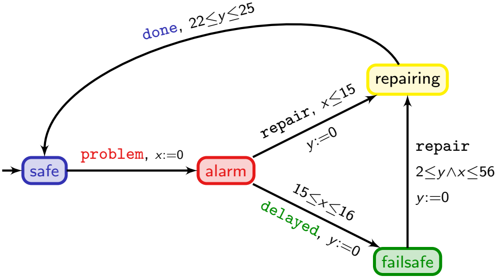


It is a reachability (and optimization) question
in a timed automaton: at least 350mn = 5h50mn!

An example of a timed automaton

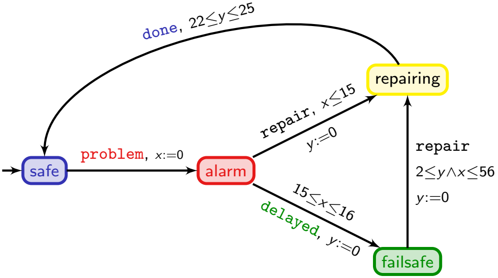


An example of a timed automaton



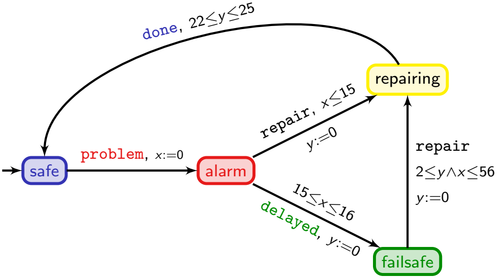
	safe
x	0
y	0

An example of a timed automaton



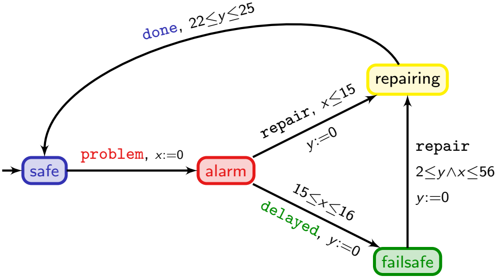
	safe	$\xrightarrow{23}$	safe
x	0		23
y	0		23

An example of a timed automaton



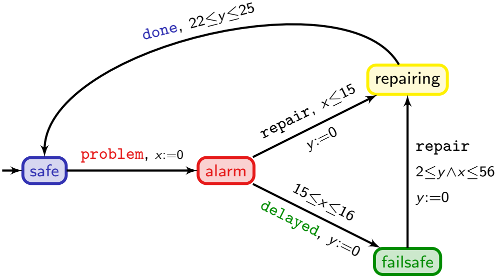
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm
x	0		23		0
y	0		23		23

An example of a timed automaton



	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm
x	0		23		0		15.6
y	0		23		23		38.6

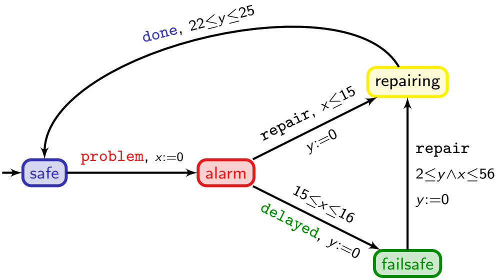
An example of a timed automaton



	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	
x	0		23		0		15.6		15.6	$\cdots$
y	0		23		23		38.6		0	

	failsafe
$\cdots$	15.6
	0

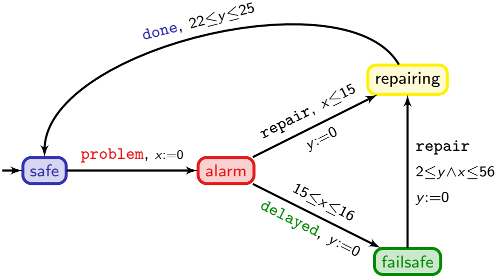
An example of a timed automaton



	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	
x	0		23		0		15.6		15.6	⋯
y	0		23		23		38.6		0	

	failsafe	$\xrightarrow{2.3}$	failsafe
⋯	15.6		17.9
	0		2.3

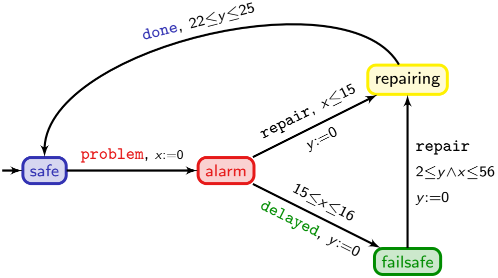
An example of a timed automaton



	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	
x	0		23		0		15.6		15.6	$\cdots$
y	0		23		23		38.6		0	

	failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\texttt{repair}}$	repairing
$\cdots$	15.6		17.9		17.9
	0		2.3		0

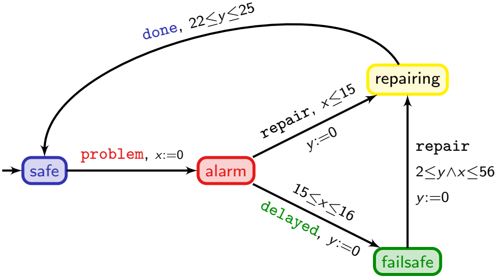
An example of a timed automaton



	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	
x	0		23		0		15.6		15.6	⋯
y	0		23		23		38.6		0	

	failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\texttt{repair}}$	repairing	$\xrightarrow{22.1}$	repairing
⋯	15.6		17.9		17.9		40
	0		2.3		0		22.1

An example of a timed automaton



	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm	$\xrightarrow{15.6}$	alarm	$\xrightarrow{\texttt{delayed}}$	failsafe	
x	0		23		0		15.6		15.6	⋯
y	0		23		23		38.6		0	

	failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\texttt{repair}}$	repairing	$\xrightarrow{22.1}$	repairing	$\xrightarrow{\texttt{done}}$	safe
⋯	15.6		17.9		17.9		40		40
	0		2.3		0		22.1		22.1

Timed automata

Theorem [AD90]

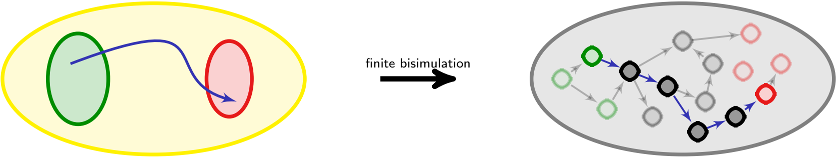
The reachability problem is decidable (and PSPACE-complete) for timed automata.

[AD90] Alur, Dill. Automata for modeling real-time systems *(ICALP'90)*.

Timed automata

Theorem [AD90]

The reachability problem is decidable (and PSPACE-complete) for timed automata.

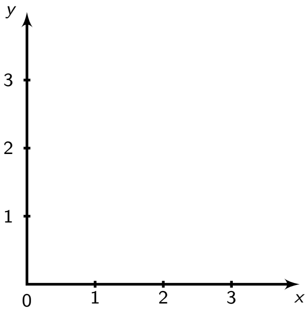


timed automaton

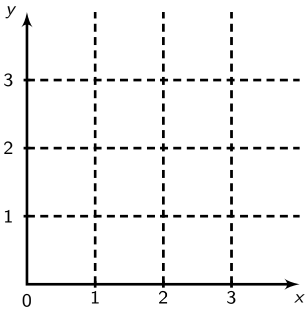
large (but finite) automaton
(region automaton)

[AD90] Alur, Dill. Automata for modeling real-time systems *(ICALP'90)*.

The region abstraction

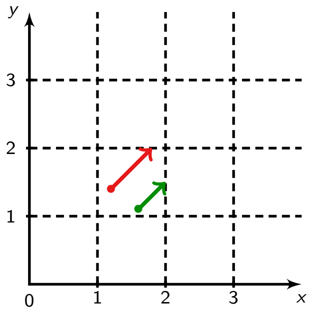


The region abstraction



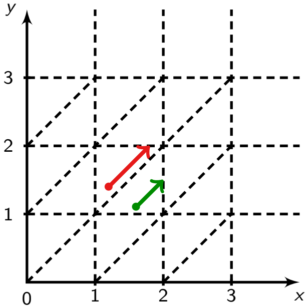
- "compatibility" between regions and constraints

The region abstraction



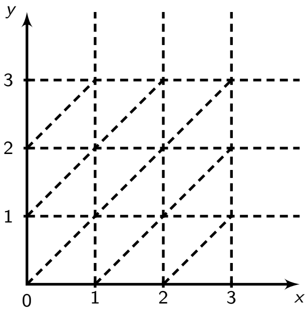
- "compatibility" between regions and constraints
- "compatibility" between regions and time elapsing

The region abstraction



- "compatibility" between regions and constraints
- "compatibility" between regions and time elapsing

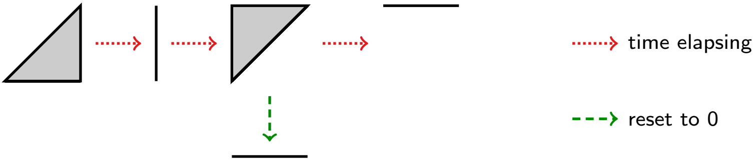
The region abstraction



- "compatibility" between regions and constraints
- "compatibility" between regions and time elapsing

$\rightsquigarrow$ an equivalence of finite index
a time-abstract bisimulation

The region abstraction



Time-optimal reachability

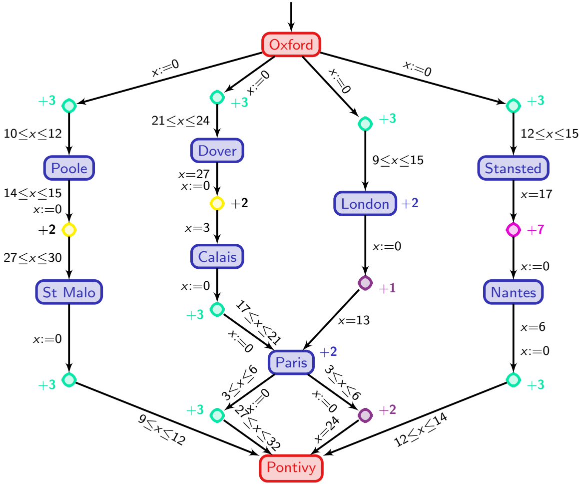
Theorem [CY92]

The time-optimal reachability problem is decidable (and PSPACE-complete) for timed automata.

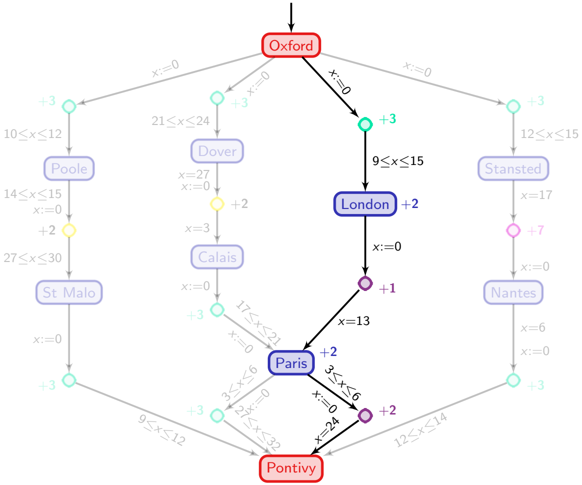
[CY92] Courcoubetis, Yannakakis. Minimum and maximum delay problems in real-time systems *(Formal Methods in System Design)*.

Outline

A third model of the system

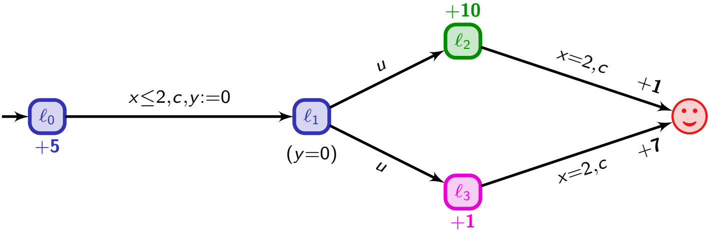


How much fuel will I use?



