Soutenance d'habilitation

"From Qualitative to Quantitative Analysis of Timed Systems"

Patricia Bouyer

January 12, 2009

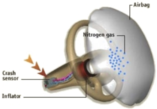
Outline

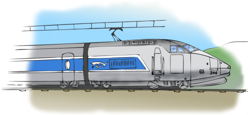
Context: verification of timed systems

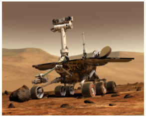
- What are timed systems?

Context: verification of timed systems

- What are timed systems?







Context: verification of timed systems

- What are timed systems?
- What we want to avoid

Context: verification of timed systems

- What are timed systems?
- What we want to avoid

Radiothérapie : nouvel incident au CHU de Toulouse

Jeudi 24 mai 2007

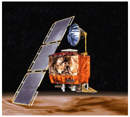
Après le centre hospitalier Jean Monnet d'Epinal, c'est au tour du CHU de Toulouse-Rangueil d'avouer la survenue d'une surexposition lors d'un traitement par radiothérapie. L'appareil de radiothérapie impliqué a été installé en avril 2006. L'erreur de calibration de l'appareil a été détectée lors du contrôle technique systématique appliqué à tous les appareils de radiothérapie.

Radiotherapy in Toulouse, 2007

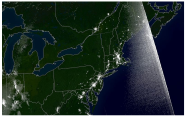
Ariane 5, 1996

Atlanta airport, 2008

Bug Pentium, 1994



Mars climate obs., 1998

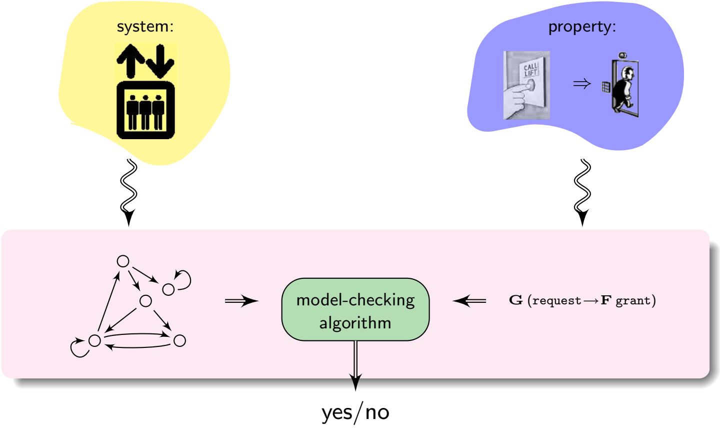


Blackout, 2003

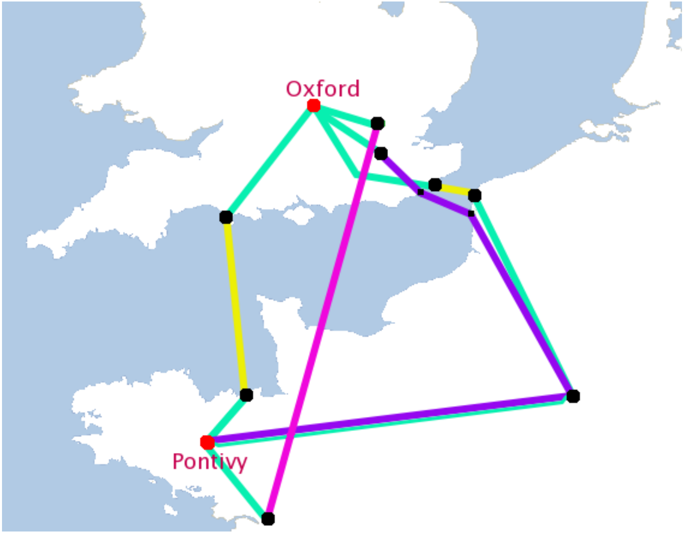
Context: verification of timed systems

- What are timed systems?
- What we want to avoid

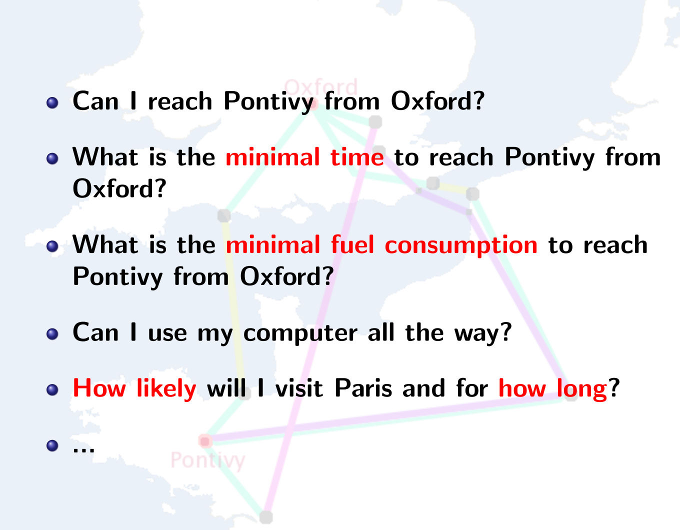
⇝ Verification by model-checking



A running example

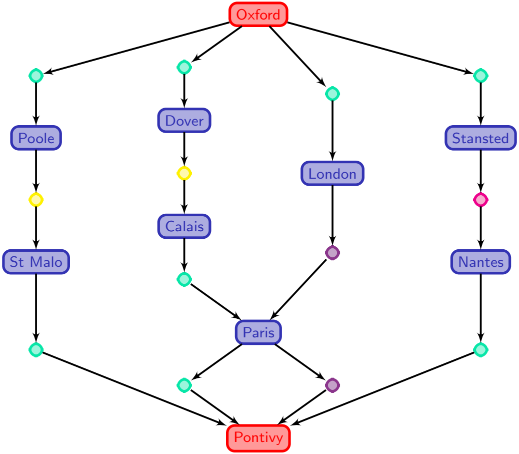


A running example: natural questions

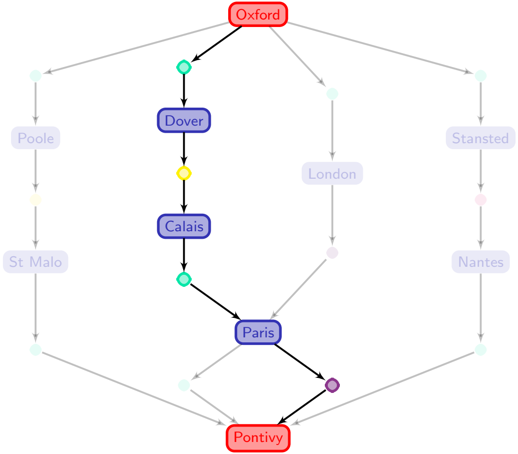


- **Can I reach Pontivy from Oxford?**
- **What is the minimal time to reach Pontivy from Oxford?**
- **What is the minimal fuel consumption to reach Pontivy from Oxford?**
- **Can I use my computer all the way?**
- **How likely will I visit Paris and for how long?**
- **...**

A first model of the system

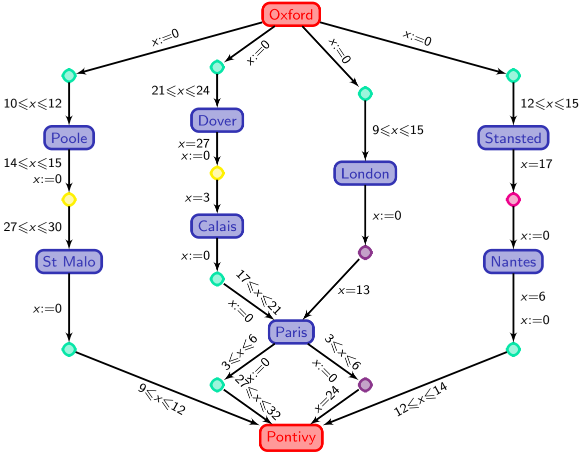


Can I reach Pontivy from Oxford?

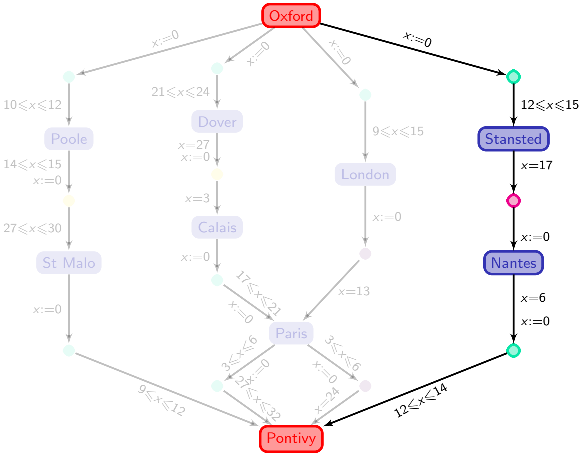


This is a reachability question in a finite graph: Yes, I can!

A second model of the system

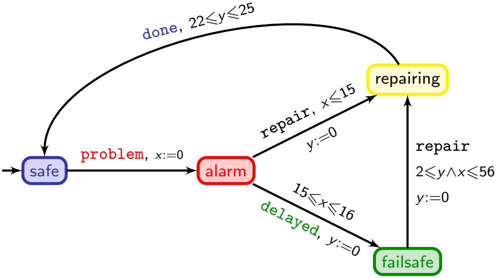


How long will that take?

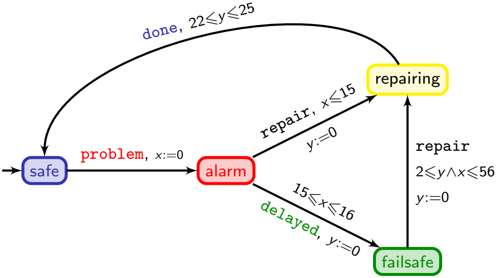


It is a reachability (and optimization) question
in a timed automaton: at least 350mn = 5h50mn!

The timed automaton model: an example

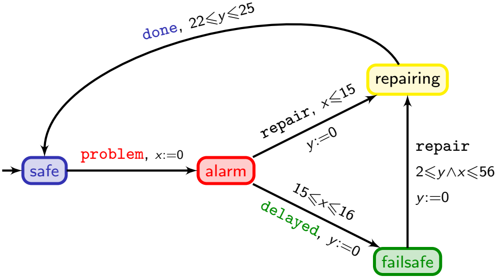


The timed automaton model: an example



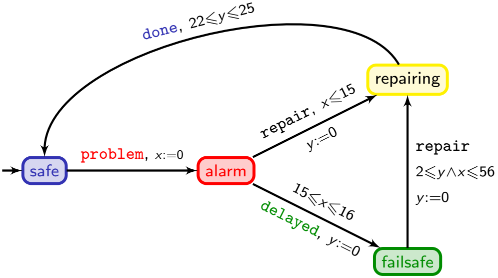
	safe
x	0
y	0

The timed automaton model: an example



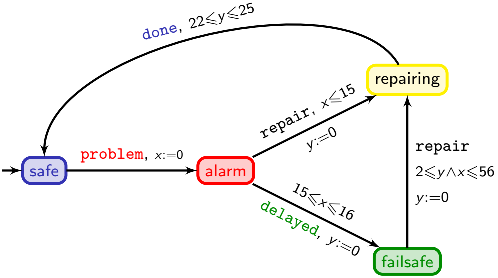
	safe	$\xrightarrow{23}$	safe
x	0		23
y	0		23

The timed automaton model: an example



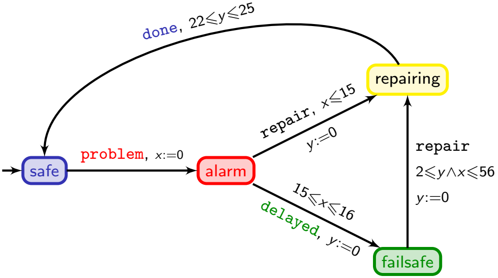
	safe	$\xrightarrow{23}$	safe	$\xrightarrow{\text{problem}}$	alarm
x	0		23		0
y	0		23		23

The timed automaton model: an example



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

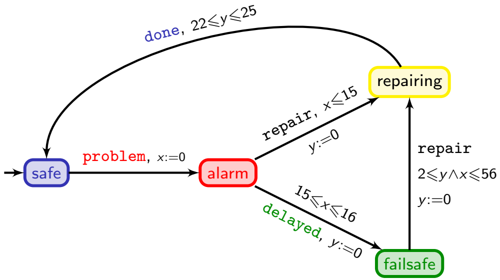
The timed automaton model: an example



	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

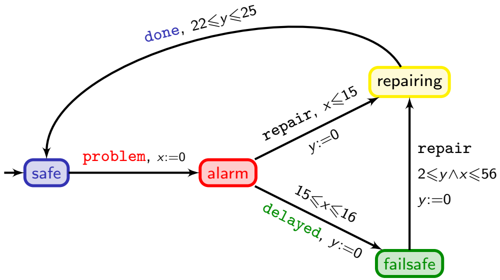
The timed automaton model: an example



	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
$\cdots$	15.6		17.9
	0		2.3

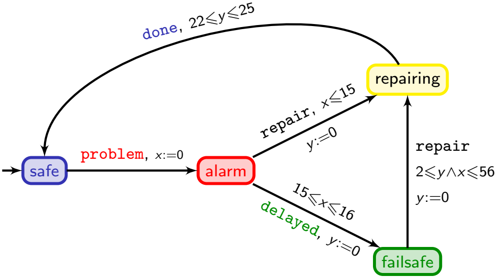
The timed automaton model: an example



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

	failsafe	$\xrightarrow{2.3}$	failsafe	$\xrightarrow{\text{repair}}$	repairing
···	15.6		17.9		17.9
	0		2.3		0

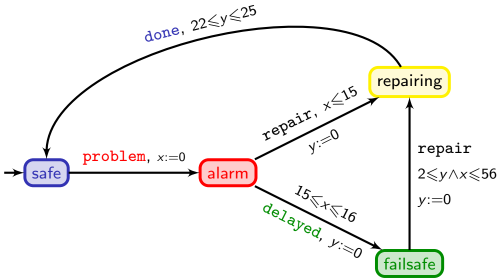
The timed automaton model: an example



	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	$\xrightarrow{22.1}$	repairing
$\cdots$	15.6		17.9		17.9		40
	0		2.3		0		22.1

The timed automaton model: an example



	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

Basics of timed automata

Theorem [AD90,AD94]

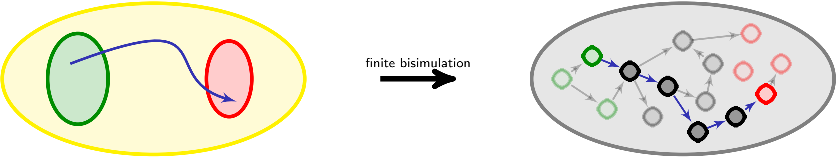
The reachability problem is decidable (and PSPACE-complete) in timed automata.

[AD90] Alur, Dill. Automata for modeling real-time systems *(ICALP'90)*.
[AD94] Alur, Dill. A theory of timed automata *(Theoretical Computer Science)*.

Basics of timed automata

Theorem [AD90,AD94]

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



timed automaton

large (but finite) automaton
(region automaton)

[AD90] Alur, Dill. Automata for modeling real-time systems *(ICALP'90)*.
[AD94] Alur, Dill. A theory of timed automata *(Theoretical Computer Science)*.

Outline

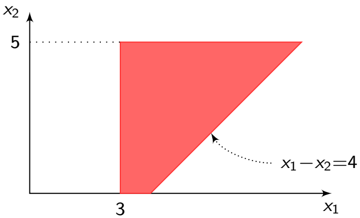
Outline

What about the practice?

- the region automaton is never computed
- instead, symbolic computations are performed using zones

Example of a zone

$$Z = (x_1 \geqslant 3) \wedge (x_2 \leqslant 5) \wedge (x_1 - x_2 \leqslant 4)$$



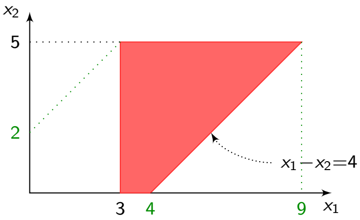
$$\begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} \infty & -3 & \infty \\ \infty & \infty & 4 \\ 5 & \infty & \infty \end{pmatrix} \end{array}$$

What about the practice?

