Modelling, analyzing, and managing resources in timed systems

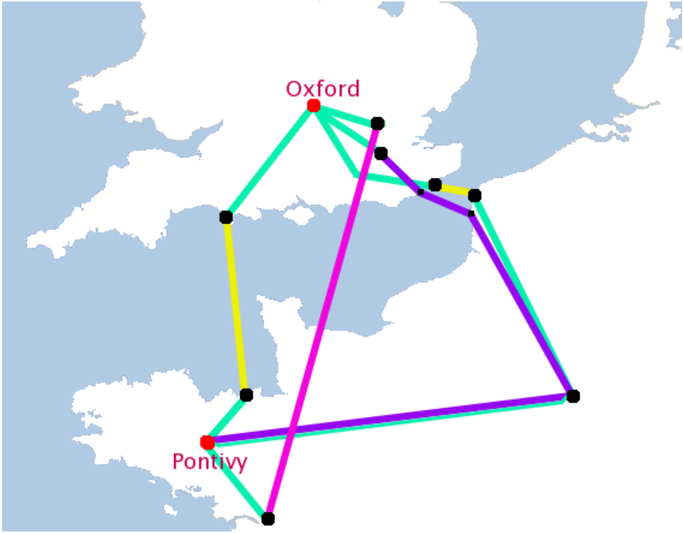
Patricia Bouyer-Decitre

LSV, CNRS & ENS Cachan, France

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

Outline

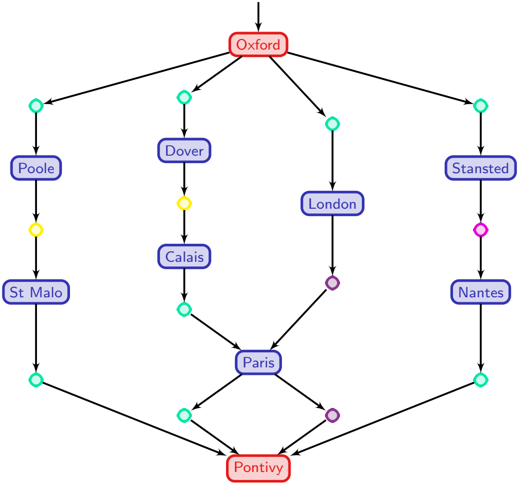
A starting example



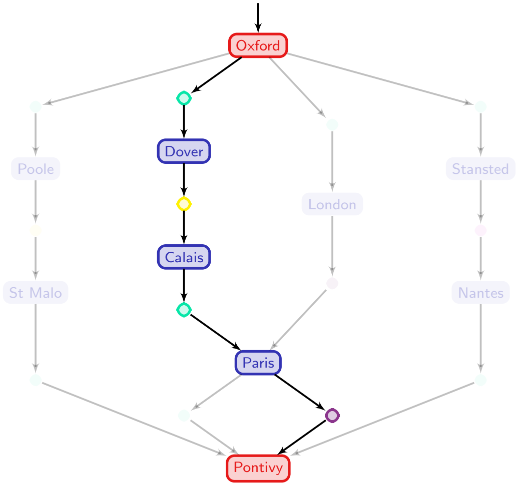
Natural questions

- **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

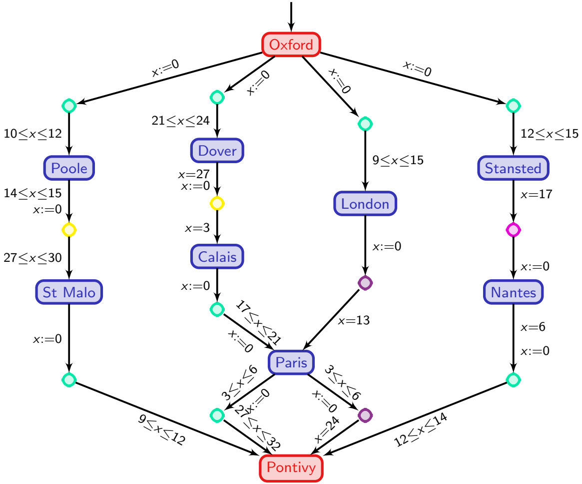


Can I reach Pontivy from Oxford?

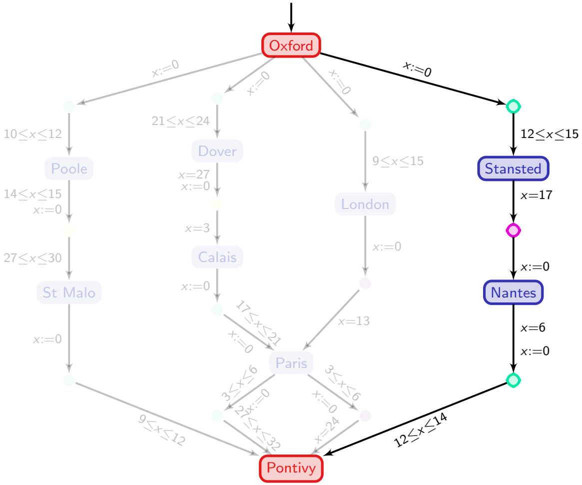


This is a reachability question in a finite graph: Yes, I can!

A second model of the system

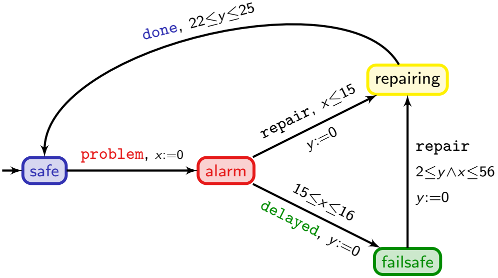


How long will that take?

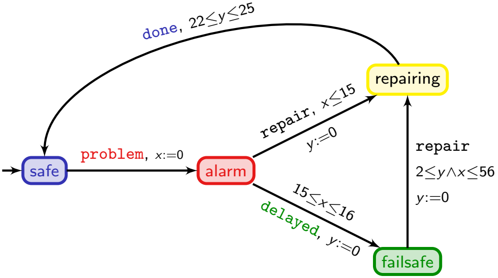


It is a reachability (and optimization) question
in a timed automaton: at least 350mn = 5h50mn!

An example of a timed automaton

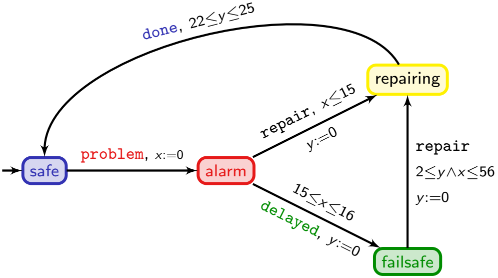


An example of a timed automaton



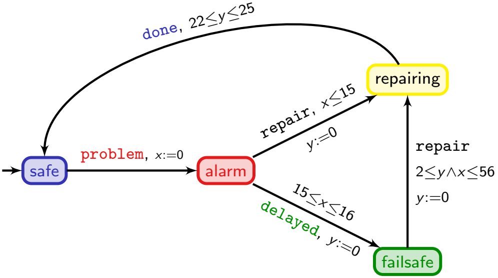
	safe
x	0
y	0

An example of a timed automaton



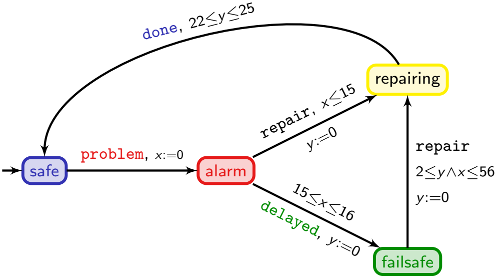
	safe	$\xrightarrow{23}$	safe
x	0		23
y	0		23

An example of a timed automaton



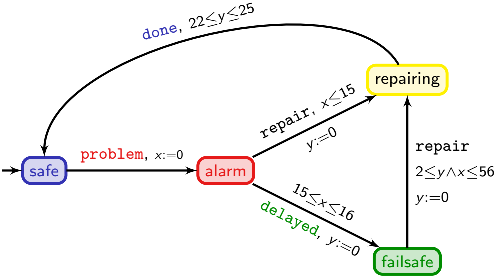
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\texttt{problem}}$	alarm
x	0		23		0
y	0		23		23

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

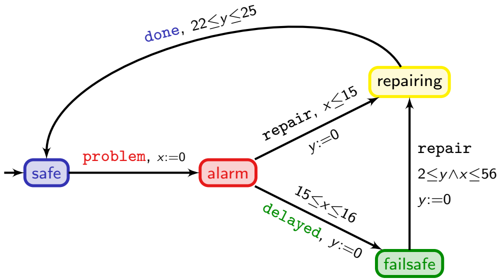
An example of a timed automaton



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

	failsafe
···	15.6
	0

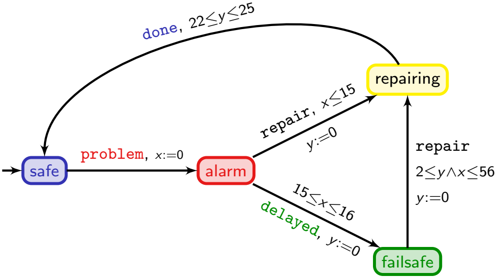
An example of a timed automaton



	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

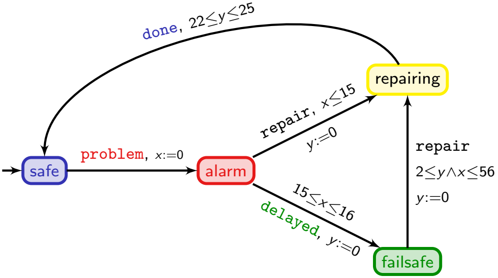
An example of a timed automaton



	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

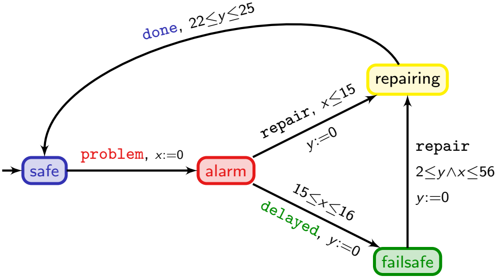
An example of a timed automaton



	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,CY92]

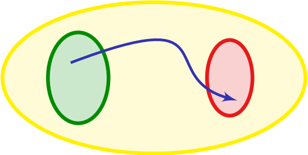
The (time-optimal) reachability problem is decidable (and PSPACE-complete) for timed automata.

[AD90] Alur, Dill. Automata for modeling real-time systems *(ICALP'90)*.
[CY92] Courcoubetis, Yannakakis. Minimum and maximum delay problems in real-time systems *(Formal Methods in System Design)*.

Timed automata

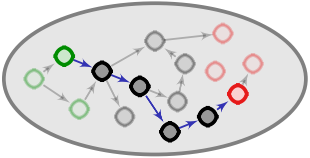
Theorem [AD90,CY92]

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



timed automaton

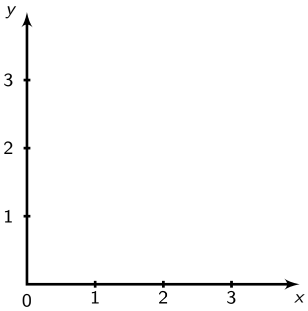
finite bisimulation



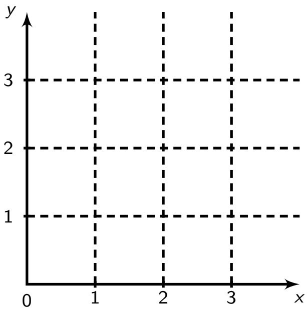
large (but finite) automaton
(region automaton)

[AD90] Alur, Dill. Automata for modeling real-time systems *(ICALP'90)*.
[CY92] Courcoubetis, Yannakakis. Minimum and maximum delay problems in real-time systems *(Formal Methods in System Design)*.

