On the reduction of energy consumption

Patricia Bouyer

LSV – CNRS & ENS de Cachan – France

Oxford University Computing Laboratory – UK

On the reduction of energy consumption
— A timed automaton approach —

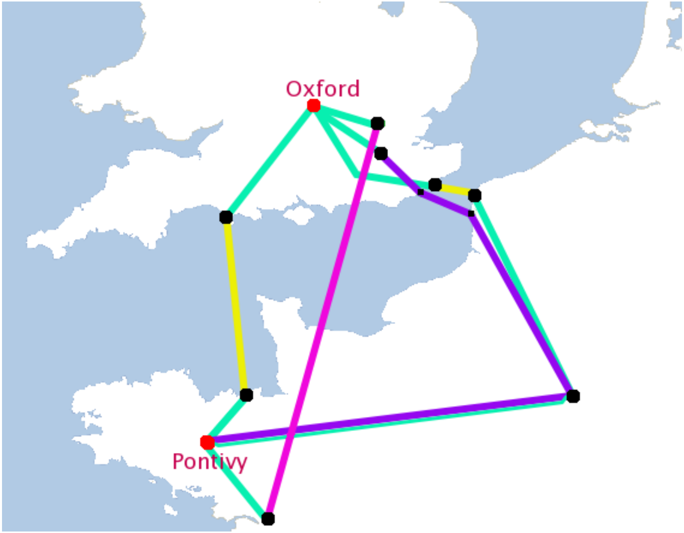
Patricia Bouyer

LSV – CNRS & ENS de Cachan – France

Oxford University Computing Laboratory – UK

Outline

A starting example



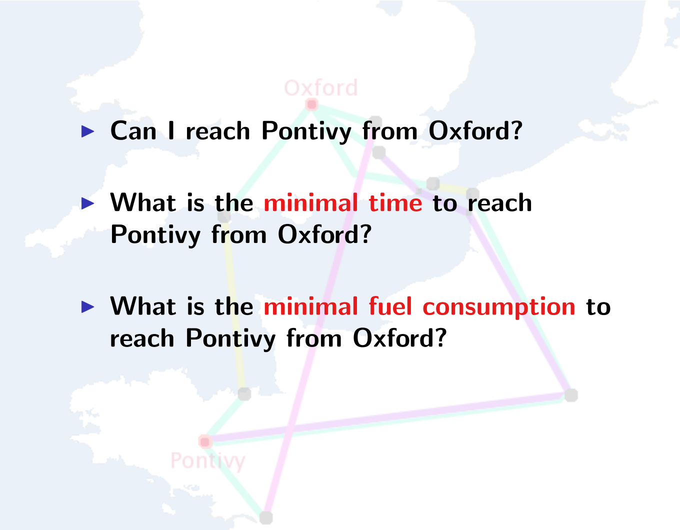
Natural questions

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

Natural questions

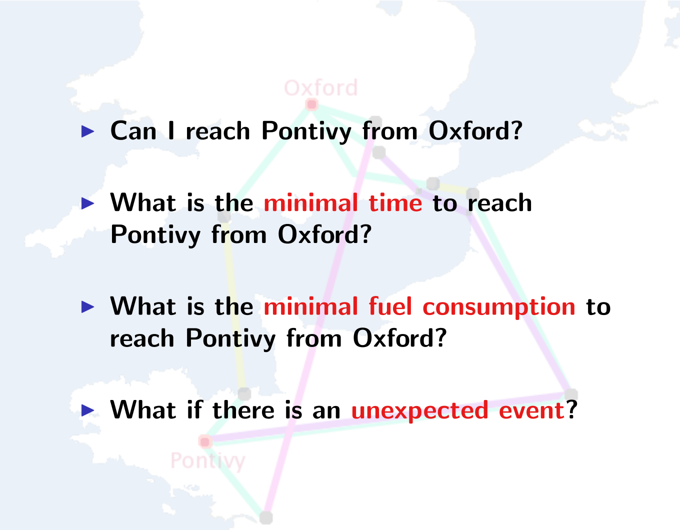
- **Can I reach Pontivy from Oxford?**
- **What is the minimal time to reach Pontivy from Oxford?**

Natural questions



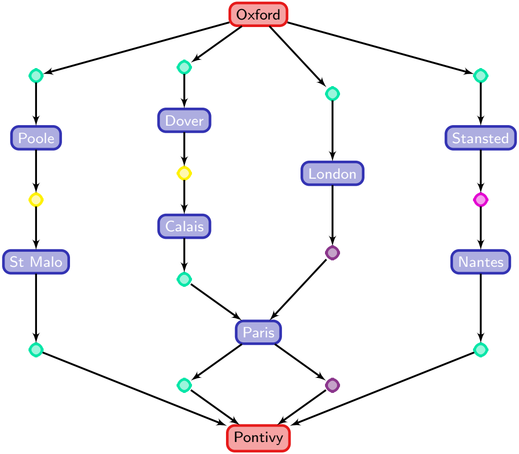
- **Can I reach Pontivy from Oxford?**
- **What is the minimal time to reach Pontivy from Oxford?**
- **What is the minimal fuel consumption to reach Pontivy from Oxford?**

