Optimal strategies in weighted timed games: undecidability and approximation

Nicolas Markey

LSV, CNRS & ENS Cachan, France

(joint work with Patricia Bouyer and Samy Jaziri)

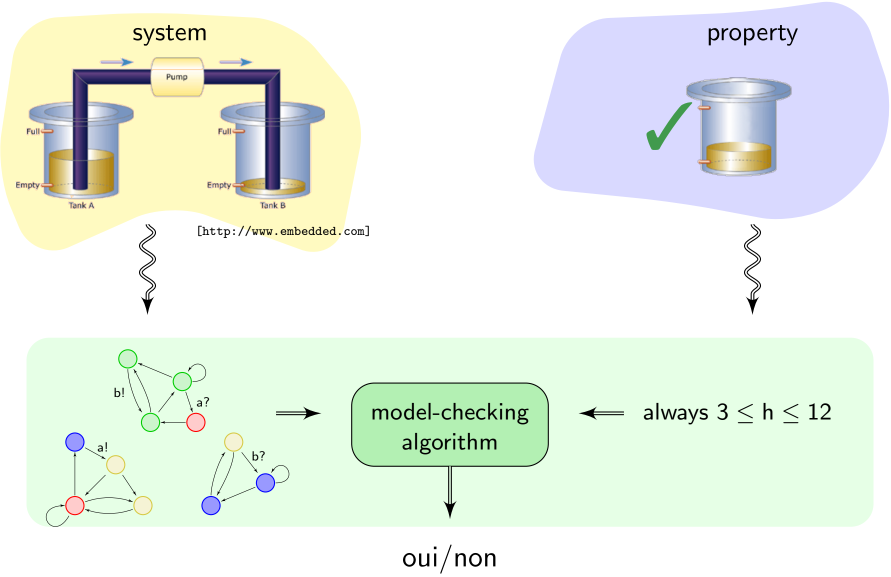
68 NQRT seminar – Rennes, France

October 1, 2015

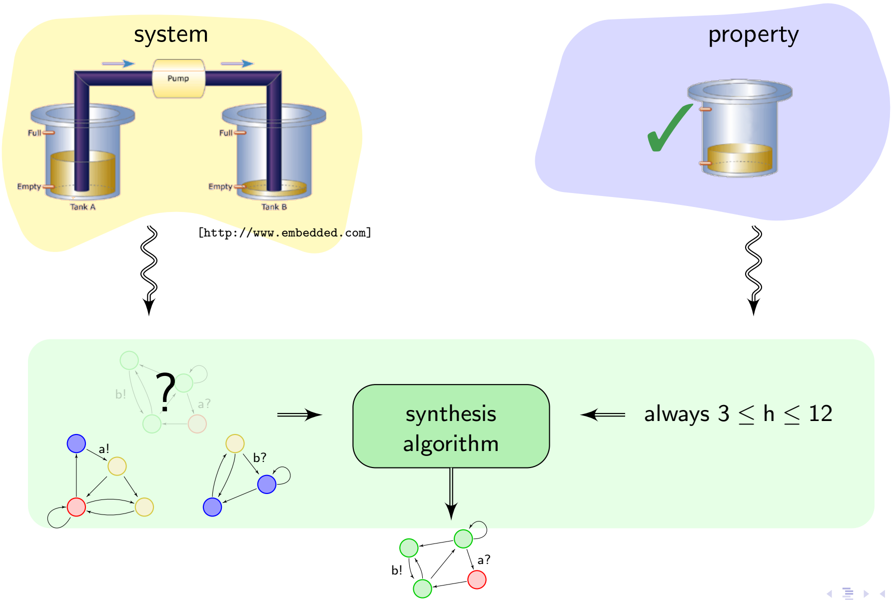
EQualIS

Cassting

Model checking and *synthesis*

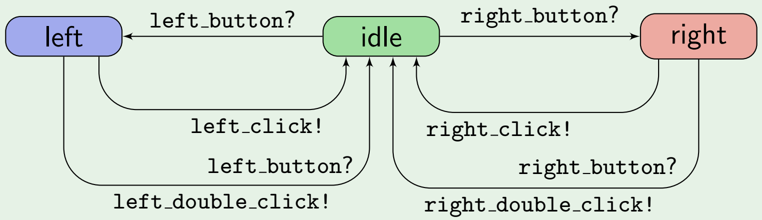


Model checking and *synthesis*



Reasoning about real-time systems

Example (A computer mouse)



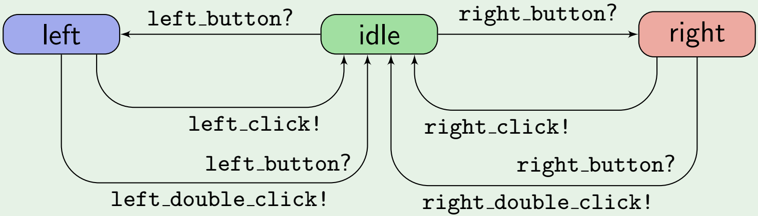
Reasoning about real-time systems

Definition ([AD90])

A timed automaton is made of

- a transition system,

Example (A computer mouse)



[AD90] Alur, Dill. Automata For Modeling Real-Time Systems. ICALP, 1990.

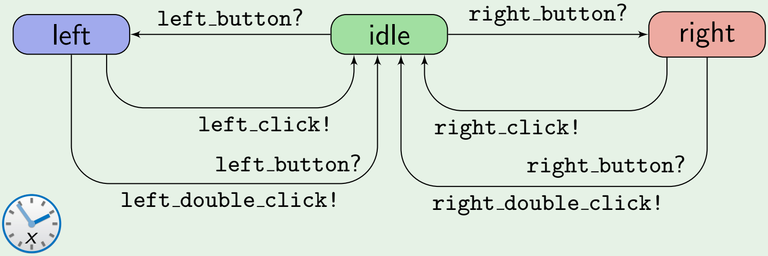
Reasoning about real-time systems

Definition ([AD90])

A timed automaton is made of

- a transition system,
- a set of clocks,

Example (A computer mouse)



[AD90] Alur, Dill. Automata For Modeling Real-Time Systems. ICALP, 1990.

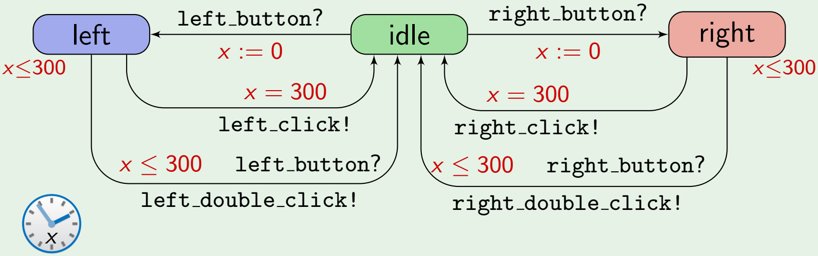
Reasoning about real-time systems

Definition ([AD90])

A timed automaton is made of

- a transition system,
- a set of clocks,
- timing constraints on states and transitions.

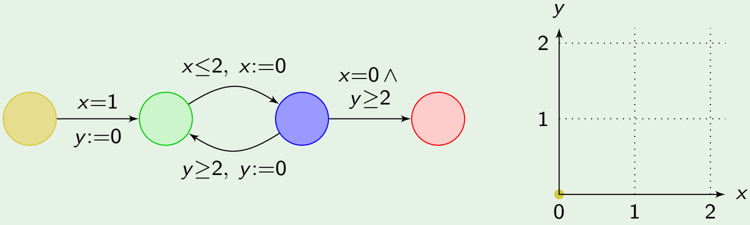
Example (A computer mouse)



[AD90] Alur, Dill. Automata For Modeling Real-Time Systems. ICALP, 1990.

