From timed to complex systems

— Stochastic timed games —

Patricia Bouyer-Decitre

LSV, CNRS & ENS Cachan, France

Based on joint works with Christel Baier, Nathalie Bertrand, Thomas Brihaye, Vojtěch Forejt, Marcus Größer and Nicolas Markey.

I am grateful to Vojtěch Forejt for some of the slides in this presentation.

Outline

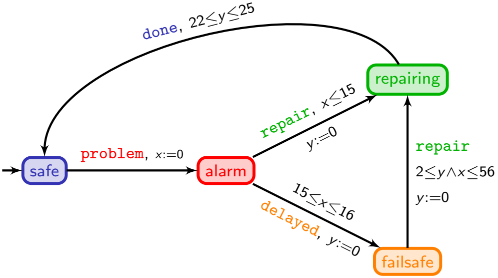
1. Timed automata

2. Timed games

3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games

4. Conclusion

An example of a timed automaton



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

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

Verification

Emptiness problem

Is the language accepted by a timed automaton empty?

- basic reachability/safety properties (final states)
- basic liveness properties (ω-regular conditions)

Verification

Emptiness problem

Is the language accepted by a timed automaton empty?

- Problem: the set of configurations is infinite
  $\leadsto$ classical methods for finite-state systems cannot be applied

Verification

Emptiness problem

Is the language accepted by a timed automaton empty?

- Problem: the set of configurations is infinite
  $\rightsquigarrow$ classical methods for finite-state systems cannot be applied
- Positive key point: variables (clocks) increase at the same speed

Verification

Emptiness problem

Is the language accepted by a timed automaton empty?

- Problem: the set of configurations is infinite
  $\leadsto$ classical methods for finite-state systems cannot be applied
- Positive key point: variables (clocks) increase at the same speed

Theorem [AD90,AD94]

The emptiness problem for timed automata is decidable and PSPACE-complete.

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

Verification

Emptiness problem

Is the language accepted by a timed automaton empty?

- Problem: the set of configurations is infinite
  $\rightsquigarrow$ classical methods for finite-state systems cannot be applied
- Positive key point: variables (clocks) increase at the same speed

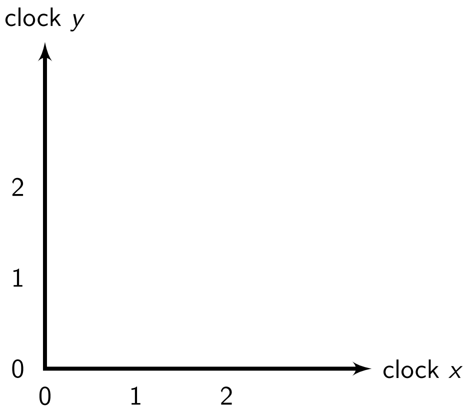
Theorem [AD90,AD94]

The emptiness problem for timed automata is decidable and PSPACE-complete.

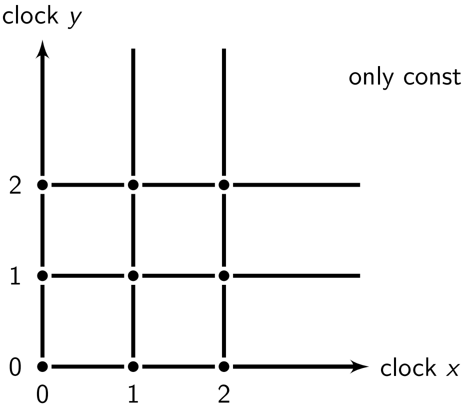
Method: construct a finite abstraction

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

The region abstraction



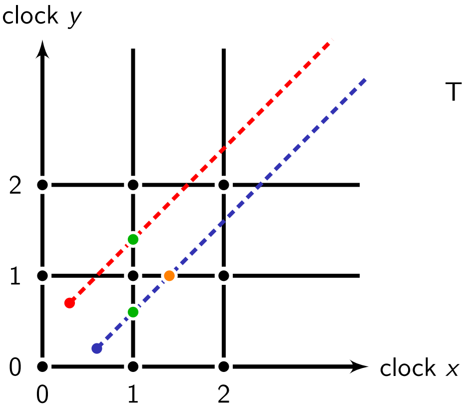
The region abstraction



only constraints: $x \sim c$ with $c \in \{0, 1, 2\}$
$y \sim c$ with $c \in \{0, 1, 2\}$

- "compatibility" between regions and constraints

The region abstraction

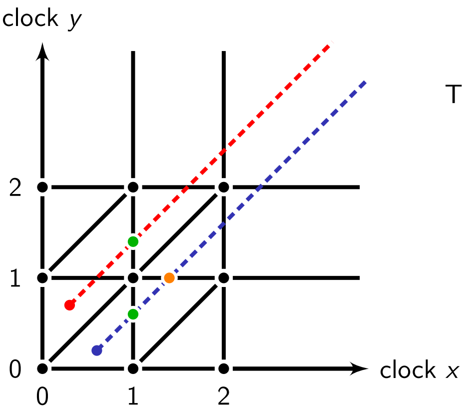


The path 

- can be fired from ●
- cannot be fired from ●

- “compatibility” between regions and constraints
- “compatibility” between regions and time elapsing