Natural questions

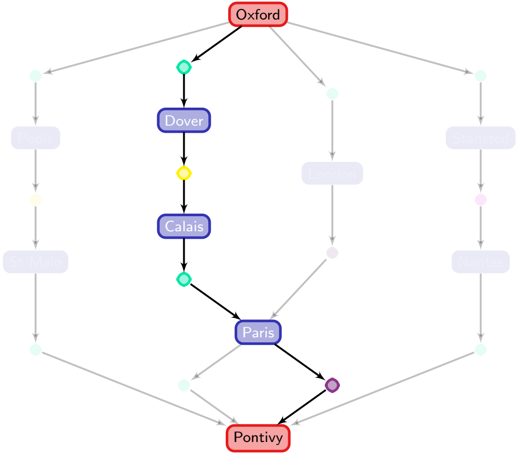


- **Can I reach Pontivy from Oxford?**
- **What is the minimal time to reach Pontivy from Oxford?**
- **What is the minimal fuel consumption to reach Pontivy from Oxford?**
- **What if there is an unexpected event?**

A first model of the system

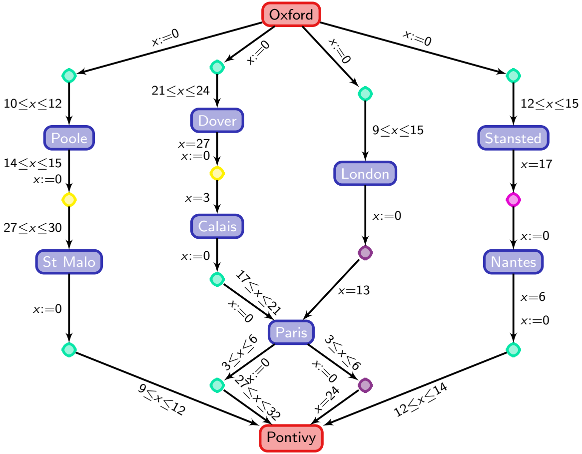


Can I reach Pontivy from Oxford?

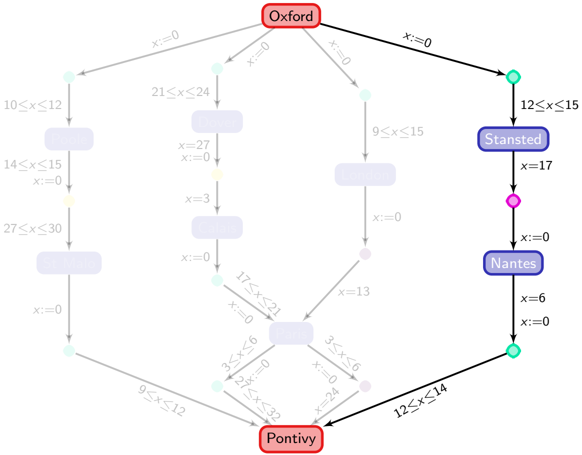


This is a reachability question in a finite graph: Yes, I can!

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!

Timed automata

Timed automata

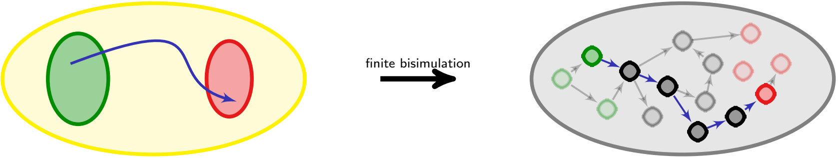
Theorem [AD90]

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

Timed automata

Theorem [AD90]

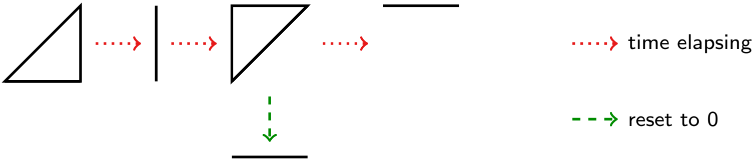
The reachability problem is decidable (and PSPACE-complete) for timed automata.



timed automaton

large (but finite) automaton
(region automaton)