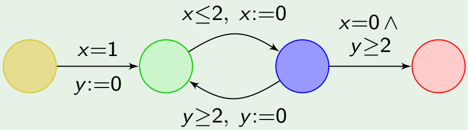
Continuous-time semantics

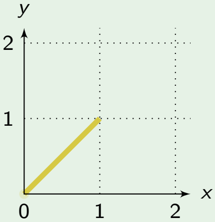
Example



Continuous-time semantics

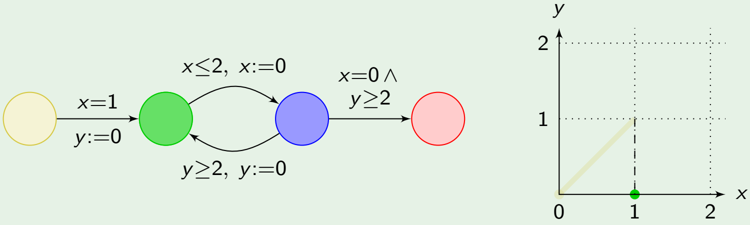
Example





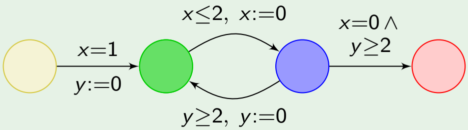
Continuous-time semantics

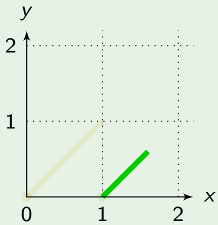
Example



Continuous-time semantics

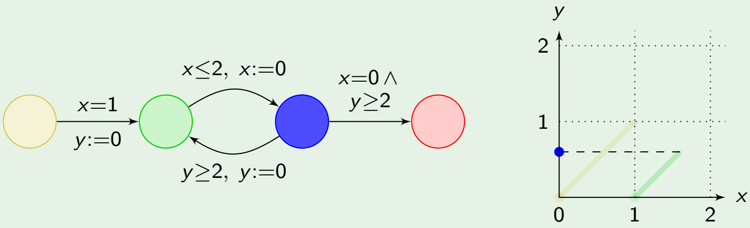
Example





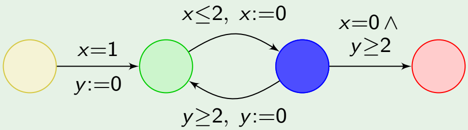
Continuous-time semantics

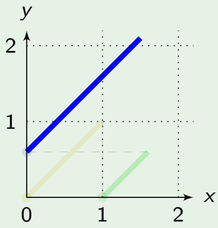
Example



Continuous-time semantics

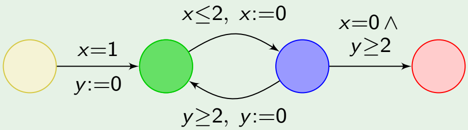
Example

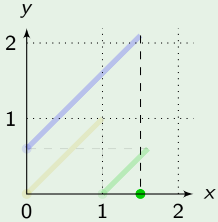




Continuous-time semantics

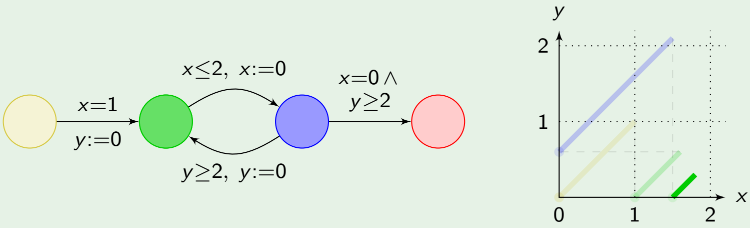
Example





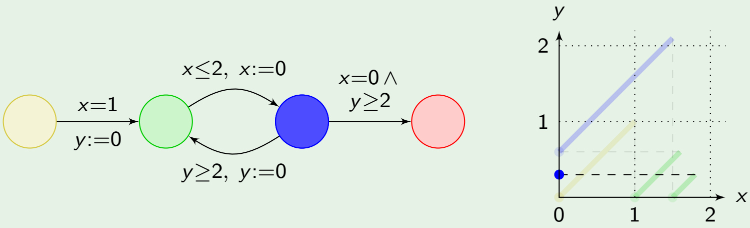
Continuous-time semantics

Example



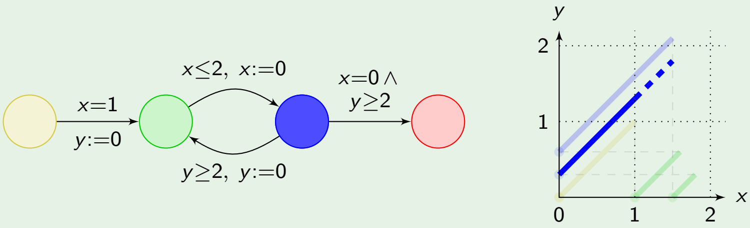
Continuous-time semantics

Example



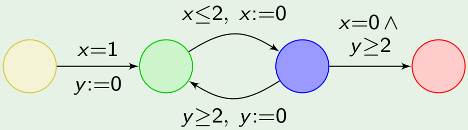
Continuous-time semantics

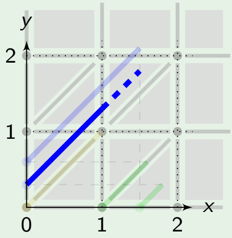
Example



Continuous-time semantics

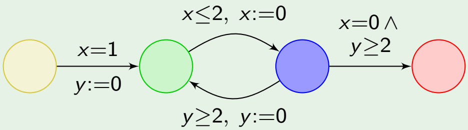
Example

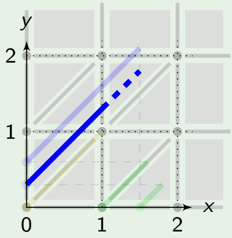




Continuous-time semantics

Example



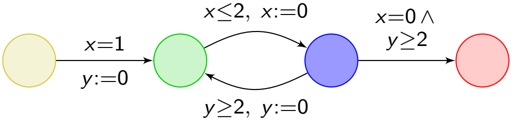


Theorem ([AD90,ACD93, ...])

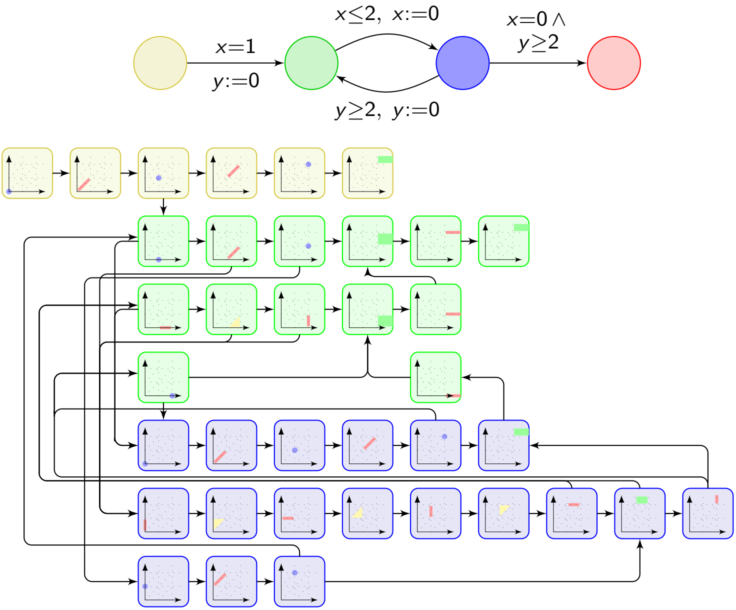
Reachability in timed automata is decidable (as well as many other important properties).

[AD90] Alur, Dill. Automata For Modeling Real-Time Systems. ICALP, 1990.
[ACD93] Alur, Courcoubetis, Dill. Model-Checking in Dense Real-Time. Inf. & Comp., 1993.

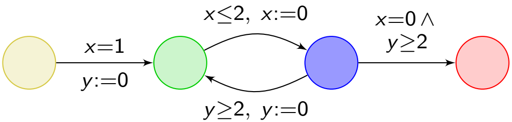
Region automaton

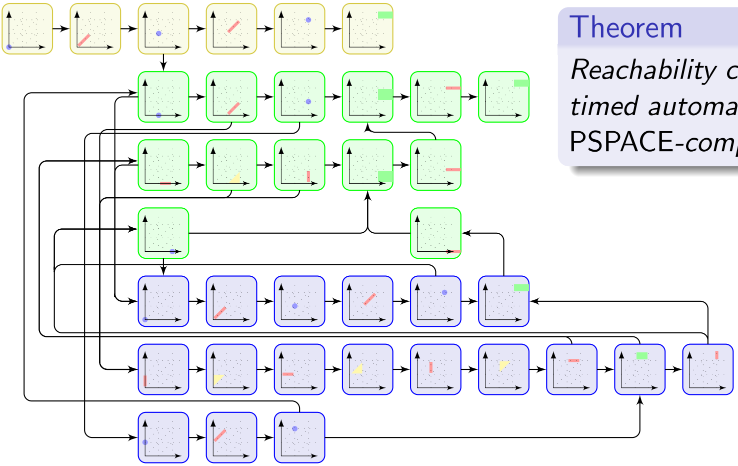


Region automaton



Region automaton





Theorem

Reachability checking in timed automata is PSPACE-*complete.*

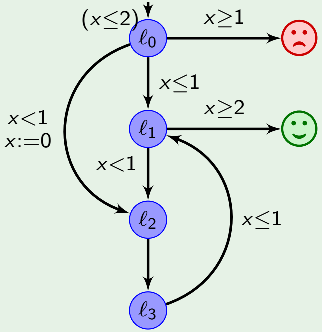
Timed games

Definition

A timed game is made of

- a timed automaton;

Example



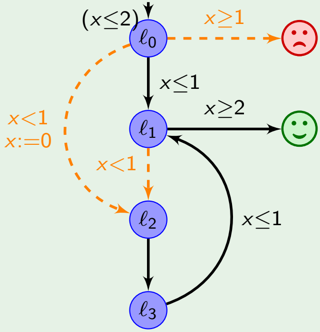
Timed games

Definition

A timed game is made of

- a timed automaton;
- a partition between controllable and uncontrollable transitions.

Example



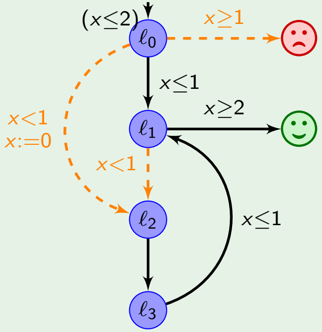
Timed games

Definition

A timed game is made of

- a timed automaton;
- a partition between controllable and uncontrollable transitions.

Example



a memoryless strategy

in $(\ell_0, x = 0)$: wait 0.5
goto ℓ_1

in (ℓ_1, x): wait until $x = 2$
goto ☺

in $(\ell_2, x \leq 1)$: wait until $x = 1$
goto ℓ_3

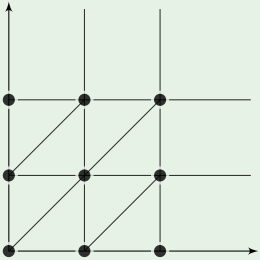
in $(\ell_3, x \leq 1)$: wait until $x = 1$
goto ℓ_1

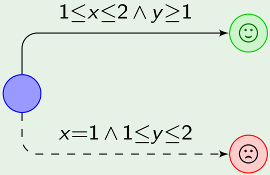
Timed games

Theorem ([AMPS98])

Deciding the winner in a timed game (e.g. for reachability objectives) is EXPTIME-complete.

Proof





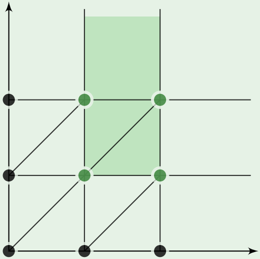
[AMPS98] Asarin, Maler, Pnueli, Sifakis. Controller synthesis for timed automata. SSC, 1998.

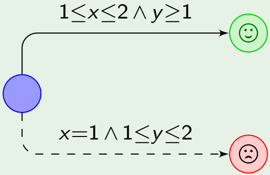
Timed games

Theorem ([AMPS98])

Deciding the winner in a timed game (e.g. for reachability objectives) is EXPTIME-complete.

Proof





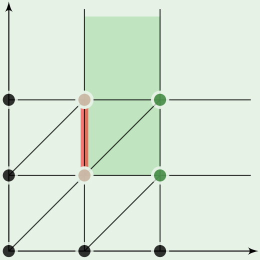
[AMPS98] Asarin, Maler, Pnueli, Sifakis. Controller synthesis for timed automata. SSC, 1998.

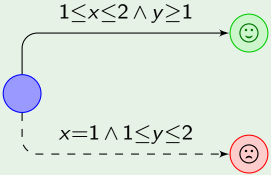
Timed games

Theorem ([AMPS98])

Deciding the winner in a timed game (e.g. for reachability objectives) is EXPTIME-*complete.*

Proof





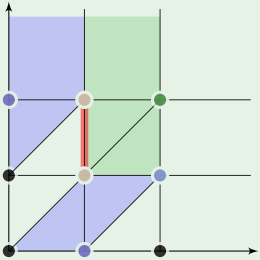
[AMPS98] Asarin, Maler, Pnueli, Sifakis. Controller synthesis for timed automata. SSC, 1998.

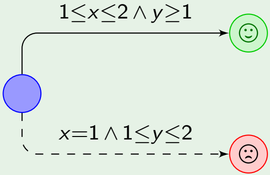
Timed games

Theorem ([AMPS98])

Deciding the winner in a timed game (e.g. for reachability objectives) is EXPTIME-complete.

Proof





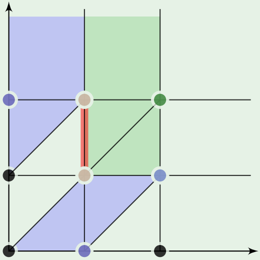
[AMPS98] Asarin, Maler, Pnueli, Sifakis. Controller synthesis for timed automata. SSC, 1998.

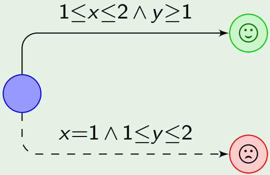
Timed games

Theorem ([AMPS98])

Deciding the winner in a timed game (e.g. for reachability objectives) is EXPTIME-complete.