The region abstraction

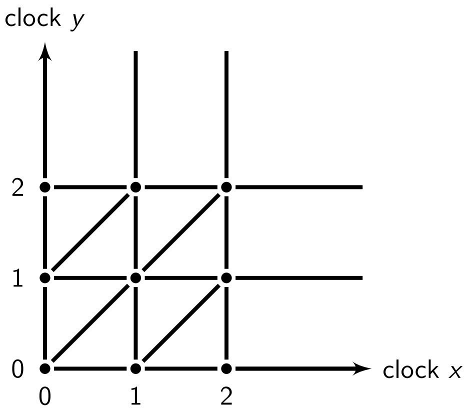


The path 

- can be fired from
- cannot be fired from

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

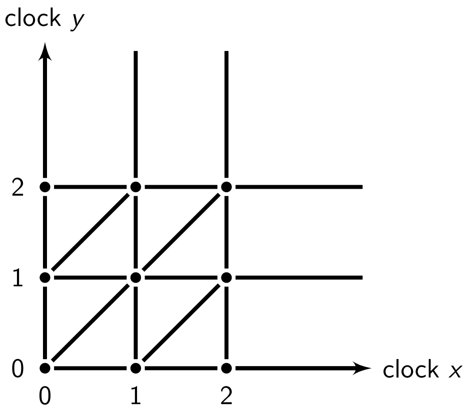
The region abstraction



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

$\rightsquigarrow$ an equivalence of finite index

The region abstraction



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

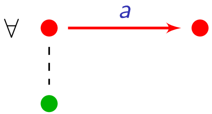
$\rightsquigarrow$ an equivalence of finite index
a time-abstract bisimulation

Time-abstract bisimulation

This is a relation between • and • such that:

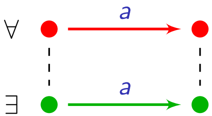
Time-abstract bisimulation

This is a relation between • and • such that:



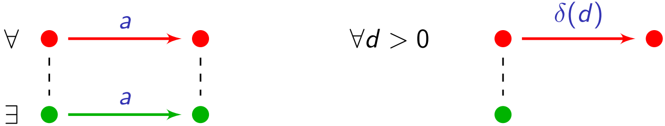
Time-abstract bisimulation

This is a relation between • and • such that:



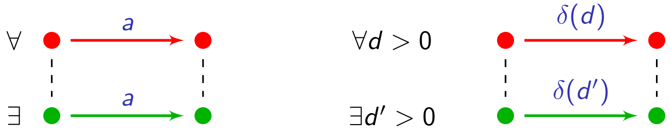
Time-abstract bisimulation

This is a relation between • and • such that:



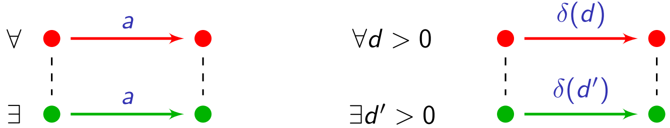
Time-abstract bisimulation

This is a relation between • and • such that:



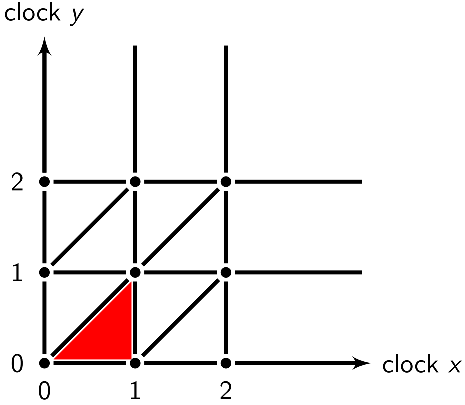
Time-abstract bisimulation

This is a relation between • and • such that:



... and vice-versa (swap • and •).

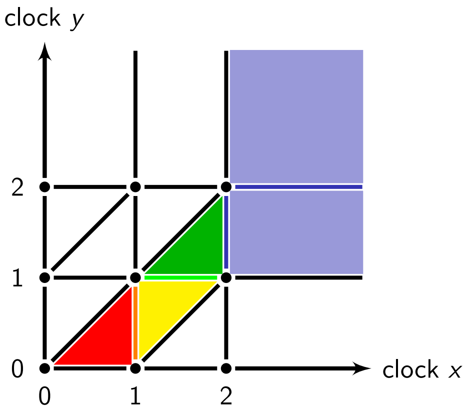
The region abstraction (2)



- region R defined by:

$$\begin{cases} 0 < x < 1 \\ 0 < y < 1 \\ y < x \end{cases}$$

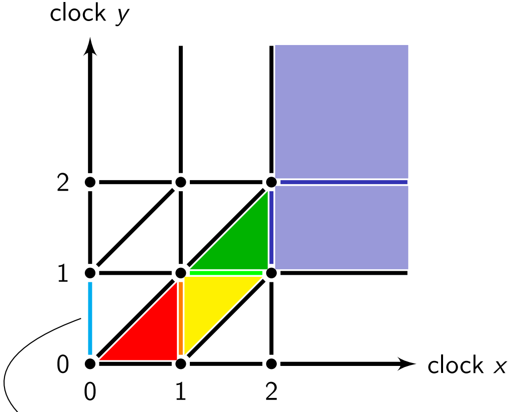
The region abstraction (2)