The region abstraction

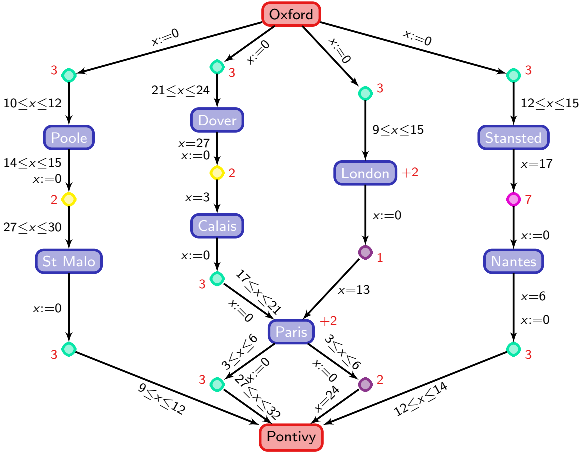


Time-optimal reachability

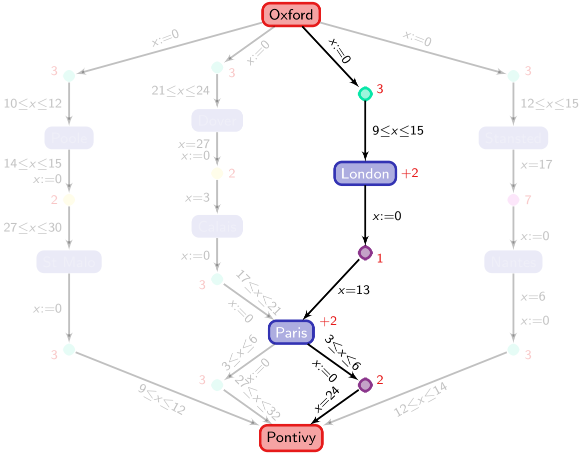
Theorem [CY92]

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

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!

Outline

2001: the space odyssey

2001: the space odyssey of weighted timed automata

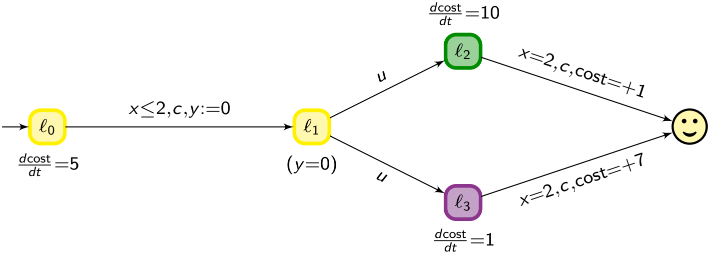
2001: the space odyssey of weighted/priced timed automata



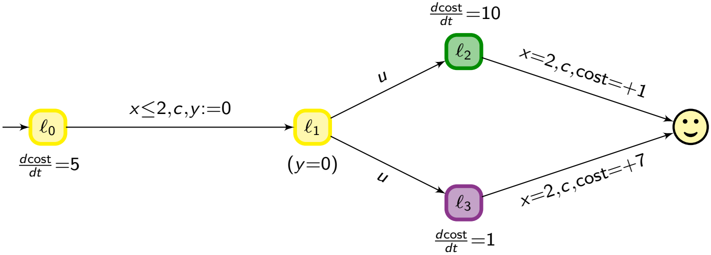

?



A simple example



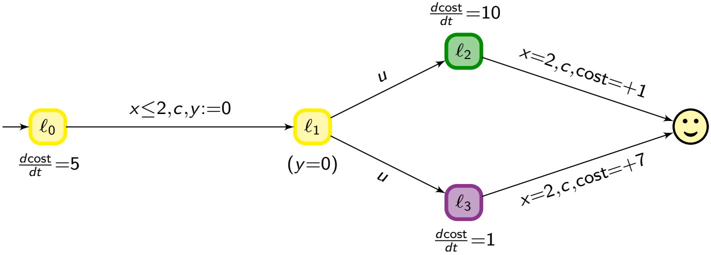
A simple example



$$(\ell_0,(0,0)) \xrightarrow{1.3} (\ell_0,(1.3,1.3)) \xrightarrow{c} (\ell_1,(1.3,0)) \xrightarrow{u} (\ell_3,(1.3,0)) \xrightarrow{0.7} (\ell_3,(2,0.7)) \xrightarrow{c} ☺$$

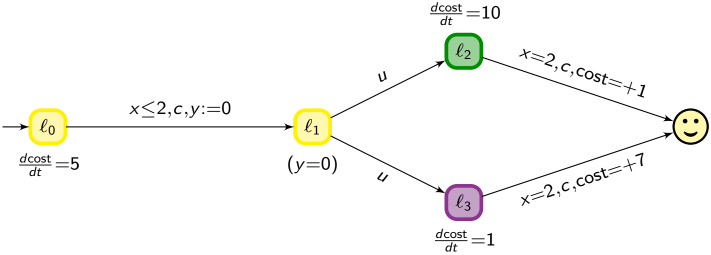
cost : $6.5 + 0 + 0 + 0.7 + 7 = 14.2$

A simple example



Question: what is the optimal cost for reaching ☺?