Proof





- regions are sufficient;
- the computation terminates.

[AMPS98] Asarin, Maler, Pnueli, Sifakis. Controller synthesis for timed automata. SSC, 1998.

Outline of the talk

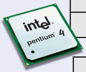
1. Introduction: timed automata and timed games
2. Measuring extra quantities in timed automata
   - Example: task graph scheduling
   - Timed automata with observer variables
3. Cost-optimal strategies
   - Optimal reachability in priced timed automata
   - Optimal reachability in priced timed games
4. Conclusions and future works

Outline of the talk

Example: task graph scheduling

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

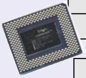
P_1 (fast):



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

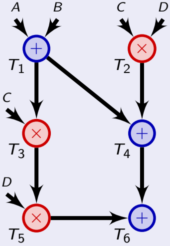
energy	
idle	10 Watt
in use	90 Watts

P_2 (slow):



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

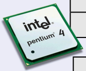
energy	
idle	20 Watts
in use	30 Watts



Example: task graph scheduling

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

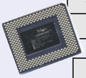
P_1 (fast):



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

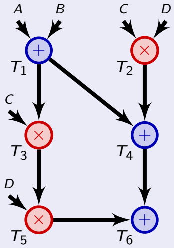
energy	
idle	10 Watt
in use	90 Watts

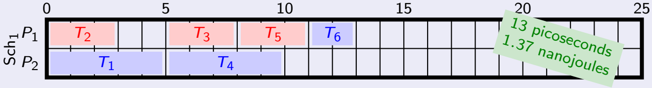
P_2 (slow):



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

energy	
idle	20 Watts
in use	30 Watts

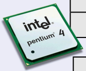




Example: task graph scheduling

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

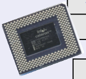
P_1 (fast):



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

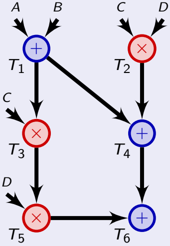
energy	
idle	10 Watt
in use	90 Watts

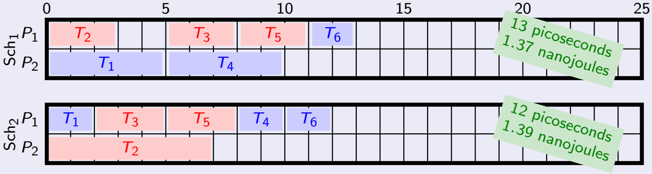
P_2 (slow):



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

energy	
idle	20 Watts
in use	30 Watts

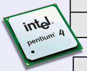




Example: task graph scheduling

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

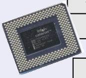
P_1 (fast):



	time
+	2 picoseconds
×	3 picoseconds

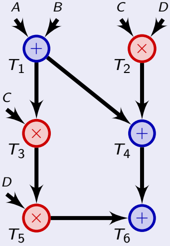
	energy
idle	10 Watt
in use	90 Watts

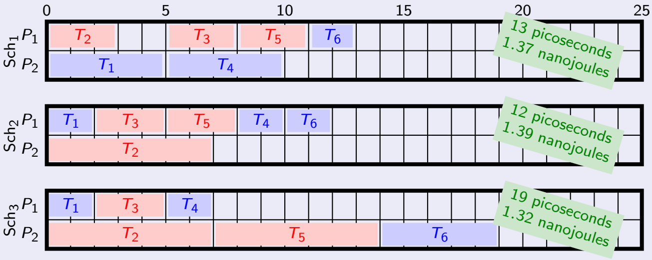
P_2 (slow):



	time
+	5 picoseconds
×	7 picoseconds

	energy
idle	20 Watts
in use	30 Watts





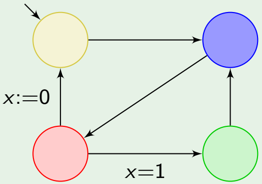
Priced timed automata

Definition ([KPSY99,ALP01,BFH$^+$01])

A priced timed automaton is made of

- a timed automaton;

Example



[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.
[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.
[BFH$^+$01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

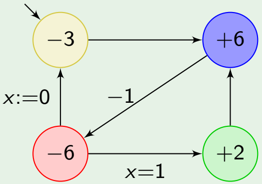
Priced timed automata

Definition ([KPSY99,ALP01,BFH+01])

A priced timed automaton is made of

- a timed automaton;
- the price of each transition and location.

Example



[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.

[BFH+01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

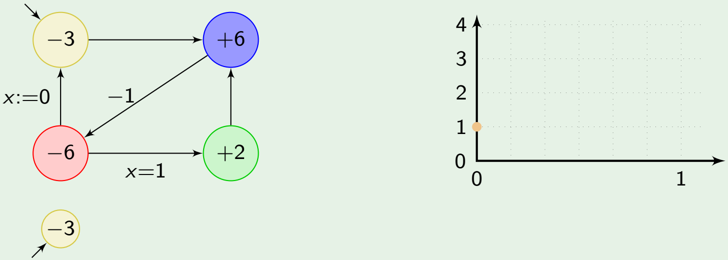
Priced timed automata

Definition ([KPSY99,ALP01,BFH$^+$01])

A priced timed automaton is made of

- a timed automaton;
- the price of each transition and location.

Example



[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.

[BFH$^+$01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

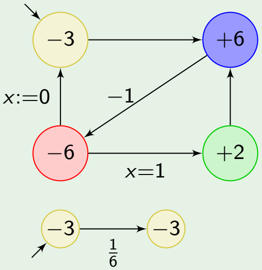
Priced timed automata

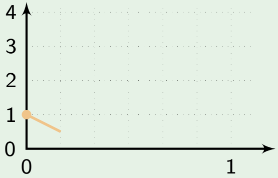
Definition ([KPSY99,ALP01,BFH$^+$01])

A priced timed automaton is made of

- a timed automaton;
- the price of each transition and location.

Example





[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.
[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.
[BFH$^+$01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

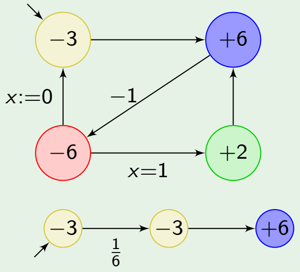
Priced timed automata

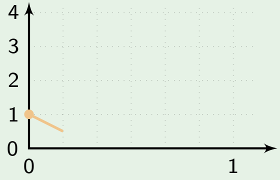
Definition ([KPSY99,ALP01,BFH$^+$01])

A priced timed automaton is made of

- a timed automaton;
- the price of each transition and location.

Example





[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.
[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.
[BFH$^+$01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

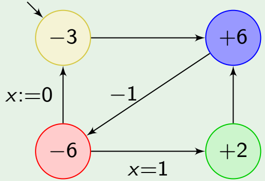
Priced timed automata

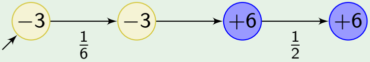
Definition ([KPSY99,ALP01,BFH$^+$01])

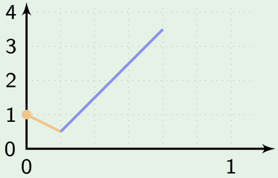
A priced timed automaton is made of

- a timed automaton;
- the price of each transition and location.

Example







[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.
[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.
[BFH$^+$01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

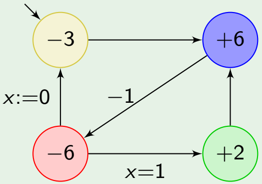
Priced timed automata

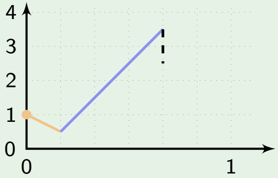
Definition ([KPSY99,ALP01,BFH$^+$01])

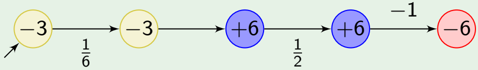
A priced timed automaton is made of

- a timed automaton;
- the price of each transition and location.

Example







[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.

[BFH$^+$01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

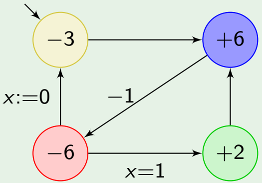
Priced timed automata

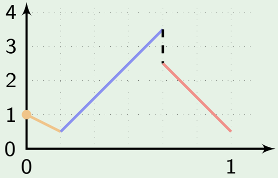
Definition ([KPSY99,ALP01,BFH⁺01])

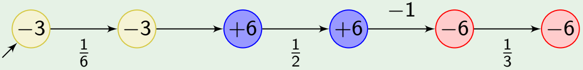
A priced timed automaton is made of

- a timed automaton;
- the price of each transition and location.

Example







[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.

[BFH⁺01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

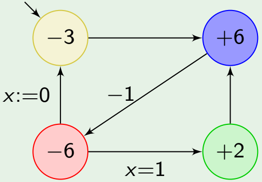
Priced timed automata

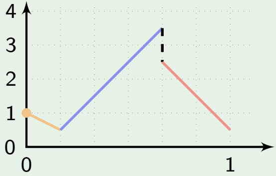
Definition ([KPSY99,ALP01,BFH$^+$01])

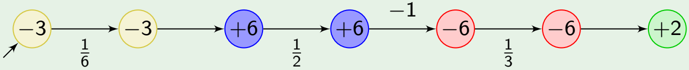
A priced timed automaton is made of

- a timed automaton;
- the price of each transition and location.

Example







[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.

[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.

[BFH$^+$01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

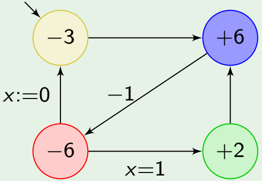
Priced timed automata

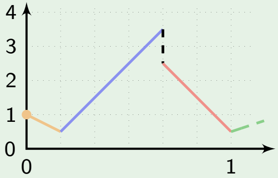
Definition ([KPSY99,ALP01,BFH$^+$01])

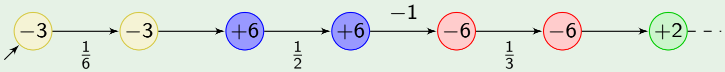
A priced timed automaton is made of

- a timed automaton;
- the price of each transition and location.

Example







[KPSY99] Kesten, Pnueli, Sifakis, Yovine. Decidable Integration Graphs. Inf. & Comp., 1999.

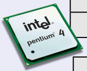
[ALP01] Alur, La Torre, Pappas. Optimal paths in weighted timed automata. HSCC, 2001.

[BFH$^+$01] Behrmann *et al.* Minimum-cost reachability in priced timed automata. HSCC, 2001.