- region R defined by:
$$\begin{cases} 0 < x < 1 \\ 0 < y < 1 \\ y < x \end{cases}$$

- time successors of R

The region abstraction (2)



- region R defined by:

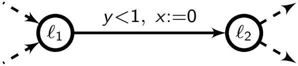
$$\begin{cases} 0 < x < 1 \\ 0 < y < 1 \\ y < x \end{cases}$$

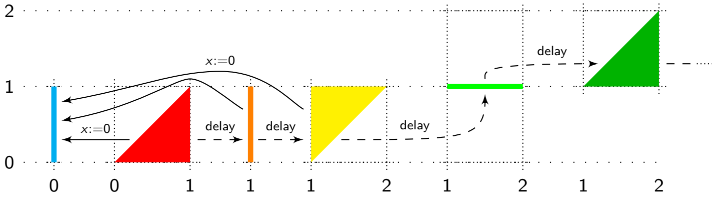
- time successors of R

image of R when resetting clock x

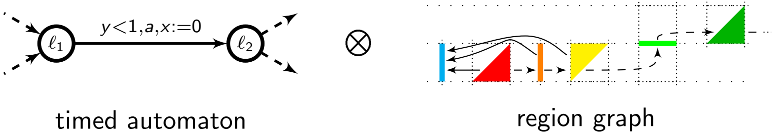
The construction of the region graph

It "mimics" the behaviours of the clocks.

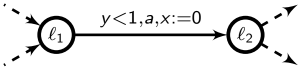




Region automaton ≡ finite bisimulation quotient

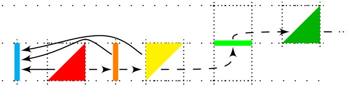


Region automaton ≡ finite bisimulation quotient

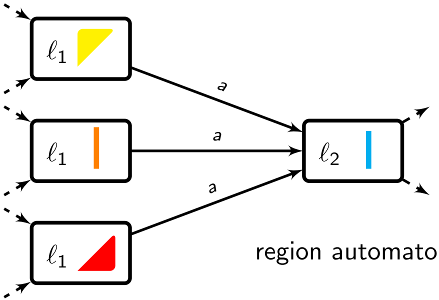


timed automaton

$\otimes$

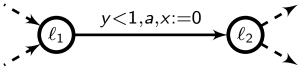


region graph



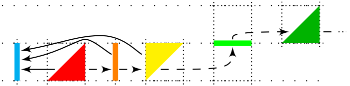
region automaton

Region automaton $\equiv$ finite bisimulation quotient

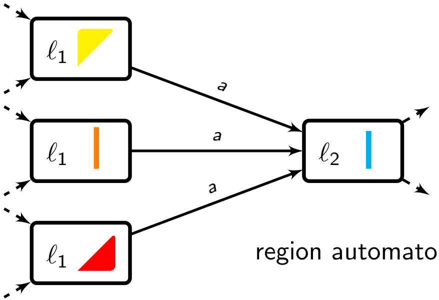


timed automaton

$\otimes$



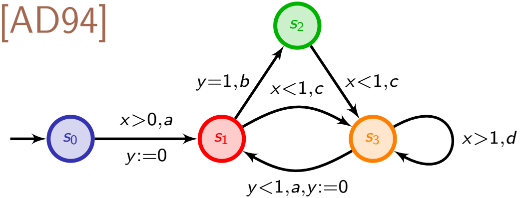
region graph



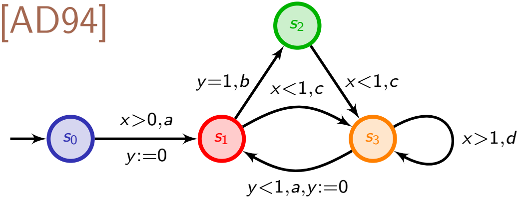
region automaton

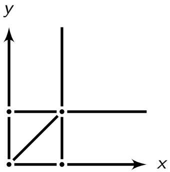
$\mathcal{L}$(reg. aut.) $=$ UNTIME($\mathcal{L}$(timed aut.))