- the region automaton is never computed
- instead, symbolic computations are performed using zones

Example of a zone

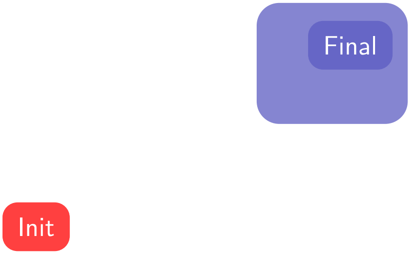
$$Z \;=\; (x_1 \geqslant 3) \;\wedge\; (x_2 \leqslant 5) \;\wedge\; (x_1 - x_2 \leqslant 4)$$



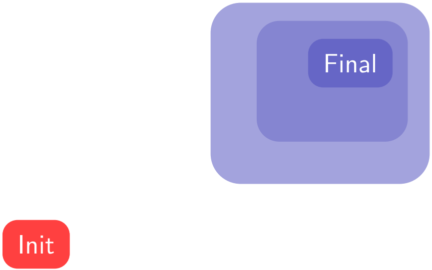
$$\begin{array}{c} \\ x_0 \\ x_1 \\ x_2 \end{array} \begin{array}{c} \begin{array}{ccc} x_0 & x_1 & x_2 \end{array} \\ \begin{pmatrix} 0 & -3 & 0 \\ 9 & 0 & 4 \\ 5 & 2 & 0 \end{pmatrix} \end{array}$$

Backward computation

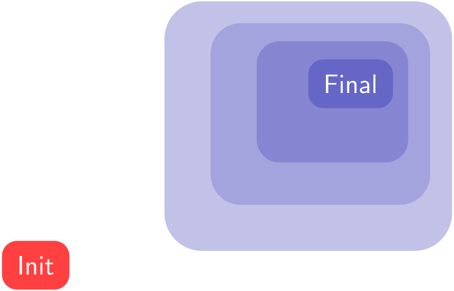
Backward computation



Backward computation



Backward computation



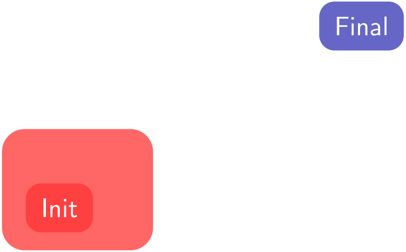
Backward computation

Backward computation

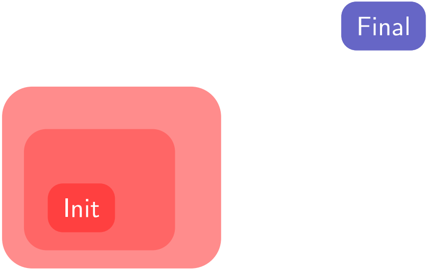
☺ the backward computation always terminates!

Forward computation

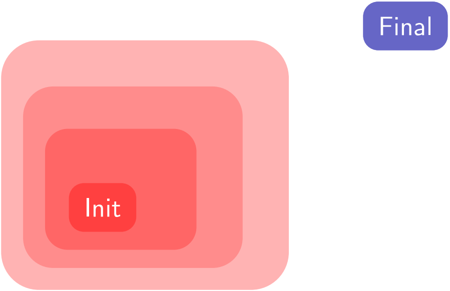
Forward computation



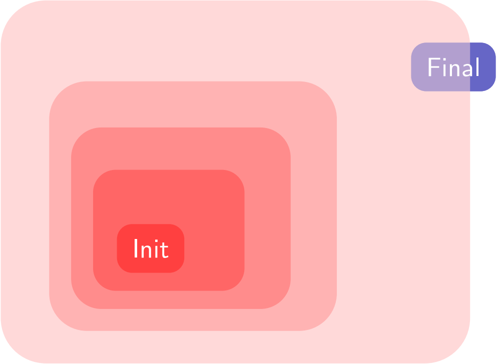
Forward computation



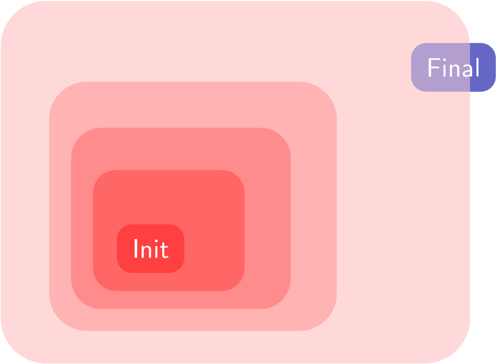
Forward computation



Forward computation

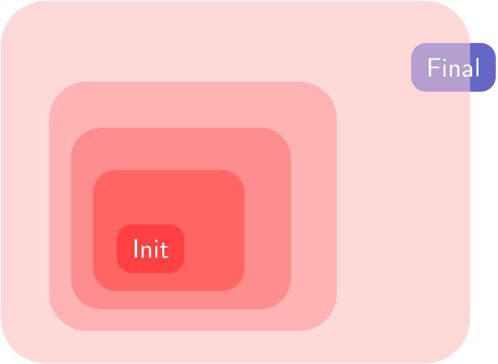


Forward computation



☹ the forward computation may not terminate...

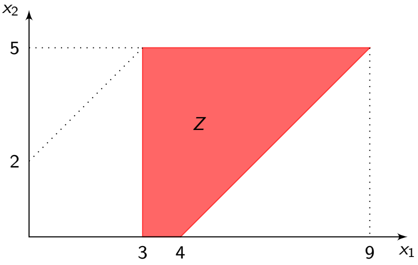
Forward computation



☹ the forward computation may not terminate...

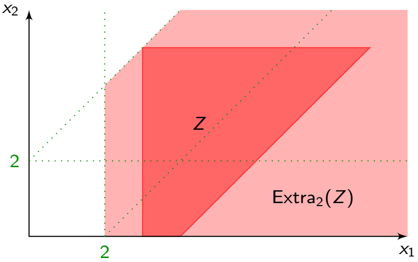
$\rightsquigarrow$ abstractions need to be used, that ensure termination...

An abstraction: the extrapolation operator



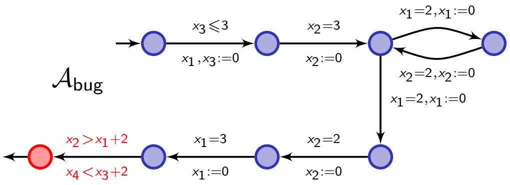
$$\begin{pmatrix} 0 & -3 & 0 \\ 9 & 0 & 4 \\ 5 & 2 & 0 \end{pmatrix}$$

An abstraction: the extrapolation operator



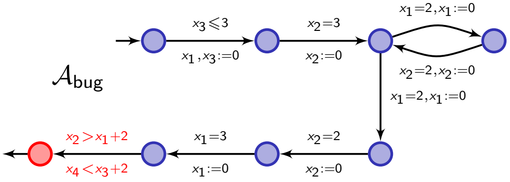
$$\begin{pmatrix} 0 & -3 & 0 \\ 9 & 0 & 4 \\ 5 & 2 & 0 \end{pmatrix} \xRightarrow{\mathsf{Extra}_2} \begin{pmatrix} 0 & -2 & 0 \\ \infty & 0 & \infty \\ \infty & 2 & 0 \end{pmatrix}$$

Results



[Bou03] Bouyer. Untameable Timed Automata! *(STACS'03)*.
[Bou04] Bouyer. Forward Analysis of Updatable Timed Automata *(Formal Methods in System Design)*.

Results

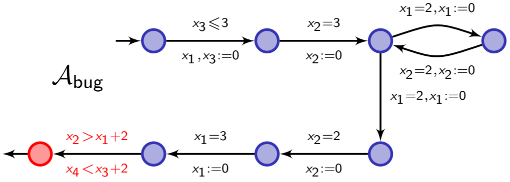


Theorem [Bou03,Bou04]

- In $\mathcal{A}_{\text{bug}}$, any extrapolation operator is incorrect!

[Bou03] Bouyer. Untameable Timed Automata! *(STACS'03)*.
[Bou04] Bouyer. Forward Analysis of Updatable Timed Automata *(Formal Methods in System Design)*.

Results



Theorem [Bou03,Bou04]

- In $\mathcal{A}_{\text{bug}}$, any extrapolation operator is incorrect!
- The extrapolation operator is correct in *diagonal-free* timed automata.

[Bou03] Bouyer. Untameable Timed Automata! *(STACS'03)*.
[Bou04] Bouyer. Forward Analysis of Updatable Timed Automata *(Formal Methods in System Design)*.

Improving further

- the extrapolation operator can be made coarser:
  - local extrapolation constants [BBFL03];
  - distinguish between lower- and upper-bounded contraints [BBLP03,BBLP06]

$\rightsquigarrow$ has leaded to a practical improvement in UPPAAL of up to 20%!

[BBFL03] Behrmann, Bouyer, Fleury, Larsen. Static Guard Analysis in Timed Automata Verification *(TACAS'03)*.
[BBLP04] Behrmann, Bouyer, Larsen, Pelánek. Lower and Upper Bounds in Zone Based Abstractions of Timed Automata *(TACAS'04)*.
[BBLP06] Behrmann, Bouyer, Larsen, Pelánek. Lower and Upper Bounds in Zone-Based Abstractions of Timed Automata *(International Journal on Software Tools for Technology Transfer)*.

Improving further

- the extrapolation operator can be made coarser:
  - local extrapolation constants [BBFL03];
  - distinguish between lower- and upper-bounded contraints [BBLP03,BBLP06]

$\rightsquigarrow$ has leaded to a practical improvement in UPPAAL of up to 20%!

Since then...

- further improvement due to better data structure manipulations...
- ... but no further algorithmic improvement!

[BBFL03] Behrmann, Bouyer, Fleury, Larsen. Static Guard Analysis in Timed Automata Verification *(TACAS'03)*.
[BBLP04] Behrmann, Bouyer, Larsen, Pelánek. Lower and Upper Bounds in Zone Based Abstractions of Timed Automata *(TACAS'04)*.
[BBLP06] Behrmann, Bouyer, Larsen, Pelánek. Lower and Upper Bounds in Zone-Based Abstractions of Timed Automata *(International Journal on Software Tools for Technology Transfer)*.

Outline

Motivation

- Checking reachability properties may not be enough:

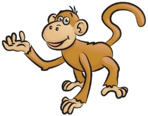
Motivation

- Checking reachability properties may not be enough:
  - basically only safety properties can be verified [ABBL98,ABBL03]

[ABBL98] Aceto, Bouyer, Burgueño, Larsen. The power of reachability testing for timed automata *(FSTTCS'98)*.
[ABBL03] Aceto, Bouyer, Burgueño, Larsen. The power of reachability testing for timed automata *(Theoretical Computer Science)*.

Motivation

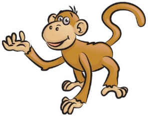
- Checking reachability properties may not be enough:
  - basically only safety properties can be verified [ABBL98,ABBL03]
  - what about liveness properties?



"the monkey will eventually write the complete works of Shakespeare"

Motivation

- Checking reachability properties may not be enough:
  - basically only safety properties can be verified [ABBL98,ABBL03]
  - what about liveness properties?

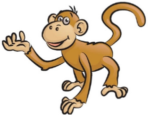


"the monkey will eventually write the complete works of Shakespeare"

"every request is eventually granted"

Motivation

- Checking reachability properties may not be enough:
  - basically only safety properties can be verified [ABBL98,ABBL03]
  - what about liveness properties? And other properties?



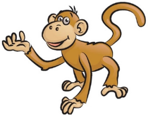
"the monkey will eventually write the complete works of Shakespeare"

"every request is eventually granted"

"the machine produces 56 items per day until it needs to be repaired"

Motivation

- Checking reachability properties may not be enough:
  - basically only safety properties can be verified [ABBL98,ABBL03]
  - what about liveness properties? And other properties?



"the monkey will eventually write the complete works of Shakespeare"

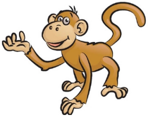
"every request is eventually granted"

"the machine produces 56 items per day until it needs to be repaired"

- Need for specification languages expressing timing constraints...

Motivation

- Checking reachability properties may not be enough:
  - basically only safety properties can be verified [ABBL98,ABBL03]
  - what about liveness properties? And other properties?



"the monkey will eventually write the complete works of Shakespeare"

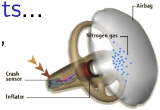
"every request is eventually granted"

"the machine produces 56 items per day until it needs to be repaired"

- Need for specification languages expressing timing constraints...

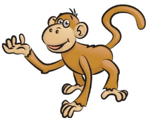
"the airbag inflates no more than 56ms after the car crashes"

$\rightsquigarrow$ critical property



Motivation

- Checking reachability properties may not be enough:
  - basically only safety properties can be verified [ABBL98,ABBL03]
  - what about liveness properties? And other properties?

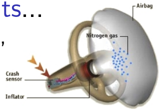


"the monkey will eventually write the complete works of Shakespeare"

"every request is eventually granted"

"the machine produces 56 items per day until it needs to be repaired"

- Need for specification languages expressing timing constraints...



"the airbag inflates no more than 56ms after the car crashes"

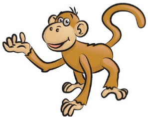
$\leadsto$ critical property

"the reponse time of the memory circuit is no more than 10^{-12}s"

$\leadsto$ performance, quality of service

Motivation

- Checking reachability properties may not be enough:
  - basically only safety properties can be verified [ABBL98,ABBL03]
  - what about liveness properties? And other properties?

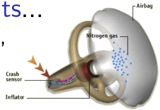


"the monkey will eventually write the complete works of Shakespeare"

"every request is eventually granted"

"the machine produces 56 items per day until it needs to be repaired"

- Need for specification languages expressing timing constraints...



"the airbag inflates no more than 56ms after the car crashes"

$\rightsquigarrow$ critical property

"the reponse time of the memory circuit is no more than 10^{-12}s"

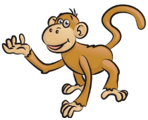
$\rightsquigarrow$ performance, quality of service

- ... and for algorithms to verify those properties.

[Pnu77] Pnueli. The temporal logic of programs *(FoCS'77)*.

Motivation

- Checking reachability properties may not be enough:
  - basically only safety properties can be verified [ABBL98,ABBL03]
  - what about liveness properties? And other properties?

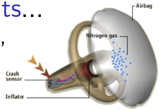


"the monkey will eventually write the complete works of Shakespeare"

"every request is eventually granted"

"the machine produces 56 items per day until it needs to be repaired"

- Need for specification languages expressing timing constraints...



"the airbag inflates no more than 56ms after the car crashes"

$\rightsquigarrow$ critical property

"the reponse time of the memory circuit is no more than 10^{-12}s"

$\rightsquigarrow$ performance, quality of service

- ... and for algorithms to verify those properties.

We will focus on timed extensions of LTL [Pnu77]

[Pnu77] Pnueli. The temporal logic of programs *(FoCS'77)*.

Two classical timed extensions of LTL: MTL and TPTL

MTL (Metric Temporal Logic) [Koy90]

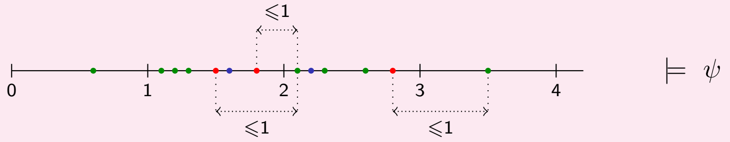
- $\psi \;=\; \mathbf{G}\,(\texttt{request} \rightarrow \mathbf{F}_{\leqslant 1}\,\texttt{grant})$

[Koy90] Koymans. Specifying real-time properties with Metric Temporal Logic *(Real-Time Systems)*.

Two classical timed extensions of LTL: MTL and TPTL

MTL (Metric Temporal Logic) [Koy90]

- $\psi \; = \; \mathbf{G}\,(\texttt{request} \rightarrow \mathbf{F}_{\leqslant 1}\,\texttt{grant})$

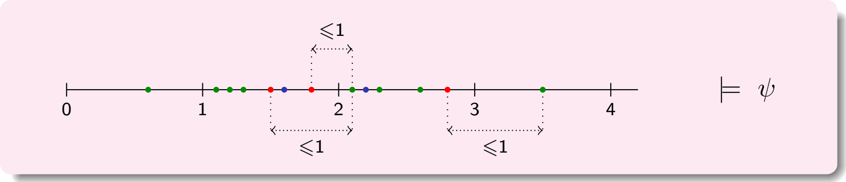


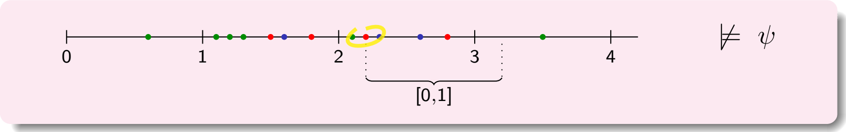
[Koy90] Koymans. Specifying real-time properties with Metric Temporal Logic *(Real-Time Systems)*.