Example: task graph scheduling

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

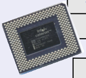
P_1 (fast):



time	
+	2 picoseconds
×	3 picoseconds

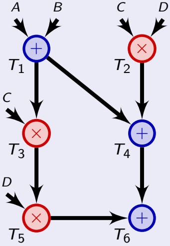
energy	
idle	10 Watt
in use	90 Watts

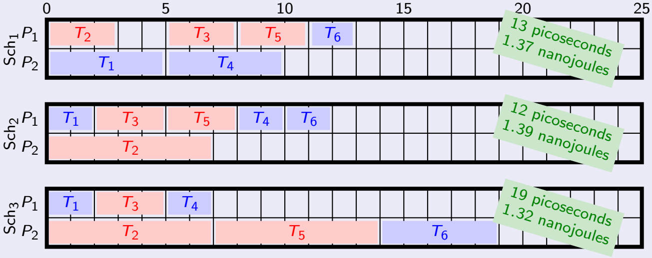
P_2 (slow):



time	
+	5 picoseconds
×	7 picoseconds

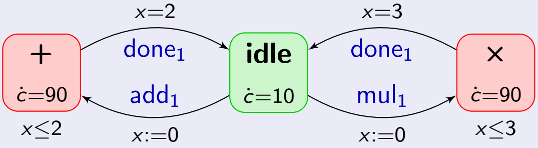
energy	
idle	20 Watts
in use	30 Watts





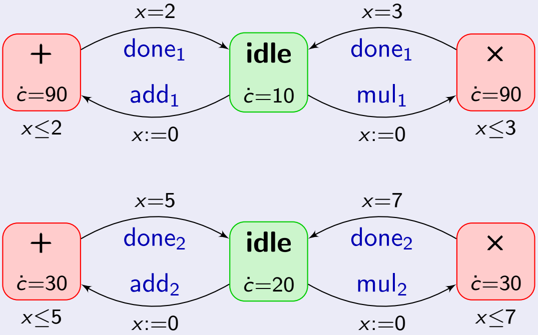
Modelling the task graph scheduling problem

Processors:

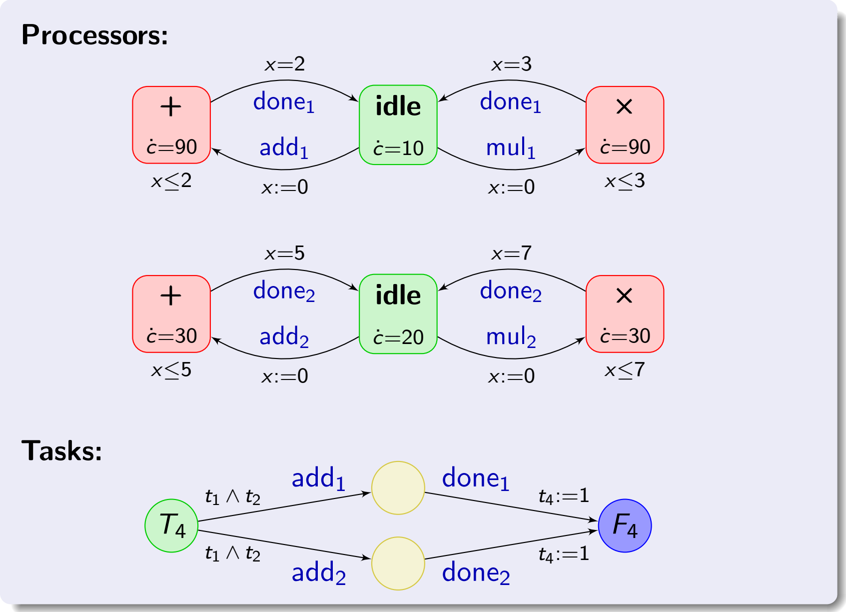


Modelling the task graph scheduling problem

Processors:



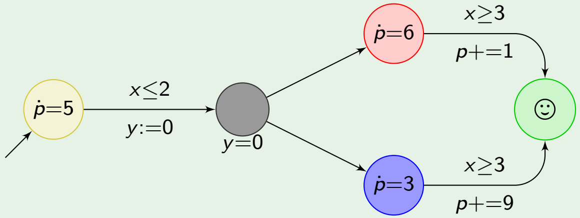
Modelling the task graph scheduling problem



Outline of the talk

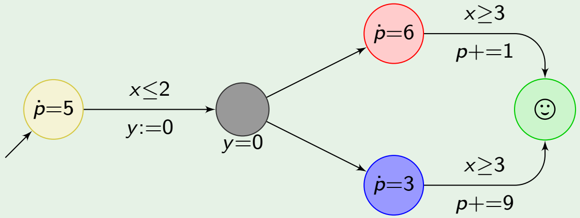
Cost-optimal reachability in priced timed automata

Example



Cost-optimal reachability in priced timed automata

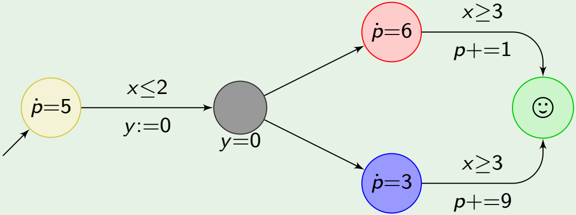
Example



Minimal cost for reaching ☺:

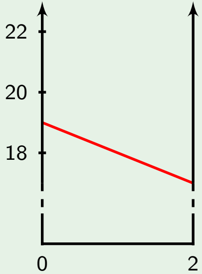
Cost-optimal reachability in priced timed automata

Example



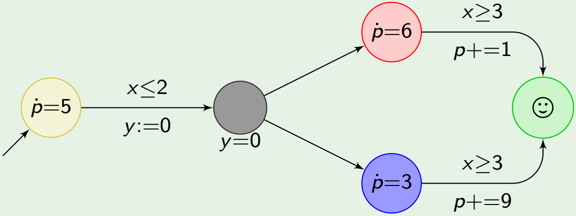
Minimal cost for reaching ☺:

$$5t + 6(3 - t) + 1$$



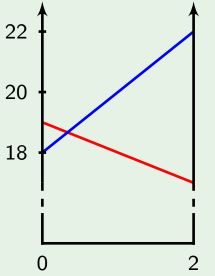
Cost-optimal reachability in priced timed automata

Example



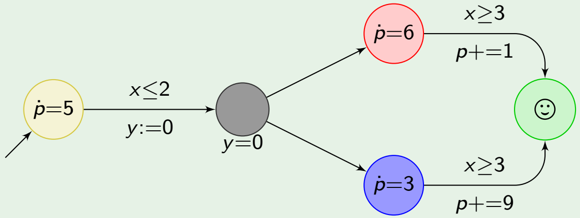
Minimal cost for reaching ☺:

$$5t + 6(3 - t) + 1$$
$$5t + 3(3 - t) + 9$$



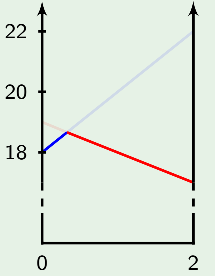
Cost-optimal reachability in priced timed automata

Example



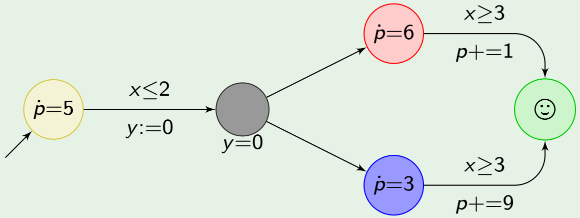
Minimal cost for reaching ☺:

$$\min \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix}$$



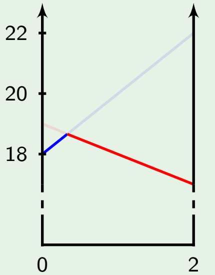
Cost-optimal reachability in priced timed automata

Example



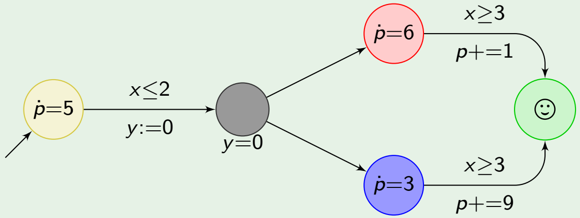
Minimal cost for reaching ☺:

$$\inf_{0\leq t\leq 2} \min \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix}$$



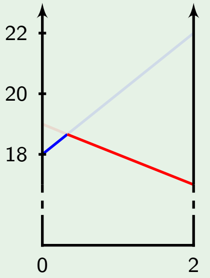
Cost-optimal reachability in priced timed automata

Example



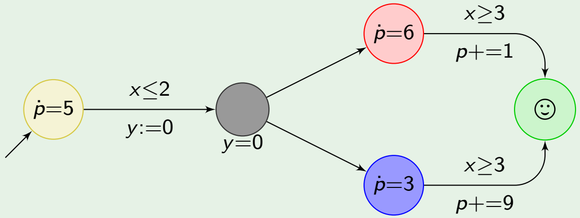
Minimal cost for reaching ☺:

$$\inf_{0\leq t\leq 2} \min \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix} = 17$$



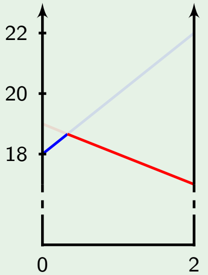
Cost-optimal reachability in priced timed automata

Example



Minimal cost for reaching ☺:

$$\inf_{0 \leq t \leq 2} \min \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix} = 17$$



The optimal *schedule* consists in

- waiting 2 time units in ◯;
- going through ◯.

Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-*complete.*

[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

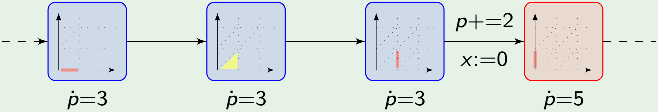
Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-*complete.*

Proof

- Regions are not precise enough;



[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE*-complete.*

Proof

- Regions are not precise enough;
- Use regions with corner-points:

[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

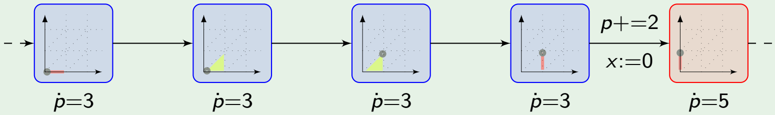
Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-complete.

Proof

- Regions are not precise enough;
- Use regions with corner-points:



[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

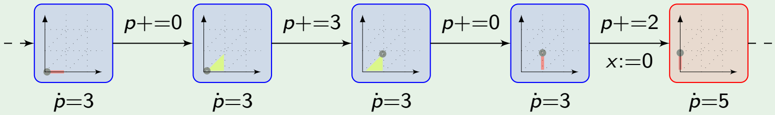
Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-complete.

Proof

- Regions are not precise enough;
- Use regions with corner-points:



[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

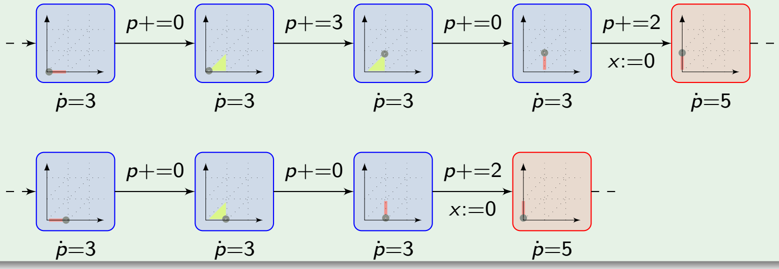
Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-complete.