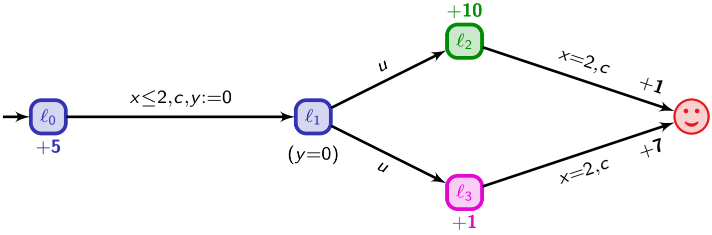
It is a quantitative (optimization) problem
in a priced/weighted timed automaton: at least 68 anti-planet units!

HSCC'01: weighted/priced timed automata



[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

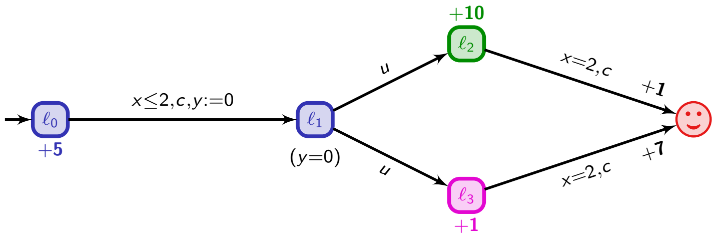
HSCC'01: weighted/priced timed automata



$$
\begin{array}{ccccccccccc}
 & \ell_0 & \xrightarrow{1.3} & \ell_0 & \xrightarrow{c} & \ell_1 & \xrightarrow{u} & \ell_3 & \xrightarrow{0.7} & \ell_3 & \xrightarrow{c} & ☺ \\
x & 0 & & 1.3 & & 1.3 & & 1.3 & & 2 & & \\
y & 0 & & 1.3 & & 0 & & 0 & & 0.7 & &
\end{array}
$$

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata

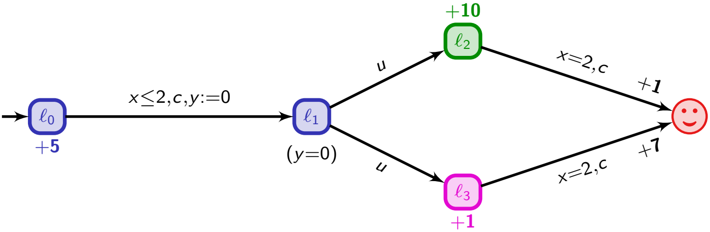


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

cost :

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata

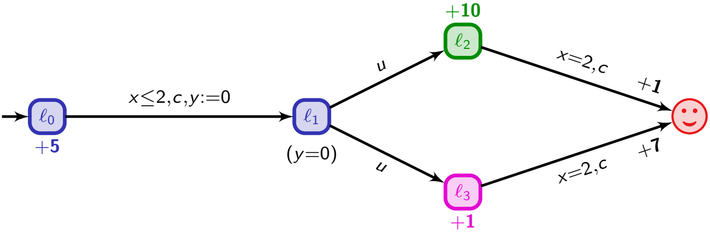


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

cost : 6.5

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata

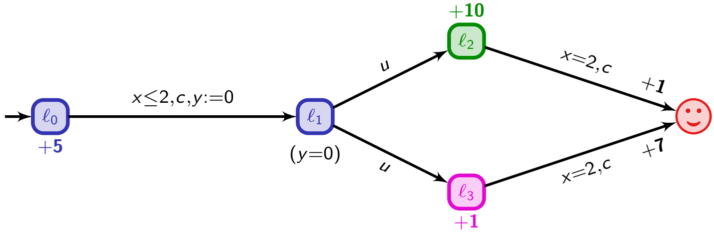


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

cost : $6.5 \quad + \quad 0$

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata

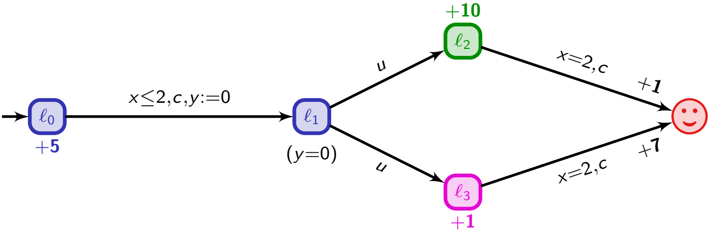


$$
\begin{array}{ccccccccccccc}
 & \ell_0 & \xrightarrow{1.3} & \ell_0 & \xrightarrow{c} & \ell_1 & \xrightarrow{u} & \ell_3 & \xrightarrow{0.7} & \ell_3 & \xrightarrow{c} & ☺ \\
x & 0 & & 1.3 & & 1.3 & & 1.3 & & 2 & & \\
y & 0 & & 1.3 & & 0 & & 0 & & 0.7 & &
\end{array}
$$

cost : $6.5 + 0 + 0$

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

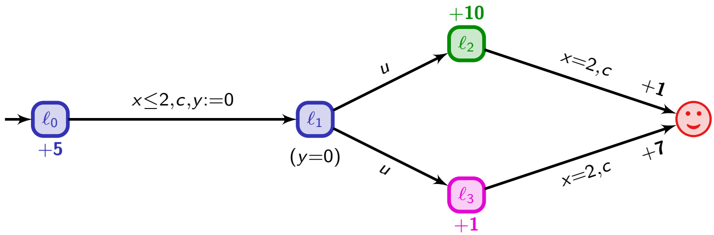
HSCC'01: weighted/priced timed automata



$$\begin{array}{ccccccccccccc}
 & \ell_0 & \xrightarrow{1.3} & \ell_0 & \xrightarrow{c} & \ell_1 & \xrightarrow{u} & \ell_3 & \xrightarrow{0.7} & \ell_3 & \xrightarrow{c} & \text{☺} \\
x & 0 & & 1.3 & & 1.3 & & 1.3 & & 2 & & \\
y & 0 & & 1.3 & & 0 & & 0 & & 0.7 & & \\
\text{cost :} & & 6.5 & + & 0 & + & 0 & + & 0.7 & & &
\end{array}$$

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

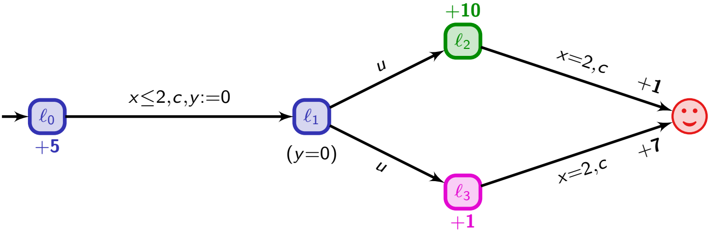
HSCC'01: weighted/priced timed automata



	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		
cost :		6.5	+	0	+	0	+	0.7	+	7	

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata

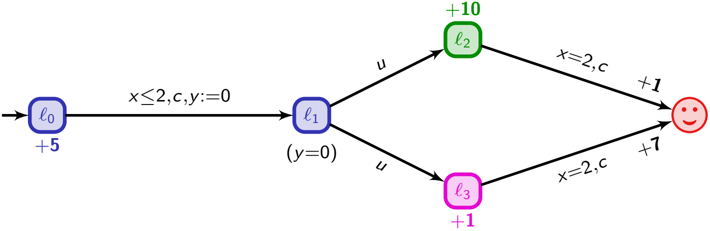


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

cost : $6.5 + 0 + 0 + 0.7 + 7 = 14.2$

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

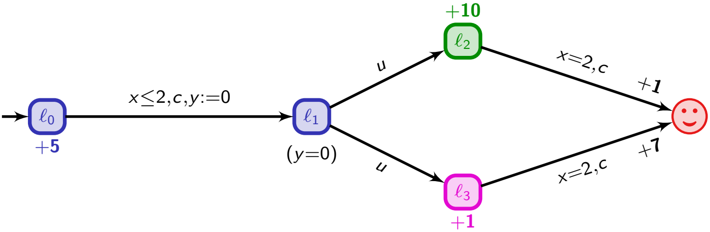
HSCC'01: weighted/priced timed automata



Question: what is the optimal cost for reaching 🙂 ?

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata

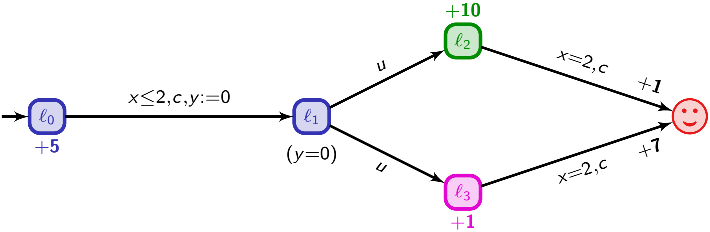


Question: what is the optimal cost for reaching ☺ ?

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

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata

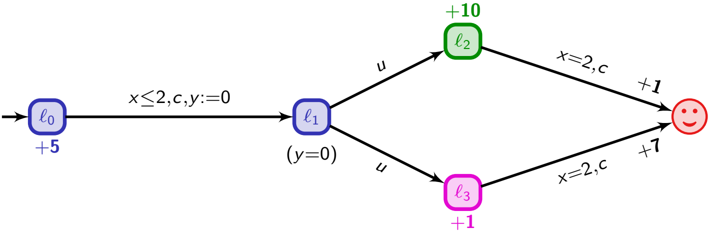


Question: what is the optimal cost for reaching ☺?

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

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata

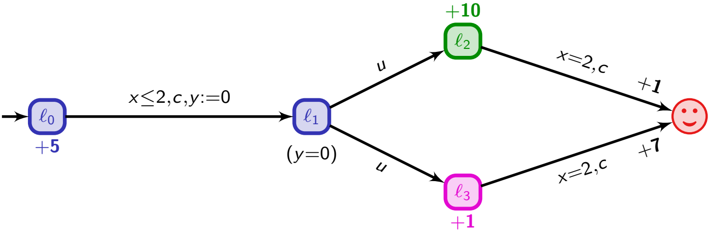


Question: what is the optimal cost for reaching 🙂?

$$\min \left(5t + 10(2-t) + 1 \; , \; 5t + (2-t) + 7 \right)$$

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata

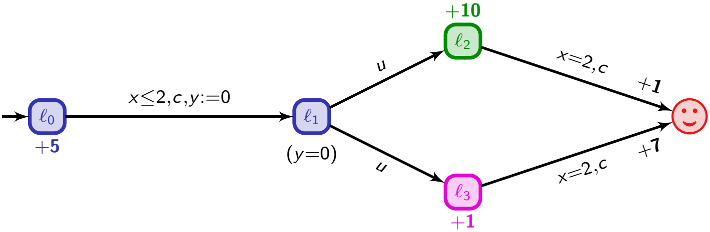


Question: what is the optimal cost for reaching ☺?

$$\inf_{0\leq t\leq 2} \min\left(5t + 10(2-t) + 1 \;,\; 5t + (2-t) + 7 \right) = 9$$

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

HSCC'01: weighted/priced timed automata



Question: what is the optimal cost for reaching ☺?

$$\inf_{0\leq t\leq 2} \min\left(5t + 10(2-t) + 1 \,,\; 5t + (2-t) + 7 \right) = 9$$

➜ **strategy:** leave immediately ℓ_0, go to ℓ_3, and wait there 2 t.u.

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

Optimal reachability

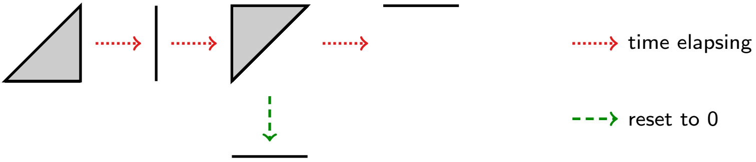
The idea "go through corners" extends in the general case.

Theorem [ALP01,BFH+01,BBBR07]

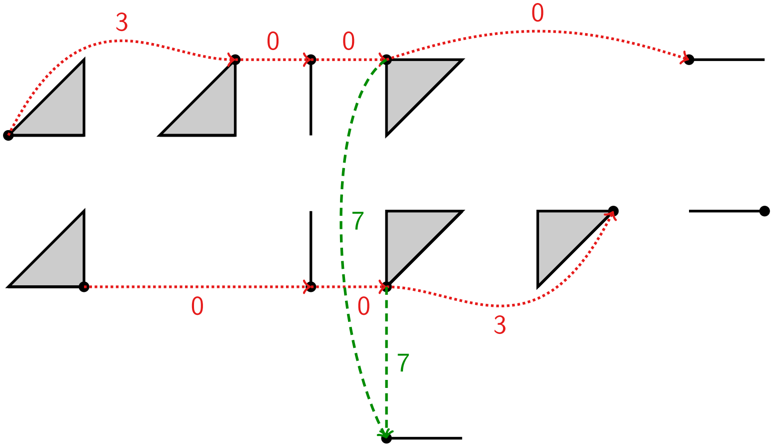
Optimal reachability is decidable (and PSPACE-complete) in timed automata.