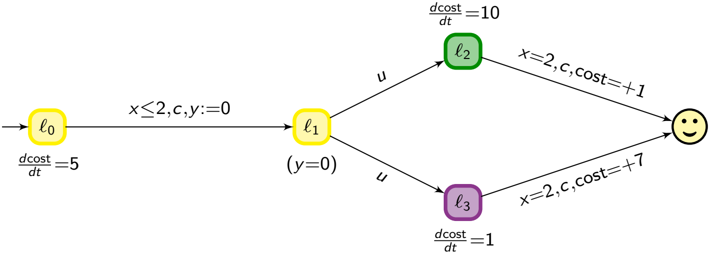
A simple example



Question: what is the optimal cost for reaching ☺?

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

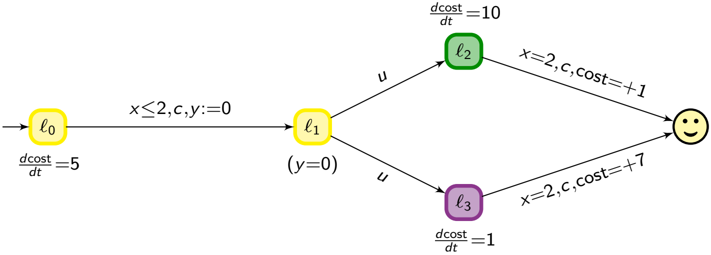
A simple example



Question: what is the optimal cost for reaching ☺?

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

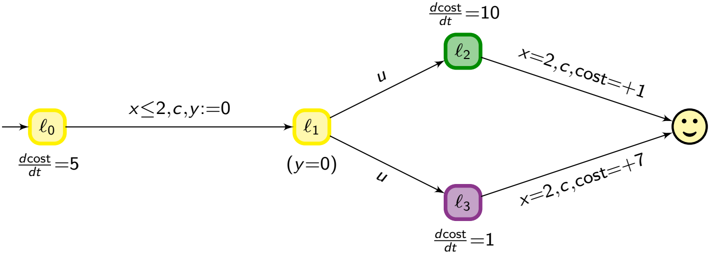
A simple example



Question: what is the optimal cost for reaching ☺?

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

A simple example



Question: what is the optimal cost for reaching ☺?

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

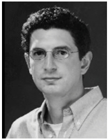
➜ **strategy:** leave immediately ℓ_0, go to ℓ_3, and wait there 2 t.u.

The idea “go through corners” extends in the general case.

Theorem

Optimal reachability is decidable in timed automata.







?

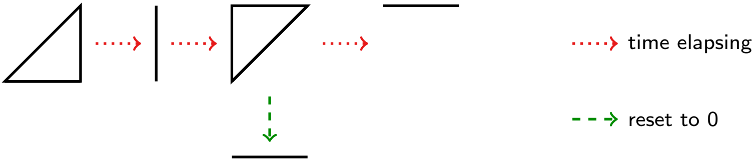



The idea "go through corners" extends in the general case.

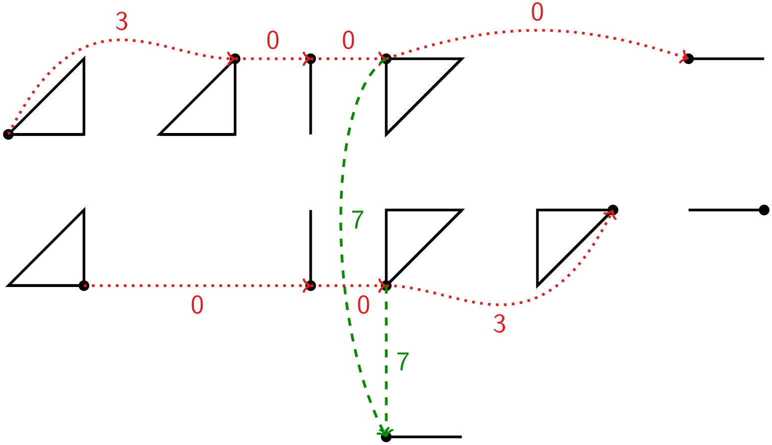
Theorem

Optimal reachability is decidable in timed automata. It is PSPACE-complete.

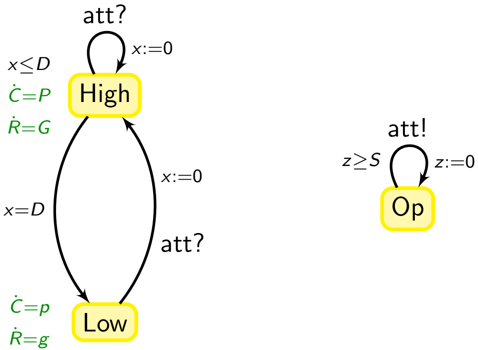
The region abstraction is not fine enough



The corner-point abstraction



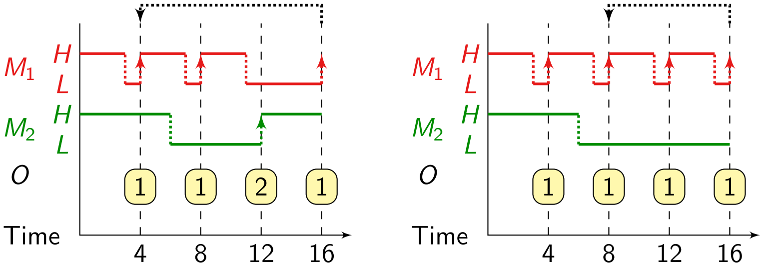
Mean-Cost Optimization



Question: How to minimize $\lim_{n\to+\infty} \frac{\text{accumulated cost}(\pi_n)}{\text{accumulated reward}(\pi_n)}$?