The region abstraction

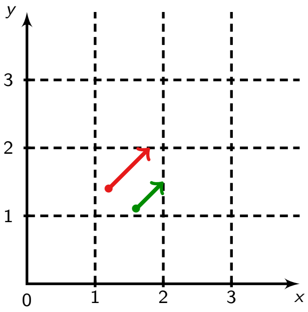


The region abstraction



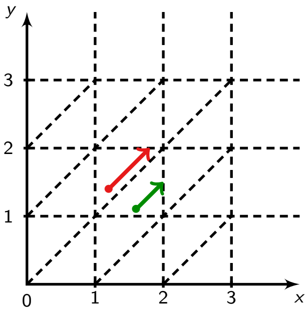
- "compatibility" between regions and constraints

The region abstraction



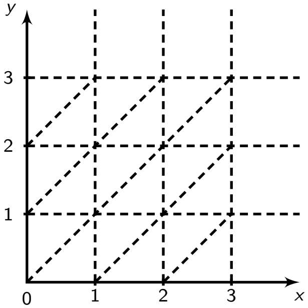
- "compatibility" between regions and constraints
- "compatibility" between regions and time elapsing

The region abstraction



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

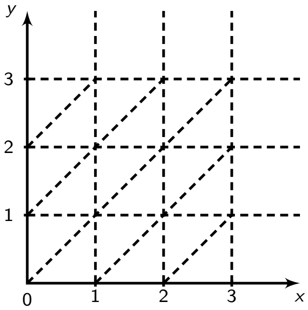
The region abstraction



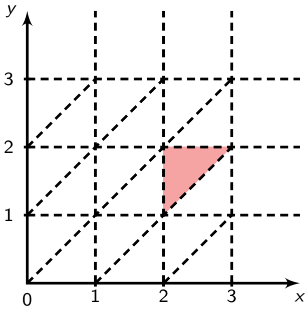
- "compatibility" between regions and constraints
- "compatibility" between regions and time elapsing

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

The region abstraction

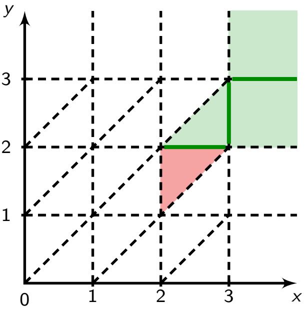


The region abstraction



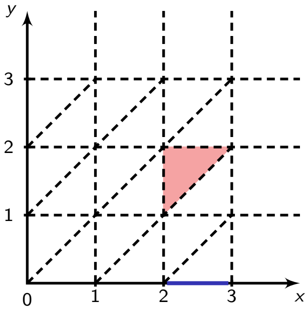
$$\begin{cases} 2<x<3 \\ 1<y<2 \\ \{x\}<\{y\} \end{cases}$$

The region abstraction



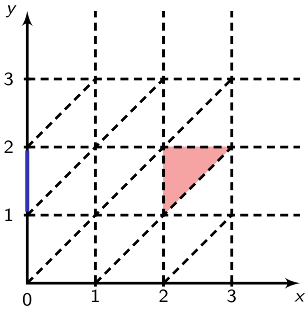
time successors

The region abstraction



reset of clock y

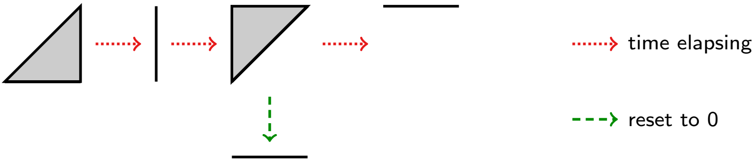
The region abstraction



reset of clock x

The region graph

A finite graph representing time elapsing and reset of clocks:



Outline

Modelling resources in timed systems

- System resources might be relevant and even crucial information

Modelling resources in timed systems

- System resources might be relevant and even crucial information
  - energy consumption,
  - memory usage,
  - price to pay,
  - bandwidth,
  - ...

Modelling resources in timed systems

- System resources might be relevant and even crucial information
  - energy consumption,
  - memory usage,
  - price to pay,
  - bandwidth,
  - ...

  $\rightsquigarrow$ timed automata are not powerful enough!

Modelling resources in timed systems

- System resources might be relevant and even crucial information
  - energy consumption,
  - memory usage,
  - price to pay,
  - bandwidth,
  - ...

  $\leadsto$ timed automata are not powerful enough!

- A possible solution: use hybrid automata

$$\begin{array}{ll} & \text{a discrete control (the mode of the system)} \\ + & \text{continuous evolution of the variables within a mode} \end{array}$$

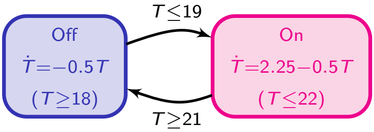
Modelling resources in timed systems

- System resources might be relevant and even crucial information
  - energy consumption,
  - memory usage,
  - price to pay,
  - bandwidth,
  - ...

    $\leadsto$ timed automata are not powerful enough!

- A possible solution: use hybrid automata

The thermostat example



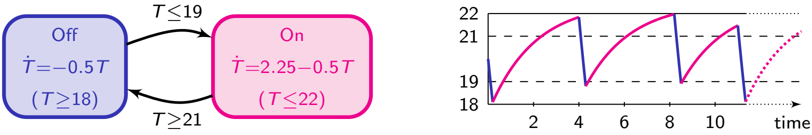
Modelling resources in timed systems

- System resources might be relevant and even crucial information
  - energy consumption,
  - memory usage,
  - price to pay,
  - bandwidth,
  - ...

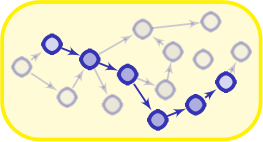
  $\leadsto$ timed automata are not powerful enough!

- A possible solution: use hybrid automata

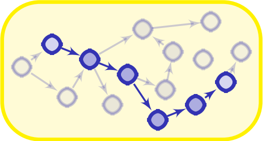
The thermostat example



Ok...

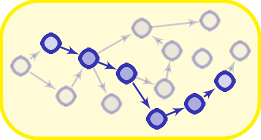


Ok...

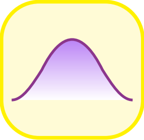


Easy...

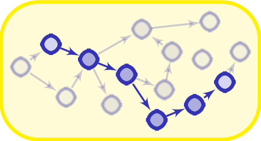
Ok...



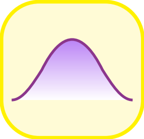
Easy...



Ok...

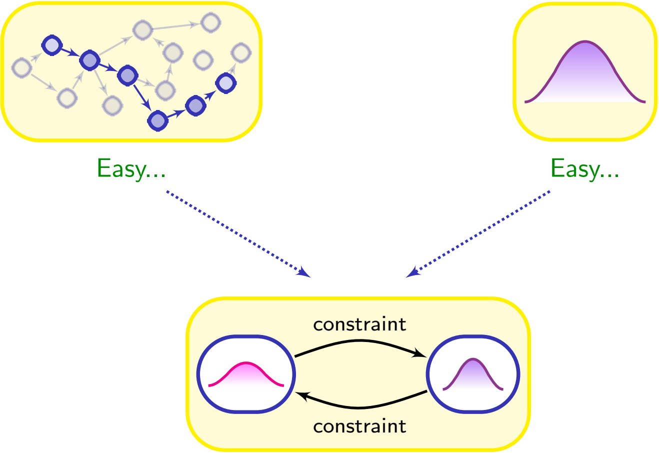


Easy...

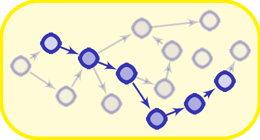


Easy...

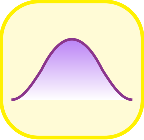
Ok... but?



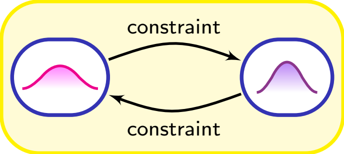
Ok... but?



Easy...



Easy...



Hard!

Modelling resources in timed systems

- System resources might be relevant and even crucial information
  - energy consumption,
  - memory usage,
  - price to pay,
  - bandwidth,
  - ...

  $\rightsquigarrow$ timed automata are not powerful enough!

- A possible solution: use hybrid automata

Theorem [HKPV95]

The reachability problem is undecidable in hybrid automata.

[HKPV95] Henzinger, Kopke, Puri, Varaiya. What's decidable wbout hybrid automata? *(SToC'95)*.

Modelling resources in timed systems

- System resources might be relevant and even crucial information
  - energy consumption,
  - memory usage,
  - price to pay,
  - bandwidth,
  - ...

  $\rightsquigarrow$ timed automata are not powerful enough!

- A possible solution: use hybrid automata

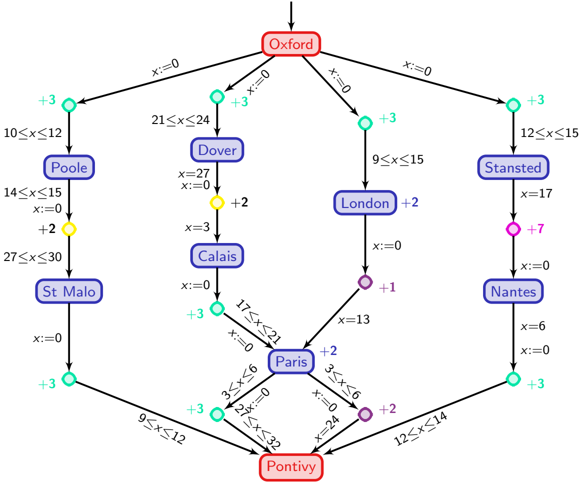
Theorem [HKPV95]

The reachability problem is undecidable in hybrid automata.

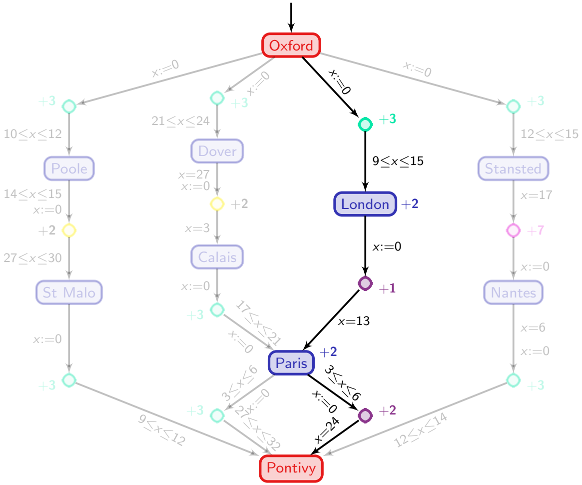
- An alternative: weighted/priced timed automata [ALP01,BFH+01]
  $\rightsquigarrow$ hybrid variables do not constrain the system
  hybrid variables are observer variables

[HKPV95] Henzinger, Kopke, Puri, Varaiya. What's decidable wbout hybrid automata? *(SToC'95)*.[ALP01] Alur, La Torre, Pappas. Optimal paths in weigh
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.

A third model of the system

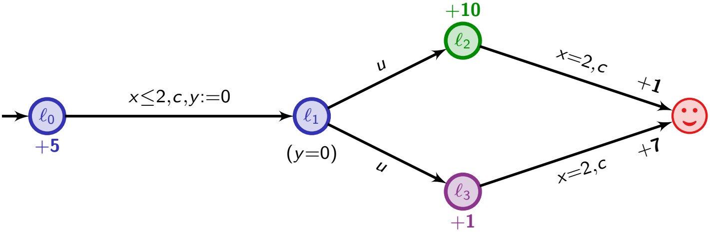


How much fuel will I use?