An example [AD94]

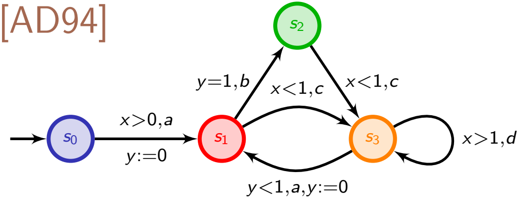


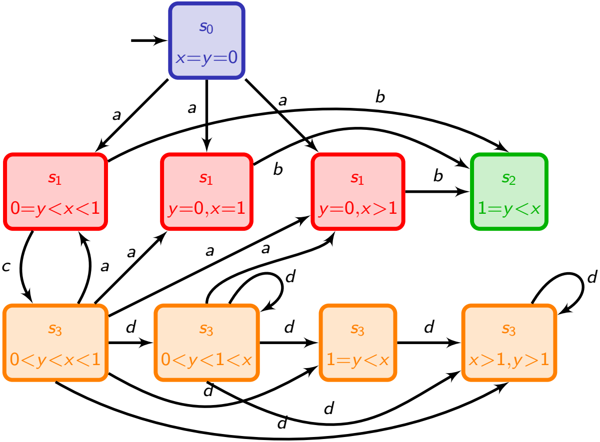
An example [AD94]

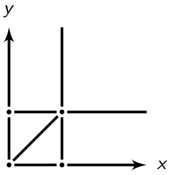


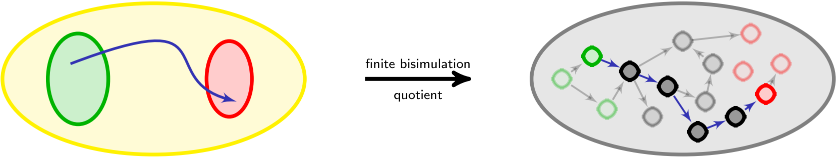


An example [AD94]



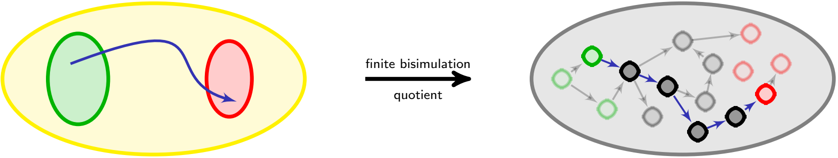






timed automaton

large (but finite) automaton
(region automaton)

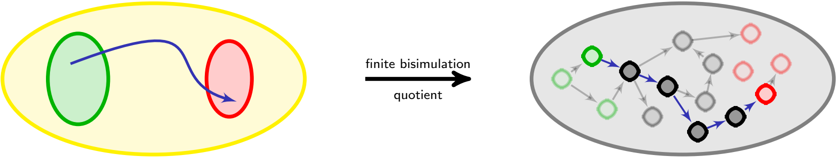


timed automaton

large (but finite) automaton
(region automaton)

- large: exponential in the number of clocks and in the constants (if encoded in binary). The number of regions is:

$$\prod_{x \in X} (2M_x + 2) \cdot |X|! \cdot 2^{|X|}$$



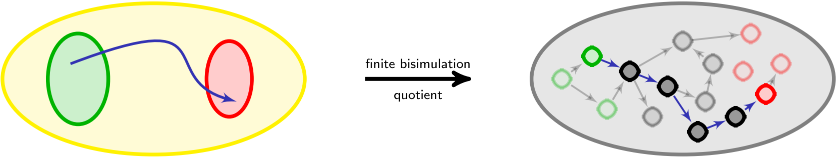
timed automaton

large (but finite) automaton
(region automaton)

- large: exponential in the number of clocks and in the constants (if encoded in binary). The number of regions is:

$$\prod_{x \in X} (2M_x + 2) \cdot |X|! \cdot 2^{|X|}$$

- It can be used to check for:



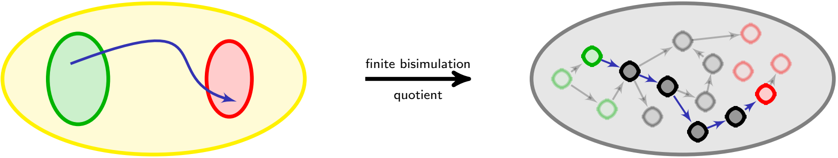
timed automaton

large (but finite) automaton
(region automaton)

- large: exponential in the number of clocks and in the constants (if encoded in binary). The number of regions is:

$$\prod_{x \in X} (2M_x + 2) \cdot |X|! \cdot 2^{|X|}$$

- It can be used to check for:
  - reachability/safety properties



timed automaton

large (but finite) automaton
(region automaton)

- large: exponential in the number of clocks and in the constants (if encoded in binary). The number of regions is:

$$\prod_{x \in X} (2M_x + 2) \cdot |X|! \cdot 2^{|X|}$$

- It can be used to check for:
  - reachability/safety properties
  - liveness properties (like Büchi properties)

Outline

1. Timed automata

2. Timed games

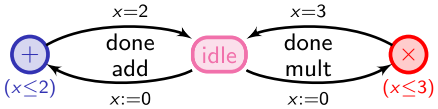
3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games

4. Conclusion

Why (timed) games?

- to model uncertainty

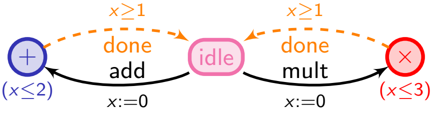
Example of a processor in the taskgraph example



Why (timed) games?