Two classical timed extensions of LTL: MTL and TPTL

MTL (Metric Temporal Logic) [Koy90]

- $\psi \;=\; \mathbf{G}\,(\texttt{request} \rightarrow \mathbf{F}_{\leqslant 1}\,\texttt{grant})$





[Koy90] Koymans. Specifying real-time properties with Metric Temporal Logic *(Real-Time Systems)*.

Two classical timed extensions of LTL: MTL and TPTL

MTL (Metric Temporal Logic) [Koy90]

- $\psi \;=\; \mathbf{G}\,(\texttt{request} \rightarrow \mathbf{F}_{\leqslant 1}\,\texttt{grant})$

[Koy90] Koymans. Specifying real-time properties with Metric Temporal Logic *(Real-Time Systems)*.

Two classical timed extensions of LTL: MTL and TPTL

MTL (Metric Temporal Logic) [Koy90]

- $\psi \;=\; \mathbf{G}\,(\texttt{request} \rightarrow \mathbf{F}_{\leqslant 1}\,\texttt{grant})$

TPTL (Timed Propositional Temporal Logic) [AH89]

[Koy90] Koymans. Specifying real-time properties with Metric Temporal Logic *(Real-Time Systems)*.
[AH89] Alur, Henzinger. A really temporal logic *(FoCS'89)*.

Two classical timed extensions of LTL: MTL and TPTL

MTL (Metric Temporal Logic) [Koy90]

- $\psi \ = \ \mathbf{G}\,(\mathtt{request} \rightarrow \mathbf{F}_{\leqslant 1}\,\mathtt{grant})$

TPTL (Timed Propositional Temporal Logic) [AH89]

- $\psi \ \equiv \ \mathbf{G}\,(\mathtt{request} \rightarrow x \cdot \mathbf{F}\,(\mathtt{grant} \wedge (x \leqslant 1)))$

[Koy90] Koymans. Specifying real-time properties with Metric Temporal Logic *(Real-Time Systems)*.
[AH89] Alur, Henzinger. A really temporal logic *(FoCS'89)*.

Two classical timed extensions of LTL: MTL and TPTL

MTL (Metric Temporal Logic) [Koy90]

- $\psi \; = \; \mathbf{G}\,(\texttt{request} \rightarrow \mathbf{F}_{\leqslant 1}\,\texttt{grant})$

TPTL (Timed Propositional Temporal Logic) [AH89]

- $\psi \; \equiv \; \mathbf{G}\,(\texttt{request} \rightarrow x \cdot \mathbf{F}\,(\texttt{grant} \wedge (x \leqslant 1)))$
- $\varphi \; = \; \mathbf{G}\,(\texttt{request} \rightarrow x \cdot \mathbf{F}\,(\texttt{ack} \wedge \mathbf{F}\,(\texttt{grant} \wedge x \leqslant 2)))$

[Koy90] Koymans. Specifying real-time properties with Metric Temporal Logic *(Real-Time Systems)*.
[AH89] Alur, Henzinger. A really temporal logic *(FoCS'89)*.

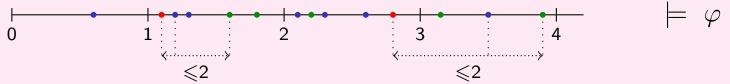
Two classical timed extensions of LTL: MTL and TPTL

MTL (Metric Temporal Logic) [Koy90]

- $\psi \ = \ \mathbf{G}\,(\mathtt{request} \rightarrow \mathbf{F}_{\leqslant 1}\,\mathtt{grant})$

TPTL (Timed Propositional Temporal Logic) [AH89]

- $\psi \ \equiv \ \mathbf{G}\,(\mathtt{request} \rightarrow x \cdot \mathbf{F}\,(\mathtt{grant} \wedge (x \leqslant 1)))$
- $\varphi \ = \ \mathbf{G}\,(\mathtt{request} \rightarrow x \cdot \mathbf{F}\,(\mathtt{ack} \wedge \mathbf{F}\,(\mathtt{grant} \wedge x \leqslant 2)))$



[Koy90] Koymans. Specifying real-time properties with Metric Temporal Logic *(Real-Time Systems)*.
[AH89] Alur, Henzinger. A really temporal logic *(FoCS'89)*.

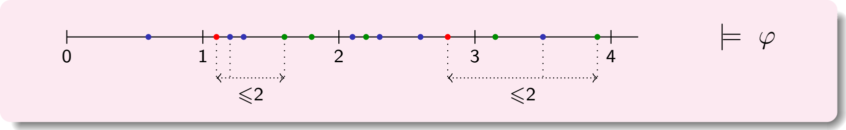
Two classical timed extensions of LTL: MTL and TPTL

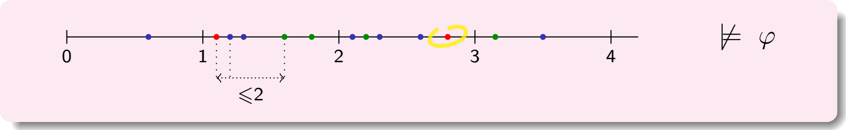
MTL (Metric Temporal Logic) [Koy90]

- $\psi \;=\; \mathbf{G}\,(\texttt{request} \rightarrow \mathbf{F}_{\leqslant 1}\,\texttt{grant})$

TPTL (Timed Propositional Temporal Logic) [AH89]

- $\psi \;\equiv\; \mathbf{G}\,(\texttt{request} \rightarrow x \cdot \mathbf{F}\,(\texttt{grant} \wedge (x \leqslant 1)))$
- $\varphi \;=\; \mathbf{G}\,(\texttt{request} \rightarrow x \cdot \mathbf{F}\,(\texttt{ack} \wedge \mathbf{F}\,(\texttt{grant} \wedge x \leqslant 2)))$





[Koy90] Koymans. Specifying real-time properties with Metric Temporal Logic *(Real-Time Systems)*.
[AH89] Alur, Henzinger. A really temporal logic *(FoCS'89)*.

Expressiveness of these logics

Conjecture (Alur & Henzinger, since 1990)

TPTL is strictly more expressive than MTL, and the TPTL formula

$$\varphi = \mathbf{G}\,(\bullet \rightarrow x \cdot \mathbf{F}\,(\bullet \wedge \mathbf{F}\,(\bullet \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

However...

$\varphi \quad \equiv$

Expressiveness of these logics

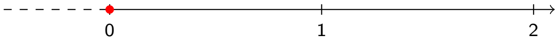
Conjecture (Alur & Henzinger, since 1990)

TPTL is strictly more expressive than MTL, and the TPTL formula

$$\varphi = \mathbf{G}\,(\textcolor{red}{\bullet} \rightarrow x \cdot \mathbf{F}\,(\textcolor{blue}{\bullet} \wedge \mathbf{F}\,(\textcolor{green}{\bullet} \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

However...



$$\varphi \quad \equiv \quad \mathbf{G}\ \textcolor{red}{\bullet}\ \rightarrow \left\{ \quad \right.$$

Expressiveness of these logics

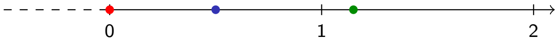
Conjecture (Alur & Henzinger, since 1990)

TPTL is strictly more expressive than MTL, and the TPTL formula

$$\varphi = \mathbf{G}\,({\color{red}\bullet} \rightarrow x \cdot \mathbf{F}\,({\color{blue}\bullet} \wedge \mathbf{F}\,({\color{green}\bullet} \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

However...



$$\varphi \quad \equiv \quad \mathbf{G}\ {\color{red}\bullet} \ \rightarrow \left\{ \begin{array}{l} \mathbf{F}_{\leqslant 1}\,{\color{blue}\bullet} \ \wedge\ \mathbf{F}_{[1,2]}\,{\color{green}\bullet} \\ \\ \\ \end{array} \right.$$

Expressiveness of these logics

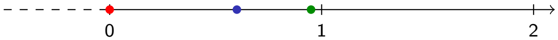
Conjecture (Alur & Henzinger, since 1990)

TPTL is strictly more expressive than MTL, and the TPTL formula

$$\varphi = \mathbf{G}\,({\color{red}\bullet} \rightarrow x \cdot \mathbf{F}\,({\color{blue}\bullet} \wedge \mathbf{F}\,({\color{green}\bullet} \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

However...



$$\varphi \equiv \mathbf{G}\ {\color{red}\bullet} \rightarrow \left\{ \begin{array}{l} \mathbf{F}_{\leqslant 1}\ {\color{blue}\bullet} \ \wedge\ \mathbf{F}_{[1,2]}\ {\color{green}\bullet} \\ \vee \\ \mathbf{F}_{\leqslant 1}\,(\ {\color{blue}\bullet}\ \wedge\ \mathbf{F}_{\leqslant 1}\ {\color{green}\bullet}\ ) \end{array} \right.$$

Expressiveness of these logics

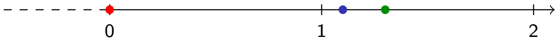
Conjecture (Alur & Henzinger, since 1990)

TPTL is strictly more expressive than MTL, and the TPTL formula

$$\varphi = \mathbf{G}\,({\color{red}\bullet} \rightarrow x \cdot \mathbf{F}\,({\color{blue}\bullet} \wedge \mathbf{F}\,({\color{green}\bullet} \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

However...



$$\varphi \quad \equiv \quad \mathbf{G}\ {\color{red}\bullet} \ \rightarrow \left\{ \begin{array}{l} \qquad \mathbf{F}_{\leqslant 1}\,{\color{blue}\bullet} \ \wedge \ \mathbf{F}_{[1,2]}\,{\color{green}\bullet} \\ \vee \\ \qquad \mathbf{F}_{\leqslant 1}\,(\ {\color{blue}\bullet} \ \wedge \ \mathbf{F}_{\leqslant 1}\,{\color{green}\bullet}\ ) \\ \vee \\ \end{array} \right.$$

Expressiveness of these logics

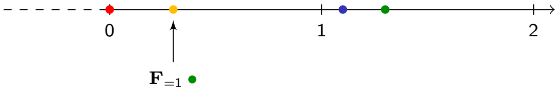
Conjecture (Alur & Henzinger, since 1990)

TPTL is strictly more expressive than MTL, and the TPTL formula

$$\varphi = \mathbf{G}\,({\color{red}\bullet} \rightarrow x \cdot \mathbf{F}\,({\color{blue}\bullet} \wedge \mathbf{F}\,({\color{green}\bullet} \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

However...



$$\varphi \quad \equiv \quad \mathbf{G}\ {\color{red}\bullet} \ \rightarrow \left\{ \begin{array}{ll} & \mathbf{F}_{\leqslant 1}\,{\color{blue}\bullet} \ \wedge \ \mathbf{F}_{[1,2]}\,{\color{green}\bullet} \\ \vee & \\ & \mathbf{F}_{\leqslant 1}\,(\,{\color{blue}\bullet} \ \wedge \ \mathbf{F}_{\leqslant 1}\,{\color{green}\bullet}\,) \\ \vee & \\ & \end{array} \right.$$

Expressiveness of these logics

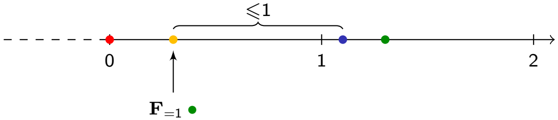
Conjecture (Alur & Henzinger, since 1990)

TPTL is strictly more expressive than MTL, and the TPTL formula

$$\varphi = \mathbf{G}\,({\color{red}\bullet} \rightarrow x\cdot \mathbf{F}\,({\color{blue}\bullet} \wedge \mathbf{F}\,({\color{green}\bullet} \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

However...



$$\varphi \quad\equiv\quad \mathbf{G}\ {\color{red}\bullet} \ \rightarrow \left\{ \begin{array}{ll} & \mathbf{F}_{\leqslant 1}\,{\color{blue}\bullet} \ \wedge\ \mathbf{F}_{[1,2]}\,{\color{green}\bullet} \\ \vee & \\ & \mathbf{F}_{\leqslant 1}\,(\,{\color{blue}\bullet}\ \wedge\ \mathbf{F}_{\leqslant 1}\,{\color{green}\bullet}\,) \\ \vee & \\ & \end{array} \right.$$

Expressiveness of these logics

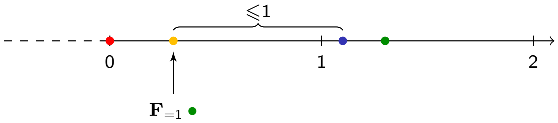
Conjecture (Alur & Henzinger, since 1990)

TPTL is strictly more expressive than MTL, and the TPTL formula

$$\varphi = \mathbf{G}\,({\color{red}\bullet} \rightarrow x \cdot \mathbf{F}\,({\color{blue}\bullet} \wedge \mathbf{F}\,({\color{green}\bullet} \wedge x \leqslant 2)))$$

cannot be expressed in MTL.

However...



$$\varphi \quad \equiv \quad \mathbf{G}\ {\color{red}\bullet} \ \rightarrow \begin{cases} \quad \mathbf{F}_{\leqslant 1}\,{\color{blue}\bullet} \ \wedge \ \mathbf{F}_{[1,2]}\,{\color{green}\bullet} \\ \vee \\ \quad \mathbf{F}_{\leqslant 1}\,(\ {\color{blue}\bullet} \ \wedge \ \mathbf{F}_{\leqslant 1}\,{\color{green}\bullet}\ ) \\ \vee \\ \quad \mathbf{F}_{\leqslant 1}\ (\ \mathbf{F}_{\leqslant 1}\,{\color{blue}\bullet} \ \wedge \ \mathbf{F}_{=1}\,{\color{green}\bullet}\ ) \end{cases}$$

Expressiveness results

Theorem [BCM05]

The conjecture is correct: TPTL is strictly more expressive than MTL.

[BCM05] Bouyer, Chevalier, Markey. On the expressiveness of TPTL and MTL *(FSTTCS'05)*.

Expressiveness results

Theorem [BCM05]

The conjecture is correct: TPTL is strictly more expressive than MTL. Also, MTL+Past is strictly more expressive than MTL.

Recall: LTL+Past and LTL are equally expressive [Kam68,GPSS80].

[Kam68] Kamp. Tense logic and the theory of linear order *(PhD UCLA)*.
[GPSS80] Gabbay, Pnueli, Shelah, Stavi. On the temporal analysis of fairness *(POPL'80)*.
[BCM05] Bouyer, Chevalier, Markey. On the expressiveness of TPTL and MTL *(FSTTCS'05)*.

Expressiveness results

Theorem [BCM05]

The conjecture is correct: TPTL is strictly more expressive than MTL.
Also, MTL+Past is strictly more expressive than MTL.

The formulas

$$x \cdot \mathbf{F}\,(\bullet \wedge (x \leqslant 1) \wedge \mathbf{G}\,((x \leqslant 1) \rightarrow \neg\bullet)) \quad \in \text{TPTL}$$
$$\mathbf{F}_{=1}\,(\neg\bullet\,\mathbf{S}\,\bullet) \quad \in \text{MTL+Past}$$

cannot be expressed in MTL.

[BCM05] Bouyer, Chevalier, Markey. On the expressiveness of TPTL and MTL *(FSTTCS'05)*.

Expressiveness results

Theorem [BCM05]

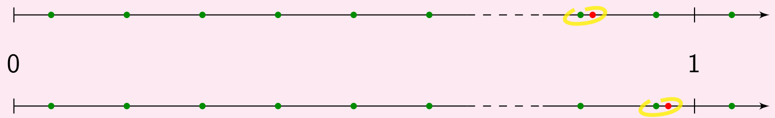
The conjecture is correct: TPTL is strictly more expressive than MTL.
Also, MTL+Past is strictly more expressive than MTL.

The formulas

$$x \cdot \mathbf{F}\,({\color{green}\bullet} \wedge (x \leqslant 1) \wedge \mathbf{G}\,((x \leqslant 1) \rightarrow \neg {\color{red}\bullet})) \quad \in \text{TPTL}$$

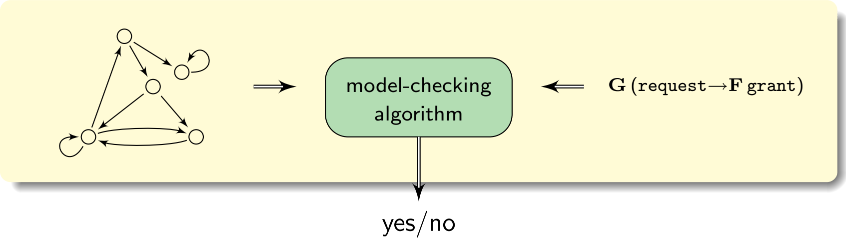
$$\mathbf{F}_{=1}\,(\neg {\color{red}\bullet}\ \mathbf{S}\ {\color{green}\bullet}) \quad \in \text{MTL+Past}$$

cannot be expressed in MTL.