Proof

- Regions are not precise enough;
- Use regions with corner-points:



[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

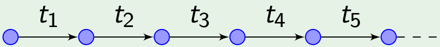
Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE*-complete.*

Proof

- optimal schedule as a linear programming problem:



[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

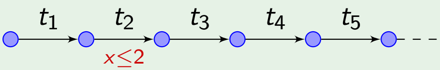
Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE*-complete.*

Proof

- optimal schedule as a linear programming problem:



$t_1 + t_2 \leq 2$

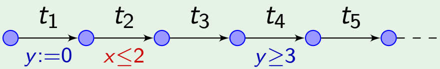
Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-*complete.*

Proof

- optimal schedule as a linear programming problem:



$$t_1 + t_2 \leq 2$$
$$t_2 + t_3 + t_4 \geq 3$$

[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

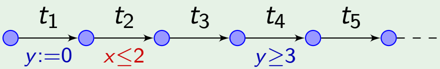
Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-complete.

Proof

- optimal schedule as a linear programming problem:



Minimize
$\sum_i c_i \cdot t_i + C_{\mathrm{disc}}$

$t_1 + t_2 \leq 2$

$t_2 + t_3 + t_4 \geq 3$

[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

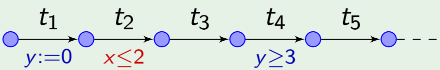
Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-*complete.*

Proof

- optimal schedule as a linear programming problem:



Minimize
$\sum_i c_i \cdot t_i + C_{\text{disc}}$

$t_1 + t_2 \leq 2$

$t_2 + t_3 + t_4 \geq 3$

$\rightsquigarrow$ infimum over bounded zone reached at a point on the frontier, with integer coordinates.

[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-*complete.*

Proof

- optimal schedule as a linear programming problem:

$$\forall \pi.\ \exists \pi_{cp}.\ cost(\pi_{cp}) \leq cost(\pi).$$

[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

Cost-optimal reachability in priced timed automata

Theorem ([BBBR07])

Optimal reachability in priced timed automata is PSPACE-complete.

Proof

- optimal schedule as a linear programming problem:

$$\forall\pi.\ \exists\pi_{cp}.\ \mathit{cost}(\pi_{cp}) \leq \mathit{cost}(\pi).$$

- approximate path in corner-point abstraction by a real run:

$$\forall\pi_{cp}.\ \exists\pi.\ \mathit{cost}(\pi) \leq \mathit{cost}(\pi_{cp}) + \epsilon.$$

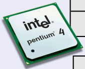
[BBBR07] Bouyer, Brihaye, Bruyère, Raskin. On the optimal reachability problem. FMSD, 2007.

Outline of the talk

Example: task graph scheduling

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

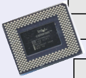
P_1 (fast):



	time
+	2 picoseconds
×	3 picoseconds

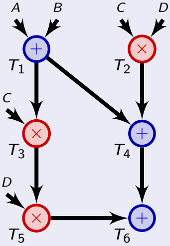
energy	
idle	10 Watt
in use	90 Watts

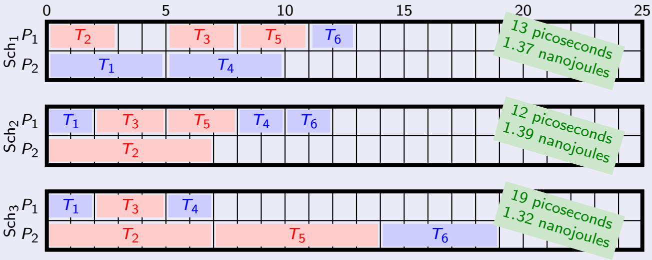
P_2 (slow):



	time
+	5 picoseconds
×	7 picoseconds

energy	
idle	20 Watts
in use	30 Watts

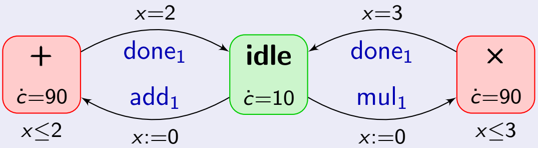




Cost-optimal reachability in priced timed games

Using games to model uncertainty over delays

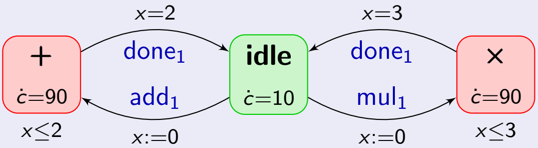
Processors with exact delays:



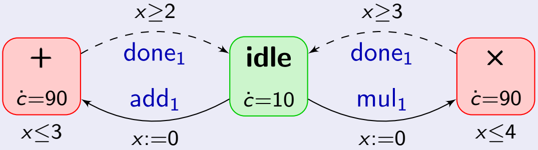
Cost-optimal reachability in priced timed games

Using games to model uncertainty over delays

Processors with exact delays:

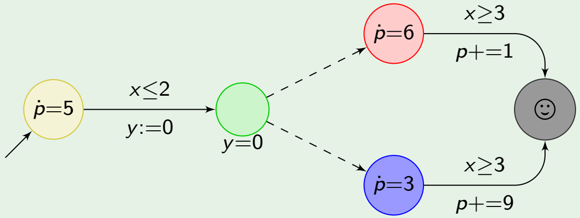


Processors with approximate delays:



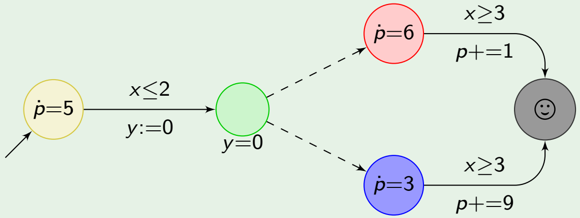
Cost-optimal reachability in priced timed games

Example



Cost-optimal reachability in priced timed games

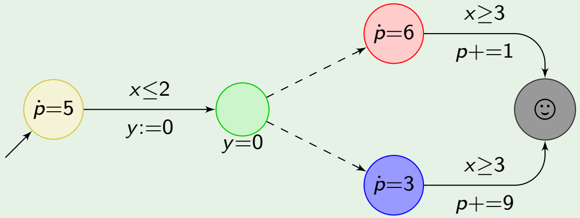
Example



Minimal cost for reaching ☺:

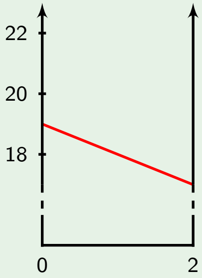
Cost-optimal reachability in priced timed games

Example



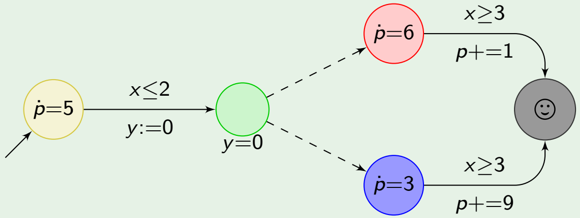
Minimal cost for reaching ☺:

$$5t + 6(3 - t) + 1$$



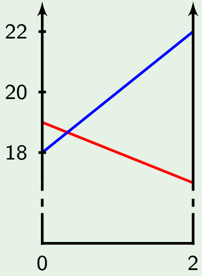
Cost-optimal reachability in priced timed games

Example



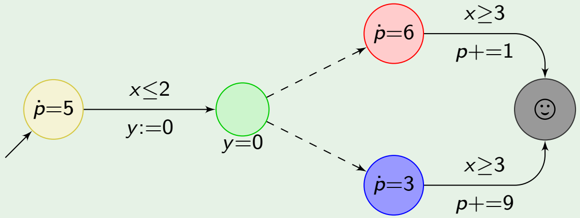
Minimal cost for reaching ☺:

$$5t + 6(3 - t) + 1$$
$$5t + 3(3 - t) + 9$$



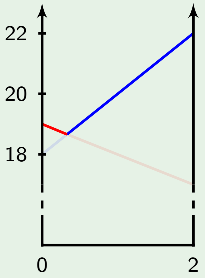
Cost-optimal reachability in priced timed games

Example



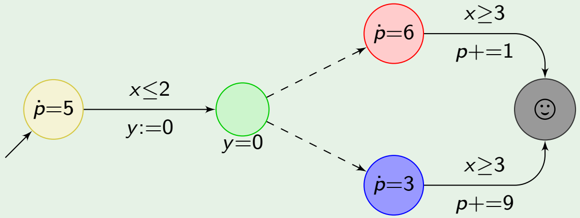
Minimal cost for reaching ☺:

$$\max \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix}$$



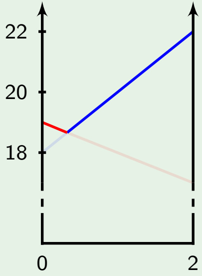
Cost-optimal reachability in priced timed games

Example



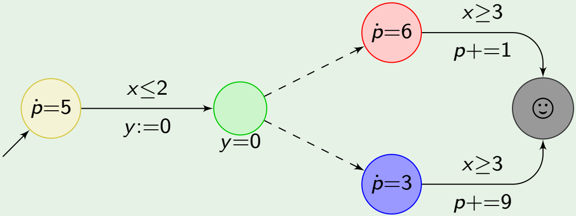
Minimal cost for reaching ☺:

$$\inf_{0\leq t\leq 2} \max \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix}$$



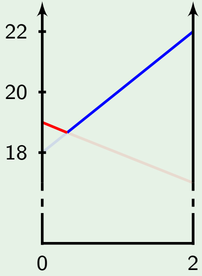
Cost-optimal reachability in priced timed games

Example



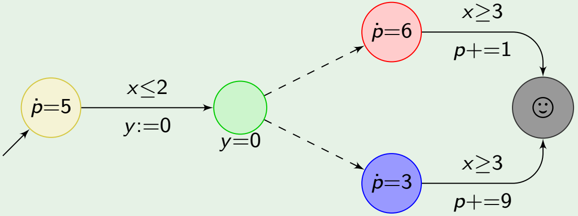
Minimal cost for reaching ☺:

$$\inf_{0\le t\le 2} \max \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix} = 18.66$$



Cost-optimal reachability in priced timed games

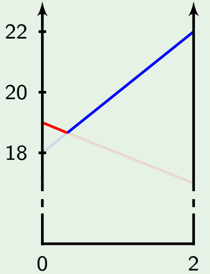
Example



Minimal cost for reaching ☺:

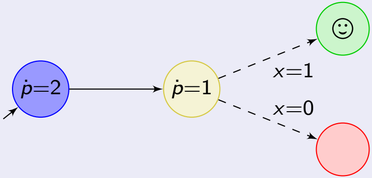
$$\inf_{0 \leq t \leq 2} \max \begin{pmatrix} 5t + 6(3-t) + 1 \\ 5t + 3(3-t) + 9 \end{pmatrix} = 18.66$$

$$(\text{with } t_{\text{opt}} = \frac{1}{3})$$



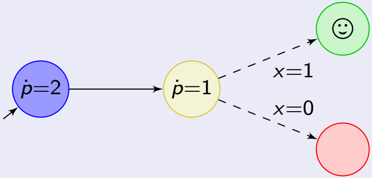
Looking for optimal strategies...