- to model uncertainty

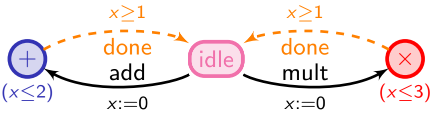
Example of a processor in the taskgraph example



Why (timed) games?

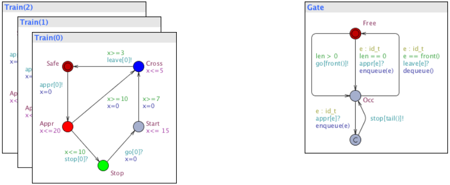
- to model uncertainty

Example of a processor in the taskgraph example



- to model an interaction with an environment

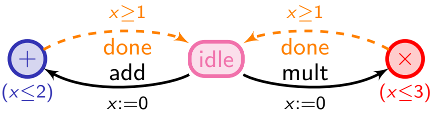
Example of the gate in the train/gate example



Why (timed) games?

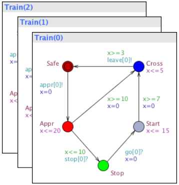
- to model uncertainty

Example of a processor in the taskgraph example



- to model an interaction with an environment

Example of the gate in the train/gate example

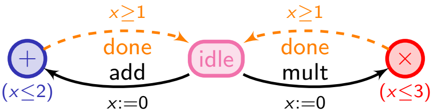


?

Why (timed) games?

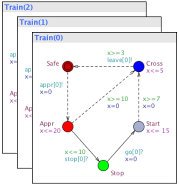
- to model uncertainty

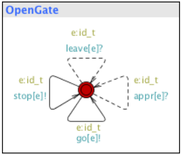
Example of a processor in the taskgraph example



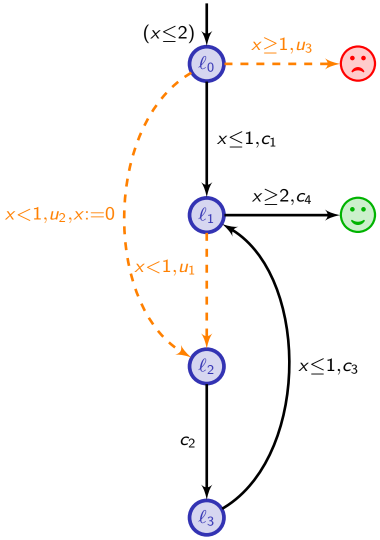
- to model an interaction with an environment

Example of the gate in the train/gate example





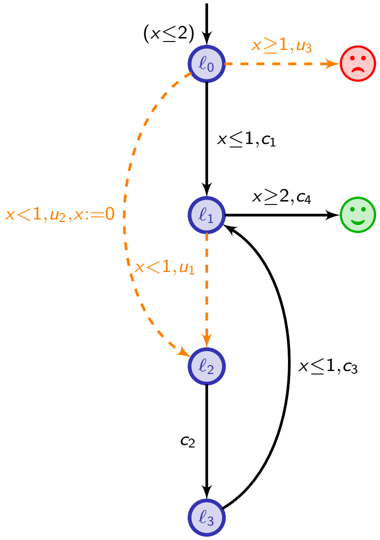
An example of a timed game



Rule of the game

- Aim: avoid ☹ and reach ☺

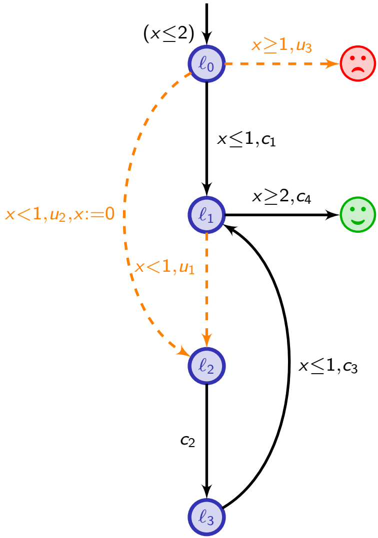
An example of a timed game



Rule of the game

- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

An example of a timed game

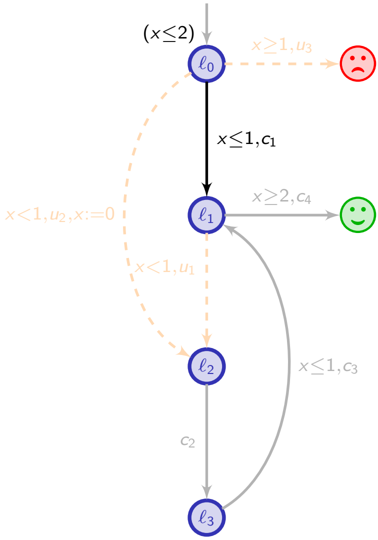


Rule of the game

- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

An example of a timed game



Rule of the game

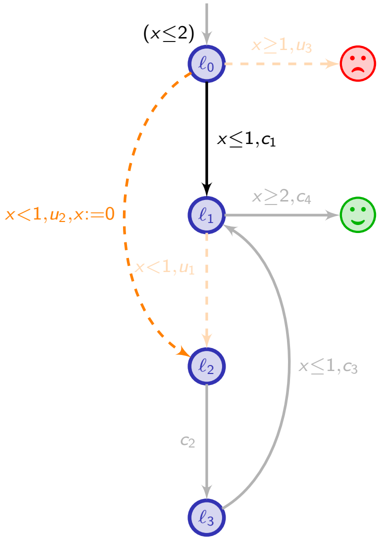
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$