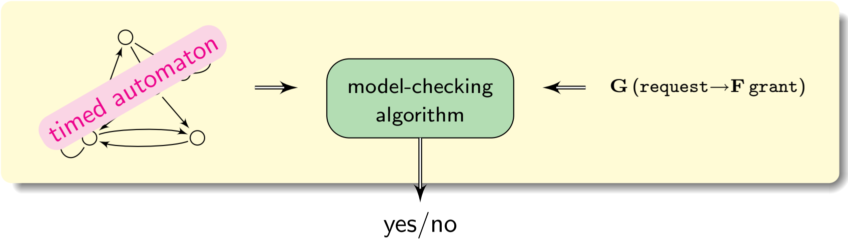


[BCM05] Bouyer, Chevalier, Markey. On the expressiveness of TPTL and MTL *(FSTTCS'05)*.

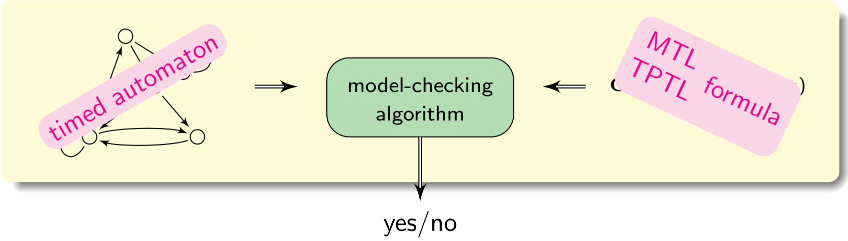
Back to the model-checking problem



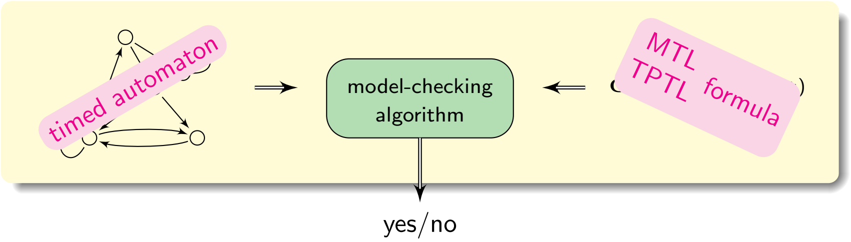
Back to the model-checking problem



Back to the model-checking problem



Back to the model-checking problem

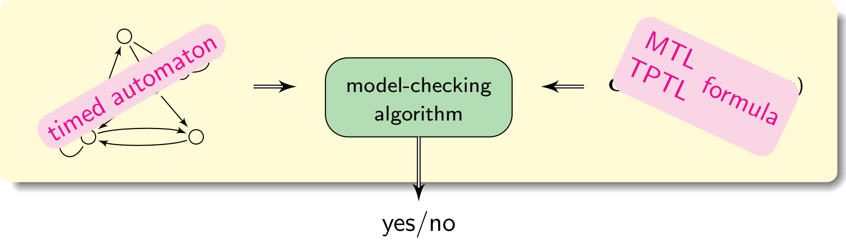


Theorem [AH94,AFH96,OW06...]

The model-checking problem is (mostly) undecidable...

[AH94] Alur, Henzinger. A really temporal logic *(Journal of the ACM)*.
[AFH96] Alur, Feder, Henzinger. The benefits of relaxing unctuality *(Journal of the ACM)*.
[OW06] Ouaknine, Worrell. On Metric Temporal Logic and faulty Turing machines *(FoSSaCS'06)*.

Back to the model-checking problem



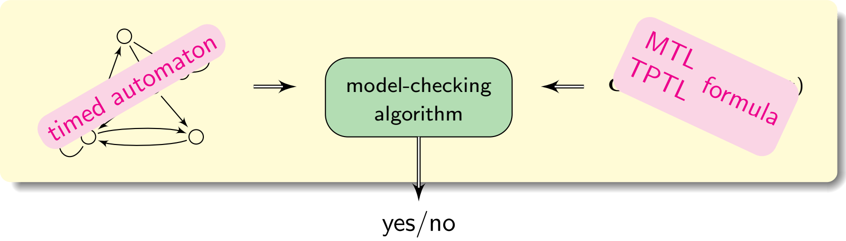
Theorem [AH94,AFH96,OW06...]

The model-checking problem is (mostly) undecidable...

- Can be rather easily explained using channel machines...

[AH94] Alur, Henzinger. A really temporal logic *(Journal of the ACM)*.
[AFH96] Alur, Feder, Henzinger. The benefits of relaxing unctuality *(Journal of the ACM)*.
[OW06] Ouaknine, Worrell. On Metric Temporal Logic and faulty Turing machines *(FoSSaCS'06)*.

Back to the model-checking problem



Theorem [AH94,AFH96,OW06...]

The model-checking problem is (mostly) undecidable...

- Can be rather easily explained using channel machines...
- ... and more tractable fragments need be defined!

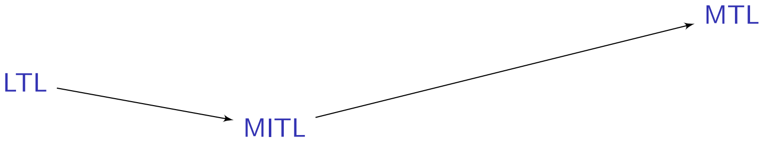
[AH94] Alur, Henzinger. A really temporal logic *(Journal of the ACM)*.
[AFH96] Alur, Feder, Henzinger. The benefits of relaxing unctuality *(Journal of the ACM)*.
[OW06] Ouaknine, Worrell. On Metric Temporal Logic and faulty Turing machines *(FoSSaCS'06)*.

The quest for tractable fragments of MTL

MTL

LTL

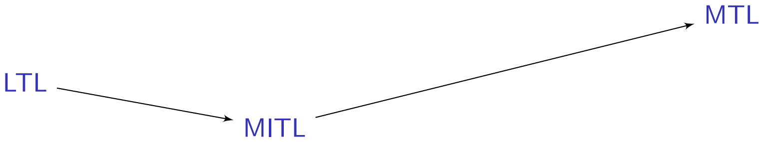
The quest for tractable fragments of MTL



- MITL [AFH96]: ban punctuality
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{\leqslant 2}\,\bullet)$ is in MITL
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is not in MITL

[AFH96] Alur, Feder, Henzinger. The benefits of relaxing unctuality *(Journal of the ACM)*.

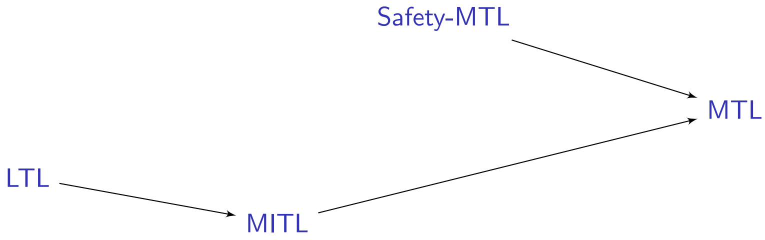
The quest for tractable fragments of MTL



- MITL [AFH96]: ban punctuality
  - timed regularity
  - verification in exponential space

[AFH96] Alur, Feder, Henzinger. The benefits of relaxing unctuality *(Journal of the ACM)*.

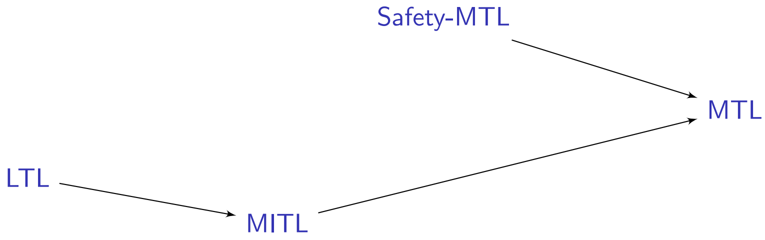
The quest for tractable fragments of MTL



- Safety-MTL [OW05]: restrict to safety properties
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{\leqslant 2}\,\bullet)$ is in Safety-MTL
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}\,\bullet)$ is not in Safety-MTL

[OW05] Ouaknine, Worrell. On the decidability of Metric Temporal Logic *(LICS'05)*.

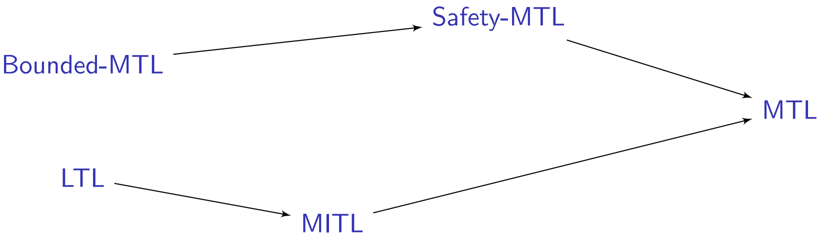
The quest for tractable fragments of MTL



- Safety-MTL [OW05]: restrict to safety properties
  - only finite counter-examples need be looked for
  - verification using alternating timed automata
  - decidable, but non-primitive recursive [OW08]

[OW05] Ouaknine, Worrell. On the decidability of Metric Temporal Logic *(LICS'05)*.
[OW08] Ouaknine, Worrell. Some recent results in Metric Temporal Logic *(FORMATS'08)*.

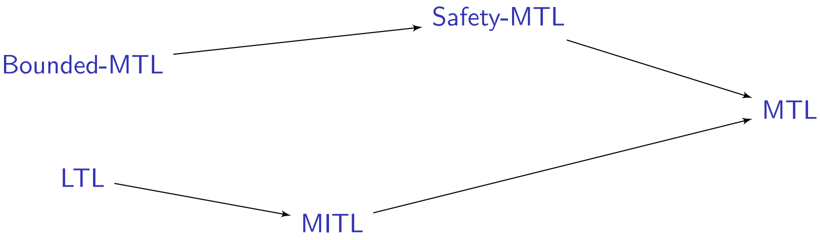
The quest for tractable fragments of MTL



- Bounded-MTL [BMOW07]: restrict to bounded future
  - $\mathbf{G}_{\leqslant 5}\,({\color{red}\bullet} \rightarrow \mathbf{F}_{\leqslant 2}\,{\color{green}\bullet})$ is in Bounded-MTL
  - $\mathbf{G}\,({\color{red}\bullet} \rightarrow \mathbf{F}_{\leqslant 2}\,{\color{green}\bullet})$ is not in Bounded-MTL

[BMOW07] Bouyer, Markey, Ouaknine, Worrell. The cost of punctuality *(LICS'07)*.

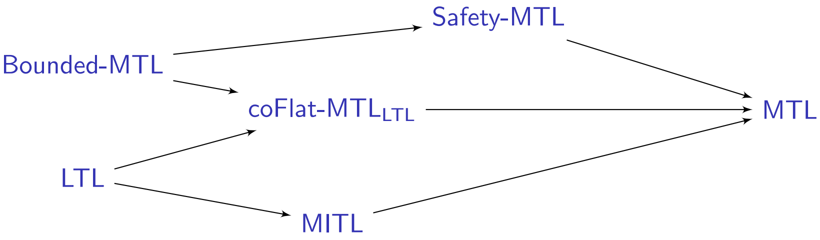
The quest for tractable fragments of MTL



- Bounded-MTL [BMOW07]: restrict to bounded future
  - expresses non-regular properties
  - only time-bounded prefixes need to be verified
  - verification in exponential space

[BMOW07] Bouyer, Markey, Ouaknine, Worrell. The cost of punctuality *(LICS'07)*.

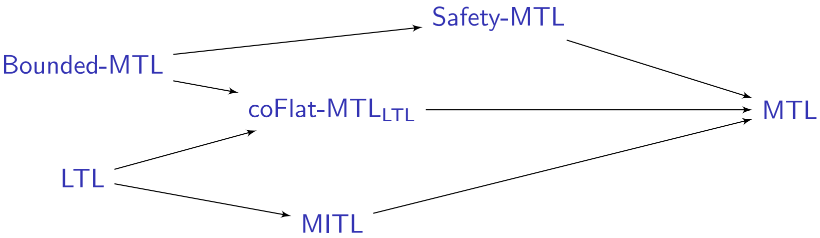
The quest for tractable fragments of MTL



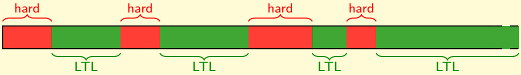
- coFlat-MTL$_{\text{LTL}}$ [BMOW07]: a flatness condition on the formula
  - $\mathbf{G}\,(\bullet \rightarrow \mathbf{F}_{=1}\,\bullet)$ is in coFlat-MTL$_{\text{LTL}}$
  - $\mathbf{F}\,\mathbf{G}_{\leqslant 1}\,\bullet$ is not in coFlat-MTL$_{\text{LTL}}$

[BMOW07] Bouyer, Markey, Ouaknine, Worrell. The cost of punctuality *(LICS'07)*.

The quest for tractable fragments of MTL



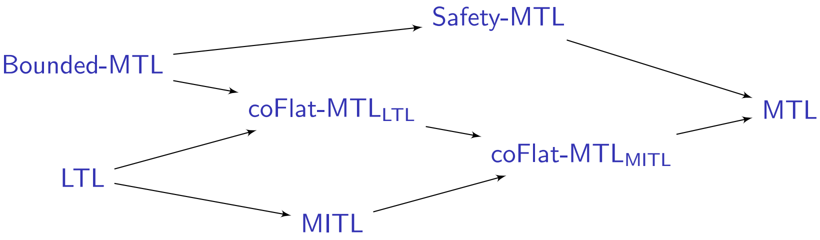
- coFlat-MTL$_{\text{LTL}}$ [BMOW07]: a flatness condition on the formula
  - expresses non-regular properties
  - only counter-examples of the following form need to be looked for:



  - verification in exponential space

[BMOW07] Bouyer, Markey, Ouaknine, Worrell. The cost of punctuality *(LICS'07)*.

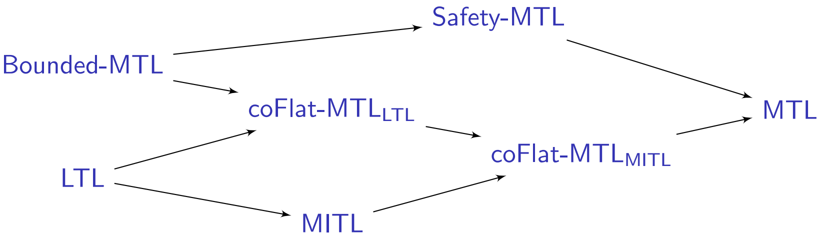
The quest for tractable fragments of MTL



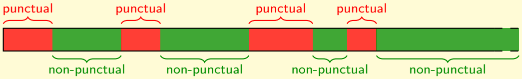
- coFlat-MTL_{MITL} [BMOW08]: a flatness condition on the formula
  - $\mathbf{F}\,\mathbf{G}_{\leqslant 1}\,\bullet$ is in coFlat-MTL_{MITL}
  - $\mathbf{F}\,\mathbf{G}_{=1}\,\bullet$ is not in coFlat-MTL_{MITL}

[BMOW08] Bouyer, Markey, Ouaknine, Worrell. On expressiveness and complexity in real-time model checking *(ICALP'08)*.