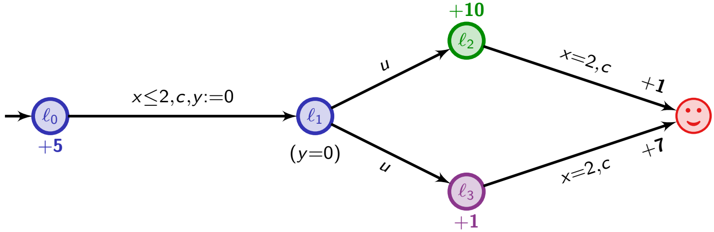
It is a quantitative (optimization) problem
in a priced timed automaton: at least 68 anti-planet units!

Weighted/priced timed automata [ALP01,BFH+01]



[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)*.

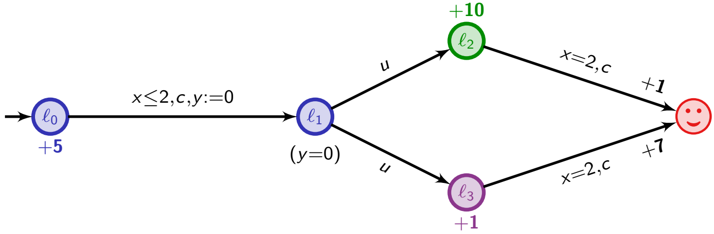
Weighted/priced timed automata [ALP01,BFH+01]



	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		

[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)*.

Weighted/priced timed automata [ALP01,BFH+01]

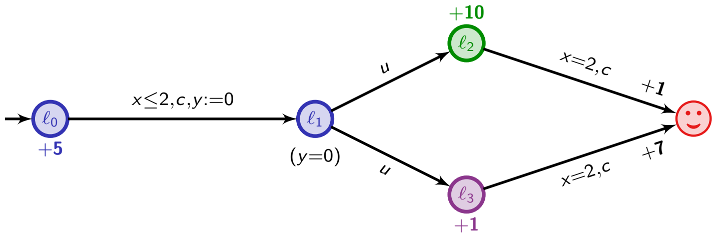


$$
\begin{array}{ccccccccccccc}
 & \ell_0 & \xrightarrow{1.3} & \ell_0 & \xrightarrow{c} & \ell_1 & \xrightarrow{u} & \ell_3 & \xrightarrow{0.7} & \ell_3 & \xrightarrow{c} & ☺ \\
x & 0 & & 1.3 & & 1.3 & & 1.3 & & 2 & & \\
y & 0 & & 1.3 & & 0 & & 0 & & 0.7 & &
\end{array}
$$

cost :

[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)*.

Weighted/priced timed automata [ALP01,BFH+01]

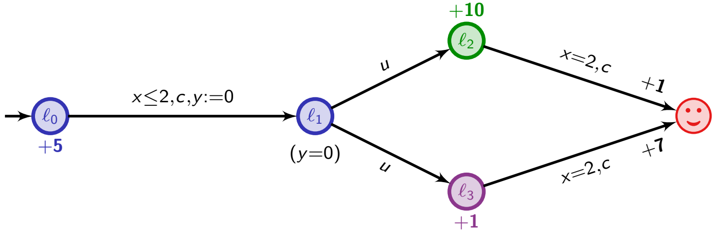


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

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)*.

Weighted/priced timed automata [ALP01,BFH+01]

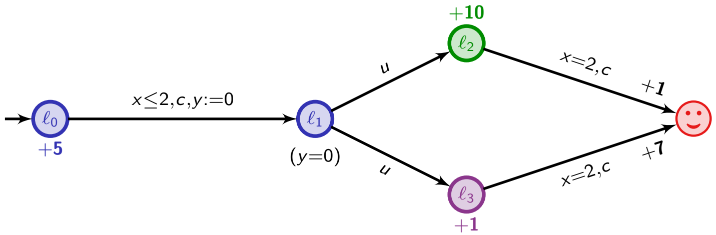


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

cost : 6.5 + 0

[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)*.

Weighted/priced timed automata [ALP01,BFH+01]

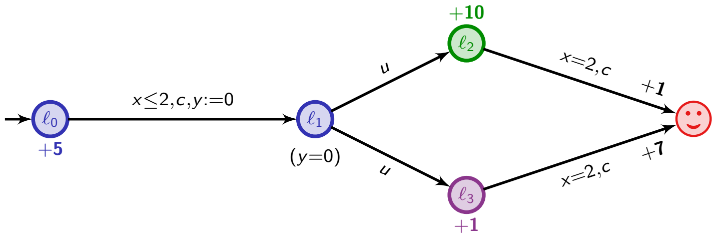


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

cost : 6.5 + 0 + 0

[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)*.

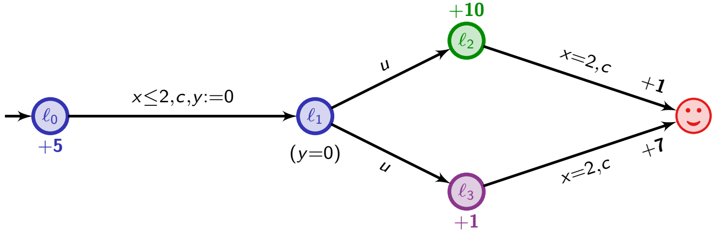
Weighted/priced timed automata [ALP01,BFH+01]



	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			

[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)*.

Weighted/priced timed automata [ALP01,BFH+01]

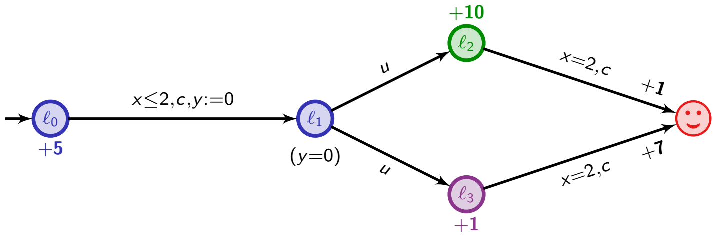


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

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)*.

Weighted/priced timed automata [ALP01,BFH+01]

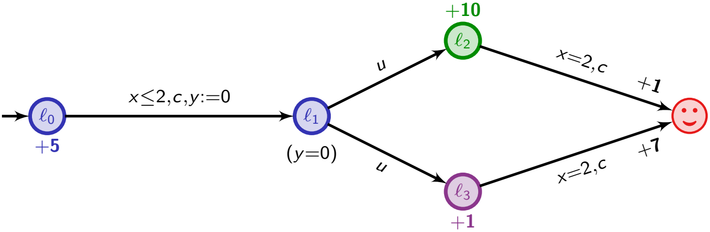


$$\begin{array}{lcccccccccccc} & \ell_0 & \xrightarrow{1.3} & \ell_0 & \xrightarrow{c} & \ell_1 & \xrightarrow{u} & \ell_3 & \xrightarrow{0.7} & \ell_3 & \xrightarrow{c} & \text{☺} \\ x & 0 & & 1.3 & & 1.3 & & 1.3 & & 2 & & \\ y & 0 & & 1.3 & & 0 & & 0 & & 0.7 & & \end{array}$$

$$\text{cost}: \quad 6.5 + 0 + 0 + 0.7 + 7 = 14.2$$

[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)*.

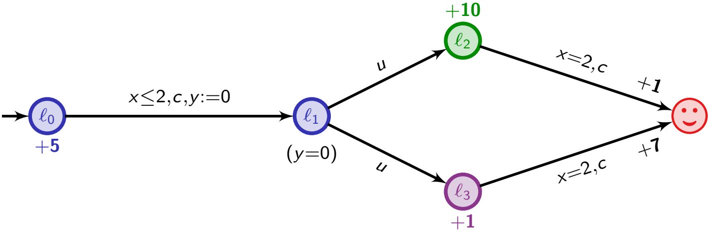
Weighted/priced timed automata [ALP01,BFH+01]



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)*.

Weighted/priced timed automata [ALP01,BFH+01]

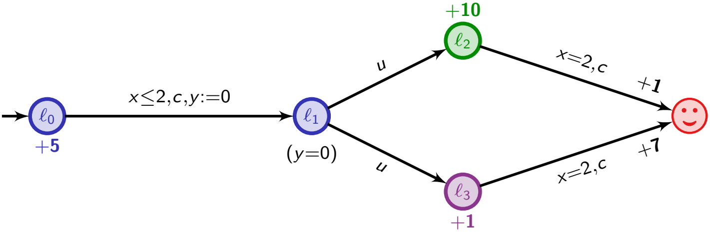


Question: what is the optimal cost for reaching ☺ ?

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

[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)*.

Weighted/priced timed automata [ALP01,BFH+01]

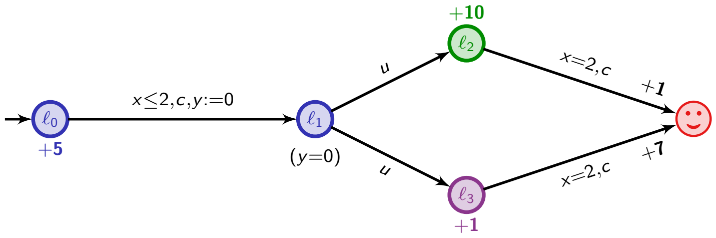


Question: what is the optimal cost for reaching ☺?

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

[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)*.

Weighted/priced timed automata [ALP01,BFH+01]

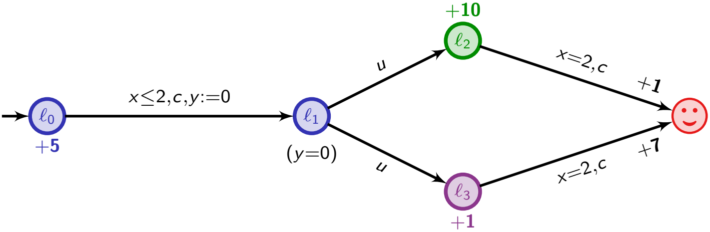


Question: what is the optimal cost for reaching ☺?

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

[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)*.

Weighted/priced timed automata [ALP01,BFH+01]

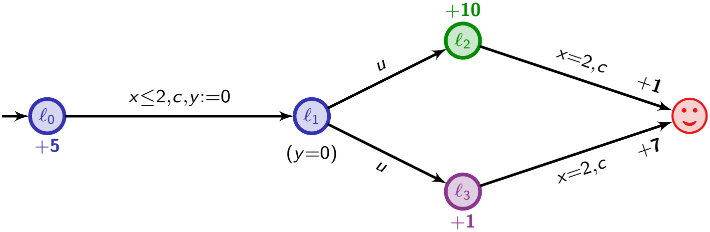


Question: what is the optimal cost for reaching 🙂 ?

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

[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)*.

Weighted/priced timed automata [ALP01,BFH+01]



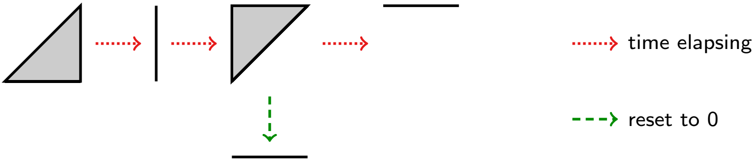
Question: what is the optimal cost for reaching 🙂 ?

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

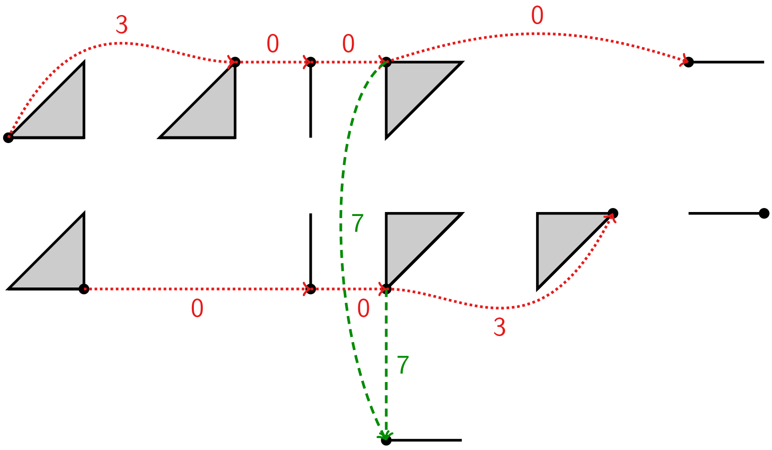
$\rightsquigarrow$ *strategy:* leave immediately ℓ_0, go to ℓ_3, and wait there 2 t.u.