An example of a timed game



Rule of the game

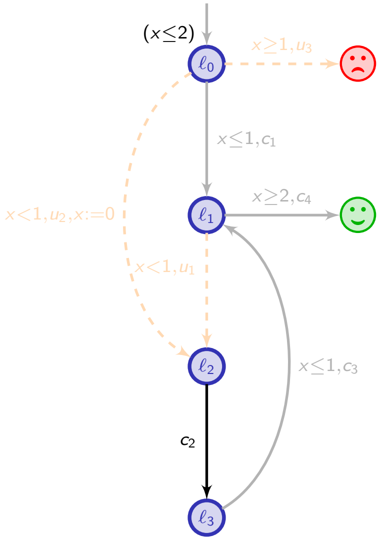
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$
  $\rightsquigarrow$ can be preempted by u_2

An example of a timed game



Rule of the game

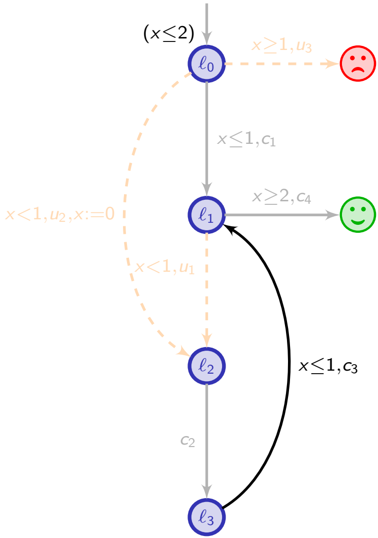
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$
  $\rightsquigarrow$ can be preempted by u_2
- from $(\ell_2, \star)$, play $(1 - \star, c_2)$

An example of a timed game



Rule of the game

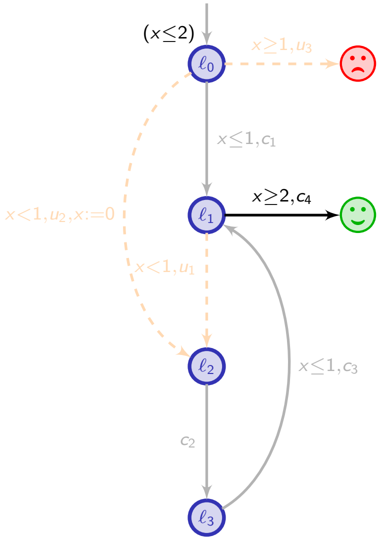
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$
  $\rightsquigarrow$ can be preempted by u_2
- from $(\ell_2, \star)$, play $(1 - \star, c_2)$
- from $(\ell_3, 1)$, play $(0, c_3)$

An example of a timed game



Rule of the game

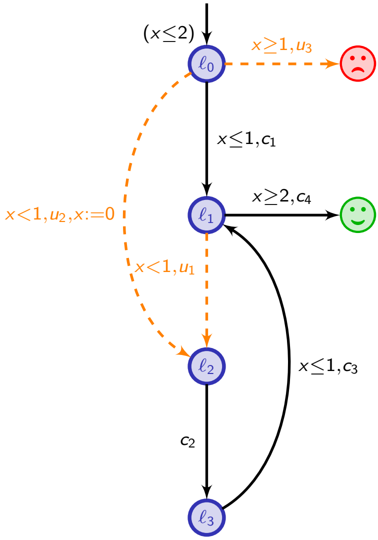
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$
  $\rightsquigarrow$ can be preempted by u_2
- from $(\ell_2, \star)$, play $(1 - \star, c_2)$
- from $(\ell_3, 1)$, play $(0, c_3)$
- from $(\ell_1, 1)$, play $(1, c_4)$

An example of a timed game



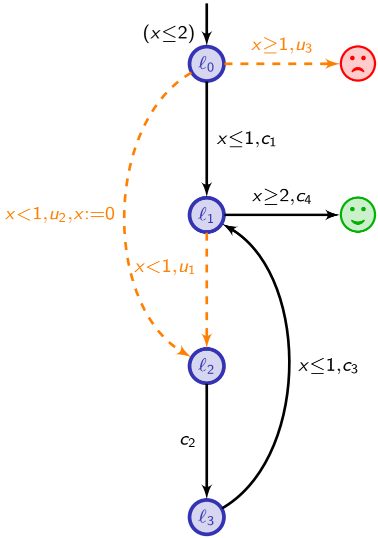
Rule of the game

- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

Problems to be considered

An example of a timed game



Rule of the game

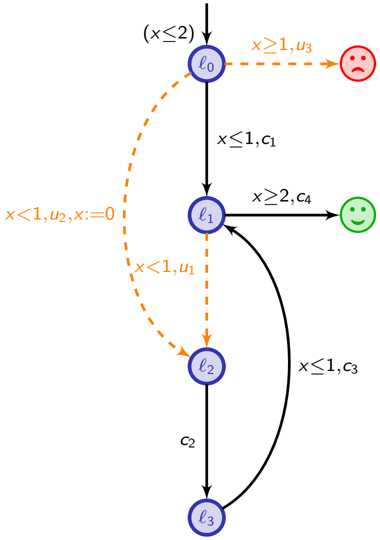
- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

Problems to be considered

- Does there exist a winning strategy?