The quest for tractable fragments of MTL



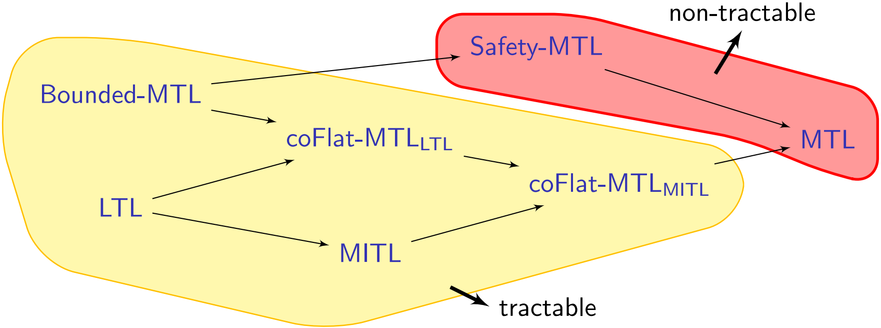
- coFlat-MTL$_{\mathsf{MITL}}$ [BMOW08]: a flatness condition on the formula
  - only counter-examples of the following form need to be looked for:



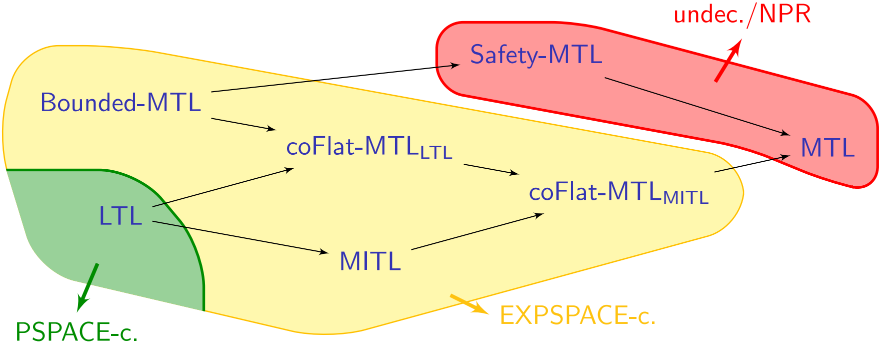
  - the largest (known) fragment for which...
  - ... verification in exponential space!

[BMOW08] Bouyer, Markey, Ouaknine, Worrell. On expressiveness and complexity in real-time model checking *(ICALP'08)*.

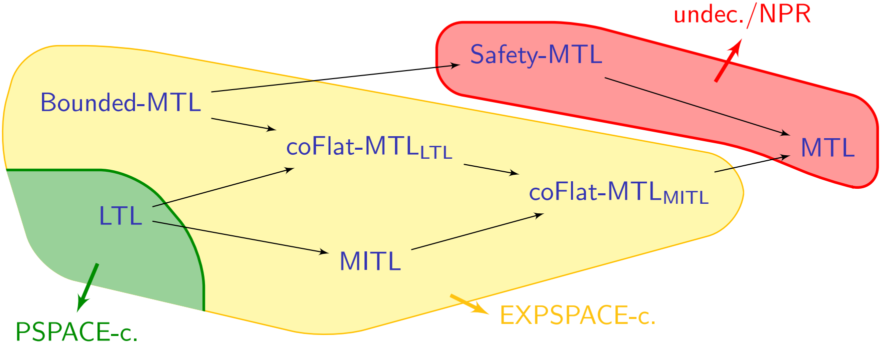
The quest for tractable fragments of MTL



The quest for tractable fragments of MTL



The quest for tractable fragments of MTL



Algorithmics needs now to be worked out!

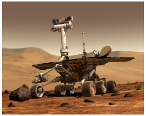
Outline

Motivation

- System resources might be relevant and even crucial information

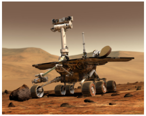
Motivation

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



Motivation

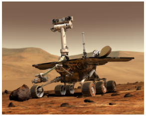
- System resources might be relevant and even crucial information
  - energy consumption,
  - memory usage,
  - price to pay,
  - bandwidth,
  - ...



$\rightsquigarrow$ timed automata are not powerful enough!

Motivation

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

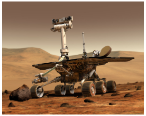
$\rightsquigarrow$ timed automata are not powerful enough!

- A possible solution: use hybrid automata

Motivation

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

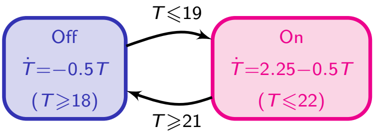




$\rightsquigarrow$ timed automata are not powerful enough!

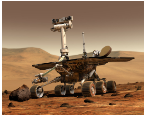
- A possible solution: use hybrid automata

The thermostat example



Motivation

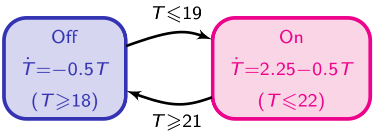
- System resources might be relevant and even crucial information
  - energy consumption,
  - memory usage,
  - price to pay,
  - bandwidth,
  - ...

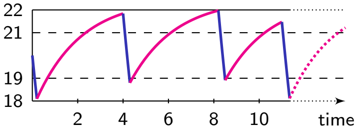


$\rightsquigarrow$ timed automata are not powerful enough!

- A possible solution: use hybrid automata

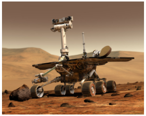
The thermostat example





Motivation

- 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

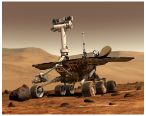
Theorem [HKPV95]

The reachability problem is undecidable in hybrid automata.

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

Motivation

- 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

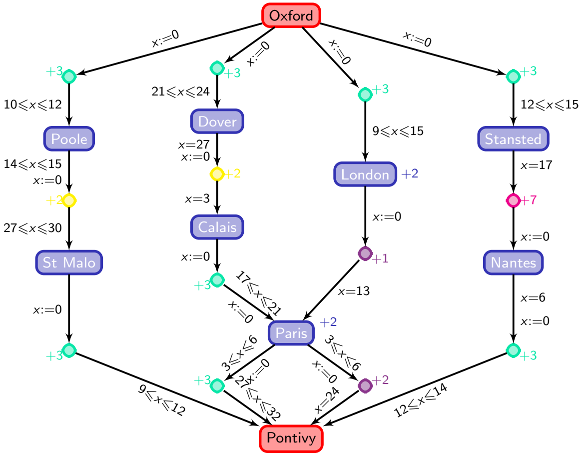
Theorem [HKPV95]

The reachability problem is undecidable in hybrid automata.

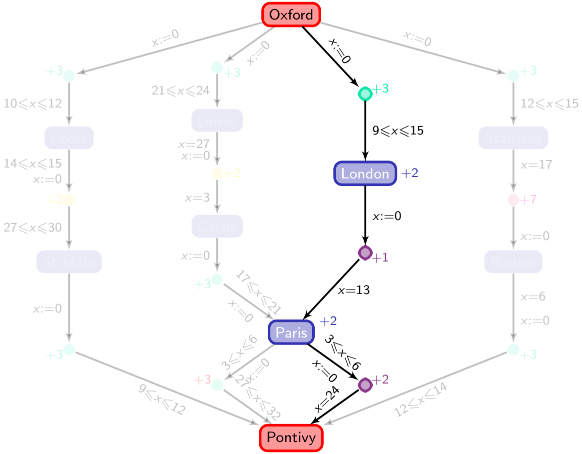
- An alternative: weighted/priced timed automata [ALP01,BFH+01]
  $\rightsquigarrow$ hybrid variables do not constrain the system

[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)*.

A third model of the system

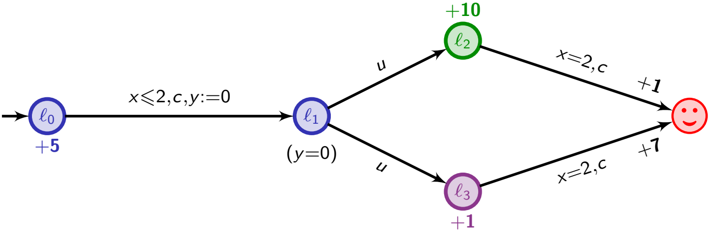


How much fuel will I use?



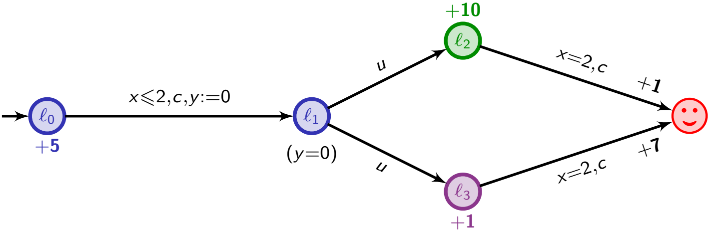
It is a quantitative (optimization) question
in a weighted timed automaton: at least 68 anti-planet units!

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)*.

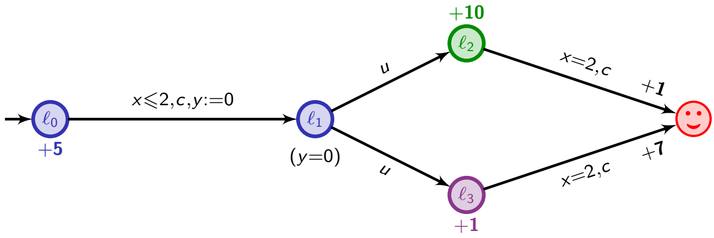
Weighted/priced timed automata [ALP01,BFH+01]



	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]

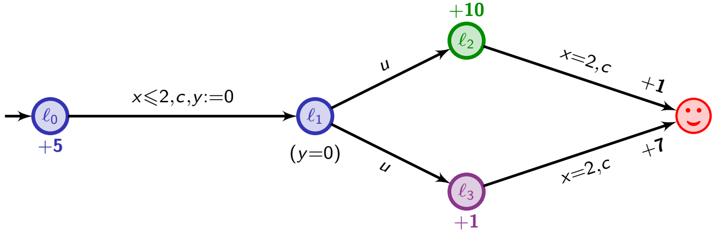


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

cost :

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

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

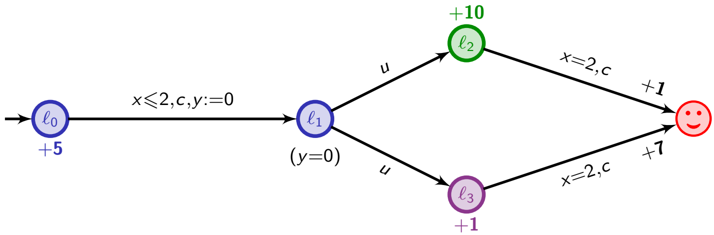


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

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]

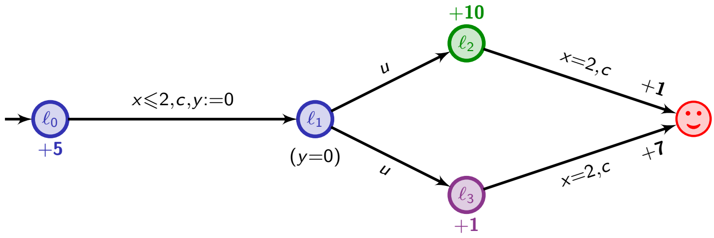


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

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]

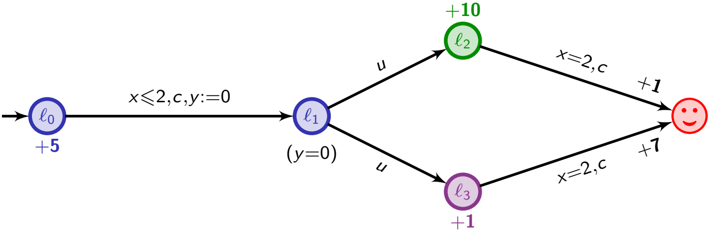


	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺
x	0		1.3		1.3		1.3		2		
y	0		1.3		0		0		0.7		

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]

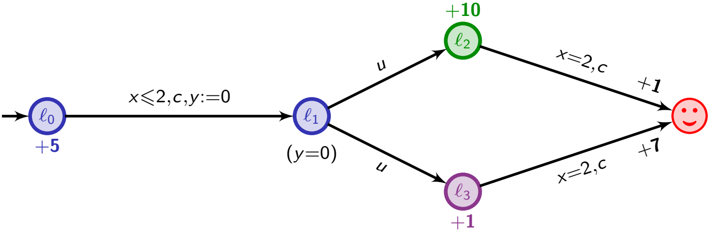


$$\begin{array}{lcccccccccccc} & \ell_0 & \xrightarrow{1.3} & \ell_0 & \xrightarrow{c} & \ell_1 & \xrightarrow{u} & \ell_3 & \xrightarrow{0.7} & \ell_3 & \xrightarrow{c} & ☺ \\ x & 0 & & 1.3 & & 1.3 & & 1.3 & & 2 & & \\ y & 0 & & 1.3 & & 0 & & 0 & & 0.7 & & \end{array}$$

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)*.

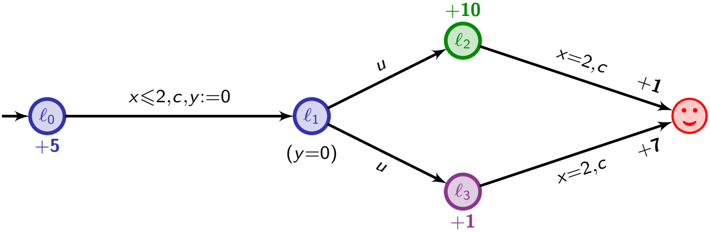
Weighted/priced timed automata [ALP01,BFH+01]



	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)*.

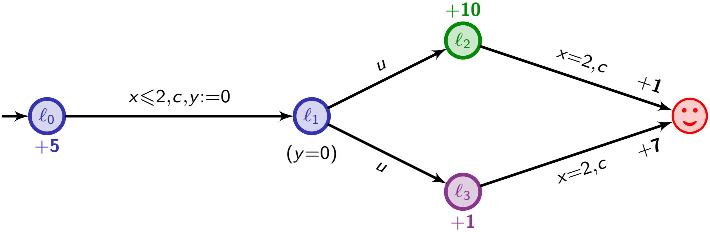
Weighted/priced timed automata [ALP01,BFH+01]



	ℓ_0	$\xrightarrow{1.3}$	ℓ_0	$\xrightarrow{c}$	ℓ_1	$\xrightarrow{u}$	ℓ_3	$\xrightarrow{0.7}$	ℓ_3	$\xrightarrow{c}$	☺		
x	0		1.3		1.3		1.3		2				
y	0		1.3		0		0		0.7				
cost :		6.5	+	0	+	0	+	0.7	+	7		=	14.2

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

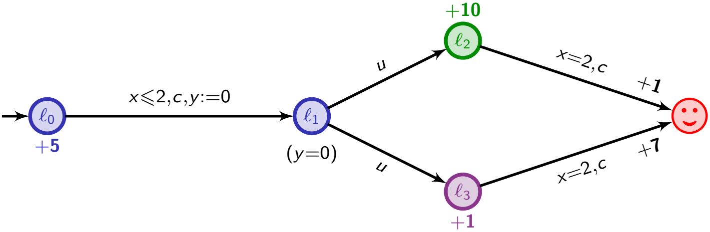
Weighted/priced timed automata [ALP01,BFH+01]



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]

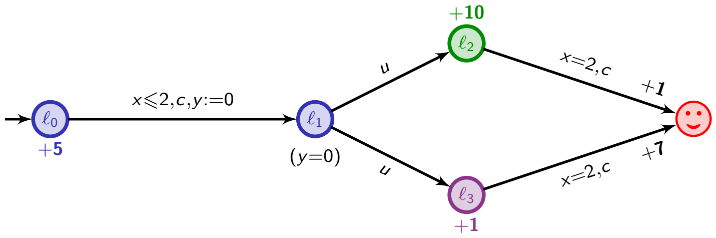


Question: what is the optimal cost for reaching 🙂 ?

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

[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]

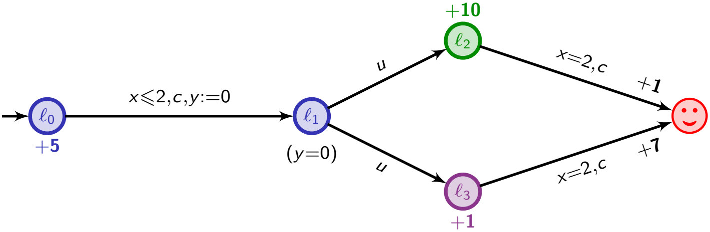


Question: what is the optimal cost for reaching 🙂 ?

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

[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]

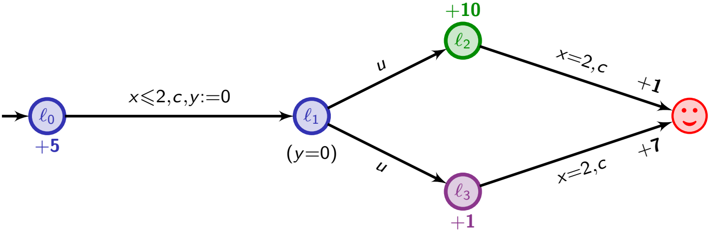


Question: what is the optimal cost for reaching ☺ ?