Optimal strategies need not exist...

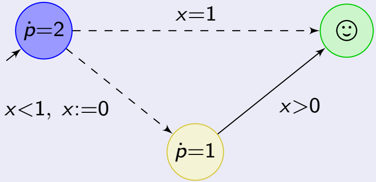


Looking for optimal strategies...

Optimal strategies need not exist...



Optimal strategies may need memory...



Cost-optimal reachability in priced timed games

Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

[BBR05] Brihaye, Bruyère, Raskin. On optimal timed strategies. FORMATS, 2005.
[BBM06] Bouyer, Brihaye, Markey. Improved undecidability results on weighted timed automa. IPL, 2006.

Cost-optimal reachability in priced timed games

Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.

[BBR05] Brihaye, Bruyère, Raskin. On optimal timed strategies. FORMATS, 2005.
[BBM06] Bouyer, Brihaye, Markey. Improved undecidability results on weighted timed automoa. IPL, 2006.

Cost-optimal reachability in priced timed games

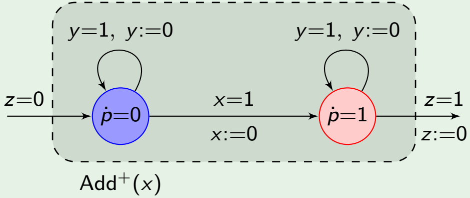
Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.

- add the value of clock x to the accumulated cost



[BBR05] Brihaye, Bruyère, Raskin. On optimal timed strategies. FORMATS, 2005.
[BBM06] Bouyer, Brihaye, Markey. Improved undecidability results on weighted timed automa. IPL, 2006.

Cost-optimal reachability in priced timed games

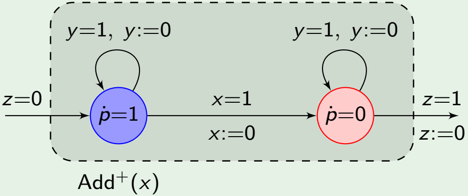
Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.

- add the value of clock x to the accumulated cost
- add $1 - x$ to the accumulated cost



[BBR05] Brihaye, Bruyère, Raskin. On optimal timed strategies. FORMATS, 2005.

[BBM06] Bouyer, Brihaye, Markey. Improved undecidability results on weighted timed automa. IPL, 2006.

Cost-optimal reachability in priced timed games

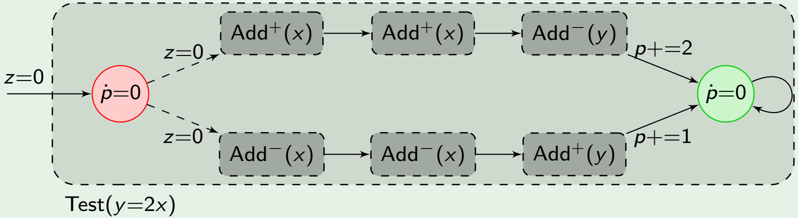
Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.

- add the value of clock x to the accumulated cost
- add $1 - x$ to the accumulated cost
- check that $y = 2x$



Cost-optimal reachability in priced timed games

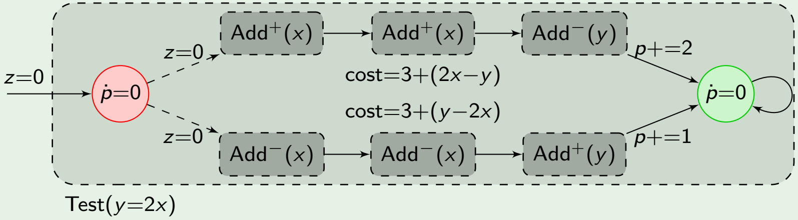
Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.

- add the value of clock x to the accumulated cost
- add $1 - x$ to the accumulated cost
- check that $y = 2x$



Cost-optimal reachability in priced timed games

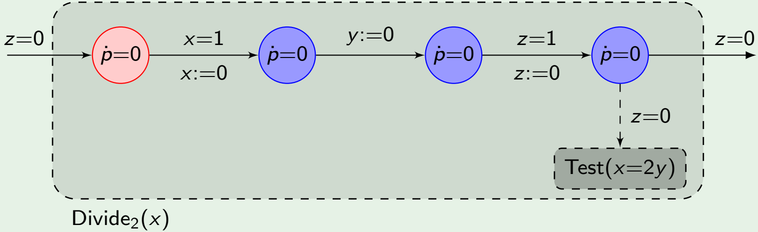
Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.

- add the value of clock x to the accumulated cost
- add $1 - x$ to the accumulated cost
- check that $y = 2x$
- divide clock x by 2



Cost-optimal reachability in priced timed games

Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.

- add the value of clock x to the accumulated cost
- add $1 - x$ to the accumulated cost
- check that $y = 2x$
- divide clock x by 2

$\rightsquigarrow$ We can use the following encoding:

$$x_1 = \frac{1}{2^{c_1}} \qquad x_2 = \frac{1}{2^{c_2}}$$

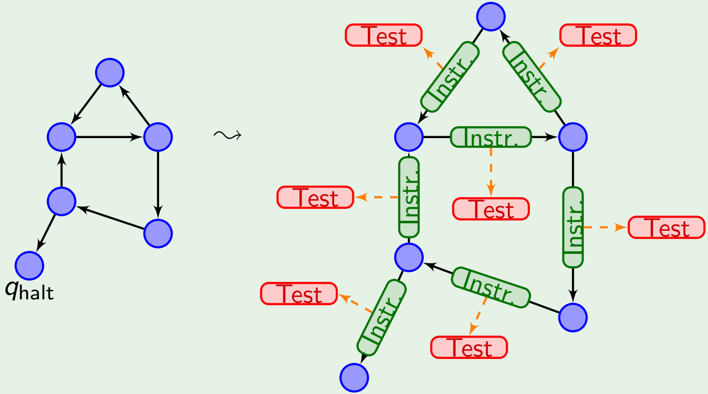
Cost-optimal reachability in priced timed games

Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.



Cost-optimal reachability in priced timed games

Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.

Lemma

The halting state is reachable
if, and only if,
there is an optimal strategy in the priced timed game.

[BBR05] Brihaye, Bruyère, Raskin. On optimal timed strategies. FORMATS, 2005.
[BBM06] Bouyer, Brihaye, Markey. Improved undecidability results on weighted timed automoa. IPL, 2006.

Cost-optimal reachability in priced timed games

Theorem ([BBR05,BBM06])

Optimal reachability in priced timed games is undecidable.

Proof

Encode a two-counter machine as a priced timed game.

Lemma

The halting state is reachable
if, and only if,
there is an optimal strategy in the priced timed game.

reach terminal location with
total weight at most 3

[BBR05] Brihaye, Bruyère, Raskin. On optimal timed strategies. FORMATS, 2005.
[BBM06] Bouyer, Brihaye, Markey. Improved undecidability results on weighted timed automa. IPL, 2006.

The value of a game

Definition

The value of a game

Definition

Cost of a path:

$\text{cost}(\pi)$ = sum of costs of all transitions until target location

The value of a game

Definition

Cost of a path:

$\text{cost}(\pi) =$ sum of costs of all transitions until target location

Cost of a strategy:

$\text{cost}(\sigma) = \sup\{\text{cost}(\pi) \mid \pi \text{ outcome of } \sigma\}$

The value of a game

Definition

Cost of a path:

$\text{cost}(\pi)$ = sum of costs of all transitions until target location

Cost of a strategy:

$\text{cost}(\sigma) = \sup\{\text{cost}(\pi) \mid \pi \text{ outcome of } \sigma\}$

Optimal cost in a priced timed game:

$\text{optcost}_{\mathcal{G}} = \inf\{\text{cost}(\sigma) \mid \sigma \text{ winning strategy in } \mathcal{G}\}$

The value of a game

Definition

Cost of a path:

$\text{cost}(\pi) = \text{sum of costs of all transitions until target location}$

Cost of a strategy:

$\text{cost}(\sigma) = \sup\{\text{cost}(\pi) \mid \pi \text{ outcome of } \sigma\}$

Optimal cost in a priced timed game:

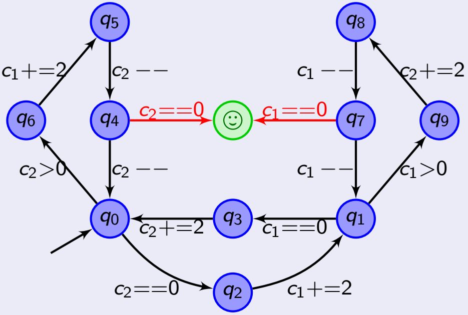
$\text{optcost}_{\mathcal{G}} = \inf\{\text{cost}(\sigma) \mid \sigma \text{ winning strategy in } \mathcal{G}\}$

The existence of a strategy with cost less than k is undecidable.

What about deciding if $\text{optcost}_{\mathcal{G}} \leq k$?

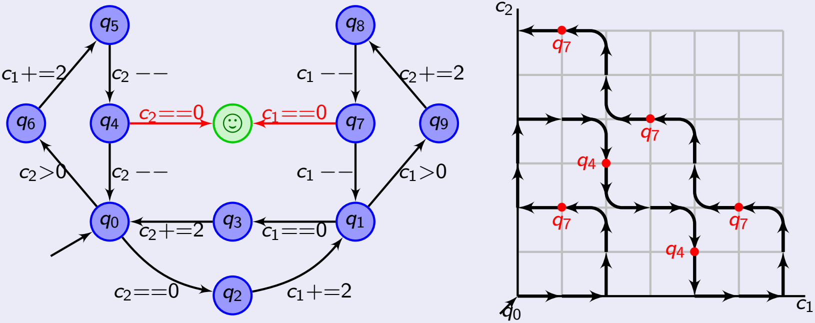
Undecidability of the value problem

Trying to reuse the previous reduction...



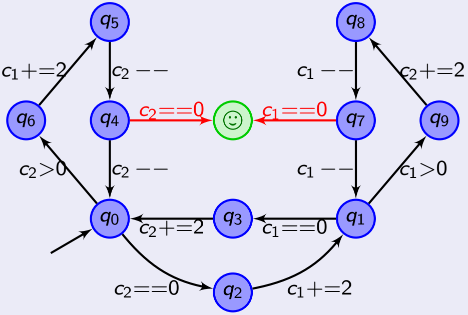
Undecidability of the value problem

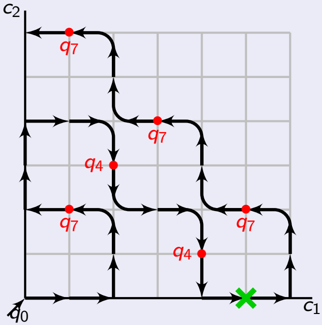
Trying to reuse the previous reduction...