[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)*.

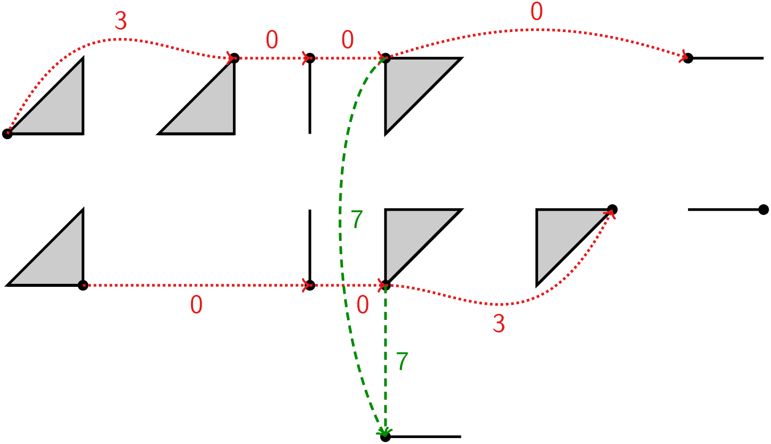
The region abstraction is not fine enough



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

$$\circ \xrightarrow{t_1} \circ \xrightarrow[x\leq 2]{t_2} \circ \xrightarrow{t_3} \circ \xrightarrow{t_4} \circ \xrightarrow{t_5} \circ \quad \cdots \qquad \left\{ \begin{array}{l} t_1+t_2\leq 2 \\ \\ \end{array} \right.$$

From timed to discrete behaviours

Optimal reachability as a linear programming problem

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

From timed to discrete behaviours

Optimal reachability as a linear programming problem

$$\circ \xrightarrow[y:=0]{t_1} \circ \xrightarrow[x\leq 2]{t_2} \circ \xrightarrow{t_3} \circ \xrightarrow[y\geq 5]{t_4} \circ \xrightarrow{t_5} \circ \quad \cdots \qquad \left\{ \begin{array}{l} t_1+t_2\leq 2 \\ t_2+t_3+t_4\geq 5 \end{array} \right.$$

Lemma

Let Z be a bounded zone and f be a function

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

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

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

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.

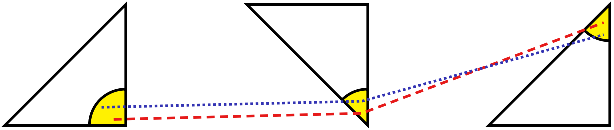
$\rightsquigarrow$ for every finite path π in $\mathcal{A}$, there exists a path Π in $\mathcal{A}_{\mathrm{cp}}$ such that

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

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

From discrete to timed behaviours

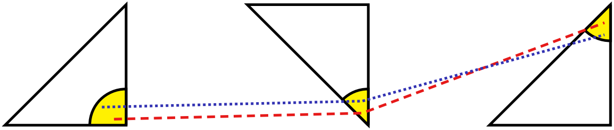
Approximation of abstract paths:



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

From discrete to timed behaviours

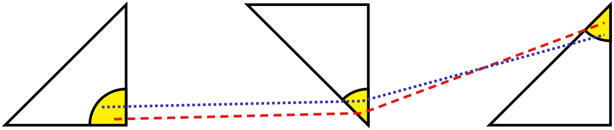
Approximation of abstract paths:



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

From discrete to timed behaviours

Approximation of abstract paths:

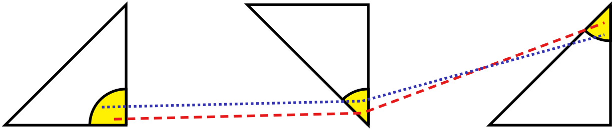


For any path Π of $\mathcal{A}_{\mathsf{cp}}$, for any $\varepsilon > 0$, there exists a path π_ε of $\mathcal{A}$ s.t.

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

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.

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

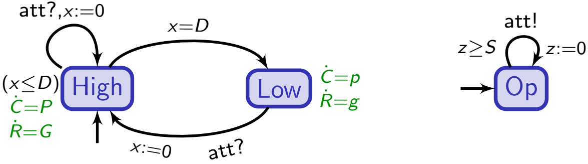
Optimal-cost reachability

Theorem [ALP01,BFH+01,BBBR07]

The optimal-cost reachability problem is decidable (and PSPACE-complete) in (priced) timed automata.

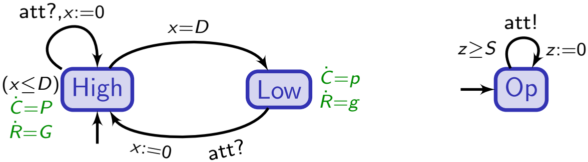
[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata *(HSCC'01)*.
[BFH+01] Behrmann, Fehnker, Hune, Larsen, Pettersson, Romijn, Vaandrager. Minimum-cost reachability in priced timed automata *(HSCC'01)*.
[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem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

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

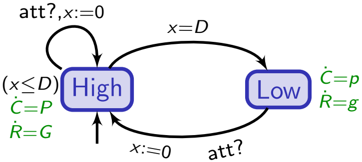


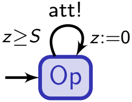
$\rightsquigarrow$ compute optimal infinite schedules that minimize

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

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

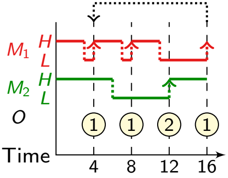
Going further 1: mean-cost optimization



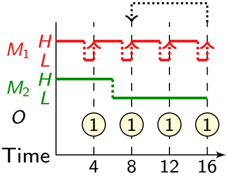


$\rightsquigarrow$ compute optimal infinite schedules that minimize

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



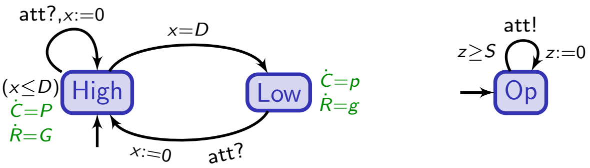
Schedule with ratio $\approx$1.455



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)*.

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.

From timed to discrete behaviours

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

Lemma

Let Z be a bounded zone and f be a function

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

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

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

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

From timed to discrete behaviours

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

Lemma

Let Z be a bounded zone and f be a function

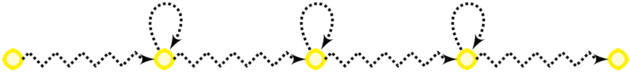
$$f : (t_1, \ldots, t_n) \mapsto \frac{\sum_{i=1}^{n} c_i t_i + c}{\sum_{i=1}^{n} r_i t_i + r}$$

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

$\rightsquigarrow$ for every finite path π in $\mathcal{A}$, there exists a path Π in $\mathcal{A}_{\text{cp}}$ s.t.

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

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



The (acyclic) linear part will be negligible!

From timed to discrete behaviours

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

Lemma

Let Z be a bounded zone and f be a function

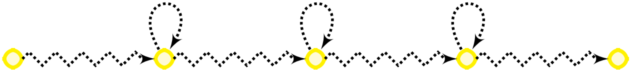
$$f : (t_1, \ldots, t_n) \mapsto \frac{\sum_{i=1}^{n} c_i t_i + c}{\sum_{i=1}^{n} r_i t_i + r}$$

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 $\mathcal{A}_{\text{cp}}$ s.t.

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

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

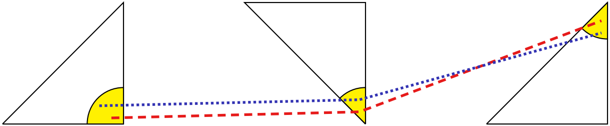


The (acyclic) linear part will be negligible!

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

From discrete to timed behaviours

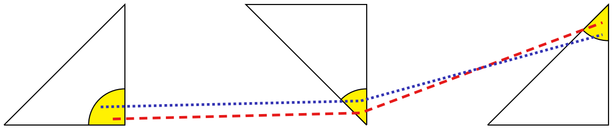
Approximation of abstract paths:



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

From discrete to timed behaviours

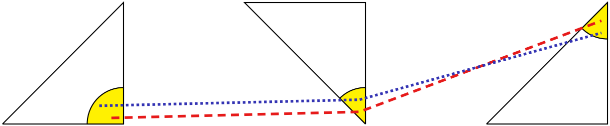
Approximation of abstract paths:



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

From discrete to timed behaviours

Approximation of abstract paths:

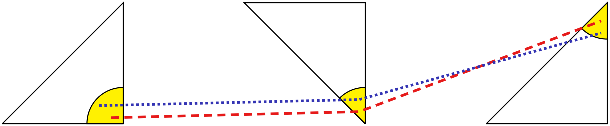


For any path Π of $\mathcal{A}_{\text{cp}}$, for any $\varepsilon > 0$, there exists a path π_ε of $\mathcal{A}$ s.t.

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

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.

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

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:

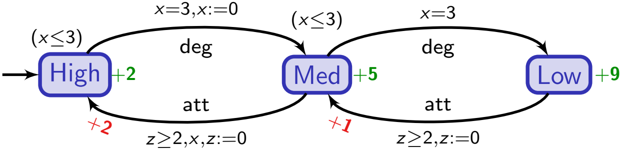
- optimal-time and optimal-cost reachability;
- optimal discrete discounted cost;
- optimal average-time and average-cost;
- optimal mean-cost.

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

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

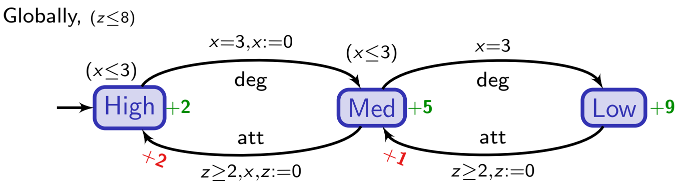
Going further 3: discounted-time cost optimization

Globally, $(z \leq 8)$



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

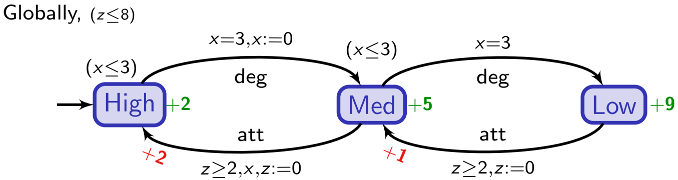
Going further 3: discounted-time cost optimization



$\rightsquigarrow$ 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



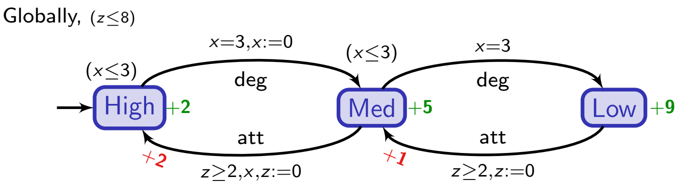
$\rightsquigarrow$ compute optimal infinite schedules that minimize

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

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)*.