An example of a timed game



Rule of the game

- Aim: avoid ☹ and reach ☺
- How do we play? According to a strategy:

$$f : \text{history} \mapsto (\text{delay}, \text{cont. transition})$$

Problems to be considered

- Does there exist a winning strategy?
- If yes, compute one (as simple as possible).

Decidability of timed games

Theorem [AMPS98,HK99]

Reachability and safety timed games are decidable and EXPTIME-complete. Furthermore memoryless and "region-based" strategies are sufficient.

[AMPS98] Asarin, Maler, Pnueli, Sifakis. Controller synthesis for timed automata *(HSSC'98)*.
[HK99] Henzinger, Kopke. Discrete-time control for rectangular hybrid automata *(Theoretical Computer Science)*.

Decidability of timed games

Theorem [AMPS98,HK99]

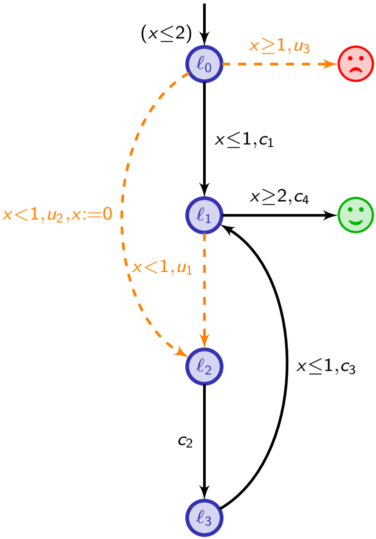
Reachability and safety timed games are decidable and EXPTIME-complete. Furthermore memoryless and "region-based" strategies are sufficient.

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

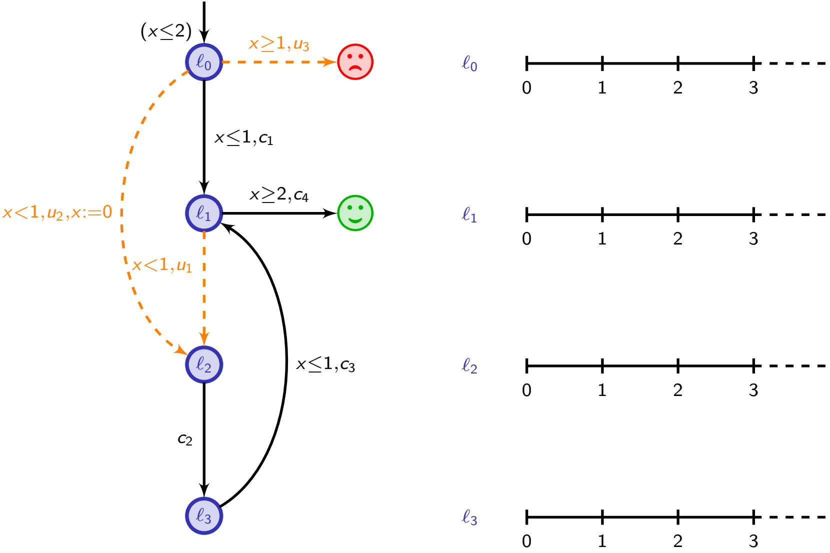
(one only needs to compute the so-called attractor)

[AMPS98] Asarin, Maler, Pnueli, Sifakis. Controller synthesis for timed automata *(HSSC'98)*.
[HK99] Henzinger, Kopke. Discrete-time control for rectangular hybrid automata *(Theoretical Computer Science)*.

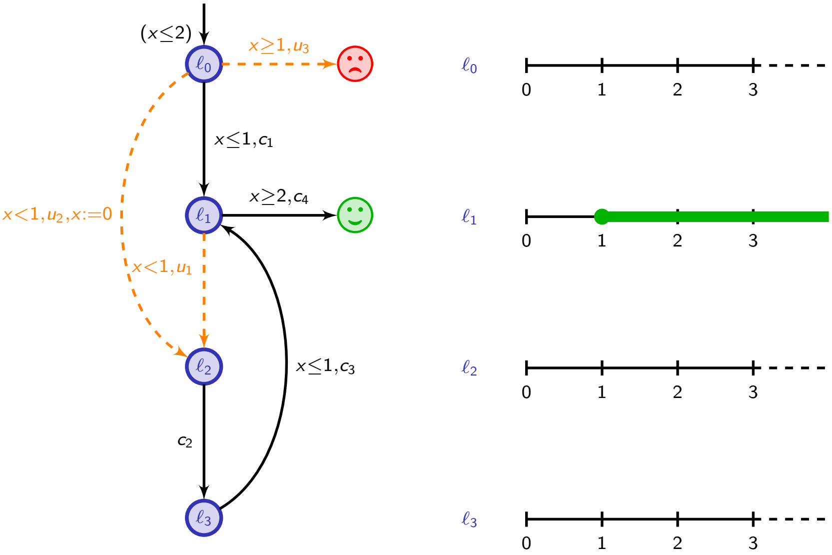
Back to the example: computing winning states