An example

Two machines $M_1(D = 3, P = 3, G = 4, p = 5, g = 3)$ and $M_2(D = 6, P = 3, G = 2, p = 5, g = 2)$.
An operator $O(4)$.



Schedule with ratio 1.455 Schedule with ratio 1.478

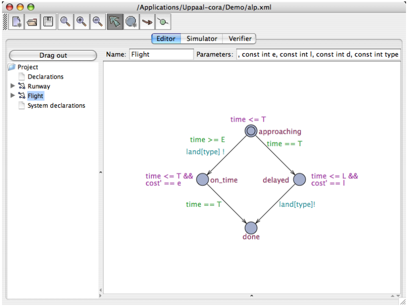
Mean-cost optimization

Theorem [BBL04]

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

☞ The corner-point abstraction is sound and complete.

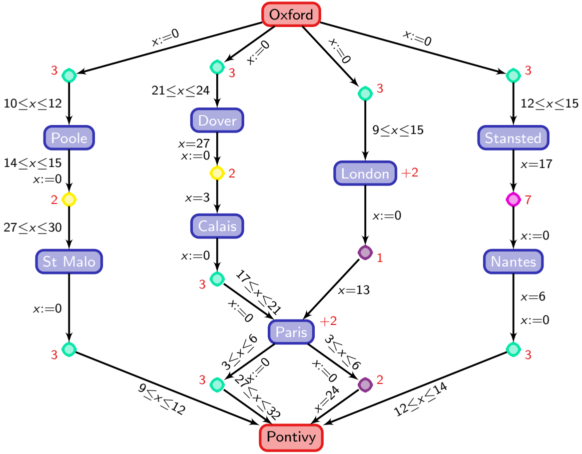
Uppaal Cora



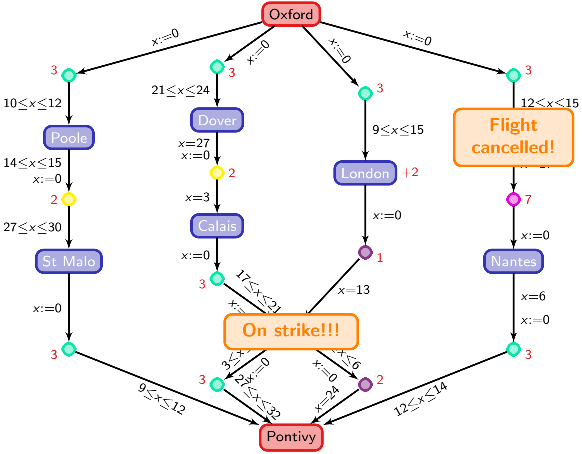
A branch of Uppaal for cost optimal reachability

Outline

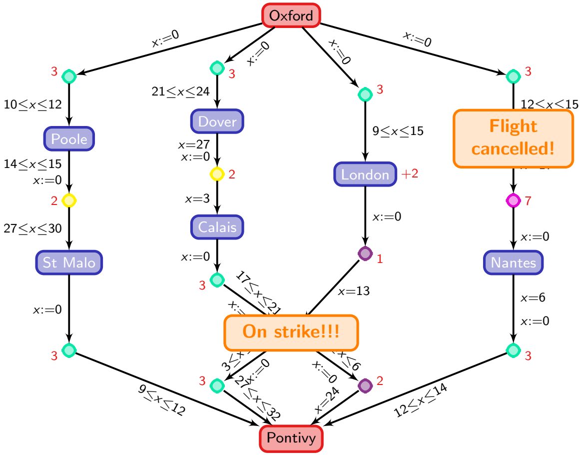
What if an unexpected event happens?



What if an unexpected event happens?

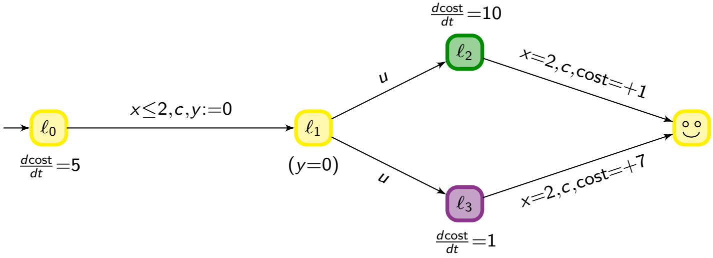


What if an unexpected event happens?