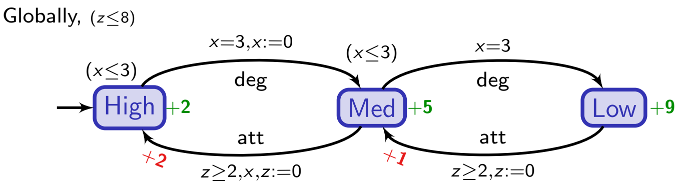
Going further 3: discounted-time cost optimization



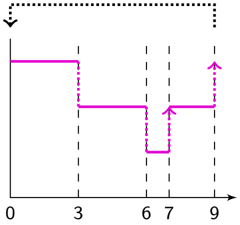
$\rightsquigarrow$ compute optimal infinite schedules that minimize
discounted cost over time

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

Going further 3: discounted-time cost optimization



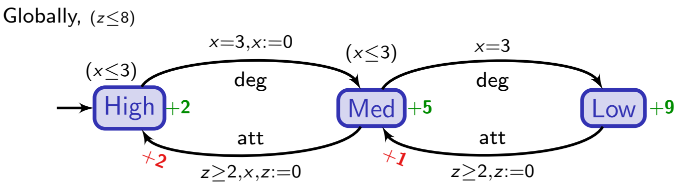
$\rightsquigarrow$ compute optimal infinite schedules that minimize
discounted cost over time



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

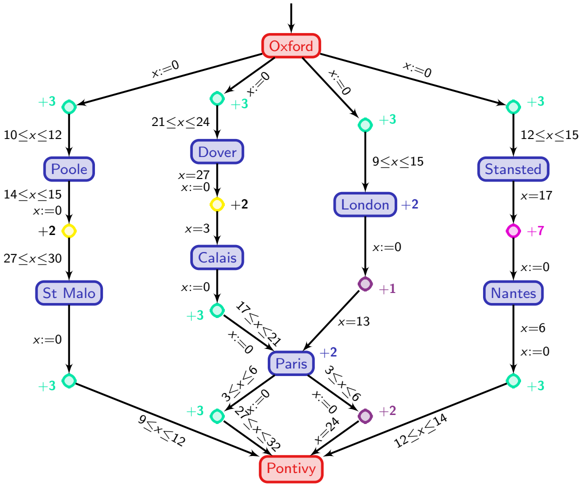
Theorem [FL08]

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

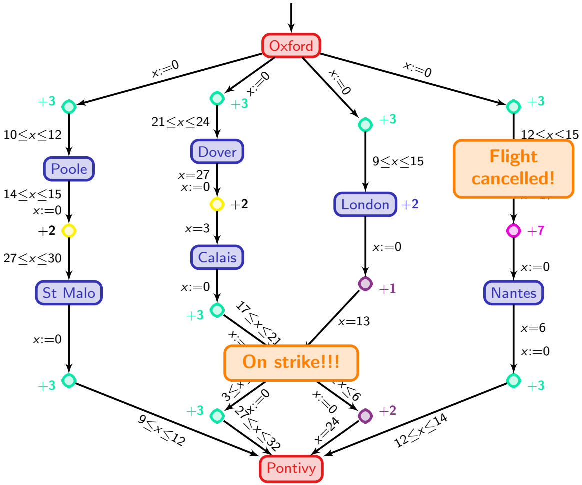
$\rightsquigarrow$ the corner-point abstraction can be used

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

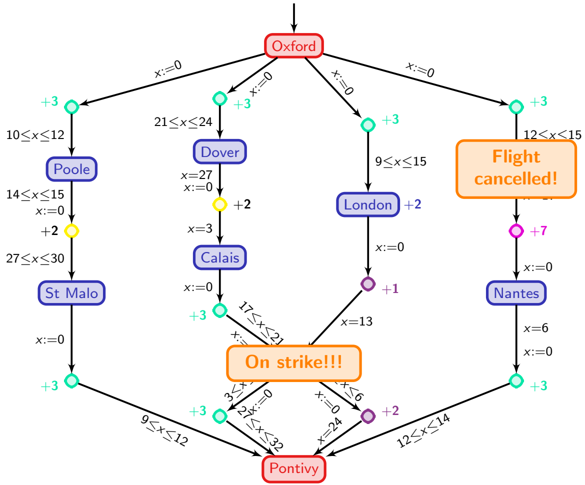
A fourth model of the system
What if there is an unexpected event?



A fourth model of the system
What if there is an unexpected event?

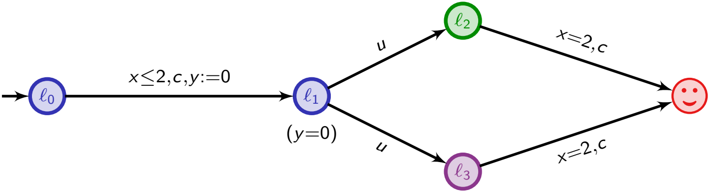


A fourth model of the system
What if there is an unexpected event?

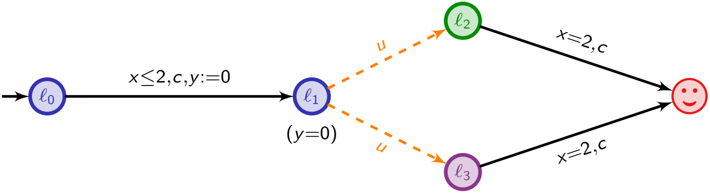


⇝ modelled as timed games

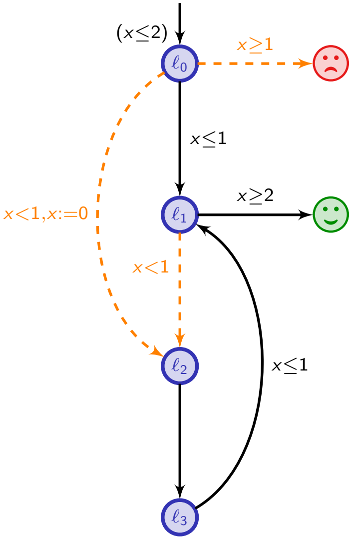
A simple example of timed game



A simple example of timed game



Another example



[CDL+05] Cassez, David, Fleury, Larsen, Lime. Efficient On-The-Fly Algorithms for the Analysis of Timed Games *(CONCUR'05)*.

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

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

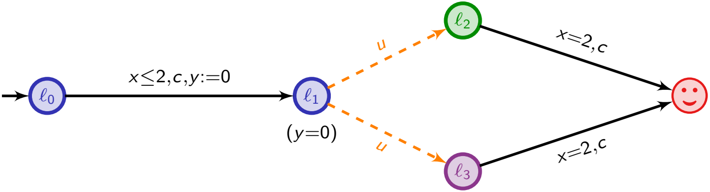
Theorem [AM99,BHPR07,JT07]

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

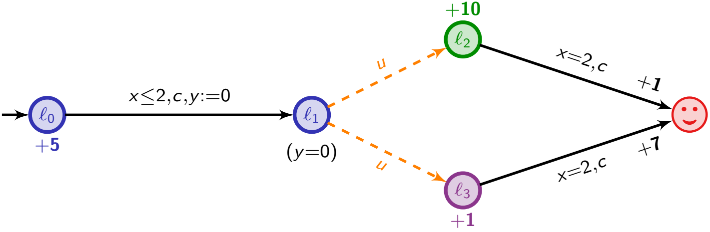
implemented in the tool Uppaal Tiga [BCD+07]

[BCD+07] Behrmann, Cougnard, David, Fleury, Larsen, Lime. Uppaal-Tiga: Time for playing games! *(CAV'07)*.

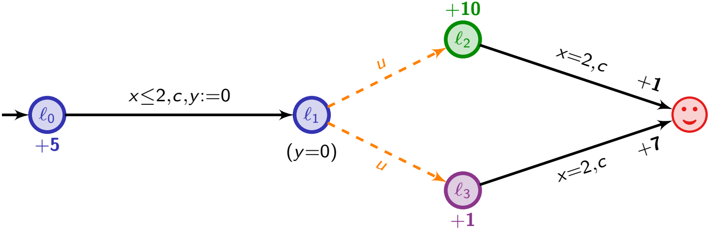
Back to the simple example



Back to the simple example

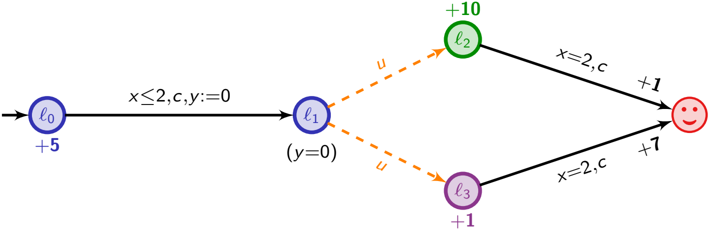


Back to the simple example



Question: what is the optimal cost we can ensure while reaching 🙂?

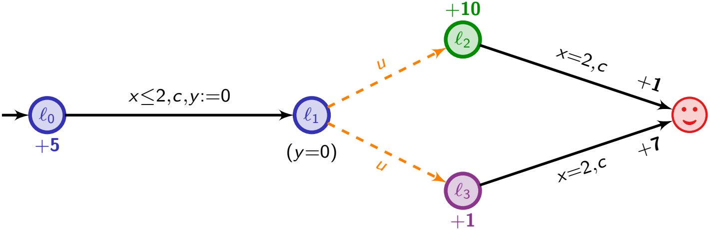
Back to the simple example



Question: what is the optimal cost we can ensure while reaching 🙂?

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

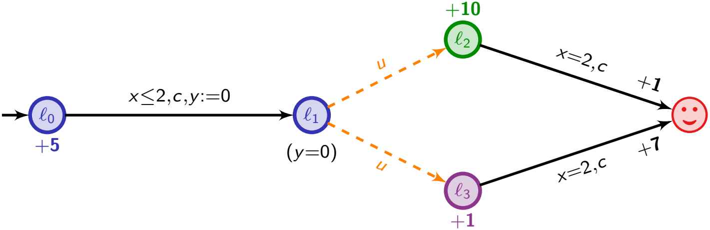
Back to the simple example



Question: what is the optimal cost we can ensure while reaching 🙂?

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

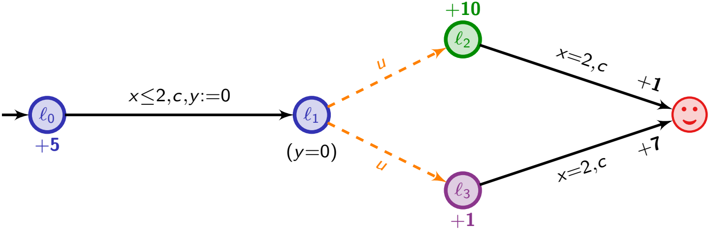
Back to the simple example



Question: what is the optimal cost we can ensure while reaching ☺ ?

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

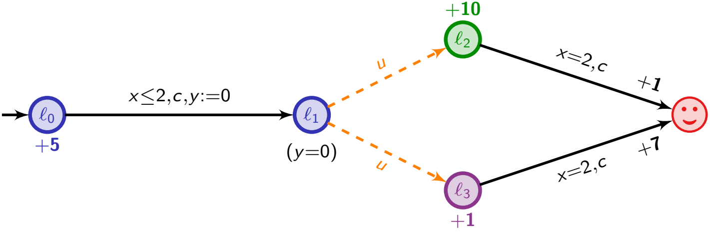
Back to the simple example



Question: what is the optimal cost we can ensure while reaching 🙂?

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

Back to the simple example



Question: what is the optimal cost we can ensure while reaching ☺?

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

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

Optimal reachability in priced timed games

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)*.

Optimal reachability in priced timed games

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)*.

Optimal reachability in priced timed games

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 PTIME-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 PTIME-hard.

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

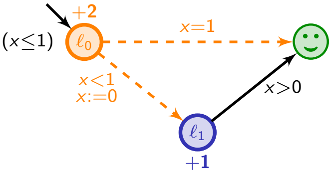
[BLMR06] Bouyer, Larsen, Markey, Rasmussen. Almost-optimal strategies in one-clock priced timed automata *(FSTTCS'06)*.

The positive side

Theorem [BLMR06]

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

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



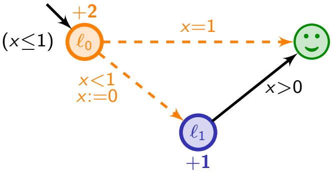
[BLMR06] Bouyer, Larsen, Markey, Rasmussen. Almost-optimal strategies in one-clock priced timed automata *(FSTTCS'06)*.