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

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

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

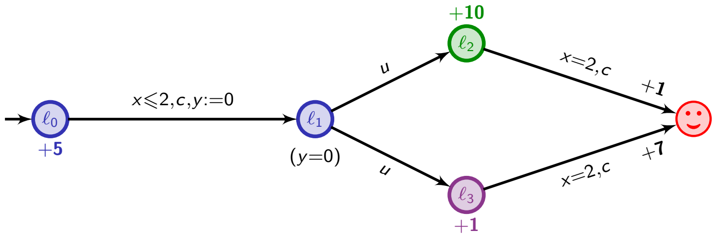


Question: what is the optimal cost for reaching 🙂 ?

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

[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 🙂?

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

$\rightsquigarrow$ *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)*.

Optimization problems in weighted timed automata

Theorem [ALP01,BFH+01,BBBR07]

The optimal reachability problem is decidable (and PSPACE-complete) in (weighted) 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)*.

Optimization problems in weighted timed automata

Theorem [ALP01,BFH+01,BBBR07]

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

Theorem [BBL04,BBL08]

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

[BBL04] Bouyer, Brinksma, Larsen. Staying alive as cheaply as possible *(HSCC'04)*.
[BBL08] Bouyer, Brinksma, Larsen. Optimal infinite scheduling for multi-priced timed automata *(Formal Methods in System Design)*.

Optimization problems in weighted timed automata

Theorem [ALP01,BFH+01,BBBR07]

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

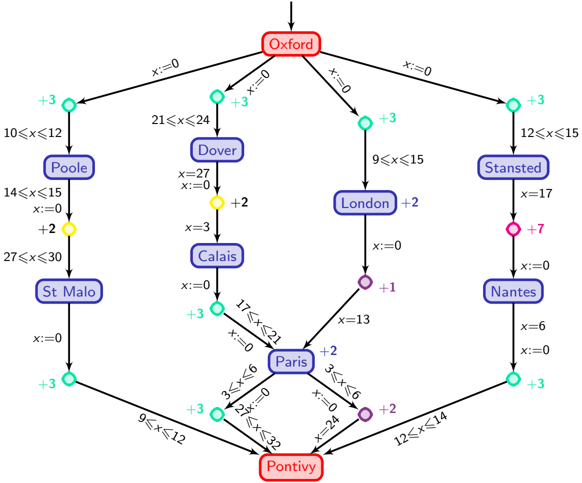
Theorem [BBL04,BBL08]

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

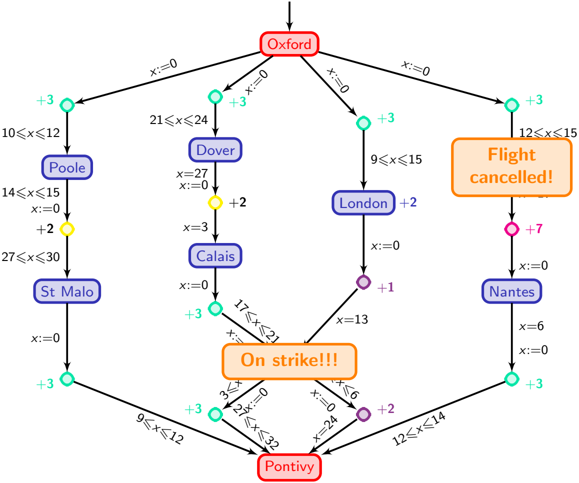
$\rightsquigarrow$ In both cases, the corner-point abstraction can be used (a refinement of the region automaton)

[BBL04] Bouyer, Brinksma, Larsen. Staying alive as cheaply as possible *(HSCC'04)*.
[BBL08] Bouyer, Brinksma, Larsen. Optimal infinite scheduling for multi-priced timed automata *(Formal Methods in System Design)*.

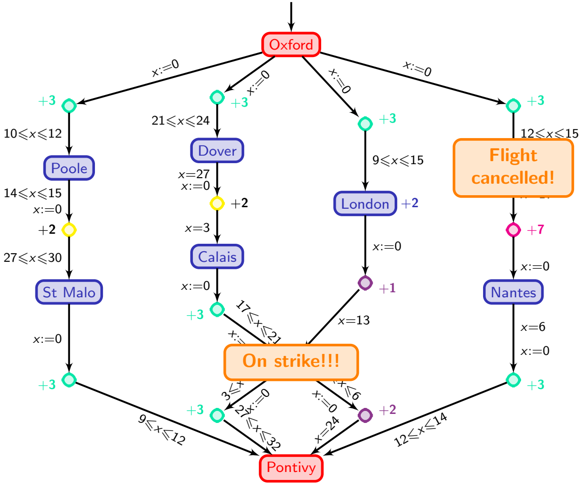
What if an unexpected event happens?



What if an unexpected event happens?

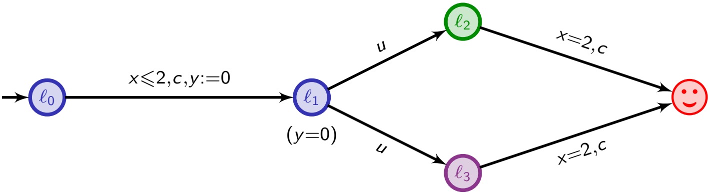


What if an unexpected event happens?

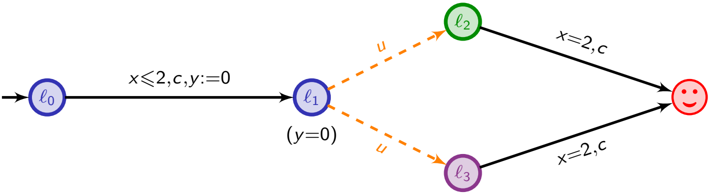


⤳ modelled as timed games

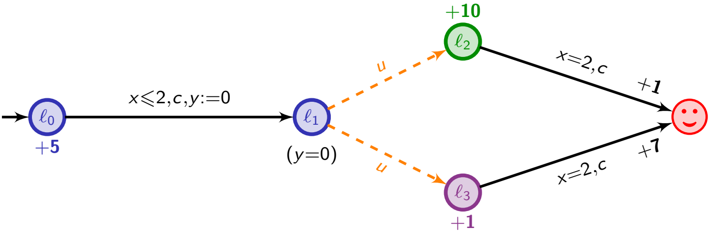
Weighted timed games



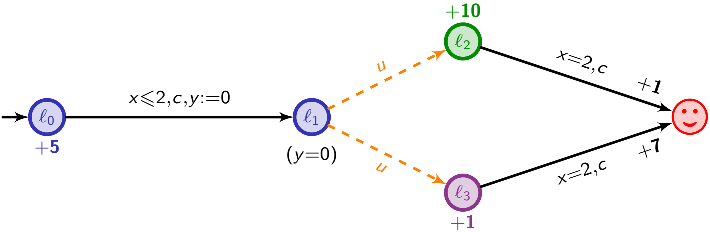
Weighted timed games



Weighted timed games

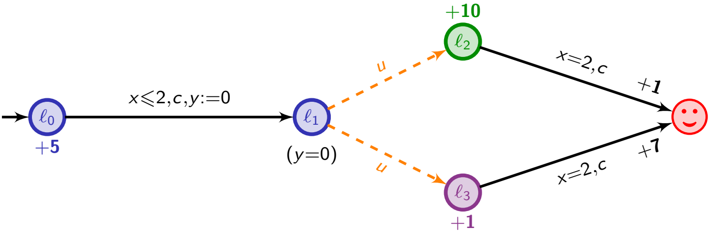


Weighted timed games



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

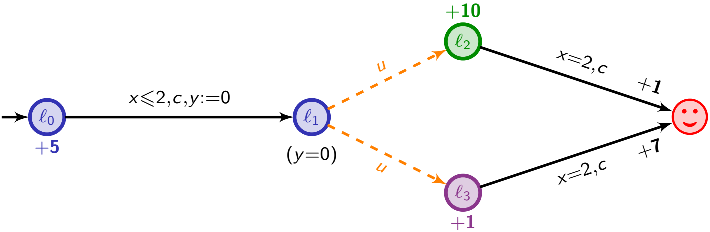
Weighted timed games



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

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

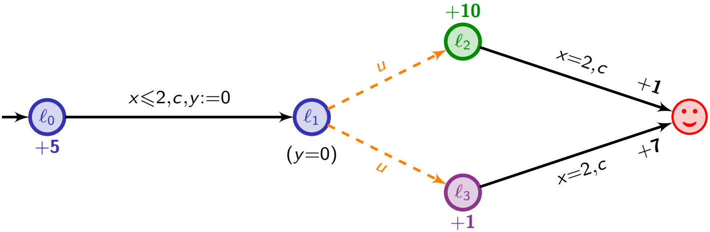
Weighted timed games



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

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

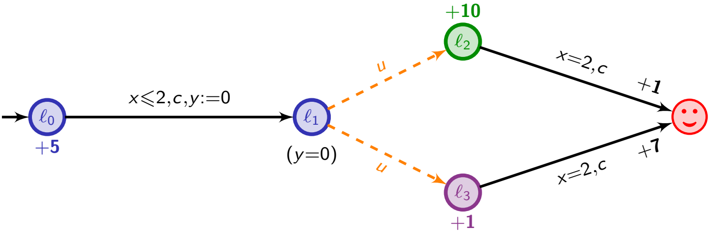
Weighted timed games



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

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

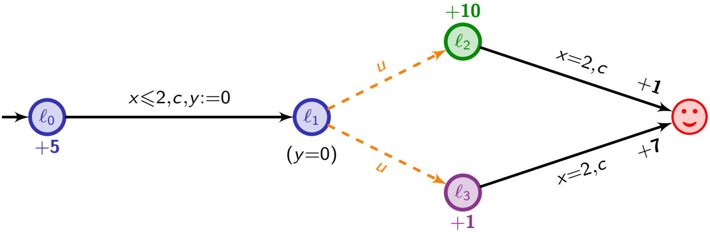
Weighted timed games



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

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

Weighted timed games



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

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

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

Optimal reachability in weighted 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 weighted 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 weighted timed games

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

e.g. [LMM02,ABM04,BCFL04]

Theorem [BBR05,BBM06]

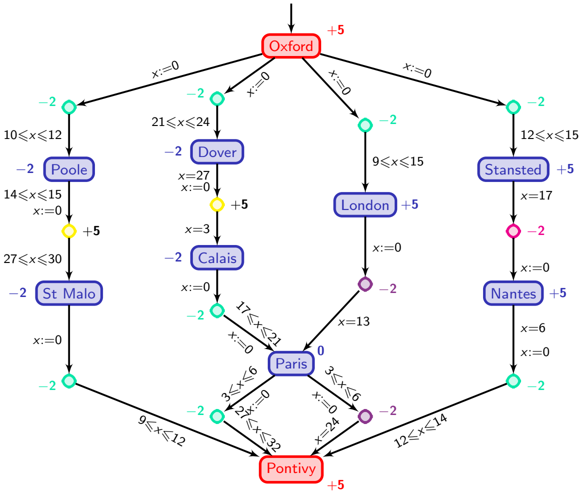
Optimal timed games are undecidable, as soon as automata have three clocks or more.

Theorem [BLMR06]

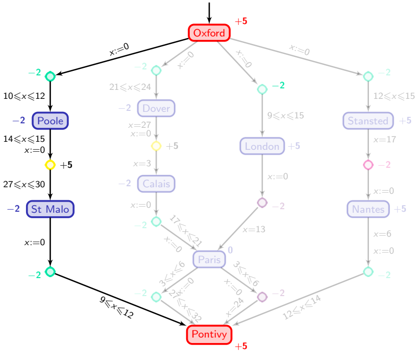
Turn-based optimal timed games are decidable in 3EXPTIME when automata have a single clock. They are PTIME-hard.

[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)*.

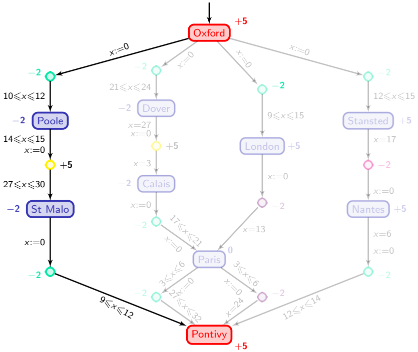
A fourth model of the system

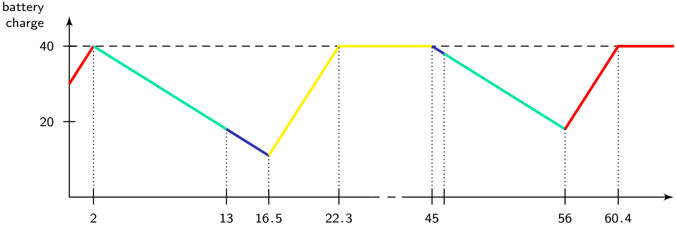


Can I work with my computer all the way?



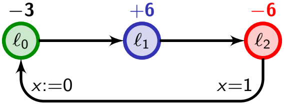
Can I work with my computer all the way?



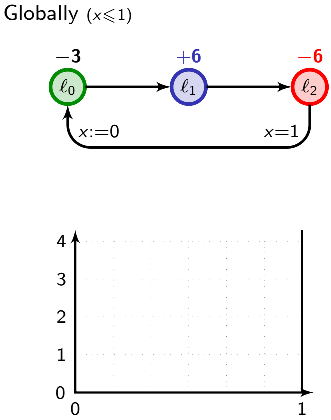


An example of resource management

Globally $(x \leqslant 1)$

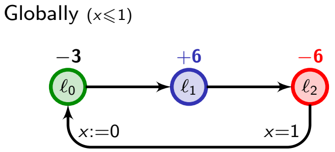


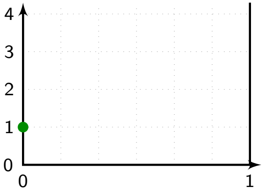
An example of resource management



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

An example of resource management

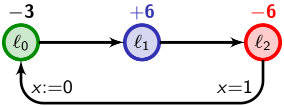


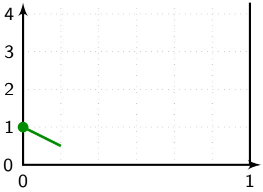


- Lower-bound problem: can we stay above 0?

An example of resource management

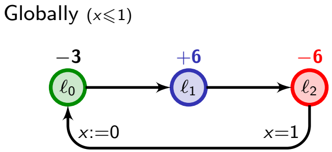
Globally $(x \leqslant 1)$

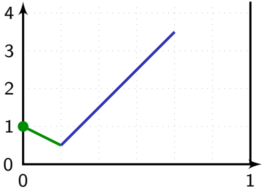




- Lower-bound problem: can we stay above 0?

An example of resource management

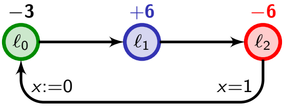


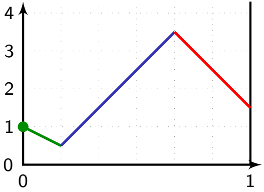


- Lower-bound problem: can we stay above 0?

An example of resource management

Globally $(x \leqslant 1)$

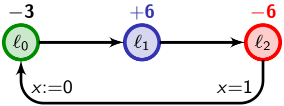


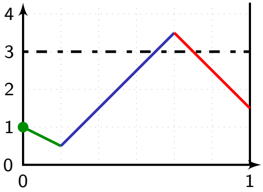


- Lower-bound problem: can we stay above 0?

An example of resource management

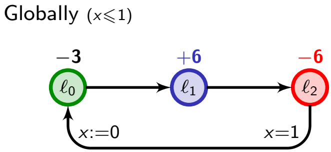
Globally $(x \leqslant 1)$

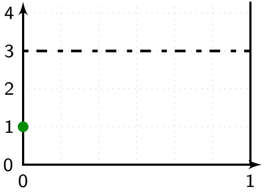




- Lower-bound problem: can we stay above 0?