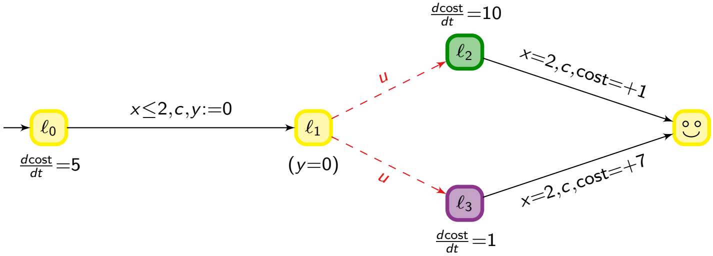


☞ modelled as timed games

A simple example of timed games



A simple example of timed games



Decidability of timed games




?

Decidability of timed games




?

Theorem [AMPS98] [HK99]

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

Decidability of timed games




?

Theorem [AMPS98] [HK99]

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

(the attractor is computable...)

Decidability of timed games




?

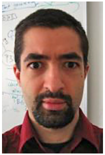
Theorem [AMPS98] [HK99]

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

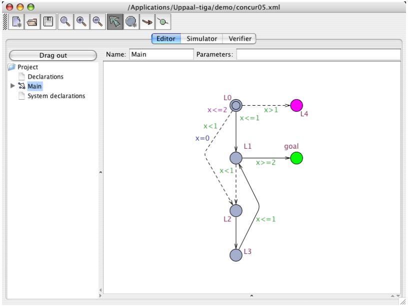
(the attractor is computable...)

☞ classical regions are sufficient for solving such problems

Uppaal Tiga

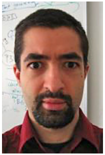


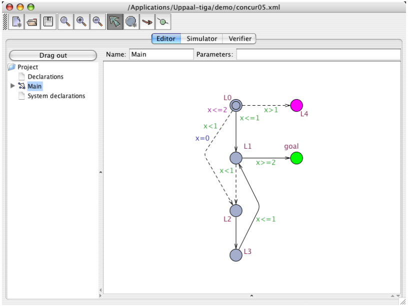


A forward on-the-fly algorithm for solving reachability timed games

☞ implemented as a branch of Uppaal

Uppaal Tiga

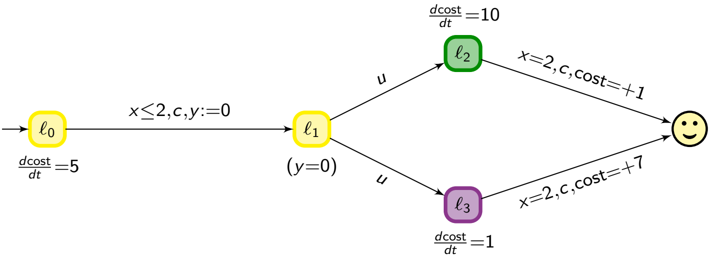



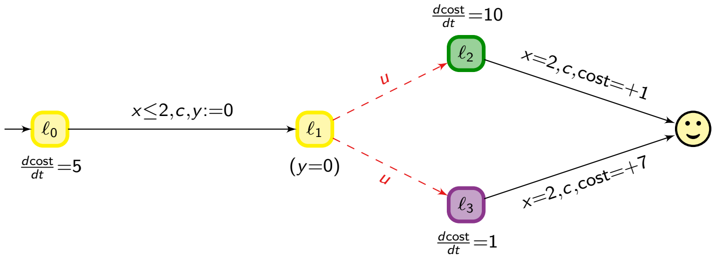
A forward on-the-fly algorithm for solving reachability timed games