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.
[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem *(Formal Methods in System Design)*.

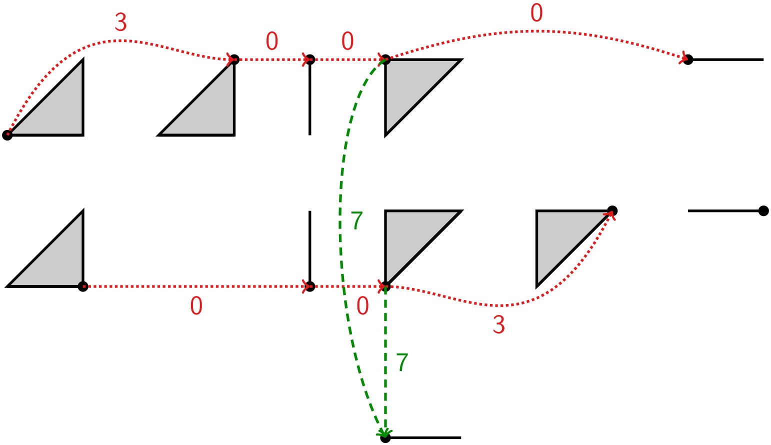
The region abstraction is not fine enough



The corner-point abstraction



The corner-point abstraction



We can somehow discretize the behaviours...

From timed to discrete behaviours

Optimal reachability as a linear programming problem

From timed to discrete behaviours

Optimal reachability as a linear programming problem

$$\circ \xrightarrow{t_1} \circ \xrightarrow{t_2} \circ \xrightarrow{t_3} \circ \xrightarrow{t_4} \circ \xrightarrow{t_5} \circ \quad \cdots$$

From timed to discrete behaviours

Optimal reachability as a linear programming problem

$$\circ \xrightarrow{t_1} \circ \xrightarrow{t_2} \circ \xrightarrow{t_3} \circ \xrightarrow{t_4} \circ \xrightarrow{t_5} \circ \quad \cdots \qquad \left\{ \begin{array}{l} t_i \leq t_{i+1} \\ \\ \\ \end{array} \right.$$

From timed to discrete behaviours

Optimal reachability as a linear programming problem

$$\circ \xrightarrow{t_1} \circ \xrightarrow[x\leq 2]{t_2} \circ \xrightarrow{t_3} \circ \xrightarrow{t_4} \circ \xrightarrow{t_5} \circ \quad \cdots \qquad \begin{cases} t_i \leq t_{i+1} \\ t_2 \leq 2 \\ \end{cases}$$

From timed to discrete behaviours

Optimal reachability as a linear programming problem

$$\circ \xrightarrow[y:=0]{t_1} \circ \xrightarrow[x\leq 2]{t_2} \circ \xrightarrow{t_3} \circ \xrightarrow[y\geq 5]{t_4} \circ \xrightarrow{t_5} \circ \quad \cdots$$

$$\begin{cases} t_i \leq t_{i+1} \\ t_2 \leq 2 \\ t_4 - t_1 \geq 5 \end{cases}$$

From timed to discrete behaviours

Optimal reachability as a linear programming problem

$$\circ \xrightarrow[y:=0]{t_1} \circ \xrightarrow[x\leq 2]{t_2} \circ \xrightarrow{t_3} \circ \xrightarrow[y\geq 5]{t_4} \circ \xrightarrow{t_5} \circ \quad \cdots \qquad \begin{cases} t_i \leq t_{i+1} \\ t_2 \leq 2 \\ t_4 - t_1 \geq 5 \end{cases}$$

Lemma

Let Z be a bounded zone and f be a function

$$f : (t_1, \ldots, t_n) \mapsto \sum_{i=1}^{n} c_i t_i + c$$

well-defined on $\overline{Z}$. Then $inf_Z f$ is obtained on the border of $\overline{Z}$ with integer coordinates.

From timed to discrete behaviours

Optimal reachability as a linear programming problem

$$\circ \xrightarrow[y:=0]{t_1} \circ \xrightarrow[x\leq 2]{t_2} \circ \xrightarrow{t_3} \circ \xrightarrow[y\geq 5]{t_4} \circ \xrightarrow{t_5} \circ \quad \cdots$$

$$\begin{cases} t_i \leq t_{i+1} \\ t_2 \leq 2 \\ t_4 - t_1 \geq 5 \end{cases}$$

Lemma

Let Z be a bounded zone and f be a function

$$f : (t_1, \ldots, t_n) \mapsto \sum_{i=1}^{n} c_i t_i + c$$

well-defined on $\overline{Z}$. Then $inf_Z f$ is obtained on the border of $\overline{Z}$ with integer coordinates.

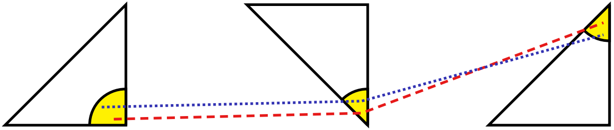
➜ for every finite path π in $\mathcal{A}$, there exists a path Π in $\mathcal{A}_{\mathrm{cp}}$ such that

$$\mathrm{cost}(\Pi) \leq \mathrm{cost}(\pi)$$

[Π is a "corner-point projection" of π]

From discrete to timed behaviours

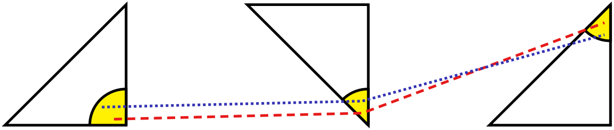
Approximation of abstract paths:



For any path Π of $\mathcal{A}_{\text{cp}}$,

From discrete to timed behaviours

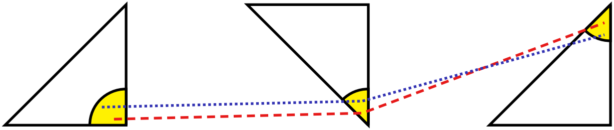
Approximation of abstract paths:



For any path Π of $\mathcal{A}_{\mathsf{cp}}$, for any $\varepsilon > 0$,

From discrete to timed behaviours

Approximation of abstract paths:

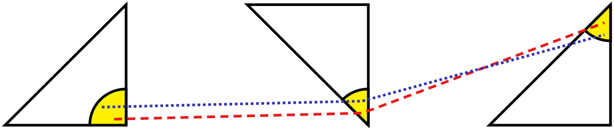


For any path Π of $\mathcal{A}_{\mathsf{cp}}$, for any $\varepsilon > 0$, there exists a path π_ε of $\mathcal{A}$ s.t.

$$\|\Pi - \pi_\varepsilon\|_\infty < \varepsilon$$

From discrete to timed behaviours

Approximation of abstract paths:



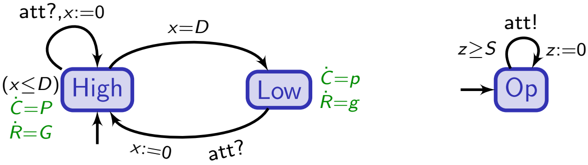
For any path Π of $\mathcal{A}_{\mathsf{cp}}$, for any $\varepsilon > 0$, there exists a path π_ε of $\mathcal{A}$ s.t.

$$\|\Pi - \pi_\varepsilon\|_\infty < \varepsilon$$

For every $\eta > 0$, there exists $\varepsilon > 0$ s.t.

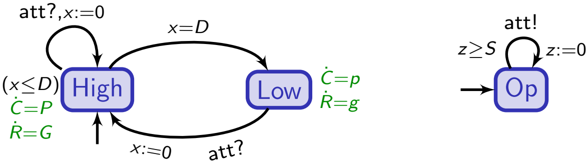
$$\|\Pi - \pi_\varepsilon\|_\infty < \varepsilon \Rightarrow |\mathsf{cost}(\Pi) - \mathsf{cost}(\pi_\varepsilon)| < \eta$$

Going further 1: mean-cost optimization



[BBL08] Bouyer, Brinksma, Larsen. Optimal infinite scheduling for multi-priced timed automata *(Formal Methods in System Designs)*.

Going further 1: mean-cost optimization

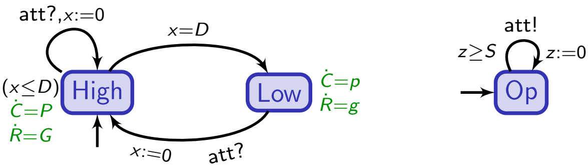


$\rightsquigarrow$ compute optimal infinite schedules that minimize

$$\text{mean-cost}(\pi) = \limsup_{n \to +\infty} \frac{\text{cost}(\pi_n)}{\text{reward}(\pi_n)}$$

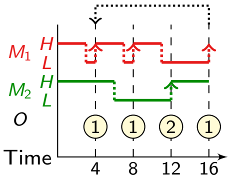
[BBL08] Bouyer, Brinksma, Larsen. Optimal infinite scheduling for multi-priced timed automata *(Formal Methods in System Designs)*.

Going further 1: mean-cost optimization

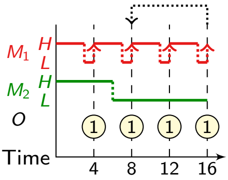


$\rightsquigarrow$ compute optimal infinite schedules that minimize

$$\text{mean-cost}(\pi) = \limsup_{n \to +\infty} \frac{\text{cost}(\pi_n)}{\text{reward}(\pi_n)}$$



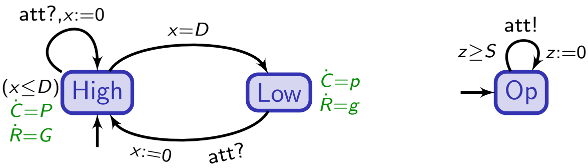
Schedule with ratio $\approx$1.455



Schedule with ratio $\approx$1.478

[BBL08] Bouyer, Brinksma, Larsen. Optimal infinite scheduling for multi-priced timed automata *(Formal Methods in System Designs)*.

Going further 1: mean-cost optimization



$\rightsquigarrow$ compute optimal infinite schedules that minimize

$$\text{mean-cost}(\pi) = \limsup_{n \to +\infty} \frac{\text{cost}(\pi_n)}{\text{reward}(\pi_n)}$$

Theorem [BBL08]

The mean-cost optimization problem is decidable (and PSPACE-complete) for priced timed automata.

$\rightsquigarrow$ the corner-point abstraction can be used

[BBL08] Bouyer, Brinksma, Larsen. Optimal infinite scheduling for multi-priced timed automata *(Formal Methods in System Designs)*.

Mean-cost optimization: from timed to discrete behaviours

- **Finite behaviours:** based on the following property

Lemma

Let Z be a bounded zone and f be a function

$$f : (t_1, \ldots, t_n) \mapsto \frac{\sum_{i=1}^{n} c_i t_i + c}{\sum_{i=1}^{n} r_i t_i + r}$$

well-defined on $\overline{Z}$. Then $inf_Z f$ is obtained on the border of $\overline{Z}$ with integer coordinates.

Mean-cost optimization: from timed to discrete behaviours

- **Finite behaviours:** based on the following property

Lemma

Let Z be a bounded zone and f be a function

$$f : (t_1, \ldots, t_n) \mapsto \frac{\sum_{i=1}^{n} c_i t_i + c}{\sum_{i=1}^{n} r_i t_i + r}$$

well-defined on $\overline{Z}$. Then $inf_Z f$ is obtained on the border of $\overline{Z}$ with integer coordinates.

$\leadsto$ for every finite path π in $\mathcal{A}$, there exists a path Π in $\mathcal{A}_{\text{cp}}$ such that

$$\text{mean-cost}(\Pi) \leq \text{mean-cost}(\pi)$$

Mean-cost optimization: from timed to discrete behaviours

- **Finite behaviours:** based on the following property

Lemma

Let Z be a bounded zone and f be a function

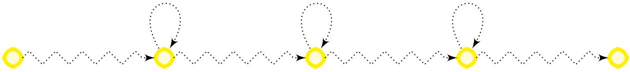
$$f : (t_1, \ldots, t_n) \mapsto \frac{\sum_{i=1}^{n} c_i t_i + c}{\sum_{i=1}^{n} r_i t_i + r}$$

well-defined on $\overline{Z}$. Then $\mathit{inf}_Z f$ is obtained on the border of $\overline{Z}$ with integer coordinates.