The positive side

Theorem [BLMR06]

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

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



- However, by unfolding and removing one by one the locations,we can synthesize memoryless almost-optimal winning strategies.

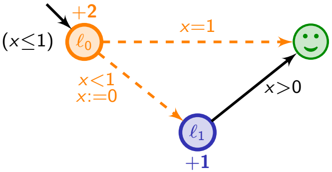
[BLMR06] Bouyer, Larsen, Markey, Rasmussen. Almost-optimal strategies in one-clock priced timed automata *(FSTTCS'06)*.

The positive side

Theorem [BLMR06]

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

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



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

[BLMR06] Bouyer, Larsen, Markey, Rasmussen. Almost-optimal strategies in one-clock priced timed automata *(FSTTCS'06)*.

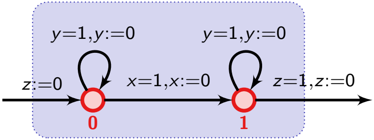
The negative side: why is that hard?

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

The negative side: why is that hard?

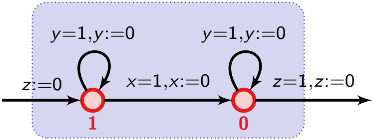
Given two clocks x and y, we can check whether $y = 2x$.

$\text{Add}^{+}(x)$



The cost is increased by x_0

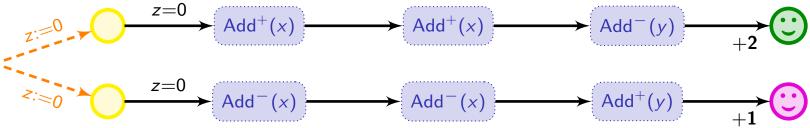
$\text{Add}^{-}(x)$



The cost is increased by $1-x_0$

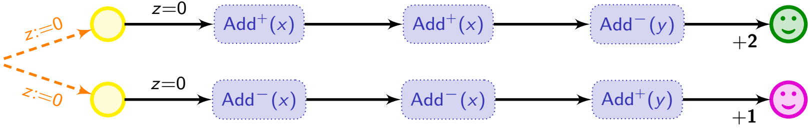
The negative side: why is that hard?

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



The negative side: why is that hard?

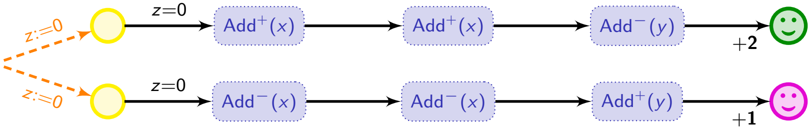
Given two clocks x and y, we can check whether $y = 2x$.



- 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 $y = 2x$.

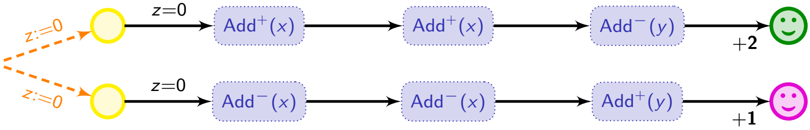


- In , cost $= 2x_0 + (1 - y_0) + 2$

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

The negative side: why is that hard?

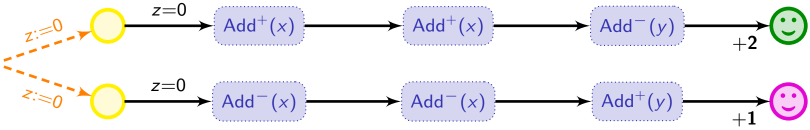
Given two clocks x and y, we can check whether $y = 2x$.



- 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?

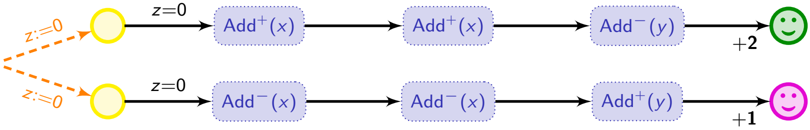
Given two clocks x and y, we can check whether $y = 2x$.



- 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 $y = 2x$.

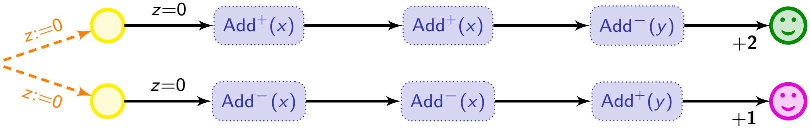


- In , cost $= 2x_0 + (1 - y_0) + 2$
  In , cost $= 2(1 - x_0) + y_0 + 1$

- 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.

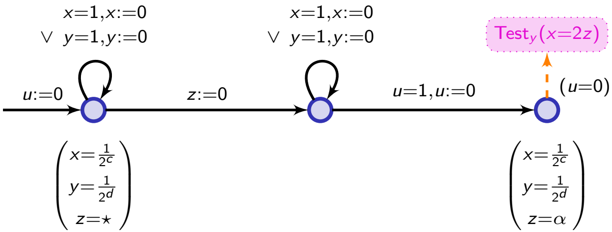
The negative side: why is that hard?

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:
$$x = \frac{1}{2^{c_1}} \quad \text{and} \quad y = \frac{1}{3^{c_2}}$$
when entering the corresponding module.

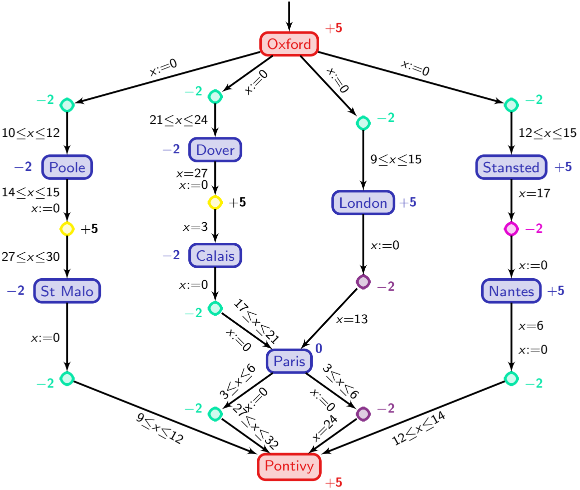
The two-counter machine has an halting computation iff player 1 has a winning strategy to ensure a cost no more than 3.

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

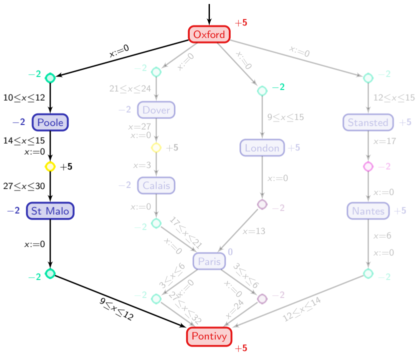


Outline

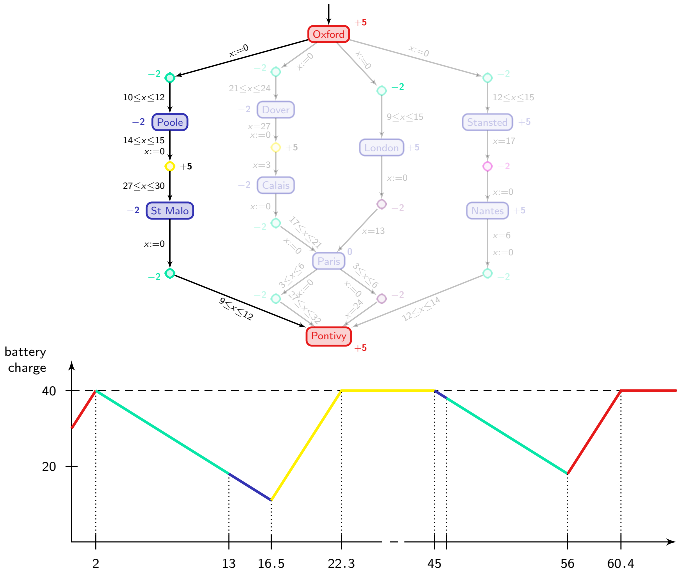
A fifth model of the system



Can I work with my computer all the way?



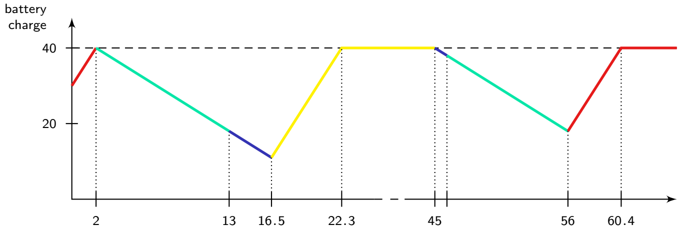
Can I work with my computer all the way?



Can I work with my computer all the way?

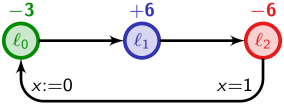
Energy is not only consumed, but can be regained.

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



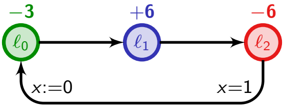
An example of resource management

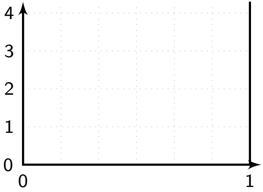
Globally $(x \leq 1)$



An example of resource management

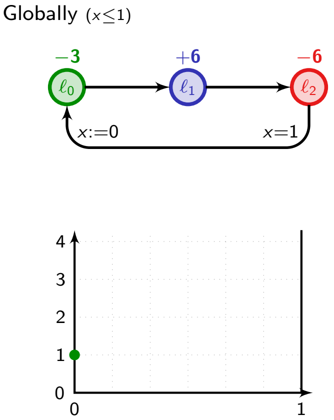
Globally $(x \leq 1)$





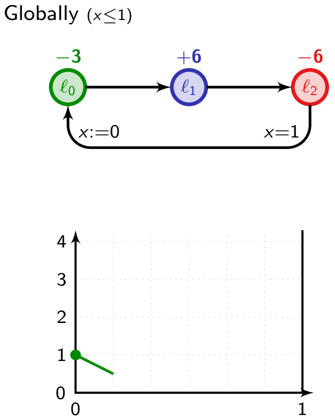
- Lower-bound problem: can we stay above 0?

An example of resource management



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

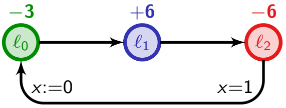
An example of resource management

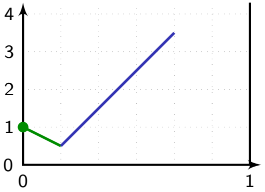


- Lower-bound problem: can we stay above 0?

An example of resource management

Globally $(x \leq 1)$

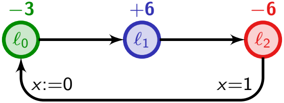


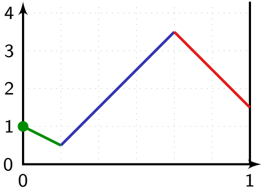


- Lower-bound problem: can we stay above 0?

An example of resource management

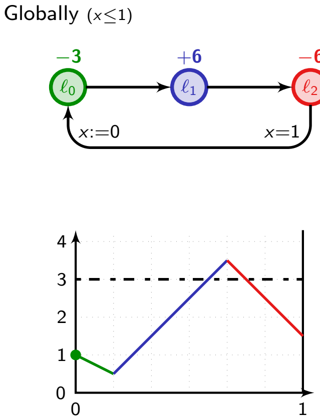
Globally $(x \leq 1)$





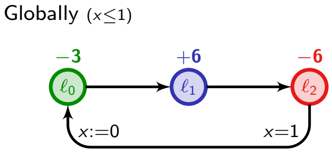
- Lower-bound problem: can we stay above 0?