Undecidability of the value problem

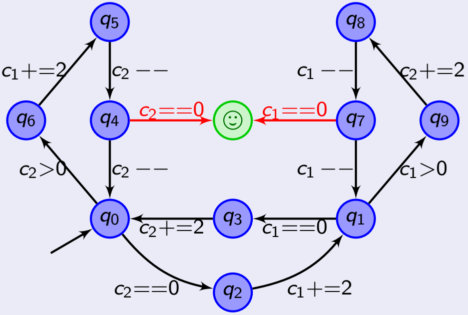
Trying to reuse the previous reduction...

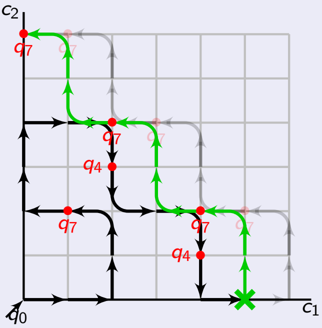




Undecidability of the value problem

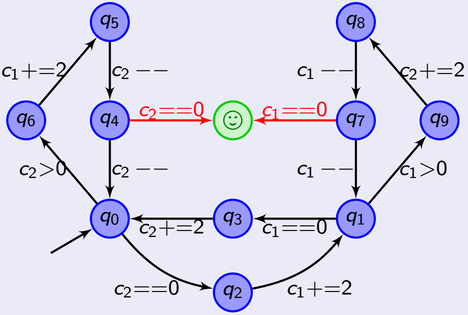
Trying to reuse the previous reduction...

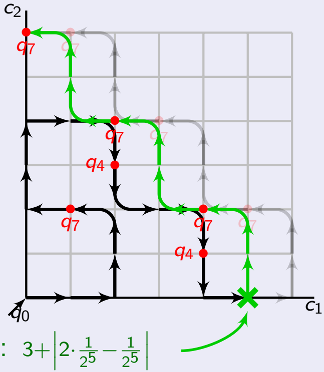




Undecidability of the value problem

Trying to reuse the previous reduction...

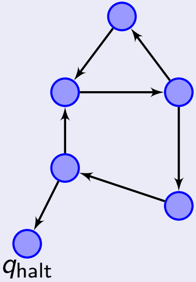




The value of the game is 3, but there is no optimal strategy...

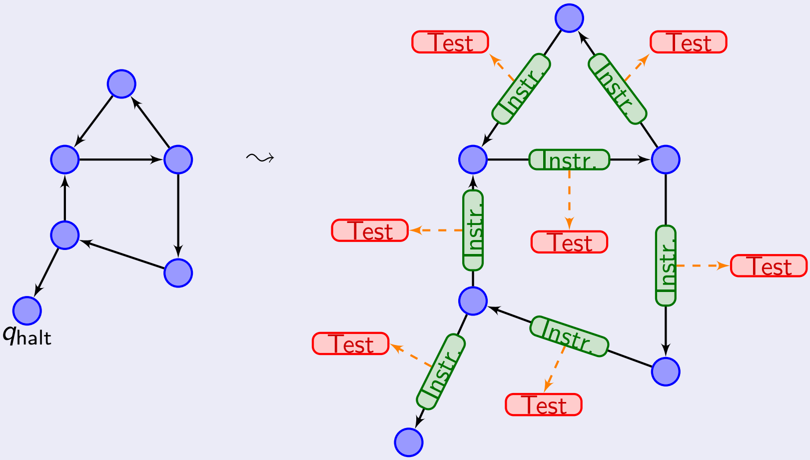
Undecidability of the value problem

Adapting the previous reduction...



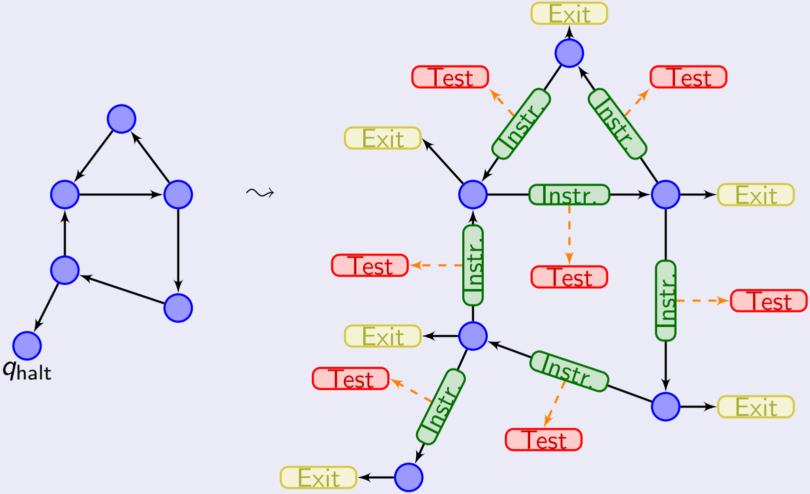
Undecidability of the value problem

Adapting the previous reduction...



Undecidability of the value problem

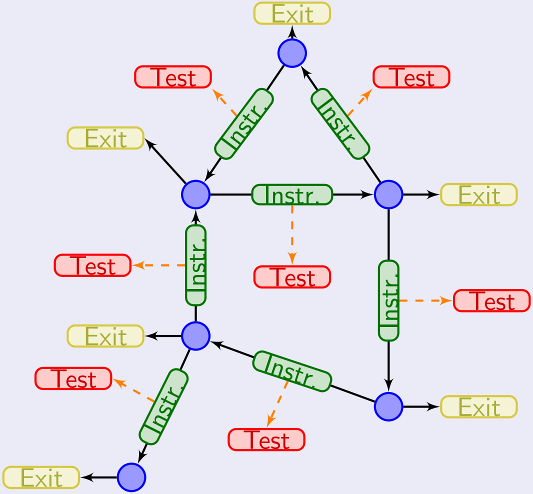
Adapting the previous reduction...



exit nodes: cost $3 + \frac{1}{2^n}$
(n = length of path)

Undecidability of the value problem

Adapting the previous reduction...



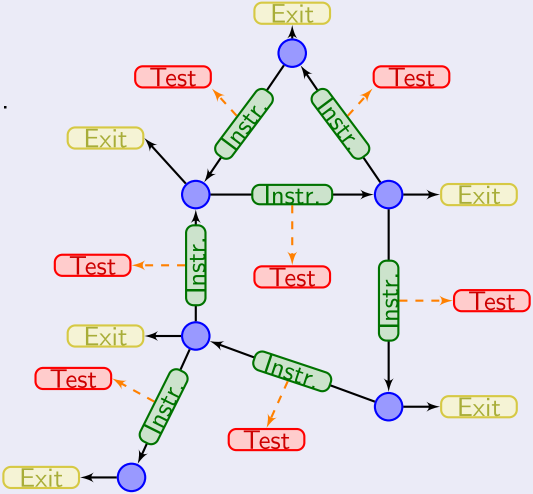
exit nodes: cost $3 + \frac{1}{2^n}$
$(n =$ length of path$)$

Undecidability of the value problem

Adapting the previous reduction...

- **if $\mathcal{M}$ does not halt:** Player 1 simulates correctly until $2^n > \frac{1}{\epsilon}$.

  $\rightsquigarrow$ $\text{cost}(\sigma) \leq 3 + \epsilon$



exit nodes: cost $3 + \frac{1}{2^n}$
(n = length of path)

Undecidability of the value problem

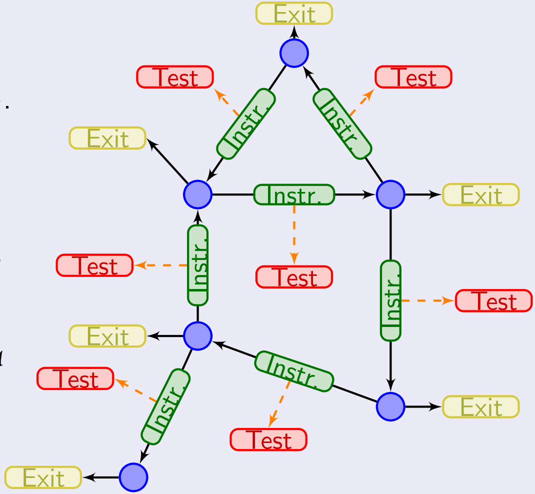
Adapting the previous reduction...

- **if $\mathcal{M}$ does not halt:** Player 1 simulates correctly until $2^n > \frac{1}{\epsilon}$.

  $\rightsquigarrow$ $\text{cost}(\sigma) \leq 3 + \epsilon$

- **if $\mathcal{M}$ halts:** correct simulation for finite duration.

  $\rightsquigarrow$ $\text{cost}(\sigma) \geq 3 + \alpha_{\mathcal{M}}$ for all σ



exit nodes: cost $3 + \frac{1}{2^n}$
(n = length of path)

Undecidability of the value problem

Theorem ([BJM15])

The value problem is undecidable in priced timed games.

[BJM15] Bouyer, Jaziri, Markey. On the Value Problem in Weighted Timed Games. CONCUR, 2015.

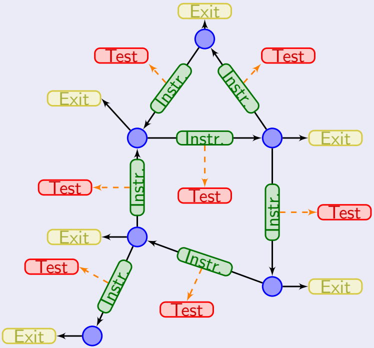
Undecidability of the value problem

Theorem ([BJM15])

The value problem is undecidable in priced timed games.

Remark

- blue nodes and intermediary instruction modules have cost zero everywhere;
- positive weights only occur in acyclic parts.



[BJM15] Bouyer, Jaziri, Markey. On the Value Problem in Weighted Timed Games. CONCUR, 2015.

Approximation of the optimal cost

Definition

A priced timed game $\mathcal{G}$ is almost-strongly non-Zeno if there exists $\kappa > 0$ for any run ρ that starts and ends in the same region:

$$\mathsf{cost}(\rho) \geq \kappa \qquad \text{or} \qquad \mathsf{cost}(\rho) = 0$$

Approximation of the optimal cost

Definition

A priced timed game $\mathcal{G}$ is almost-strongly non-Zeno if there exists $\kappa > 0$ for any run ρ that starts and ends in the same region:

$$\mathsf{cost}(\rho) \geq \kappa \qquad \text{or} \qquad \mathsf{cost}(\rho) = 0$$

Theorem ([BJM15])

The optimal cost of almost-strongly non-Zeno priced timed automata can be approximated.

[BJM15] Bouyer, Jaziri, Markey. On the Value Problem in Weighted Timed Games. CONCUR, 2015.