$\rightsquigarrow$ for every finite path π in $\mathcal{A}$, there exists a path Π in $\mathcal{A}_{\mathrm{cp}}$ such that

$$\text{mean-cost}(\Pi) \leq \text{mean-cost}(\pi)$$

- **Infinite behaviours:** decompose each sufficiently long projection into cycles



The linear part will be negligible!

Mean-cost optimization: from timed to discrete behaviours

- **Finite behaviours:** based on the following property

Lemma

Let Z be a bounded zone and f be a function

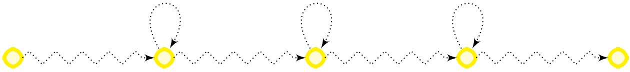
$$f : (t_1, \ldots, t_n) \mapsto \frac{\sum_{i=1}^{n} c_i t_i + c}{\sum_{i=1}^{n} r_i t_i + r}$$

well-defined on $\overline{Z}$. Then $\mathit{inf}_Z f$ is obtained on the border of $\overline{Z}$ with integer coordinates.

$\rightsquigarrow$ for every finite path π in $\mathcal{A}$, there exists a path Π in $\mathcal{A}_{\mathsf{cp}}$ such that

$$\text{mean-cost}(\Pi) \leq \text{mean-cost}(\pi)$$

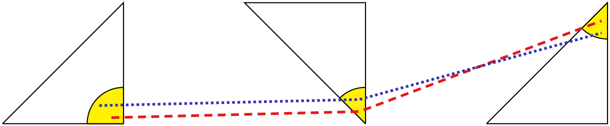
- **Infinite behaviours:** decompose each sufficiently long projection into cycles



The linear part will be negligible!

$\rightsquigarrow$ the optimal cycle of $\mathcal{A}_{\mathsf{cp}}$ is better than any infinite path of $\mathcal{A}$!

Mean-cost optimization: from discrete to timed behaviours



For any path Π of $\mathcal{A}_{\mathsf{cp}}$, for any $\varepsilon > 0$, there exists a path π_ε of $\mathcal{A}$ s.t.

$$\|\Pi - \pi_\varepsilon\|_\infty < \varepsilon$$

For every $\eta > 0$, there exists $\varepsilon > 0$ s.t.

$$\|\Pi - \pi_\varepsilon\|_\infty < \varepsilon \Rightarrow |\text{mean-cost}(\Pi) - \text{mean-cost}(\pi_\varepsilon)| < \eta$$

Going further 2: concavely-priced cost functions

$\rightsquigarrow$ A general abstract framework for quantitative timed systems

Theorem [JT08]

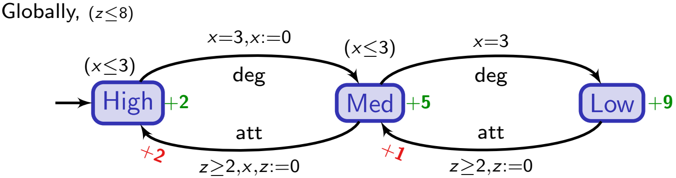
Optimal cost in concavely-priced timed automata is computable, if we restrict to quasi-concave price functions. For the following cost functions, the (decision) problem is even PSPACE-complete:

- optimal-time and optimal-cost reachability;
- optimal discrete discounted cost;
- optimal average-time and average-cost;
- optimal mean-cost.

$\rightsquigarrow$ a slight extension of the corner-point abstraction can be used

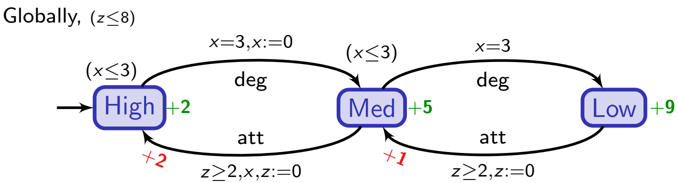
[JT08] Judziński, Trivedi. Concavely-priced timed automata *(FORMATS'08)*.

Going further 3: discounted-time cost optimization



[FL08] Fahrenberg, Larsen. Discount-optimal infinite runs in priced timed automata *(INFINITY'08)*.

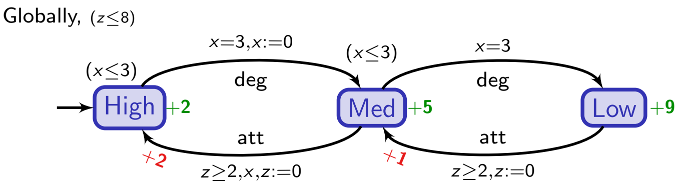
Going further 3: discounted-time cost optimization



$\leadsto$ compute optimal infinite schedules that minimize
discounted cost over time

[FL08] Fahrenberg, Larsen. Discount-optimal infinite runs in priced timed automata *(INFINITY'08)*.

Going further 3: discounted-time cost optimization



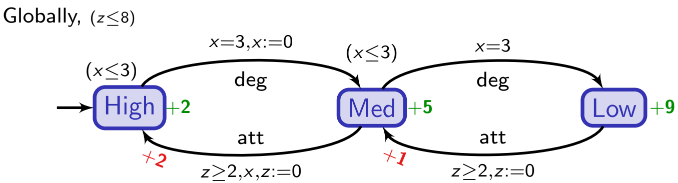
$\rightsquigarrow$ compute optimal infinite schedules that minimize

$$\text{discounted-cost}_{\lambda}(\pi) = \sum_{n \geq 0} \lambda^{T_n} \int_{t=0}^{\tau_{n+1}} \lambda^t \text{cost}(\ell_n)\,\mathrm{d}t + \lambda^{T_{n+1}} \text{cost}(\ell_n \xrightarrow{a_{n+1}} \ell_{n+1})$$

if $\pi = (\ell_0, v_0) \xrightarrow{\tau_1, a_1} (\ell_1, v_1) \xrightarrow{\tau_2, a_2} \cdots$ and $T_n = \sum_{i \leq n} \tau_i$

[FL08] Fahrenberg, Larsen. Discount-optimal infinite runs in priced timed automata *(INFINITY'08)*.

Going further 3: discounted-time cost optimization

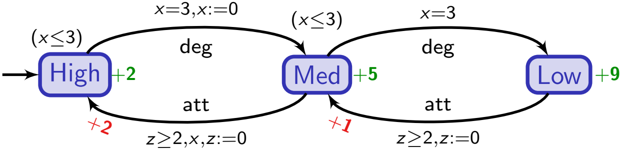


$\rightsquigarrow$ compute optimal infinite schedules that minimize
discounted cost over time

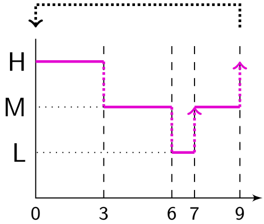
[FL08] Fahrenberg, Larsen. Discount-optimal infinite runs in priced timed automata *(INFINITY'08)*.

Going further 3: discounted-time cost optimization

Globally, $(z \leq 8)$



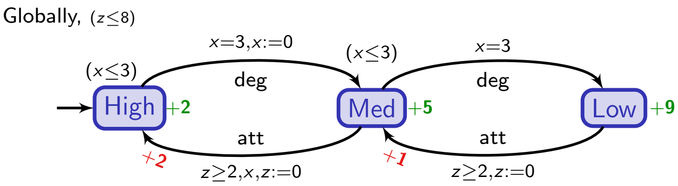
$\rightsquigarrow$ compute optimal infinite schedules that minimize
discounted cost over time



if $\lambda = e^{-1}$, the discounted cost of that infinite schedule is ≈ 2.16

[FL08] Fahrenberg, Larsen. Discount-optimal infinite runs in priced timed automata *(INFINITY'08)*.

Going further 3: discounted-time cost optimization



$\rightsquigarrow$ compute optimal infinite schedules that minimize
discounted cost over time

Theorem [FL08]

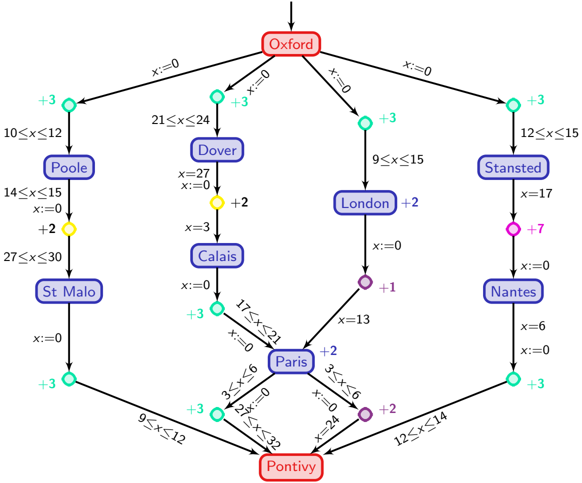
The optimal discounted cost is computable in EXPTIME in priced timed automata.

$\rightsquigarrow$ the corner-point abstraction can be used

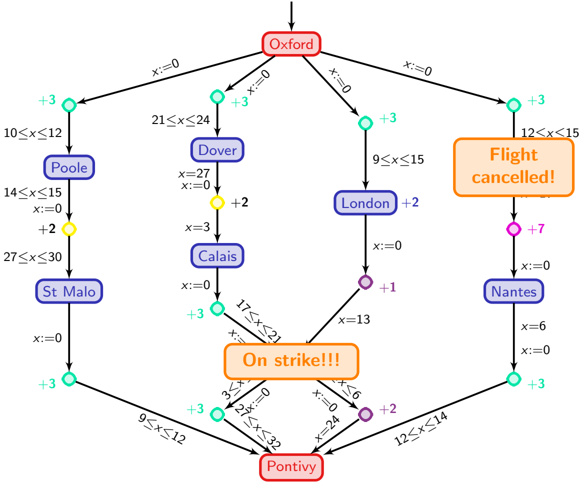
[FL08] Fahrenberg, Larsen. Discount-optimal infinite runs in priced timed automata *(INFINITY'08)*.

Outline

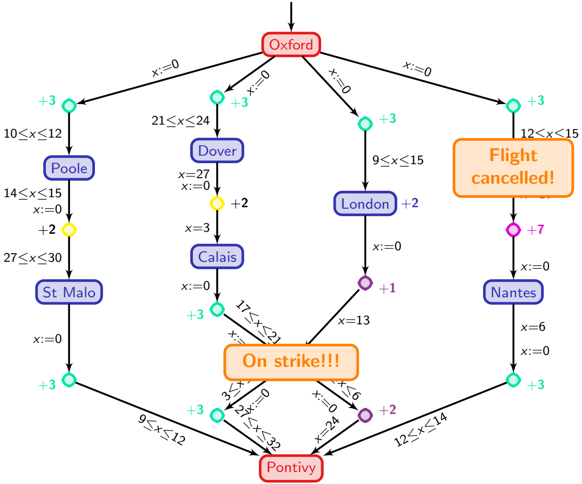
What if an unexpected event happens?



What if an unexpected event happens?

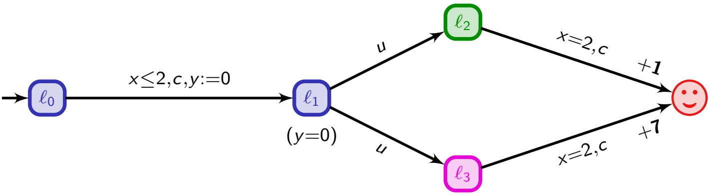


What if an unexpected event happens?