☞ implemented as a branch of Uppaal

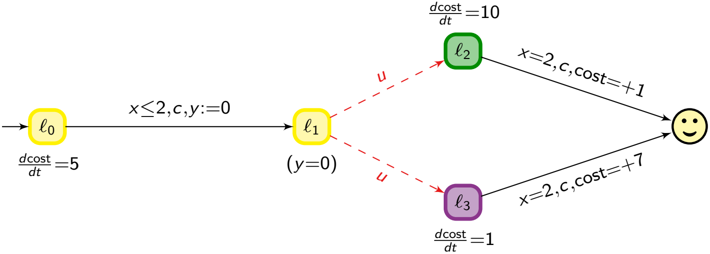
Back to the simple example



Back to the simple example

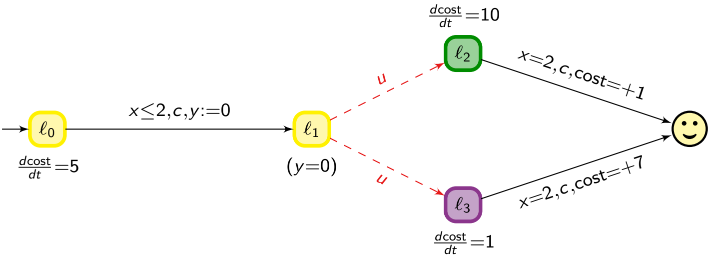


Back to the simple example



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

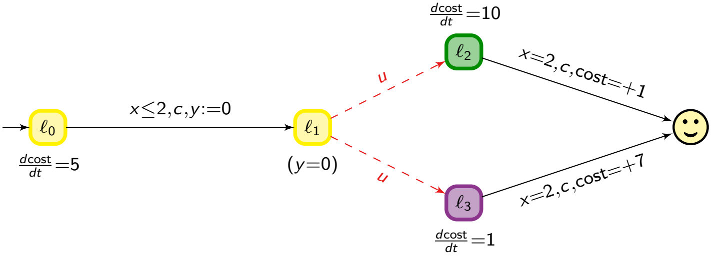
Back to the simple example



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

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

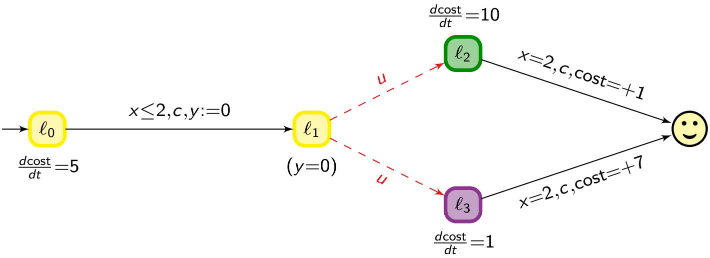
Back to the simple example



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

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

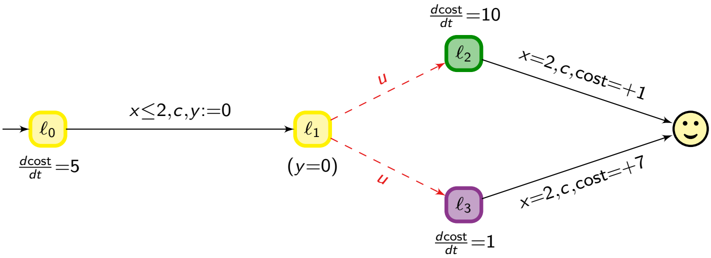
Back to the simple example



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

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

Back to the simple example

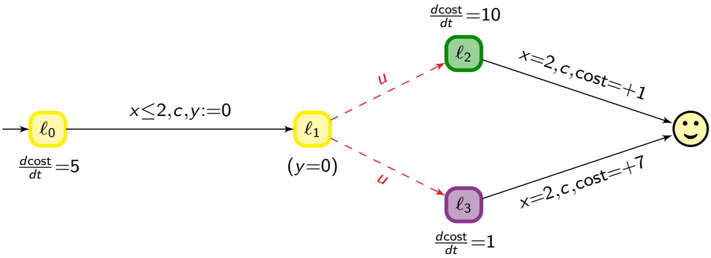


Question: what is the optimal cost we can ensure in state ℓ_0?

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

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

Back to the simple example



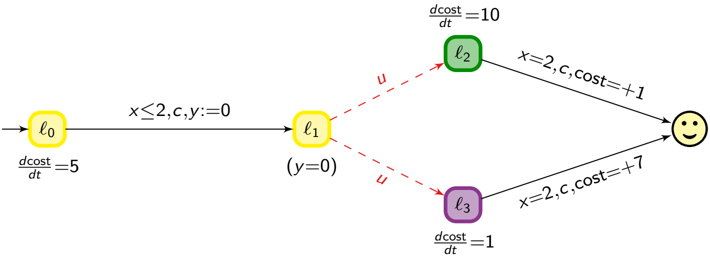
Question: what is the optimal cost we can ensure in state ℓ_0?

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

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

- How to automatically compute such optimal costs?

Back to the simple example



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

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

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

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

A fairly hot topic!

- optimal time is computable in timed games

A fairly hot topic!

- optimal time is computable in timed games


- case of acyclic games



A fairly hot topic!

- optimal time is computable in timed games


- case of acyclic games



- general case


?


  - complexity of k-step games
  - under a strongly non-*zeno* assumption, optimal cost is computable

A fairly hot topic!

- optimal time is computable in timed games


- case of acyclic games

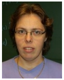


- general case


?


  - complexity of k-step games
  - under a strongly non-*zeno* assumption, optimal cost is computable

- general case




  - structural properties of strategies (e.g. memory)
  - under a strongly non-*zeno* assumption, optimal cost is computable

A fairly hot topic!

- general case



  - with five clocks, optimal cost is not computable!
  - with one clock and one stopwatch cost, optimal cost is computable

A fairly hot topic!

- general case

  - with five clocks, optimal cost is not computable!
  - with one clock and one stopwatch cost, optimal cost is computable

- general case

  - with three clocks, optimal cost is not computable

A fairly hot topic!