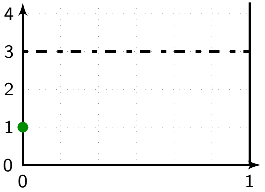
An example of resource management



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

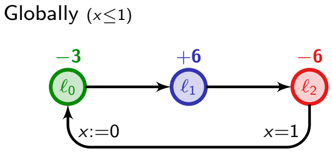
An example of resource management

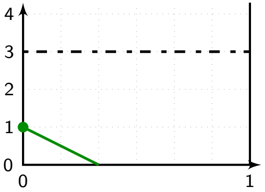




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

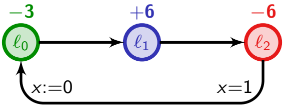


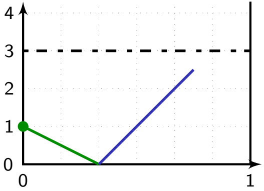


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

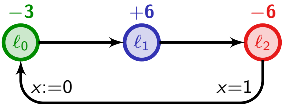


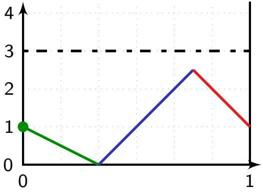


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

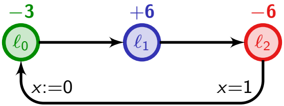


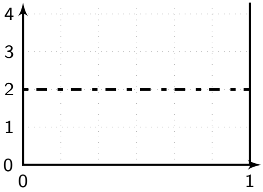


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

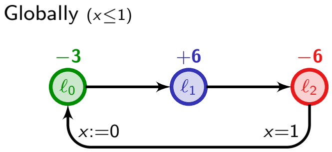
Globally $(x \leq 1)$

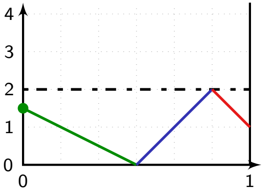




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

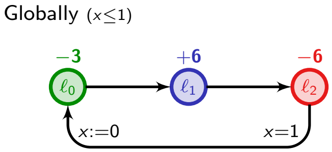
An example of resource management

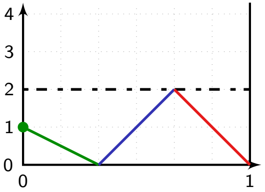




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

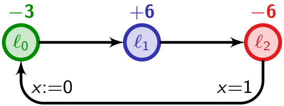


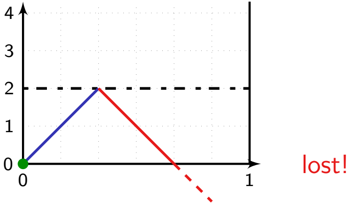


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

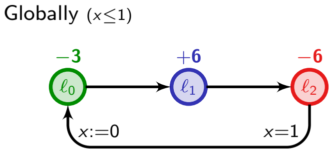
Globally $(x \leq 1)$

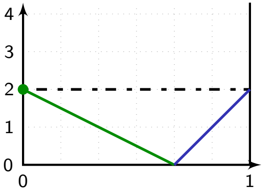




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

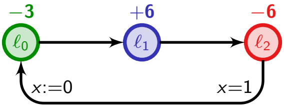


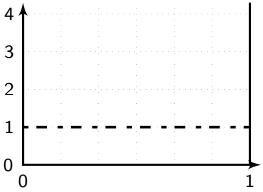


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leq 1)$

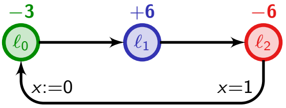


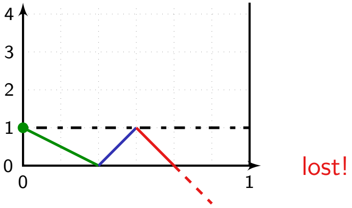


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

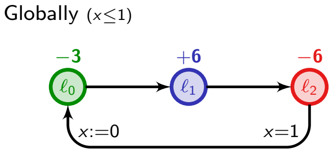
Globally $(x \leq 1)$

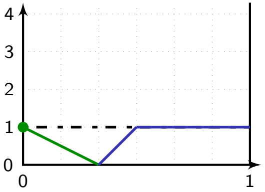




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

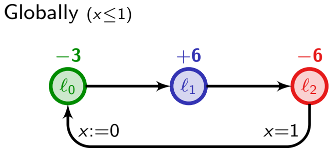
An example of resource management

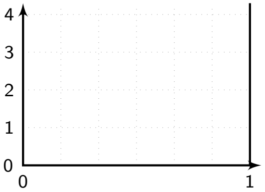




- Lower-bound problem
- Lower-upper-bound problem
- Lower-weak-upper-bound problem: can we "weakly" stay within bounds?

An example of resource management





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

Only partial results so far [BFLMS08]

0 clock!	**exist. problem**	**univ. problem**	**games**
L	$\in$ PTIME	$\in$ PTIME	$\in$ UP $\cap$ co-UP PTIME-hard
L+W	$\in$ PTIME	$\in$ PTIME	$\in$ NP $\cap$ co-NP PTIME-hard
L+U	$\in$ PSPACE NP-hard	$\in$ PTIME	EXPTIME-c.

[BFLMS08] Bouyer, Fahrenberg, Larsen, Markey, Srba. Infinite runs in weighted timed automata with energy constraints *(FORMATS'08)*.

Only partial results so far [BFLMS08]

1 clock	**exist. problem**	**univ. problem**	**games**
L	$\in$ PTIME	$\in$ PTIME	?
L+W	$\in$ PTIME	$\in$ PTIME	?
L+U	?	?	undecidable

[BFLMS08] Bouyer, Fahrenberg, Larsen, Markey, Srba. Infinite runs in weighted timed automata with energy constraints *(FORMATS'08)*.

Only partial results so far [BFLMS08]

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

[BFLMS08] Bouyer, Fahrenberg, Larsen, Markey, Srba. Infinite runs in weighted timed automata with energy constraints *(FORMATS'08)*.

Relation 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?

[DGR09] Doyen, Gentilini, Raskin. Faster Pseudo-Polynomial Algorithms for Mean-Payoff Games.

Relation 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.

[DGR09] Doyen, Gentilini, Raskin. Faster Pseudo-Polynomial Algorithms for Mean-Payoff Games.

Relation 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).

[DGR09] Doyen, Gentilini, Raskin. Faster Pseudo-Polynomial Algorithms for Mean-Payoff Games.

Relation 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:

[DGR09] Doyen, Gentilini, Raskin. Faster Pseudo-Polynomial Algorithms for Mean-Payoff Games.

Relation 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.

Corollary

Mean-payoff games (and hence parity games) and **L**-games have the same complexity (log-space reducibility).

$\rightsquigarrow$ a way to improve complexity of mean-payoff games [DGR09]

[DGR09] Doyen, Gentilini, Raskin. Faster Pseudo-Polynomial Algorithms for Mean-Payoff Games.

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

Theorem

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

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.

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.

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

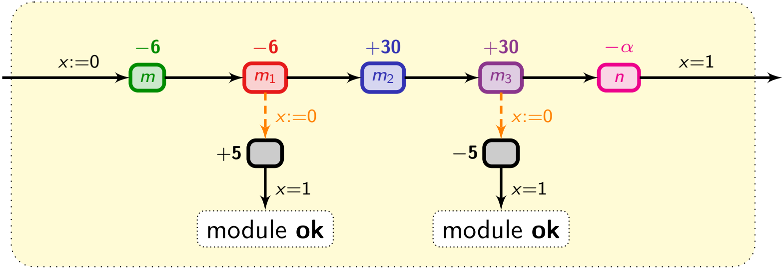
$$e = 5 - \frac{1}{2^{c_1} \cdot 3^{c_2}}$$

  when entering the corresponding module.

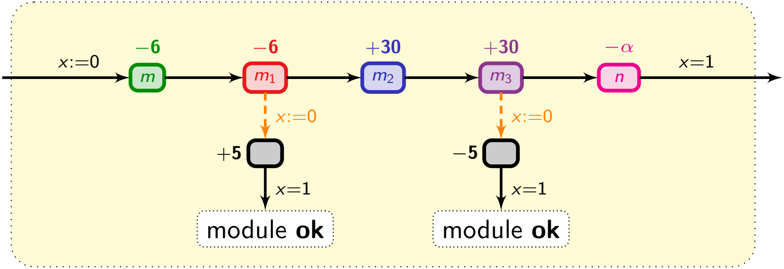
There is an infinite execution in the two-counter machine iff there is a strategy in the single-clock timed game under which the energy level remains between 0 and 5.

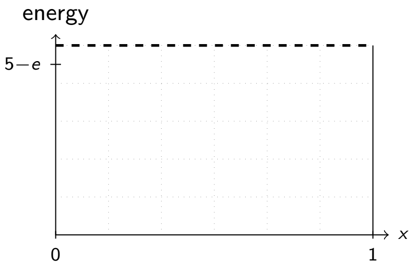
$\rightsquigarrow$ We present a generic construction for incrementing/decrementing the counters.

Generic module for incrementing/decrementing

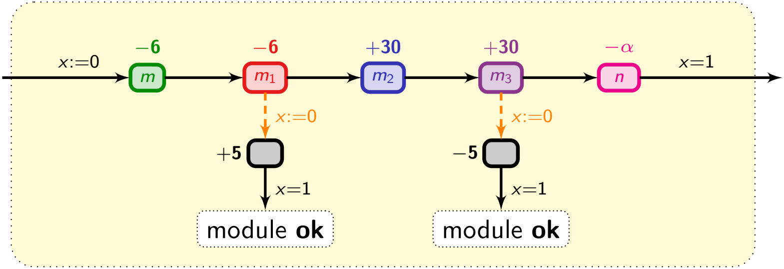


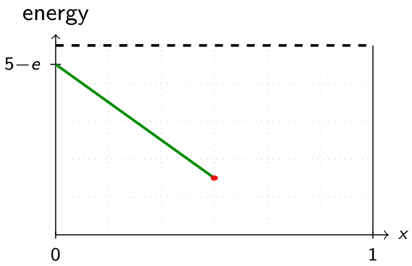
Generic module for incrementing/decrementing



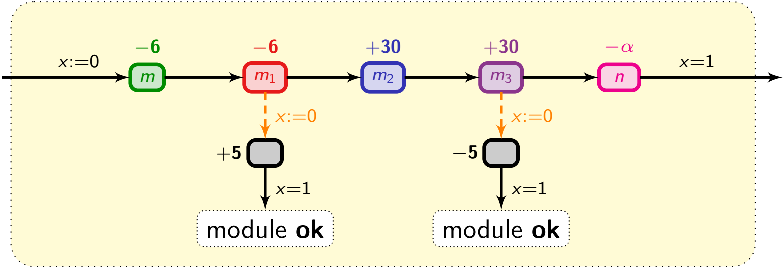


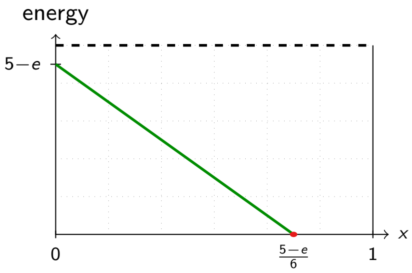
Generic module for incrementing/decrementing



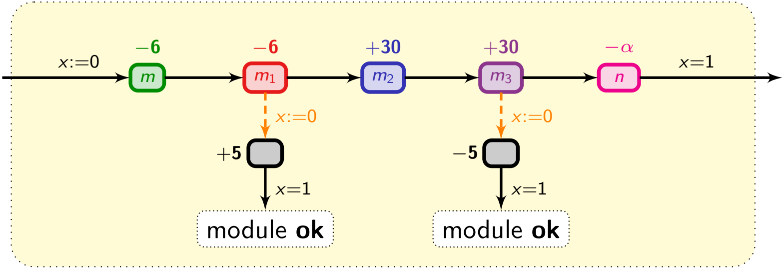


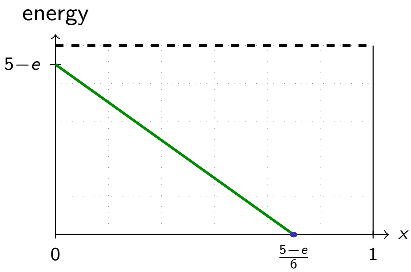
Generic module for incrementing/decrementing