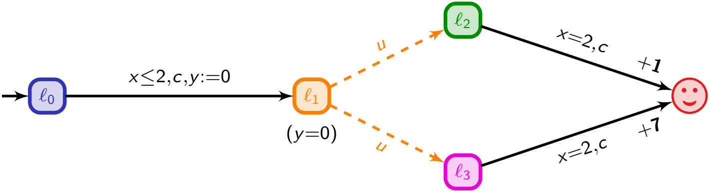


⤳ modelled as timed games

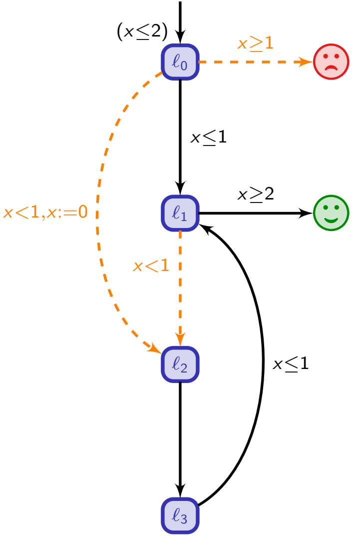
A simple example of timed game



A simple example of timed game



Another example



Decidability of timed games

Theorem [AMPS98,HK99]

Safety and reachability control in timed automata are decidable and EXPTIME-complete.

[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]

Safety and reachability control in timed automata are decidable and EXPTIME-complete.

(the attractor is computable...)

[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]

Safety and reachability control in timed automata are decidable and EXPTIME-complete.

(the attractor is computable...)

$\rightsquigarrow$ 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]

Safety and reachability control in timed automata are decidable and EXPTIME-complete.

(the attractor is computable...)

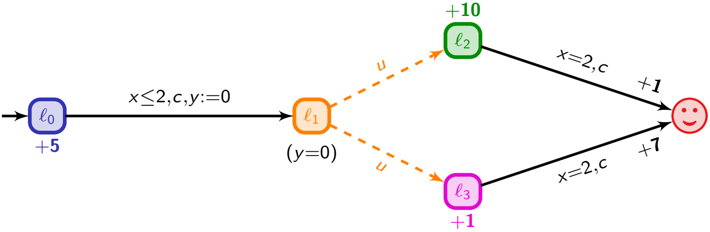
$\rightsquigarrow$ classical regions are sufficient for solving such problems

Theorem [AM99,BHPR07,JT07]

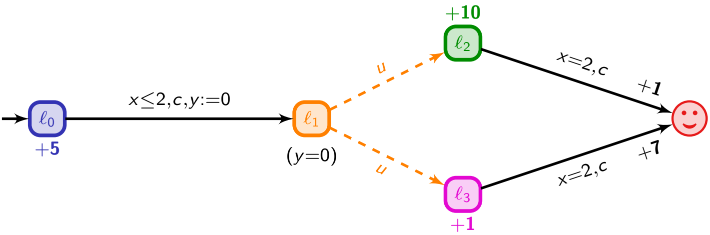
Optimal-time reachability timed games are decidable and EXPTIME-complete.

[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] Jurdzinński, Trivedi. Reachability-time games on timed automata *(ICALP'07)*.

Back to the first simple example

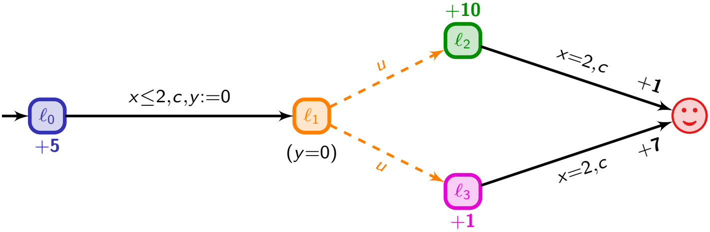


Back to the first simple example



Question: what is the optimal cost we can ensure from ℓ_0?

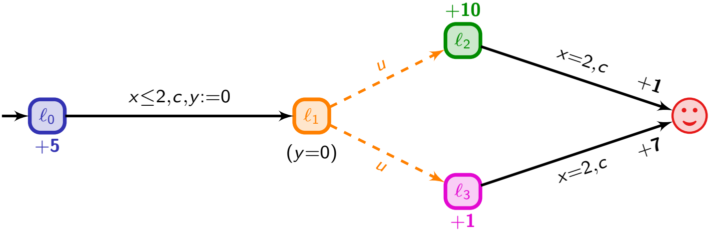
Back to the first simple example



Question: what is the optimal cost we can ensure from ℓ_0?

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

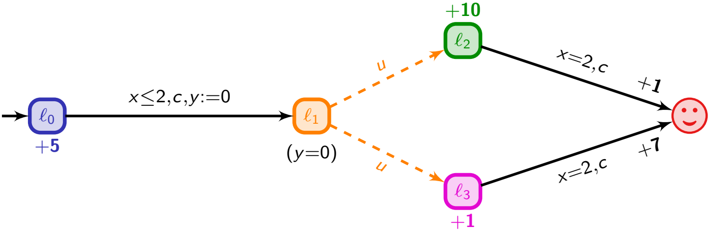
Back to the first simple example



Question: what is the optimal cost we can ensure from ℓ_0?

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

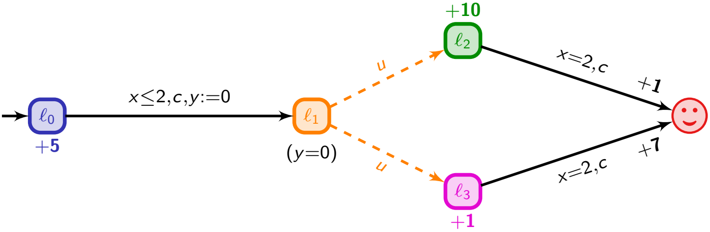
Back to the first simple example



Question: what is the optimal cost we can ensure from ℓ_0?

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

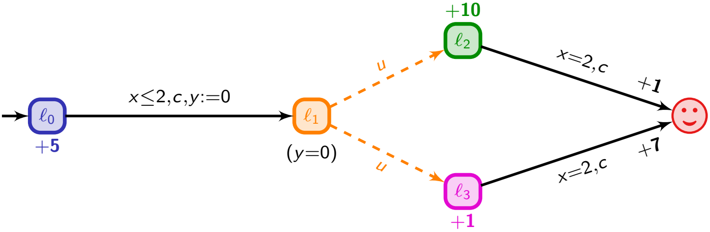
Back to the first simple example



Question: what is the optimal cost we can ensure from ℓ_0?

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

Back to the first simple example

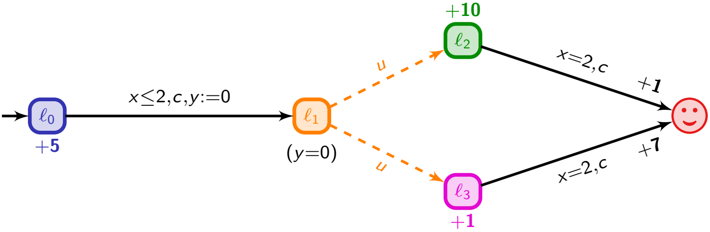


Question: what is the optimal cost we can ensure from ℓ_0?

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

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

Back to the first simple example



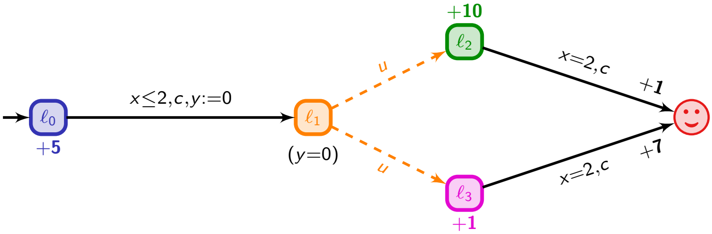
Question: what is the optimal cost we can ensure from ℓ_0?

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

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

- How to automatically compute such optimal costs?

Back to the first simple example



Question: what is the optimal cost we can ensure from ℓ_0?

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

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

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

Results

This topic has been fairly hot these last couple of years...

e.g. [LMM02,ABM04,BCFL04]

[LMM02] La Torre, Mukhopadhyay, Murano. Optimal-reachability and control for acyclic weighted timed automata *(TCS@02)*.
[ABM04] Alur, Bernardsky, Madhusudan. Optimal reachability in weighted timed games *(ICALP'04)*.
[BCFL04] Bouyer, Cassez, Fleury, Larsen. Optimal strategies in priced timed game automata *(FSTTCS'04)*.

Results

This topic has been fairly hot these last couple of years...

e.g. [LMM02,ABM04,BCFL04]

Theorem [BBR05,BBM06]

Optimal timed games are undecidable, as soon as automata have three clocks or more.

[BBR05] Brihaye, Bruyère, Raskin. On optimal timed strategies *(FORMATS'05)*.
[BBM06] Bouyer, Brihaye, Markey. Improved undecidability results on weighted timed automata *(Information Processing Letters)*.
[BLMR06] Bouyer, Larsen, Markey, Rasmussen. Almost-optimal strategies in one-clock priced timed automata *(FSTTCS'06)*.

Results

This topic has been fairly hot these last couple of years...

e.g. [LMM02,ABM04,BCFL04]

Theorem [BBR05,BBM06]

Optimal timed games are undecidable, as soon as automata have three clocks or more.

Theorem [BLMR06]

Turn-based optimal timed games are decidable in 3EXPTIME when automata have a single clock. They are P-hard.

[BBR05] Brihaye, Bruyère, Raskin. On optimal timed strategies *(FORMATS'05)*.
[BBM06] Bouyer, Brihaye, Markey. Improved undecidability results on weighted timed automata *(Information Processing Letters)*.
[BLMR06] Bouyer, Larsen, Markey, Rasmussen. Almost-optimal strategies in one-clock priced timed automata *(FSTTCS'06)*.

The positive side

Theorem [BLMR06]

Turn-based optimal timed games are decidable in 3EXPTIME when automata have a single clock. They are P-hard.

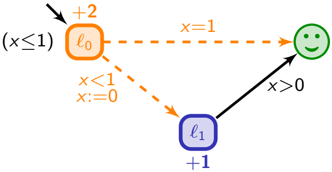
- Key: resetting the clock somehow resets the history...

The positive side

Theorem [BLMR06]

Turn-based optimal timed games are decidable in 3EXPTIME when automata have a single clock. They are P-hard.

- Key: resetting the clock somehow resets the history...
- Memoryless strategies can be non-optimal...

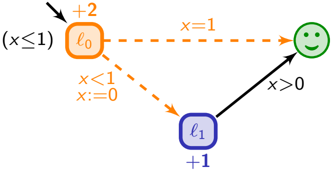


The positive side

Theorem [BLMR06]

Turn-based optimal timed games are decidable in 3EXPTIME when automata have a single clock. They are P-hard.

- Key: resetting the clock somehow resets the history...
- Memoryless strategies can be non-optimal...



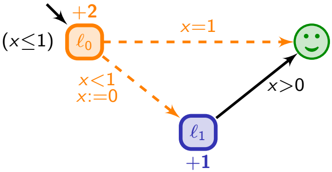
- However, we can synthesize memoryless almost-optimal winning strategies.

The positive side

Theorem [BLMR06]

Turn-based optimal timed games are decidable in 3EXPTIME when automata have a single clock. They are P-hard.

- Key: resetting the clock somehow resets the history...
- Memoryless strategies can be non-optimal...



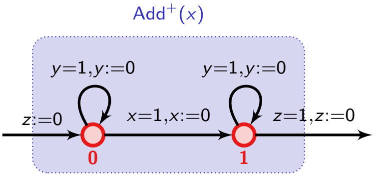
- However, we can synthesize memoryless almost-optimal winning strategies.
- Rather involved proof (by unfolding and removing one by one locations) of correctness for a simple algorithm.

The negative side: why is that hard?

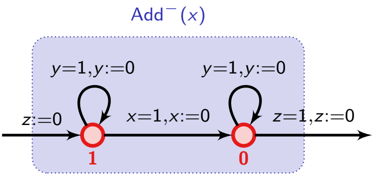
Given two clocks x and y, we can check whether $y = 2x$.

The negative side: why is that hard?

Given two clocks x and y, we can check whether $y = 2x$.



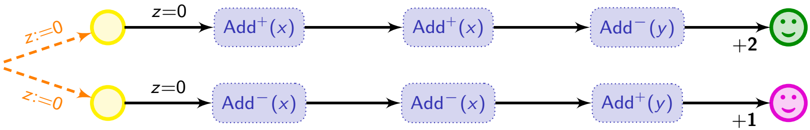
The cost is increased by x_0



The cost is increased by $1-x_0$

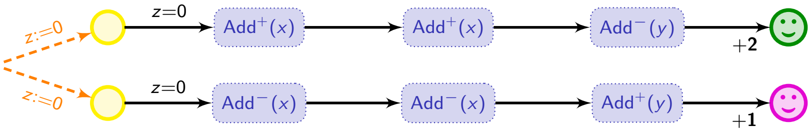
The negative side: why is that hard?