- general case

  - with five clocks, optimal cost is not computable!
  - with one clock and one stopwatch cost, optimal cost is computable

- general case

  - with three clocks, optimal cost is not computable

- the single-clock case

  - with one clock, optimal cost is computable

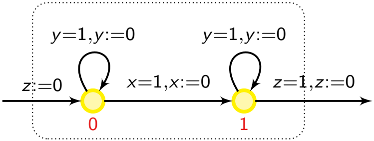
Why is that hard?

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

Why is that hard?

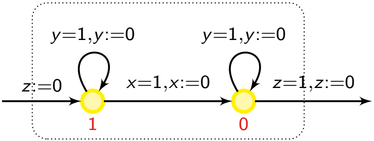
Given two clocks x and y, we can check whether $y = 2x$

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



The cost is increased by x_0

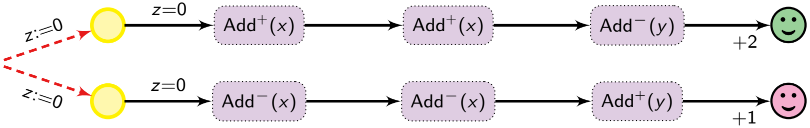
$\mathsf{Add}^-(x)$



The cost is increased by $1-x_0$

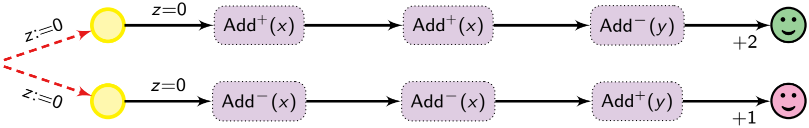
Why is that hard?

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



Why is that hard?

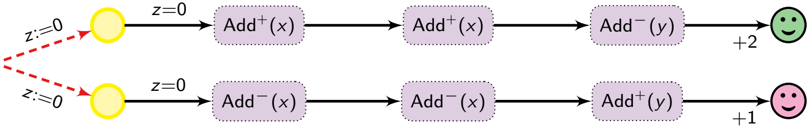
Given two clocks x and y, we can check whether $y = 2x$



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

Why is that hard?

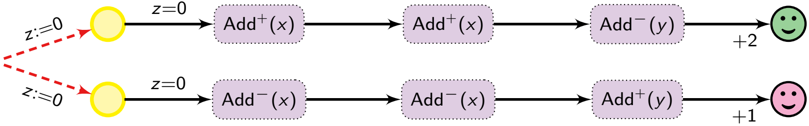
Given two clocks x and y, we can check whether $y = 2x$



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

Why is that hard?

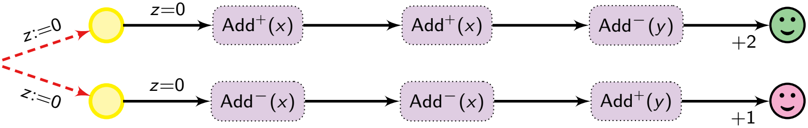
Given two clocks x and y, we can check whether $y = 2x$



- 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

Why is that hard?

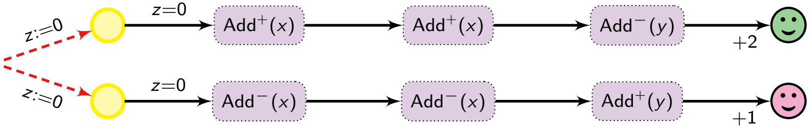
Given two clocks x and y, we can check whether $y = 2x$



- 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

Why is that hard?

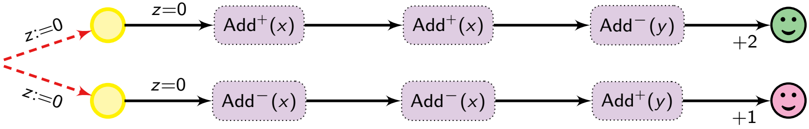
Given two clocks x and y, we can check whether $y = 2x$



- 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 $= 3$

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 $y_0 = 2x_0$

Outline

Conclusion

Priced timed automata, a model and framework to represent quantitative constraints on timed systems.

Conclusion

Priced timed automata, a model and framework to represent quantitative constraints on timed systems.

Not mentioned here

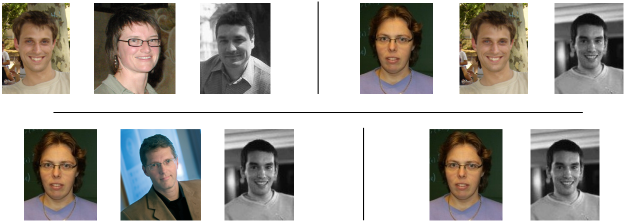
- all works on model-checking issues (extensions of CTL, LTL)
  - very few decidability results

Conclusion

Priced timed automata, a model and framework to represent quantitative constraints on timed systems.

Not mentioned here

- all works on model-checking issues (extensions of CTL, LTL)
  - very few decidability results



Further work

- approximate optimal timed games to circumvent undecidability results