An example of resource management

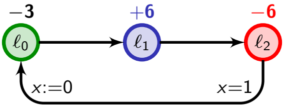


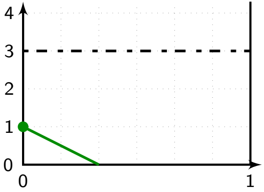


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leqslant 1)$

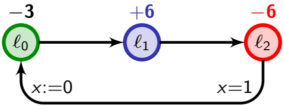


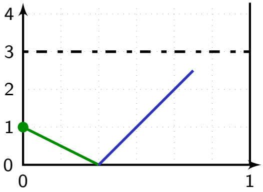


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leqslant 1)$

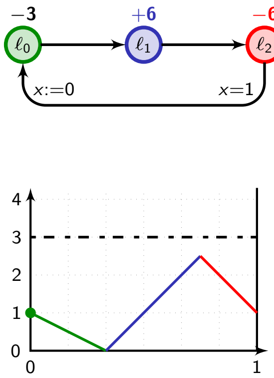




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

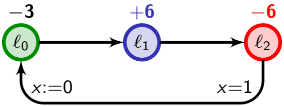
Globally $(x \leqslant 1)$

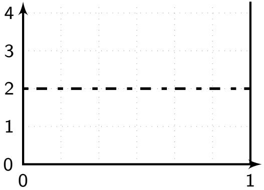


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

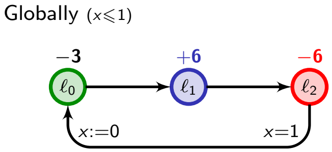
Globally $(x\leqslant 1)$

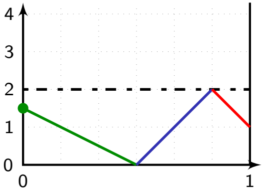




- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

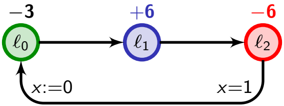


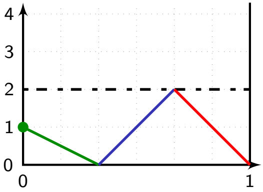


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

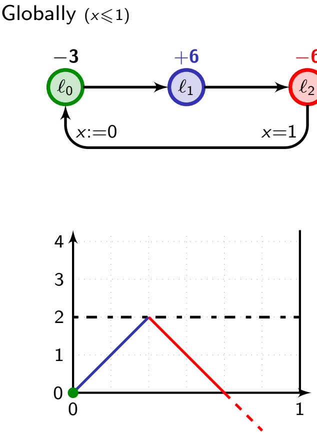
Globally $(x \leqslant 1)$





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

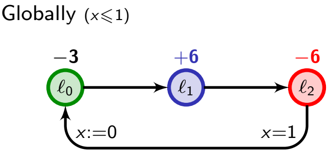
An example of resource management

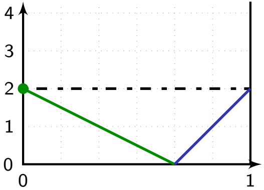


lost!

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

An example of resource management

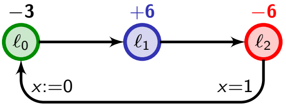


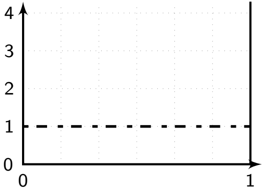


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

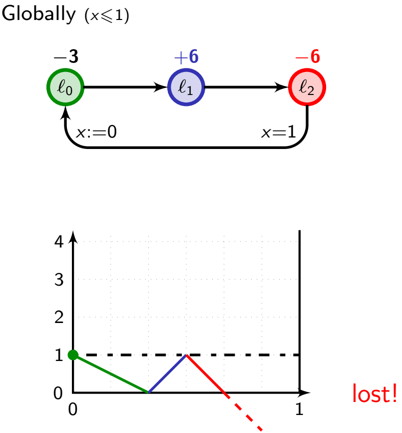
Globally $(x \leqslant 1)$





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

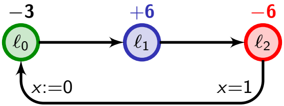
An example of resource management

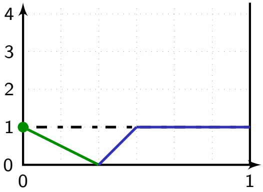


- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?

An example of resource management

Globally $(x \leqslant 1)$

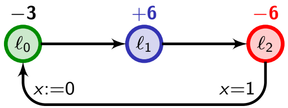


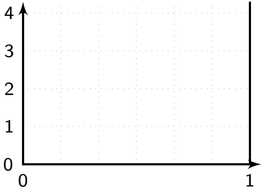


- Lower-bound problem
- Lower-upper-bound problem
- Lower-weak-upper-bound problem: can we "weakly" stay within bounds?

An example of resource management

Globally $(x \leqslant 1)$





- 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)*.

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

Theorem

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

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

Theorem

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

We encode the behaviour of a two-counter machine:

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

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

  when entering the corresponding module.

Single-clock **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

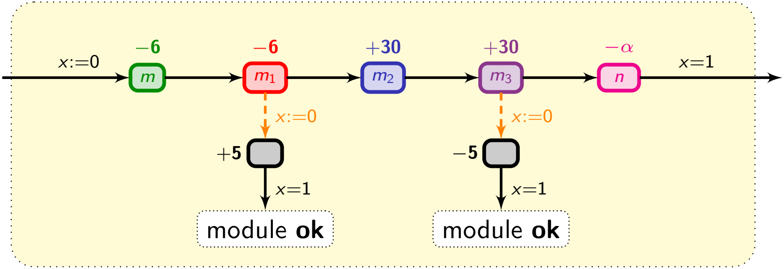
$$e = 5 - \frac{1}{2^{c_1} \cdot 3^{c_2}}$$

  when entering the corresponding module.

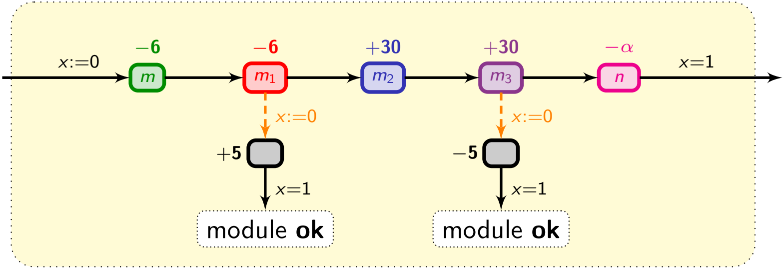
There is an infinite execution in the two-counter machine iff there is a strategy in the single-clock timed game under which the energy level remains between 0 and 5.

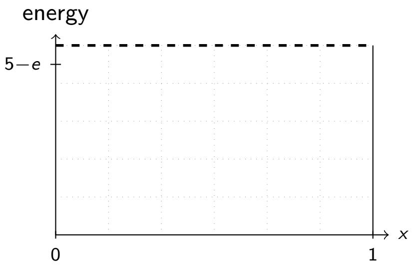
$\rightsquigarrow$ We present a generic construction for incrementing/decrementing the counters.

Generic module for incrementing/decrementing

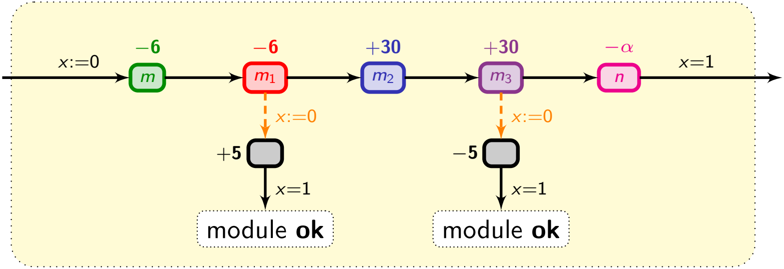


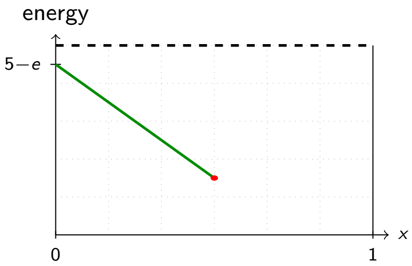
Generic module for incrementing/decrementing



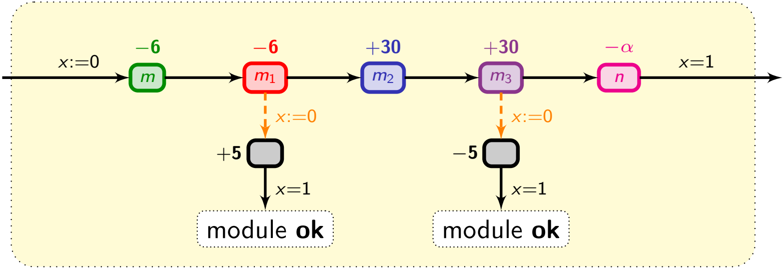


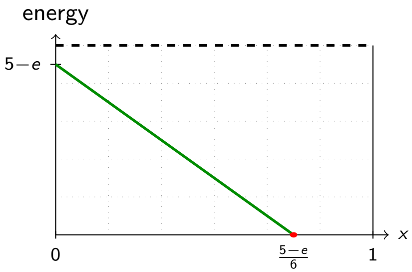
Generic module for incrementing/decrementing



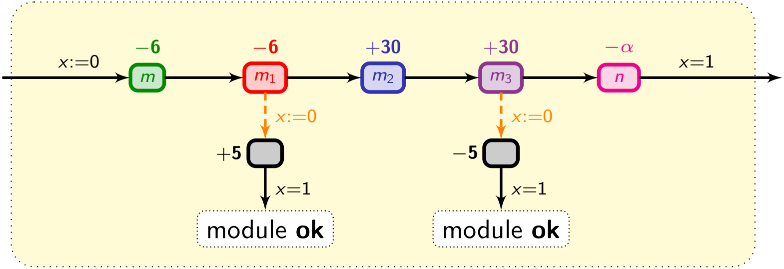


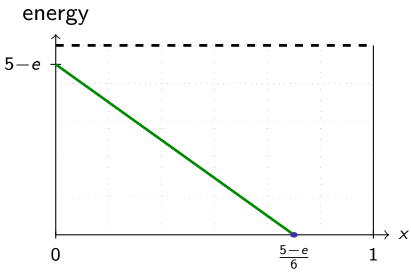
Generic module for incrementing/decrementing



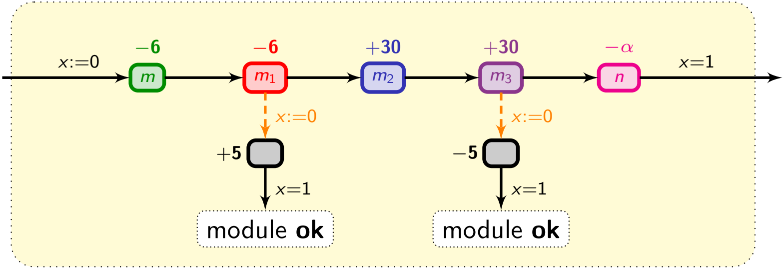


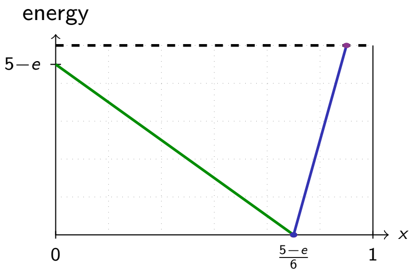
Generic module for incrementing/decrementing



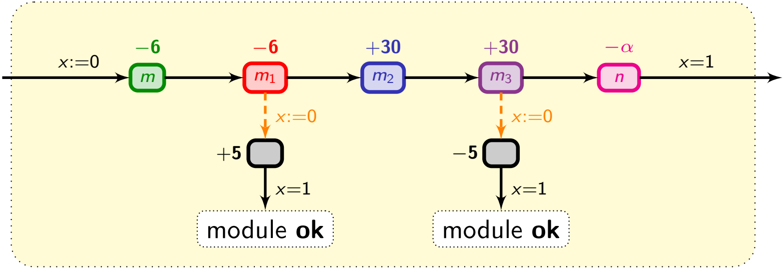


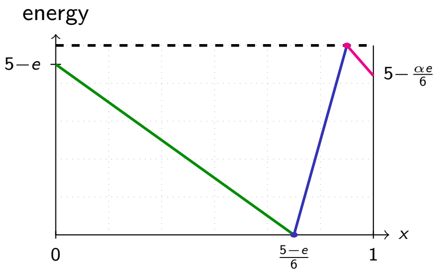
Generic module for incrementing/decrementing



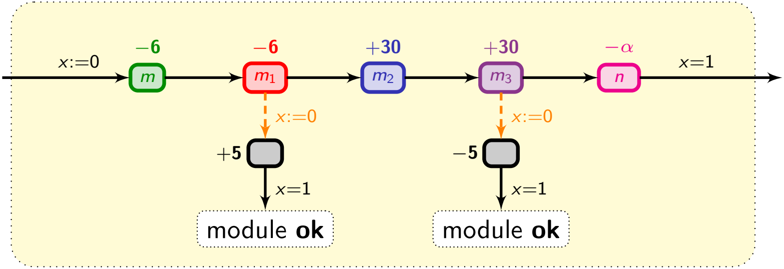


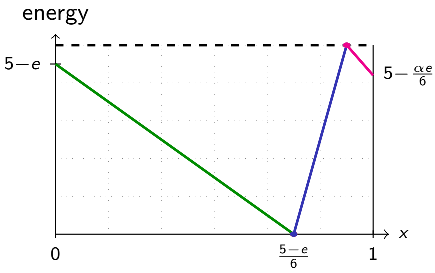
Generic module for incrementing/decrementing





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

Partial conclusion and perspectives

Weighted timed automata, an interesting model for representing resources in timed systems

Partial conclusion and perspectives

Weighted timed automata, an interesting model for representing resources in timed systems

- ☺ some optimization problems are (fully) decidable (with a reasonable complexity)

Partial conclusion and perspectives

Weighted timed automata, an interesting model for representing resources in timed systems

- ☺ some optimization problems are (fully) decidable (with a reasonable complexity)

  $\rightsquigarrow$ algorithmics needs to be further developed

Partial conclusion and perspectives

Weighted timed automata, an interesting model for representing resources in timed systems

- ☺ some optimization problems are (fully) decidable
  (with a reasonable complexity)
  $\leadsto$ algorithmics needs to be further developed

- ☹ many problems are undecidable in general
  (we did not mention the model-checking problem, but it is mostly undecidable)

Partial conclusion and perspectives

Weighted timed automata, an interesting model for representing resources in timed systems

☺ some optimization problems are (fully) decidable
(with a reasonable complexity)
$\rightsquigarrow$ algorithmics needs to be further developed

☹ many problems are undecidable in general
(we did not mention the model-checking problem, but it is mostly undecidable)

- restriction of the underlying timed automaton to one clock and development of specific algorithms

Partial conclusion and perspectives

Weighted timed automata, an interesting model for representing resources in timed systems

☺ some optimization problems are (fully) decidable
(with a reasonable complexity)
$\rightsquigarrow$ algorithmics needs to be further developed

☹ many problems are undecidable in general
(we did not mention the model-checking problem, but it is mostly undecidable)

- restriction of the underlying timed automaton to one clock and development of specific algorithms
- development of approximation schemes

Partial conclusion and perspectives

Weighted timed automata, an interesting model for representing resources in timed systems

- ☺ some optimization problems are (fully) decidable (with a reasonable complexity)

  $\rightsquigarrow$ algorithmics needs to be further developed

- ☹ many problems are undecidable in general

  (we did not mention the model-checking problem, but it is mostly undecidable)

  - restriction of the underlying timed automaton to one clock and development of specific algorithms
  - development of approximation schemes

- Many open problems to be solved, e.g. in resource management

Partial conclusion and perspectives

Weighted timed automata, an interesting model for representing resources in timed systems

☺ some optimization problems are (fully) decidable
(with a reasonable complexity)
$\leadsto$ algorithmics needs to be further developed

☹ many problems are undecidable in general
(we did not mention the model-checking problem, but it is mostly undecidable)

- restriction of the underlying timed automaton to one clock and development of specific algorithms
- development of approximation schemes

- Many open problems to be solved, e.g. in resource management
- Compute equilibria in weighted timed games
$\leadsto$ towards a theory of timed games

Outline

Motivation

The goal

Define a (meaningful) measure on runs of timed automata that will tell us how likely properties are satisfied.

Motivation

The goal

Define a (meaningful) measure on runs of timed automata that will tell us how likely properties are satisfied.

In the running example: "how likely will I visit Paris?"
"how long should I expect be waiting in Paris?"

Motivation

The goal

Define a (meaningful) measure on runs of timed automata that will tell us how likely properties are satisfied.

In the running example: "how likely will I visit Paris?"
"how long should I expect be waiting in Paris?"

- A relaxed semantics for timed automata
  - removes behaviours that are unlikely to happen and could unfairly violate/validate a property
  - relaxes (some of the) assumptions made in timed automata, like the infinite precision of the clocks
  - related works include implementability issues, robust semantics, *etc.*

Motivation

The goal

Define a (meaningful) measure on runs of timed automata that will tell us how likely properties are satisfied.