Given two clocks x and y, we can check whether $y = 2x$.



The negative side: why is that hard?

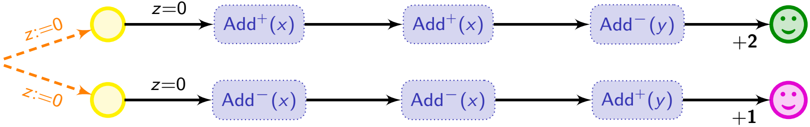
Given two clocks x and y, we can check whether $y = 2x$.



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

The negative side: why is that hard?

Given two clocks x and y, we can check whether $y = 2x$.

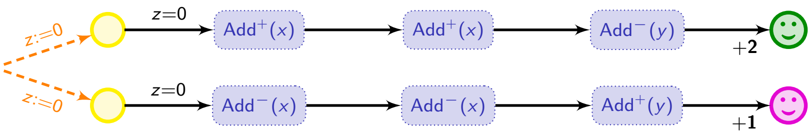


- In , cost $= 2x_0 + (1 - y_0) + 2$

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

The negative side: why is that hard?

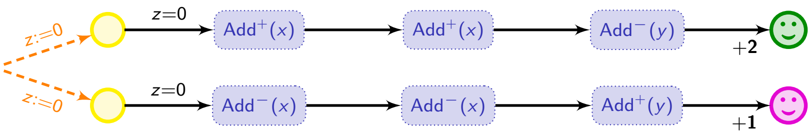
Given two clocks x and y, we can check whether $y = 2x$.



- In , cost $= 2x_0 + (1 - y_0) + 2$
  In , cost $= 2(1 - x_0) + y_0 + 1$
- if $y_0 < 2x_0$, player 2 chooses the first branch: cost > 3

The negative side: why is that hard?

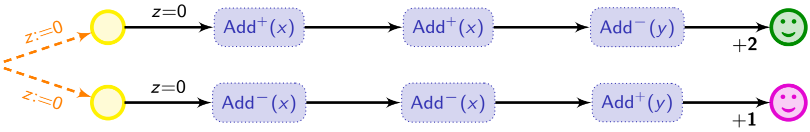
Given two clocks x and y, we can check whether $y = 2x$.



- In , cost $= 2x_0 + (1 - y_0) + 2$
  In , cost $= 2(1 - x_0) + y_0 + 1$
- if $y_0 < 2x_0$, player 2 chooses the first branch: cost > 3
  if $y_0 > 2x_0$, player 2 chooses the second branch: cost > 3

The negative side: why is that hard?

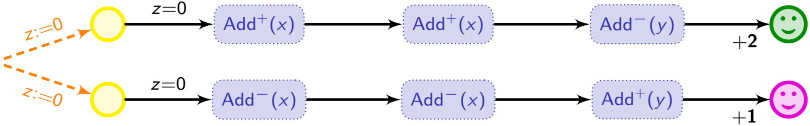
Given two clocks x and y, we can check whether $y = 2x$.



- In , cost $= 2x_0 + (1 - y_0) + 2$
  In , cost $= 2(1 - x_0) + y_0 + 1$
- if $y_0 < 2x_0$, player 2 chooses the first branch: cost > 3
  if $y_0 > 2x_0$, player 2 chooses the second branch: cost > 3
  if $y_0 = 2x_0$, in both branches, cost $= 3$

The negative side: why is that hard?

Given two clocks x and y, we can check whether $y = 2x$.



- In , cost $= 2x_0 + (1 - y_0) + 2$
  In , cost $= 2(1 - x_0) + y_0 + 1$
- if $y_0 < 2x_0$, player 2 chooses the first branch: cost > 3
  if $y_0 > 2x_0$, player 2 chooses the second branch: cost > 3
  if $y_0 = 2x_0$, in both branches, cost $= 3$
- Player 1 has a winning strategy with cost ≤ 3 iff $y_0 = 2x_0$

The negative side: why is that hard?

Player 1 will simulate a two-counter machine:

- each instruction is encoded as a module;
- the values c_1 and c_2 of the counters are encoded by the values of two clocks:

$$x = \frac{1}{2^{c_1}} \qquad \text{and} \qquad y = \frac{1}{3^{c_2}}$$

  when entering the corresponding module.

The negative side: why is that hard?

Player 1 will simulate a two-counter machine:

- each instruction is encoded as a module;
- the values c_1 and c_2 of the counters are encoded by the values of two clocks:

$$x = \frac{1}{2^{c_1}} \qquad \text{and} \qquad y = \frac{1}{3^{c_2}}$$

  when entering the corresponding module.

The two-counter machine has an halting computation iff player 1 has a winning strategy to ensure a cost no more than 3.

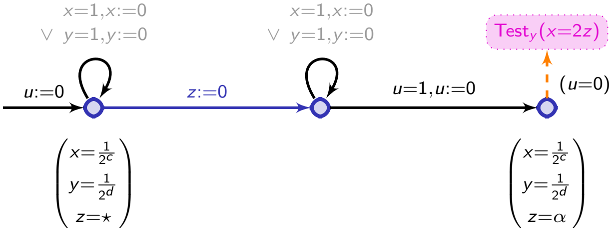
The negative side: why is that hard?

Player 1 will simulate a two-counter machine:

- each instruction is encoded as a module;
- the values c_1 and c_2 of the counters are encoded by the values of two clocks:
$$x = \frac{1}{2^{c_1}} \qquad \text{and} \qquad y = \frac{1}{3^{c_2}}$$
when entering the corresponding module.

The two-counter machine has an halting computation iff player 1 has a winning strategy to ensure a cost no more than 3.

Globally, $(x\leq 1, y\leq 1, u\leq 1)$



Going further: other cost functions

An easy adaptation of the previous undecidability proof yields:

Theorem

Optimal mean-cost games are undecidable.

[JT08] Jurdziński, Trivedi. Average-time games *(FSTTCS'08)*.

Going further: other cost functions

An easy adaptation of the previous undecidability proof yields:

Theorem

Optimal mean-cost games are undecidable.

Theorem [JT08]

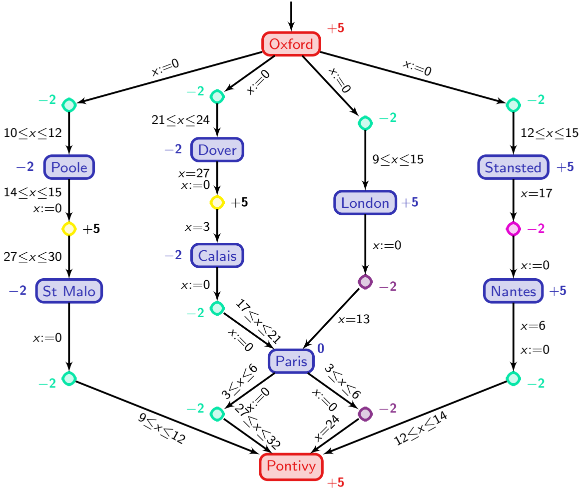
Turn-based optimal average-time games are decidable and EXPTIME-complete.

$\leadsto$ talk of ~~Ashutosh Trivedi~~ in the next session
Marcin Jurdziǹski

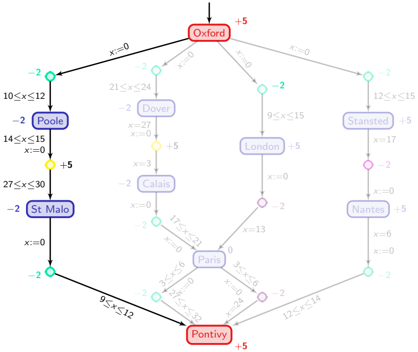
[JT08] Jurdziński, Trivedi. Average-time games *(FSTTCS'08)*.

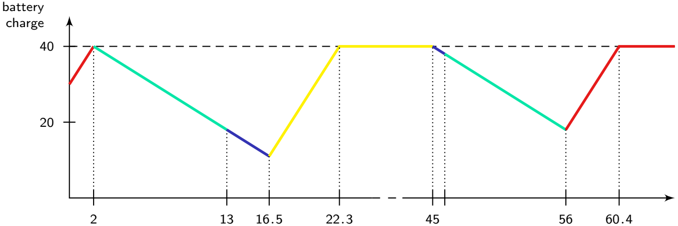
Outline

A fourth model of the system



Can I work on my computer all the way?





The motivation

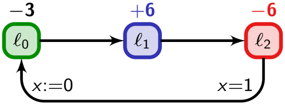
Energy is not only consumed, but can be regained.

$\leadsto$ the aim is to continuously satisfy some energy constraints.

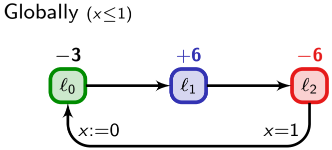
[BFL+08] Bouyer, Fahrenberg, Larsen, Markey, Srba. Infinite runs in weighted timed automata with energy constraints *(FORMATS'08)*.

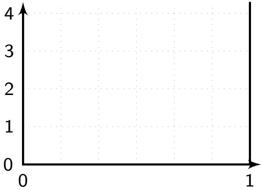
An example

Globally $(x \leq 1)$



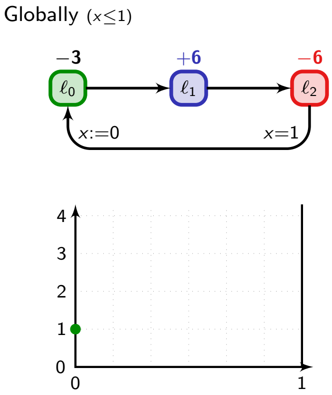
An example





- Lower-bound problem: can we stay above 0?

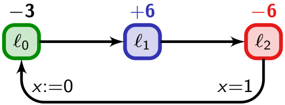
An example

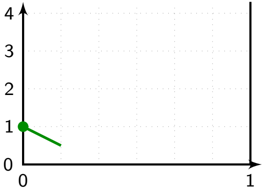


- Lower-bound problem: can we stay above 0?

An example

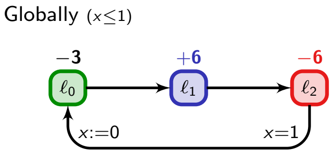
Globally $(x \leq 1)$

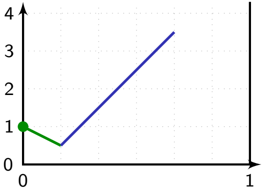




- Lower-bound problem: can we stay above 0?

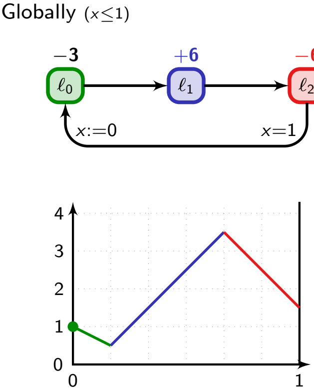
An example





- Lower-bound problem: can we stay above 0?

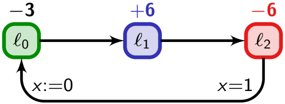
An example

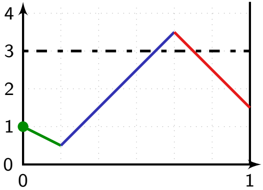


- Lower-bound problem: can we stay above 0?

An example

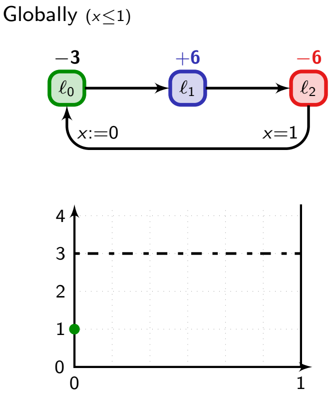
Globally $(x \leq 1)$





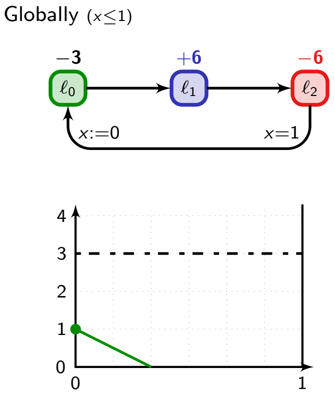
- Lower-bound problem: can we stay above 0?