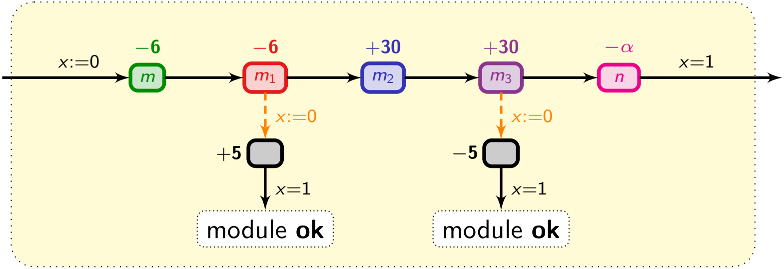


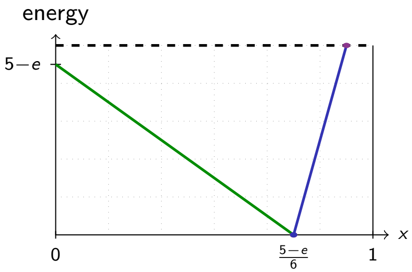
Generic module for incrementing/decrementing



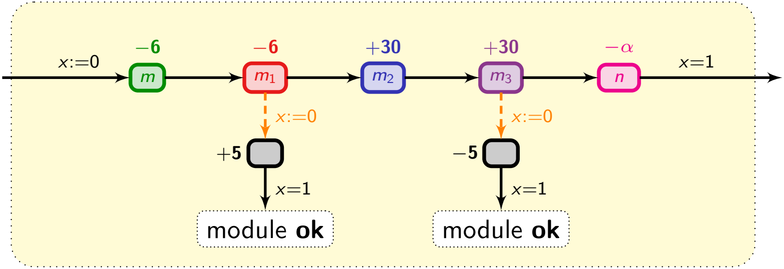


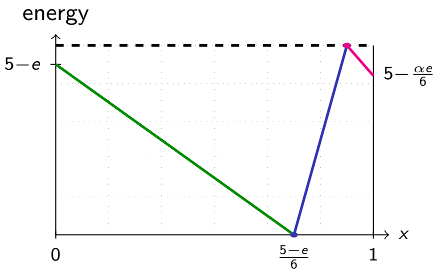
Generic module for incrementing/decrementing



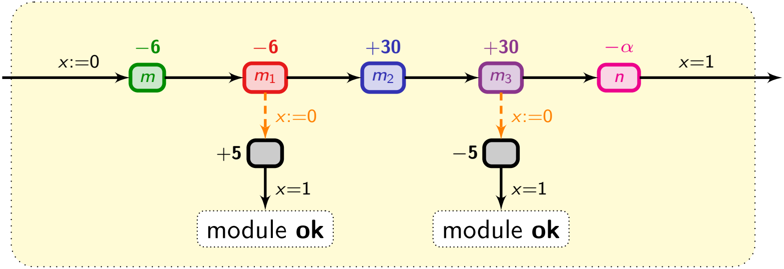


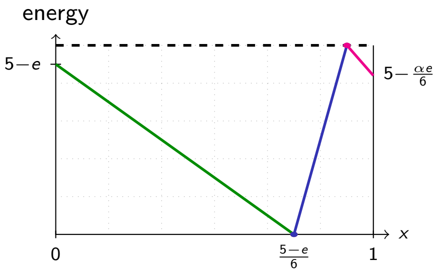
Generic module for incrementing/decrementing





Generic module for incrementing/decrementing





- α=3: increment c_1
- α=2: increment c_2
- α=12: decrement c_1
- α=18: decrement c_2

Outline

Some applications

Tools

- Uppaal (timed automata)
- Uppaal Cora (priced timed automata)
- Uppaal Tiga (timed games)

Case studies

- A lacquer production scheduling problem [BBHM05]
- Task graph scheduling problems [AKM03]
- An oil pump control problem [CJL+09]

[BBHM05] Behrmann, Brinksma, Hendriks, Mader. Scheduling lacquer production by reachability analysis - A case study *(IFAC'05)*.
[AKM03] Abdeddaïm, Kerbaa, Maler. Task graph scheduling using timed automata *(IPDPS'03)*.
[CJL+09] Cassez, Jessen, Larsen, Raskin, Reynier. Automatic synthesis of robust and optimal controllers - An industrial case study *(HSCC'09)*.

Task graph scheduling problems

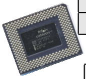
Compute $D\times(C\times(A+B))+(A+B)+(C\times D)$ using two processors:

P_1 (fast):

time	
$+$	2 picoseconds
$\times$	3 picoseconds

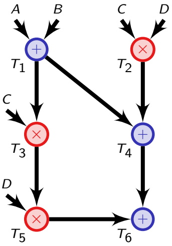
energy	
idle	10 Watt
in use	90 Watts

P_2 (slow):



time	
$+$	5 picoseconds
$\times$	7 picoseconds

energy	
idle	20 Watts
in use	30 Watts



[BFLM10] Bouyer, Fahrenberg, Larsen, Markey. Quantitative Analysis of Real-Time Systems using Priced Timed Automata.

Task graph scheduling problems

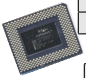
Compute $D\times(C\times(A+B))+(A+B)+(C\times D)$ using two processors:

P_1 (fast):

time	
+	2 picoseconds
×	3 picoseconds

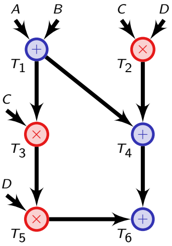
energy	
idle	10 Watt
in use	90 Watts

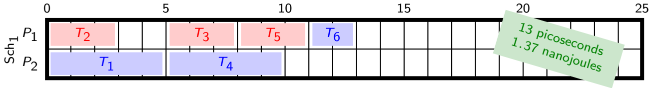
P_2 (slow):



time	
+	5 picoseconds
×	7 picoseconds

energy	
idle	20 Watts
in use	30 Watts





[BFLM10] Bouyer, Fahrenberg, Larsen, Markey. Quantitative Analysis of Real-Time Systems using Priced Timed Automata.

Task graph scheduling problems

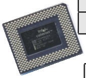
Compute $D\times(C\times(A+B))+(A+B)+(C\times D)$ using two processors:

P_1 (fast):

time	
+	2 picoseconds
×	3 picoseconds

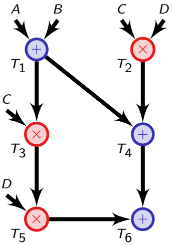
energy	
idle	10 Watt
in use	90 Watts

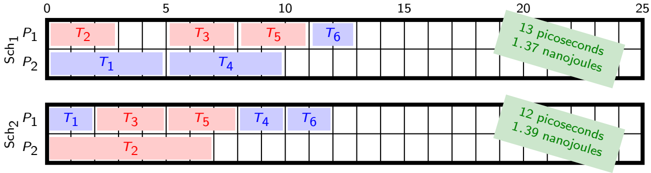
P_2 (slow):



time	
+	5 picoseconds
×	7 picoseconds

energy	
idle	20 Watts
in use	30 Watts





[BFLM10] Bouyer, Fahrenberg, Larsen, Markey. Quantitative Analysis of Real-Time Systems using Priced Timed Automata.

Task graph scheduling problems

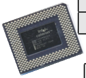
Compute $D\times(C\times(A+B))+(A+B)+(C\times D)$ using two processors:

P_1 (fast):

time	
$+$	2 picoseconds
$\times$	3 picoseconds

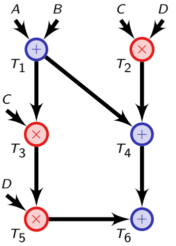
energy	
idle	10 Watt
in use	90 Watts

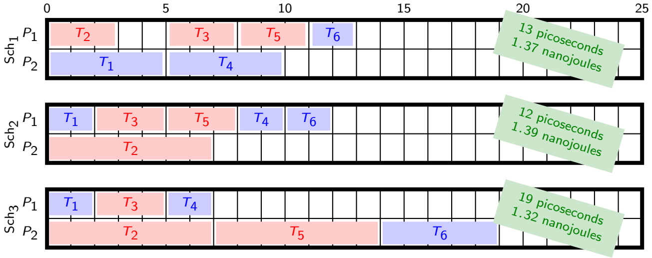
P_2 (slow):



time	
$+$	5 picoseconds
$\times$	7 picoseconds

energy	
idle	20 Watts
in use	30 Watts



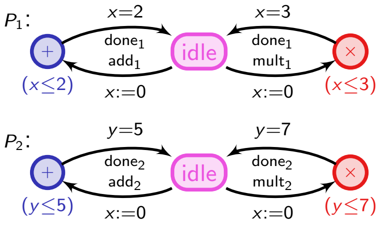


[BFLM10] Bouyer, Fahrenberg, Larsen, Markey. Quantitative Analysis of Real-Time Systems using Priced Timed Automata.

Modelling the task graph scheduling problem

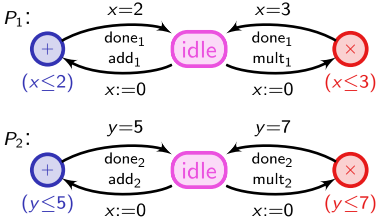
Modelling the task graph scheduling problem

- Processors

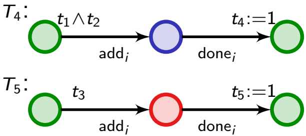


Modelling the task graph scheduling problem

- Processors

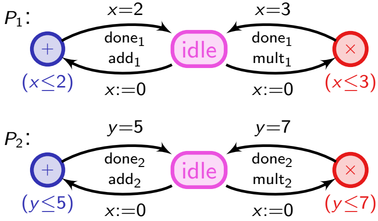


- Tasks

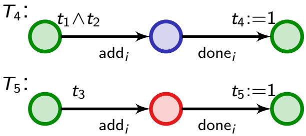


Modelling the task graph scheduling problem

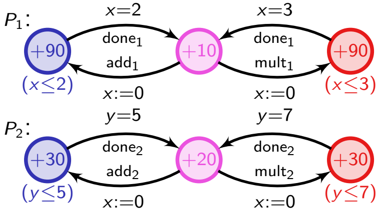
- Processors



- Tasks

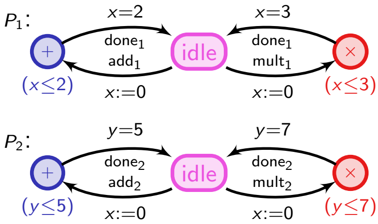


- Modelling energy

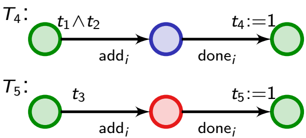


Modelling the task graph scheduling problem

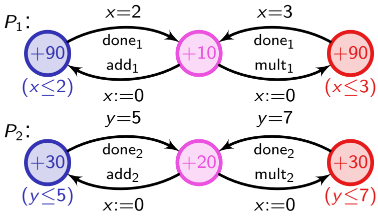
- Processors



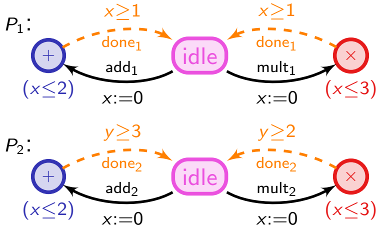
- Tasks



- Modelling energy



- Modelling uncertainty



Conclusion

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

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

Conclusion

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

    ... and not all of them have been answered!
- 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]
  - 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 for modelling resources in timed systems
  - natural (optimization/management) questions have been posed...

    ... and not all of them have been answered!
- 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]
  - various tools have been developed:

    Uppaal, Uppaal Cora, Uppaal Tiga
- Current and further work:
  - further cost functions (e.g. exponential)
  - 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?
  - computation of equilibria
  - ...