In the running example: "how likely will I visit Paris?"
"how long should I expect be waiting in Paris?"

- A relaxed semantics for timed automata
  - removes behaviours that are unlikely to happen and could unfairly violate/validate a property
  - relaxes (some of the) assumptions made in timed automata, like the infinite precision of the clocks
  - related works include implementability issues, robust semantics, *etc.*
- A new timed and probabilistic model
  - models a purely probabilistic environment
  - related models include continuous-time Markov chains, and probabilistic timed automata

Methodology

Rough idea

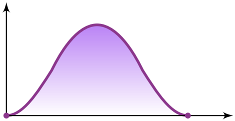
Build a stochastic process based on a timed automaton by randomizing all possible evolutions of the system.

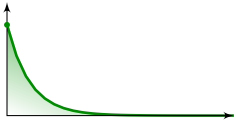
Methodology

Rough idea

Build a stochastic process based on a timed automaton by randomizing all possible evolutions of the system.

- We put (continuous) distributions over delays...





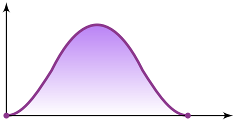
- ... and discrete distributions over transitions.

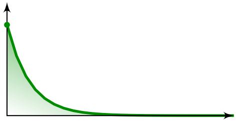
Methodology

Rough idea

Build a stochastic process based on a timed automaton by randomizing all possible evolutions of the system.

- We put (continuous) distributions over delays...





- ... and discrete distributions over transitions.

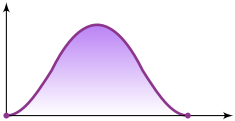
  $\rightsquigarrow$ this naturally defines a probability measure $\mathbb{P}$ over sets of runs.

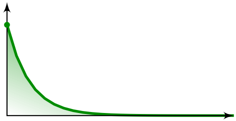
Methodology

Rough idea

Build a stochastic process based on a timed automaton by randomizing all possible evolutions of the system.

- We put (continuous) distributions over delays...





- ... and discrete distributions over transitions.

  $\rightsquigarrow$ this naturally defines a probability measure $\mathbb{P}$ over sets of runs.

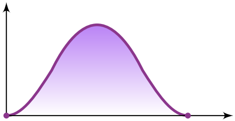
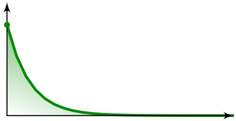
$\mathbb{P}(\mathcal{A} \models \varphi)$ measures "how likely $\mathcal{A}$ satisfies φ"

Methodology

Rough idea

Build a stochastic process based on a timed automaton by randomizing all possible evolutions of the system.

- We put (continuous) distributions over delays...

- ... and discrete distributions over transitions.

  $\rightsquigarrow$ this naturally defines a probability measure $\mathbb{P}$ over sets of runs.

$\mathbb{P}(\mathcal{A} \models \varphi)$ measures "how likely $\mathcal{A}$ satisfies φ"

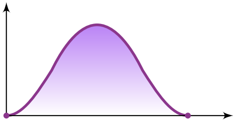
Two natural questions:

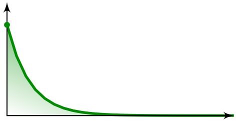
Methodology

Rough idea

Build a stochastic process based on a timed automaton by randomizing all possible evolutions of the system.

- We put (continuous) distributions over delays...





- ... and discrete distributions over transitions.

  $\rightsquigarrow$ this naturally defines a probability measure $\mathbb{P}$ over sets of runs.

$\mathbb{P}(\mathcal{A} \models \varphi)$ measures "how likely $\mathcal{A}$ satisfies φ"

Two natural questions:

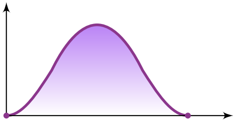
$\rightsquigarrow$ decide whether $\mathbb{P}(\mathcal{A} \models \varphi) = 1$ (qualitative question)

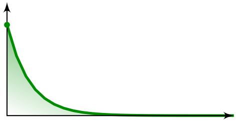
Methodology

Rough idea

Build a stochastic process based on a timed automaton by randomizing all possible evolutions of the system.

- We put (continuous) distributions over delays...





- ... and discrete distributions over transitions.

  $\rightsquigarrow$ this naturally defines a probability measure $\mathbb{P}$ over sets of runs.

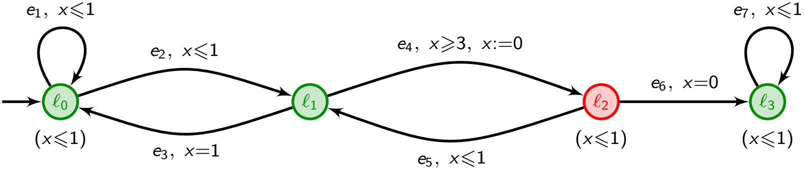
$\mathbb{P}(\mathcal{A} \models \varphi)$ measures "how likely $\mathcal{A}$ satisfies φ"

Two natural questions:

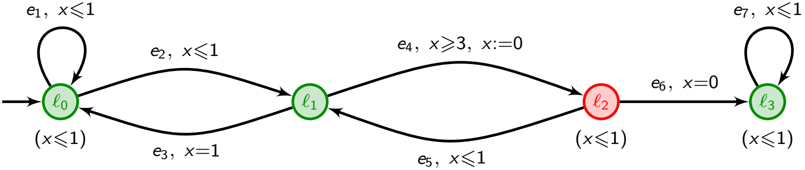
$\rightsquigarrow$ decide whether $\mathbb{P}(\mathcal{A} \models \varphi) = 1$ (qualitative question)

$\rightsquigarrow$ compute (an approximation of) $\mathbb{P}(\mathcal{A} \models \varphi)$ (quantitative question)

An example

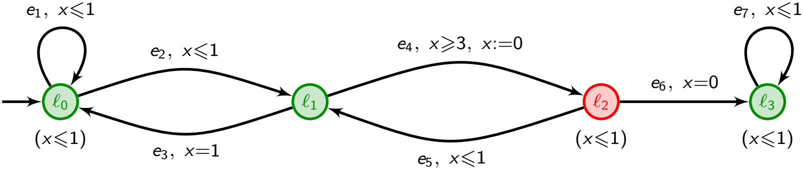

$e_1,\ x \leqslant 1$
$e_2,\ x \leqslant 1$
$e_4,\ x \geqslant 3,\ x{:=}0$
$e_7,\ x \leqslant 1$
$e_6,\ x=0$
ℓ_0
ℓ_1
ℓ_2
ℓ_3
$(x \leqslant 1)$
$(x \leqslant 1)$
$(x \leqslant 1)$
$e_3,\ x=1$
$e_5,\ x \leqslant 1$

An example



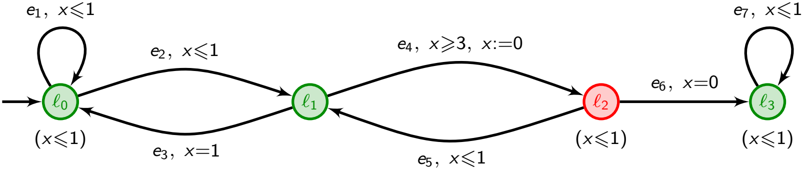
$\mathcal{A} \not\models \mathbf{G}\,(\textcolor{green}{\bullet} \Rightarrow \mathbf{F}\,\textcolor{red}{\bullet})$

An example



$\mathcal{A} \not\models \mathbf{G}\,(\bullet \Rightarrow \mathbf{F}\,\bullet)$ but $\mathbb{P}\big(\mathcal{A} \models \mathbf{G}\,(\bullet \Rightarrow \mathbf{F}\,\bullet)\big) = 1$

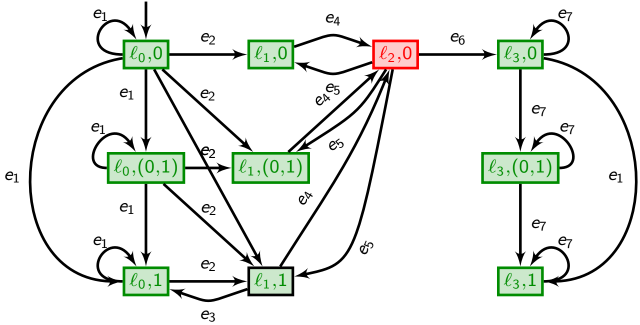
An example



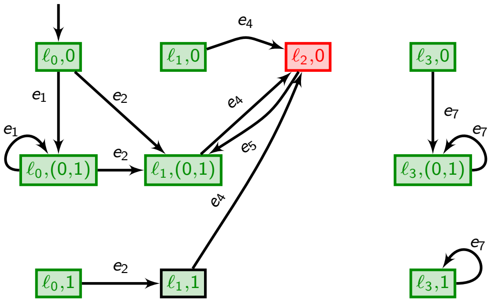
$$\mathcal{A} \not\models \mathbf{G}\,(\bullet \Rightarrow \mathbf{F}\,\bullet) \qquad \text{but} \qquad \mathbb{P}\big(\mathcal{A} \models \mathbf{G}\,(\bullet \Rightarrow \mathbf{F}\,\bullet)\big) = 1$$

Indeed, almost surely, paths are of the form $e_1^* e_2 \big(e_4 e_5\big)^\omega$

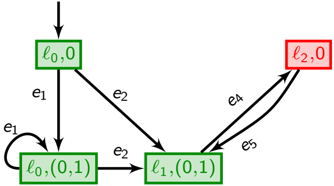
The classical region automaton



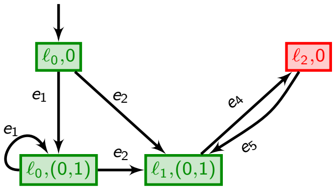
The pruned region automaton



The pruned region automaton

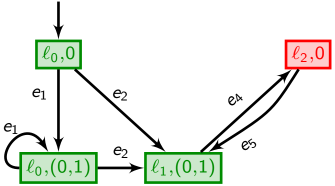


The pruned region automaton



... viewed as a finite Markov chain $MC(\mathcal{A})$

The pruned region automaton



... viewed as a finite Markov chain $MC(\mathcal{A})$

Proposition

For single-clock timed automata,

$$\mathbb{P}(\mathcal{A} \models \varphi) = 1 \quad \text{iff} \quad \mathbb{P}(MC(\mathcal{A}) \models \varphi) = 1$$

(this is independent of the choice of the distributions...)

Probabilistic model-checking

Theorem [BBBBG08]

For single-clock timed automata, the almost-sure model-checking

- of LTL is PSPACE-complete;
- of ω-regular properties is NLOGSPACE-complete;
- of the non-zenoness property is in NLOGSPACE.

[BBBBG08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[BBBM08] Bertrand, Bouyer, Brihaye, Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics *(QEST'08)*.

Probabilistic model-checking

Theorem [BBBBG08]

For single-clock timed automata, the almost-sure model-checking

- of LTL is PSPACE-complete;
- of ω-regular properties is NLOGSPACE-complete;
- of the non-zenoness property is in NLOGSPACE.

Theorem [BBBM08]

In single-clock timed automata, we can compute (an approximation of) the probability of satisfying an LTL/ω-regular property (for exponential distributions).

[BBBBG08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[BBBM08] Bertrand, Bouyer, Brihaye, Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics *(QEST'08)*.

Probabilistic model-checking

Theorem [BBBBG08]

For single-clock timed automata, the almost-sure model-checking

- of LTL is PSPACE-complete;
- of ω-regular properties is NLOGSPACE-complete;
- of the non-zenoness property is in NLOGSPACE.

Theorem [BBBM08]

In single-clock timed automata, we can compute (an approximation of) the probability of satisfying an LTL/ω-regular property (for exponential distributions).

$\leadsto$ none of these results extend to two-clock timed automata...

[BBBBG08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[BBBM08] Bertrand, Bouyer, Brihaye, Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics *(QEST'08)*.

Perspectives

- Further study this timed and probabilistic model
  - timed automata with an arbitrary number of clocks
    (hint: restrict the distributions)
  - model-checking timed properties like those in MTL
    - measurability of MTL properties
    - can we get a better complexity than NPR?
  - performance analysis (expected time, mean waiting time, *etc.*)
    (this model, a $\frac{1}{2}$-player model, generalizes continuous-time Markov chains)

Perspectives

- Further study this timed and probabilistic model
  - timed automata with an arbitrary number of clocks
    (hint: restrict the distributions)
  - model-checking timed properties like those in MTL
    - measurability of MTL properties
    - can we get a better complexity than NPR?
  - performance analysis (expected time, mean waiting time, *etc.*)
    (this model, a $\frac{1}{2}$-player model, generalizes continuous-time Markov chains)

- Study the $1\frac{1}{2}$- and $2\frac{1}{2}$-player models
  (to model non-determinism and interaction)
  - *preliminary result:* quantitative model-checking of the $2\frac{1}{2}$-player model is undecidable

Perspectives

- Further study this timed and probabilistic model
  - timed automata with an arbitrary number of clocks
    (hint: restrict the distributions)
  - model-checking timed properties like those in MTL
    - measurability of MTL properties
    - can we get a better complexity than NPR?
  - performance analysis (expected time, mean waiting time, *etc.*)
    (this model, a $\frac{1}{2}$-player model, generalizes continuous-time Markov chains)
- Study the $1\frac{1}{2}$- and $2\frac{1}{2}$-player models
  (to model non-determinism and interaction)
  - *preliminary result:* quantitative model-checking of the $2\frac{1}{2}$-player model is undecidable
- Design an irrefutable example

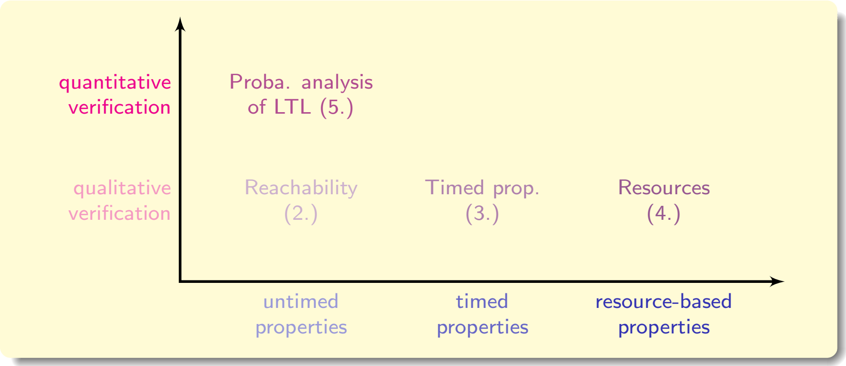
Outline

Summary

A progressive lift from qualitative to quantitative considerations:

Summary

A progressive lift from qualitative to quantitative considerations:



Perspectives

Modelling resources in timed systems

- study further the resource management problem
- develop approximation schemes
  (undecidability relies on the infinite precision of the system)
- compute equilibria

Perspectives

Modelling resources in timed systems

- study further the resource management problem
- develop approximation schemes
  (undecidability relies on the infinite precision of the system)
- compute equilibria

Probabilities and timed automata

- investigate further the $\frac{1}{2}$-player model
  - model with several clocks
    (*hint:* restrict the allowed distributions)
  - quantitative properties (time and resources)
  - performance analysis: expected time, *etc.*
- add non-determinism and interaction ($1\frac{1}{2}$- and $2\frac{1}{2}$-player models)
  (preliminary results: undecidability rather far away...)
- design an irrefutable example

Perspectives

Adequation of timed models to timed systems

Two main approaches: relaxed satisfaction
robust satisfaction

[BMR08] Bouyer, Markey, Reynier. Robust analysis of timed automata *via* channel machines *(FoSSaCS'08)*.

Perspectives

Adequation of timed models to timed systems

Two main approaches: relaxed satisfaction
robust satisfaction

- Relaxed satisfaction: *cf* probabilities and timed automata

[BMR08] Bouyer, Markey, Reynier. Robust analysis of timed automata *via* channel machines *(FoSSaCS'08)*.

Perspectives

Adequation of timed models to timed systems

Two main approaches: relaxed satisfaction
robust satisfaction

- Relaxed satisfaction: *cf* probabilities and timed automata
- Robust satisfaction:
  - develop further the purely channel-machine approach of [BMR08]
  - synthesize systems that are robustly correct by construction
  - further think of other notions of robustness

[BMR08] Bouyer, Markey, Reynier. Robust analysis of timed automata *via* channel machines *(FoSSaCS'08)*.