An example



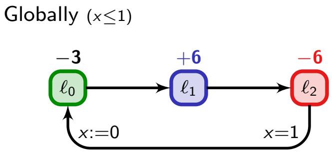
- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

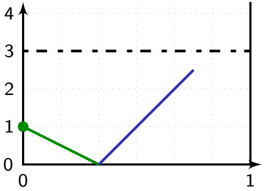
An example



- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example

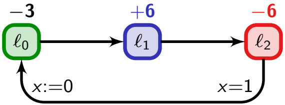


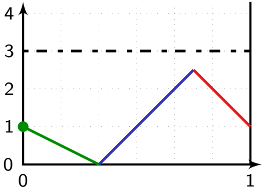


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example

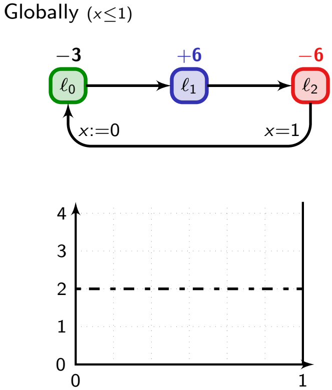
Globally $(x \leq 1)$





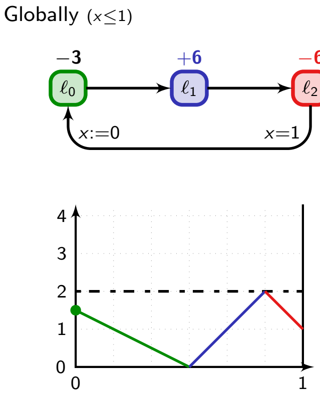
- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example



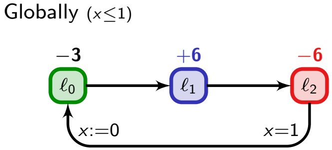
- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

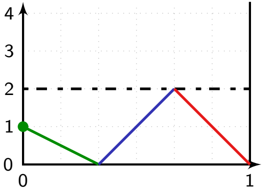
An example



- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example

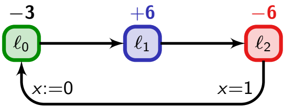


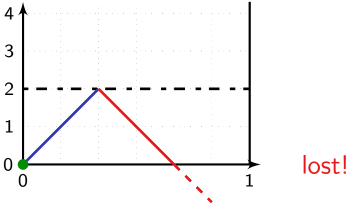


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example

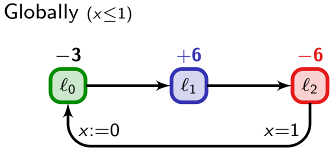
Globally $(x \leq 1)$

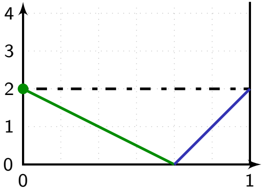




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example

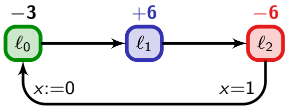


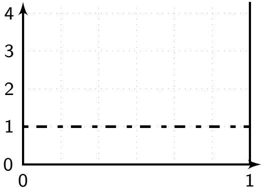


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example

Globally $(x \leq 1)$

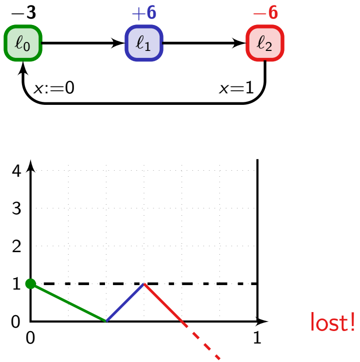




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

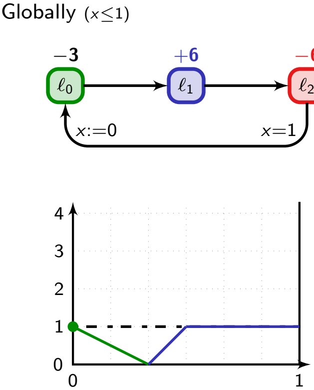
An example

Globally $(x \leq 1)$



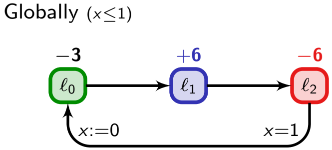
- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

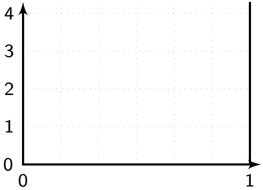
An example



- Lower-bound problem
- Lower-upper-bound problem
- Lower-weak-upper-bound problem: can we "weakly" stay within bounds?

An example





- Lower–bound problem $\rightsquigarrow$ **L**
- Lower-upper-bound problem $\rightsquigarrow$ **L+U**
- Lower-weak-upper-bound problem $\rightsquigarrow$ **L+W**

Results in the untimed case

	exist. problem	**univ. problem**	**games**
L	$\in$ P	$\in$ P	$\in$ UP $\cap$ coUP P-hard
L$+$**W**	$\in$ P	$\in$ P	$\in$ NP $\cap$ coNP P-hard
L$+$**U**	$\in$ PSPACE NP-hard	$\in$ P	EXPTIME-c.

Results in the untimed case

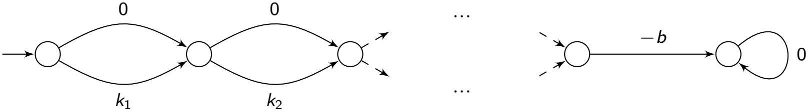
	exist. problem	**univ. problem**	**games**
L	$\in$ P	$\in$ P	$\in$ UP $\cap$ coUP P-hard
L+W	$\in$ P	$\in$ P	$\in$ NP $\cap$ coNP P-hard
L+U	$\in$ PSPACE NP-hard	$\in$ P	EXPTIME-c.

- Bellman-Ford algorithm

Results in the untimed case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	$\in$ UP $\cap$ coUP P-hard
L+W	$\in$ P	$\in$ P	$\in$ NP $\cap$ coNP P-hard
L+U	$\in$ PSPACE NP-hard	$\in$ P	EXPTIME-c.

- PSPACE: guess an infinite path in the graph augmented with the energy level.
- NP-hardness: encode SUBSET-SUM:



Results in the untimed case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	$\in$ UP $\cap$ coUP P-hard
L+W	$\in$ P	$\in$ P	$\in$ NP $\cap$ coNP P-hard
L+U	$\in$ PSPACE NP-hard	$\in$ P	EXPTIME-c.

- EXPTIME: play the game in the graph augmented with the energy level.
- EXPTIME-hardness: encode COUNTDOWN-GAME [JLS07].

Results in the untimed case

	exist. problem	**univ. problem**	**games**
L	$\in$ P	$\in$ P	$\in$ UP $\cap$ coUP P-hard
L+W	$\in$ P	$\in$ P	$\in$ NP $\cap$ coNP P-hard
L+U	$\in$ PSPACE NP-hard	$\in$ P	EXPTIME-c.

- Mean-payoff games

Equivalence with mean-payoff games

Definition

Mean-payoff games: in a weighted game graph, does there exists a strategy s.t. the mean-cost of any play is nonnegative?

Equivalence with mean-payoff games

Definition

Mean-payoff games: in a weighted game graph, does there exists a strategy s.t. the mean-cost of any play is nonnegative?

Lemma

L-games and **L**+**W**-games are determined, and memoryless strategies are sufficient to win.

Equivalence with mean-payoff games

Definition

Mean-payoff games: in a weighted game graph, does there exists a strategy s.t. the mean-cost of any play is nonnegative?

Lemma

L-games and **L**+**W**-games are determined, and memoryless strategies are sufficient to win.

- from mean-payoff games to **L**-games or **L**+**W**-games: play in the same game graph G with initial credit $-M \geq 0$ (where M is the sum of negative costs in G).

Equivalence with mean-payoff games

Definition

Mean-payoff games: in a weighted game graph, does there exists a strategy s.t. the mean-cost of any play is nonnegative?

Lemma

L-games and **L**+**W**-games are determined, and memoryless strategies are sufficient to win.

- from mean-payoff games to **L**-games or **L**+**W**-games: play in the same game graph G with initial credit $-M \geq 0$ (where M is the sum of negative costs in G).
- from **L**-games to mean-payoff games: transform the game as follows:

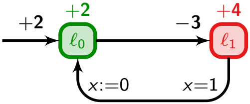
Results for the single-clock case

	exist. problem	**univ. problem**	**games**
L	$\in$ P	$\in$ P	?
L+**W**	$\in$ P	$\in$ P	?
L+**U**	?	?	undecidable

Results for the single-clock case

	exist. problem	**univ. problem**	**games**
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

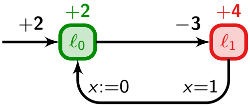
- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

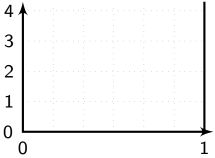


Results for the single-clock case

	exist. problem	**univ. problem**	**games**
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

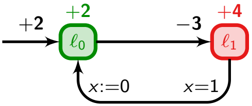


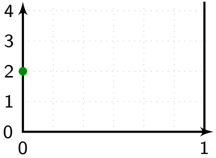


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

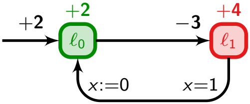


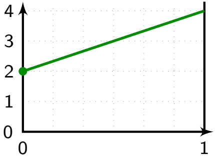


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

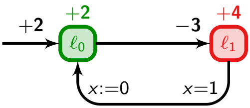


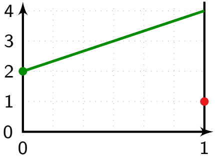


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

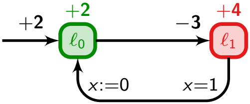


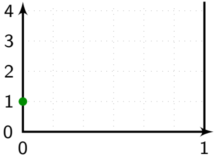


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

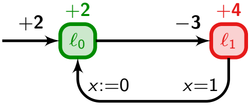


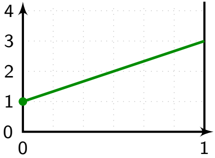


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

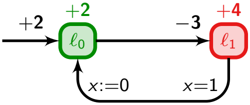


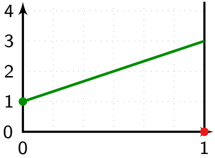


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

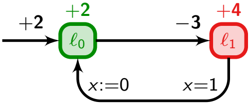


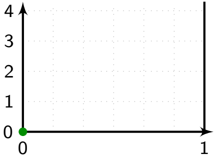


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

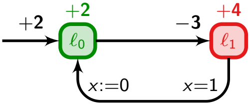


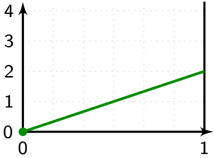


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

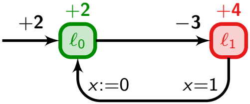


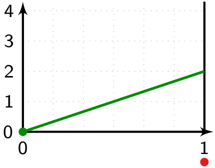


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!



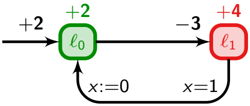


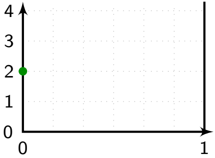
lost!

Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

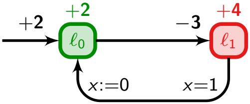


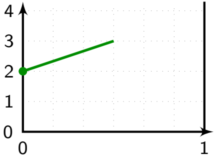


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

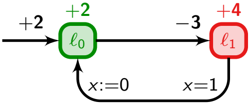


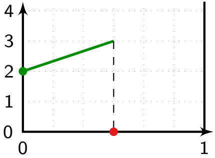


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

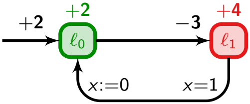


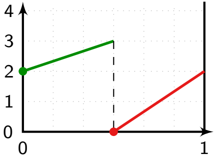


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!

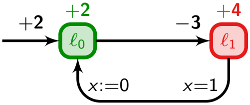


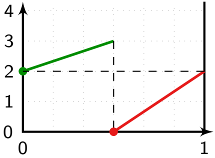


Results for the single-clock case

	exist. problem	univ. problem	games
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- the corner-point abstraction can be used (wait in the most profitable location) ... but only if discrete costs are not used!!