Approximation of the optimal cost

Definition

A priced timed game $\mathcal{G}$ is almost-strongly non-Zeno if there exists $\kappa > 0$ for any run ρ that starts and ends in the same region:

$$\mathsf{cost}(\rho) \geq \kappa \qquad \text{or} \qquad \mathsf{cost}(\rho) = 0$$

Theorem ([BJM15])

The optimal cost of almost-strongly non-Zeno priced timed automata can be approximated: for every $\epsilon > 0$, we can compute

- *values v_ϵ^+ and v_ϵ^- such that*

$$|v_\epsilon^+ - v_\epsilon^-| < \epsilon \qquad v_\epsilon^- \leq optcost_{\mathcal{G}} \leq v_\epsilon^+$$

- *a strategy σ_ϵ such that*

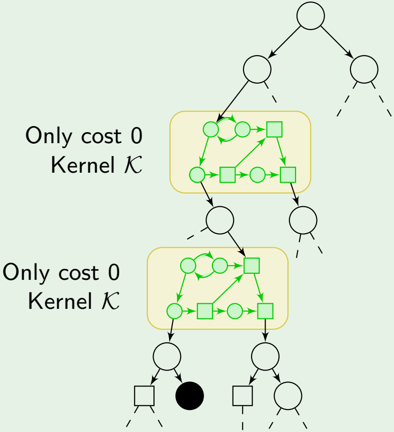
$$optcost_{\mathcal{G}} \leq cost(\sigma_\epsilon) \leq optcost_{\mathcal{G}} + \epsilon.$$

[BJM15] Bouyer, Jaziri, Markey. On the Value Problem in Weighted Timed Games. CONCUR, 2015.

Approximation of the optimal cost

Proof

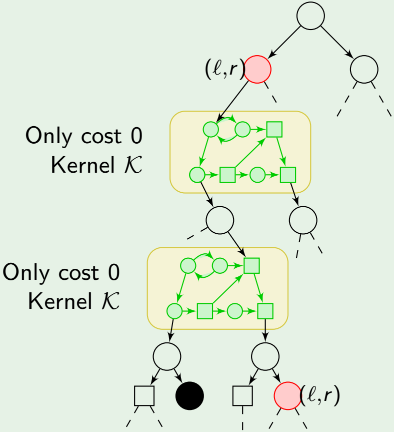
- **semi-unfolding** of region automaton (seen as a timed game)



Approximation of the optimal cost

Proof

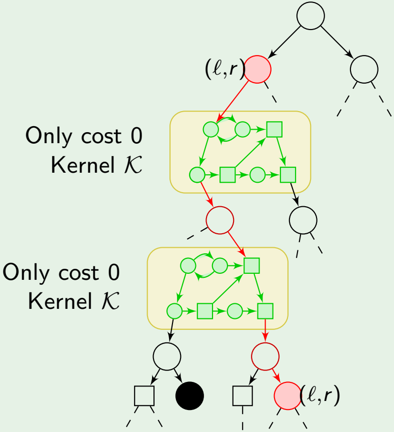
- **semi-unfolding** of region automaton (seen as a timed game)



Approximation of the optimal cost

Proof

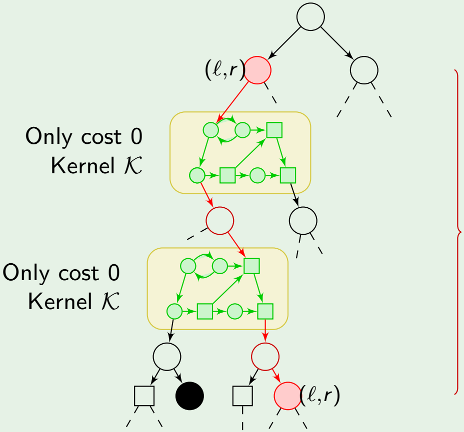
- **semi-unfolding** of region automaton (seen as a timed game)



Approximation of the optimal cost

Proof

- **semi-unfolding** of region automaton (seen as a timed game)



Hypothesis:

cost > 0

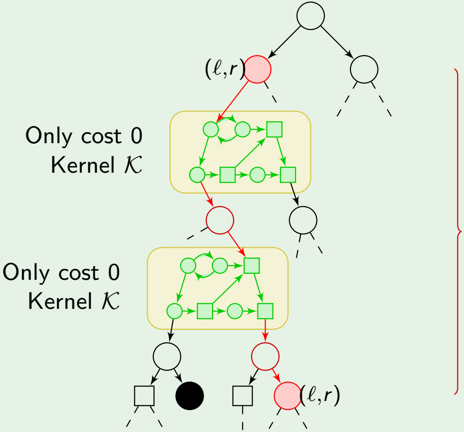
$\downarrow$

cost $\geq \kappa$

Approximation of the optimal cost

Proof

- **semi-unfolding** of region automaton (seen as a timed game)



Hypothesis:

$$\text{cost} > 0$$

$$\downarrow$$

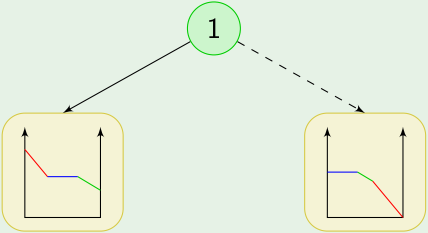
$$\text{cost} \geq \kappa$$

$\rightsquigarrow$ bounded depth

Approximation of the optimal cost

Proof

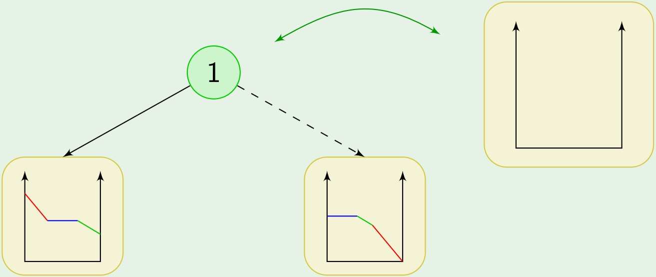
- **semi-unfolding** of region automaton (seen as a timed game)
- compute **exact** optimal cost in tree-like parts



Approximation of the optimal cost

Proof

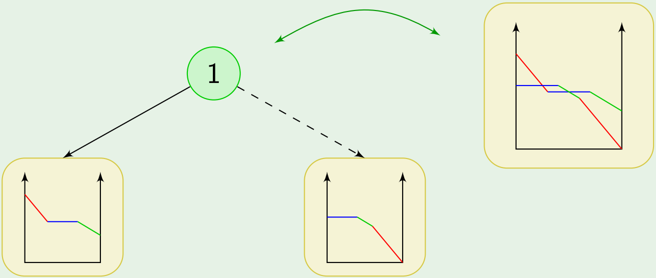
- **semi-unfolding** of region automaton (seen as a timed game)
- compute **exact** optimal cost in tree-like parts



Approximation of the optimal cost

Proof

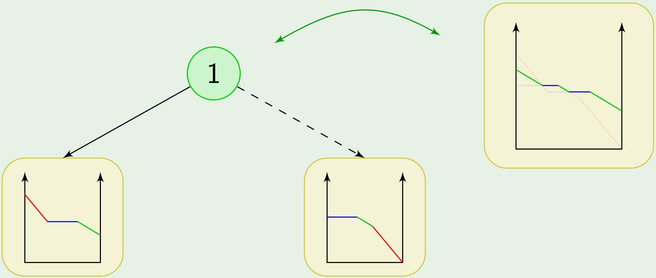
- **semi-unfolding** of region automaton (seen as a timed game)
- compute **exact** optimal cost in tree-like parts



Approximation of the optimal cost

Proof

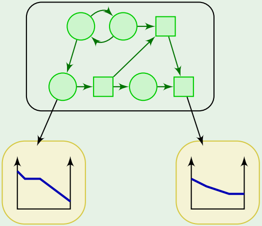
- **semi-unfolding** of region automaton (seen as a timed game)
- compute **exact** optimal cost in tree-like parts



Approximation of the optimal cost

Proof

- **semi-unfolding** of region automaton (seen as a timed game)
- compute **exact** optimal cost in tree-like parts
- compute **approximate** optimal cost in kernels



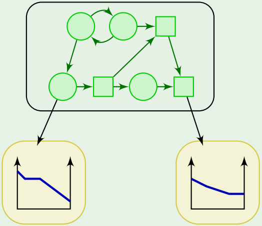
Output cost functions f

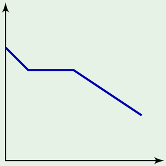
Approximation of the optimal cost

Proof

- **semi-unfolding** of region automaton (seen as a timed game)
- compute **exact** optimal cost in tree-like parts
- compute **approximate** optimal cost in kernels

Under- and over-approximate by piecewise constant functions f_ϵ^- and f_ϵ^+





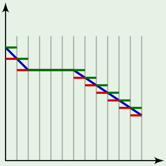
Output cost functions f

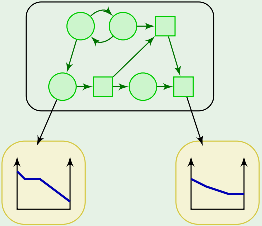
Approximation of the optimal cost

Proof

- **semi-unfolding** of region automaton (seen as a timed game)
- compute **exact** optimal cost in tree-like parts
- compute **approximate** optimal cost in kernels

Under- and over-approximate by piecewise constant functions f_ϵ^- and f_ϵ^+



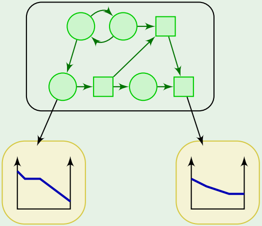


Output cost functions f

Approximation of the optimal cost

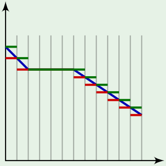
Proof

- **semi-unfolding** of region automaton (seen as a timed game)
- compute **exact** optimal cost in tree-like parts
- compute **approximate** optimal cost in kernels



Output cost functions f

Under- and over-approximate by piecewise constant functions f_ϵ^- and f_ϵ^+

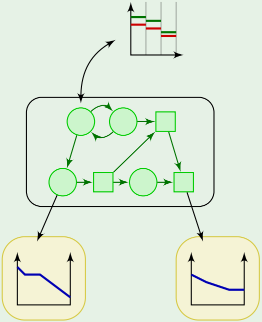


$\rightsquigarrow$ reachability timed game in small regions

Approximation of the optimal cost

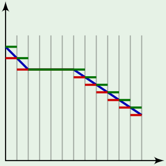
Proof

- **semi-unfolding** of region automaton (seen as a timed game)
- compute **exact** optimal cost in tree-like parts
- compute **approximate** optimal cost in kernels



Output cost functions f

Under- and over-approximate by piecewise constant functions f_ϵ^- and f_ϵ^+



$\leadsto$ reachability timed game in small regions

Outline of the talk

Conclusions and future directions

Priced timed automata and games

- convenient for modelling resources;
- 1-player setting remains tractable (sort of);
- 2-player setting undecidable, but approximable.
- approximation algorithms are a convenient trade-off.

Conclusions and future directions

Priced timed automata and games

- convenient for modelling resources;
- 1-player setting remains tractable (sort of);
- 2-player setting undecidable, but approximable.
- approximation algorithms are a convenient trade-off.

Future work

- improve approximation technique (in terms of complexity);
- extend results to whole class of priced timed games;
- average energy and energy constraints;
- robust analysis of priced timed games;
- develop a tool.