Results for the single-clock case

	exist. problem	**univ. problem**	**games**
L	$\in$ P	$\in$ P	?
L+W	$\in$ P	$\in$ P	?
L+U	?	?	undecidable

- simulation of a two-counter machine

Single-clock $\mathbf{L}+\mathbf{U}$-games

Theorem

The single-clock $\mathbf{L}+\mathbf{U}$-games are undecidable.

Single-clock **L**+**U**-games

Theorem

The single-clock **L**+**U**-games are undecidable.

We encode the behaviour of a two-counter machine:

- each instruction is encoded as a module;
- the values c_1 and c_2 of the counters are encoded by the energy level

$$e = 5 - \frac{1}{2^{c_1} \cdot 3^{c_2}}$$

when entering the corresponding module.

Single-clock $\mathbf{L}+\mathbf{U}$-games

Theorem

The single-clock $\mathbf{L}+\mathbf{U}$-games are undecidable.

We encode the behaviour of a two-counter machine:

- each instruction is encoded as a module;
- the values c_1 and c_2 of the counters are encoded by the energy level

$$e = 5 - \frac{1}{2^{c_1} \cdot 3^{c_2}}$$

  when entering the corresponding module.

There is an infinite execution in the two-counter machine iff there is a strategy in the single-clock timed game under which the energy level remains between 0 and 5.

Single-clock **L**+**U**-games

Theorem

The single-clock **L**+**U**-games are undecidable.

We encode the behaviour of a two-counter machine:

- each instruction is encoded as a module;
- the values c_1 and c_2 of the counters are encoded by the energy level

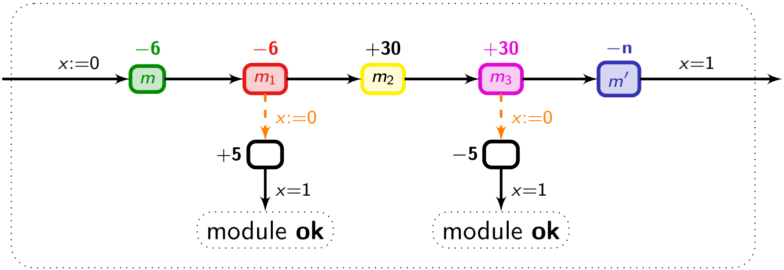
$$e = 5 - \frac{1}{2^{c_1} \cdot 3^{c_2}}$$

when entering the corresponding module.

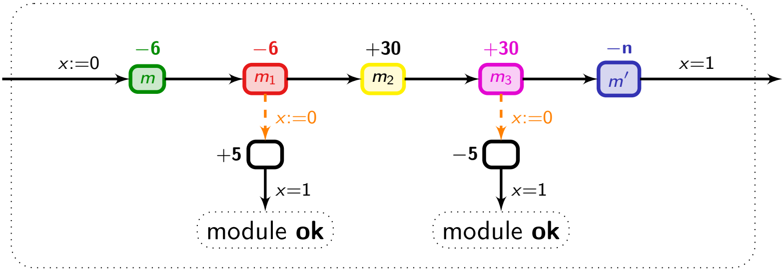
There is an infinite execution in the two-counter machine iff there is a strategy in the single-clock timed game under which the energy level remains between 0 and 5.

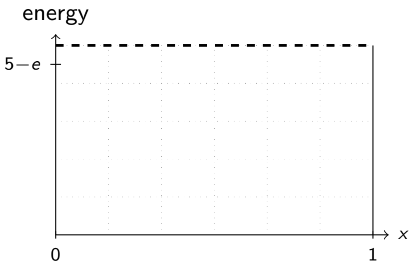
$\rightsquigarrow$ We present a generic construction for incrementing/decrementing the counters.

Generic module for incrementing/decrementing

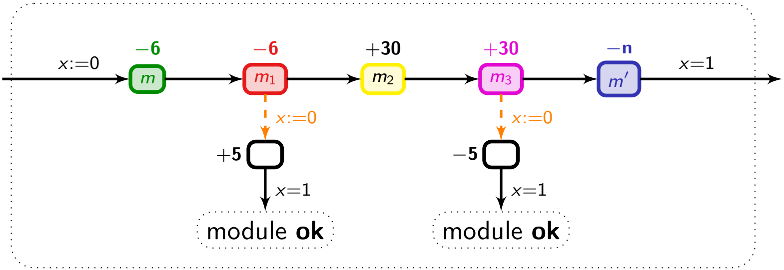


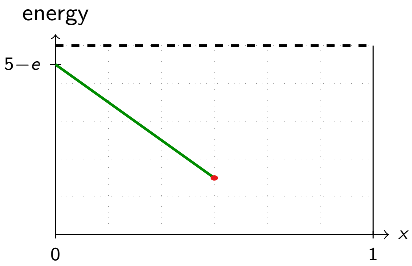
Generic module for incrementing/decrementing



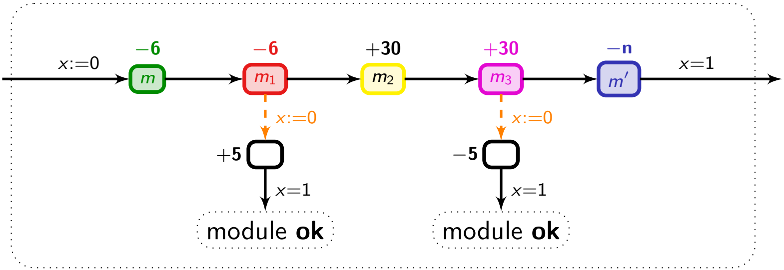


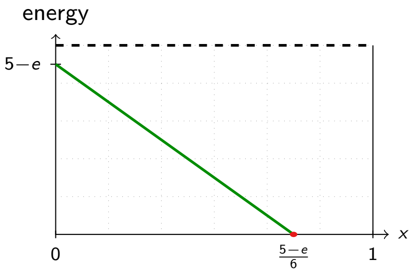
Generic module for incrementing/decrementing



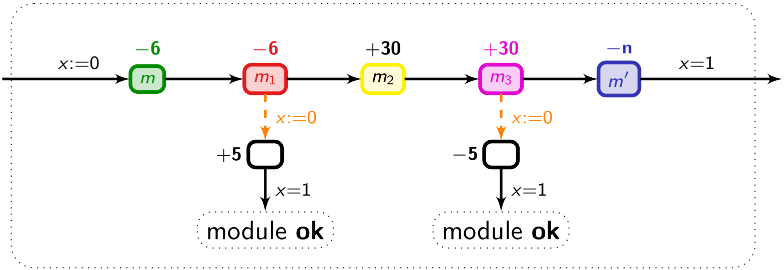


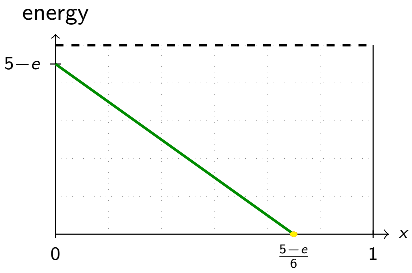
Generic module for incrementing/decrementing



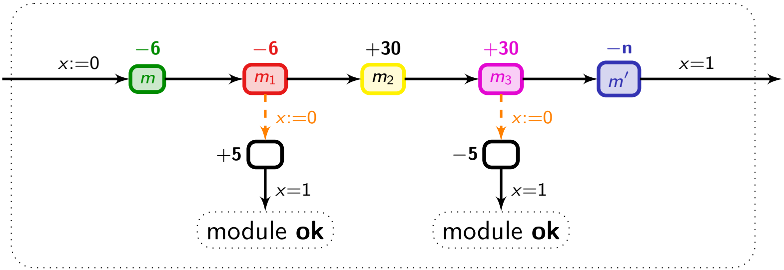


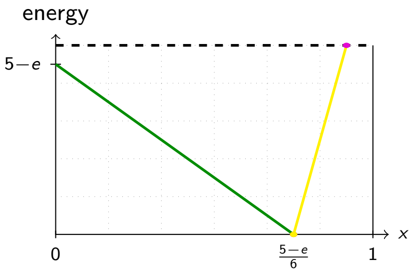
Generic module for incrementing/decrementing



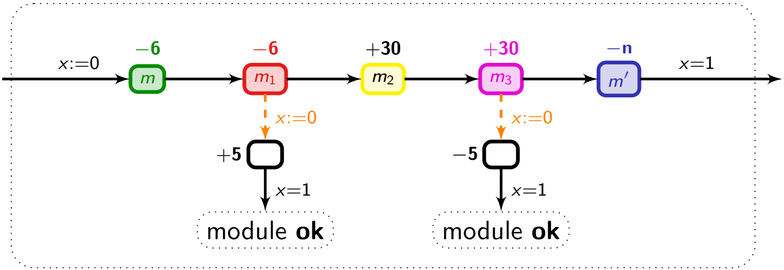


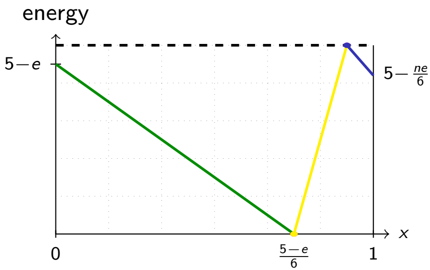
Generic module for incrementing/decrementing



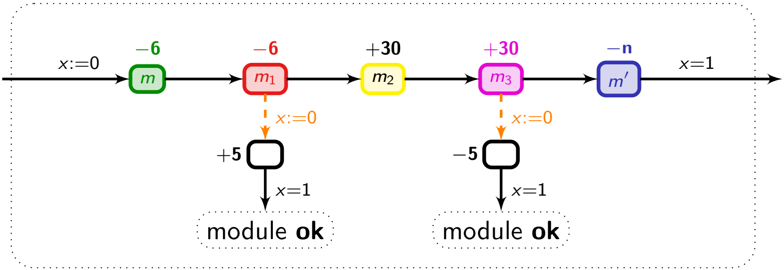


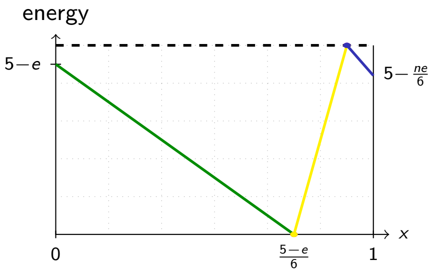
Generic module for incrementing/decrementing





Generic module for incrementing/decrementing





- **n**=3: increment c_1
- **n**=2: increment c_2
- **n**=12: decrement c_1
- **n**=18: decrement c_2

Results for the general case

	exist. problem	**univ. problem**	**games**
L	?	?	?
L+W	?	?	?
L+U	?	?	undecidable

Outline

Conclusion

- Priced/weighted timed automata, a model for representing quantitative constraints on timed systems:
  - useful in embedded systems verification
  - natural (optimization) questions have been posed...

  ... and not all of them have been answered yet!

Conclusion

- Priced/weighted timed automata, a model for representing quantitative constraints on timed systems:
  - useful in embedded systems verification
  - natural (optimization) questions have been posed...

    ... and not all of them have been answered yet!
- Not mentioned here:
  - all works on model-checking issues (extensions of CTL, LTL)
  - models based on hybrid automata
    - weighted o-minimal hybrid games [BBC07]
    - weighted strong reset hybrid games [BBJLR07]

      $\rightsquigarrow$ talk of Michał Rutkowski in the next session
  - various tools have been developed:

    Uppaal, Uppaal Cora, Uppaal Tiga

Conclusion

- Priced/weighted timed automata, a model for representing quantitative constraints on timed systems:
  - useful in embedded systems verification
  - natural (optimization) questions have been posed...

    ... and not all of them have been answered yet!
- Not mentioned here:
  - all works on model-checking issues (extensions of CTL, LTL)
  - models based on hybrid automata
    - weighted o-minimal hybrid games [BBC07]
    - weighted strong reset hybrid games [BBJLR07]

      $\rightsquigarrow$ talk of Michał Rutkowski in the next session
  - various tools have been developed:

    Uppaal, Uppaal Cora, Uppaal Tiga
- Current and further work:
  - computation of approximate optimal values
  - further investigation of safe games + several cost variables?
  - discounted-time optimal games
  - link between discounted-time games and mean-cost games?
  - ...