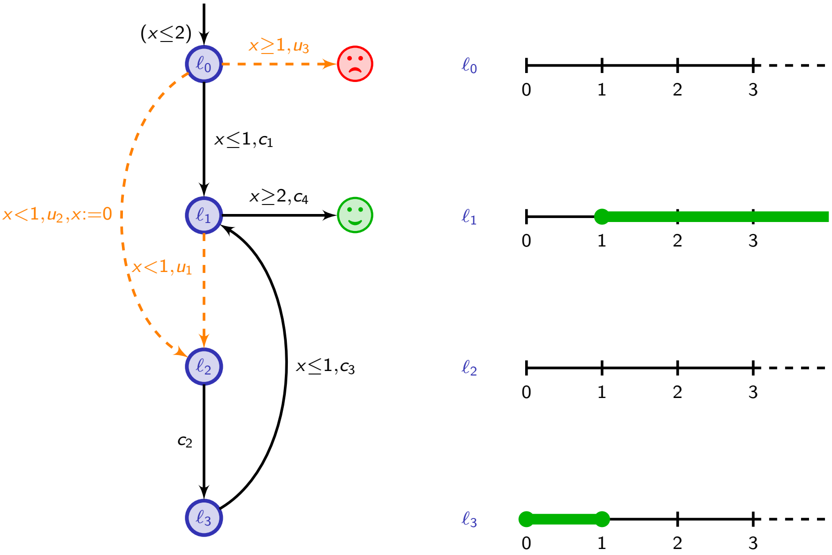
Back to the example: computing winning states



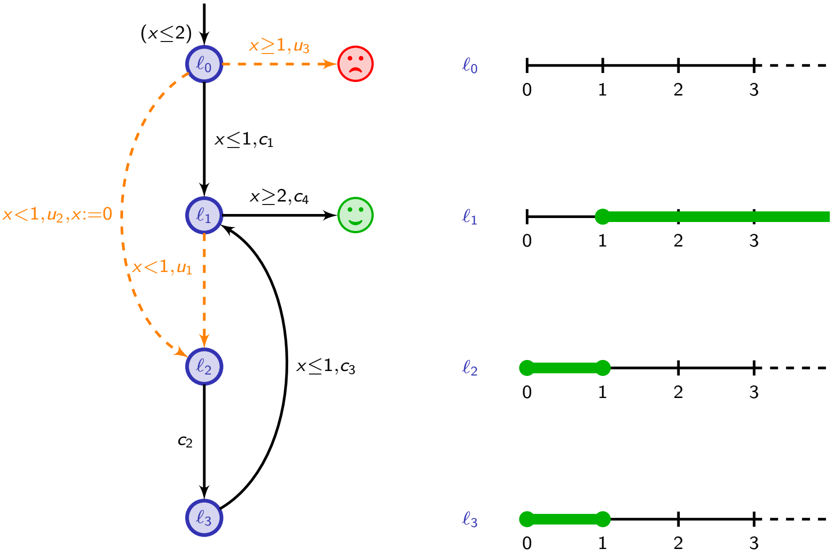
Back to the example: computing winning states



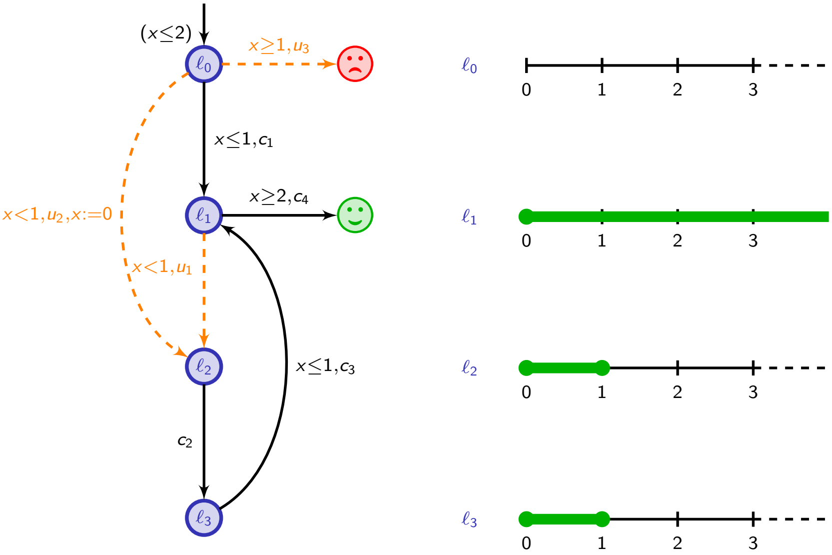
Back to the example: computing winning states



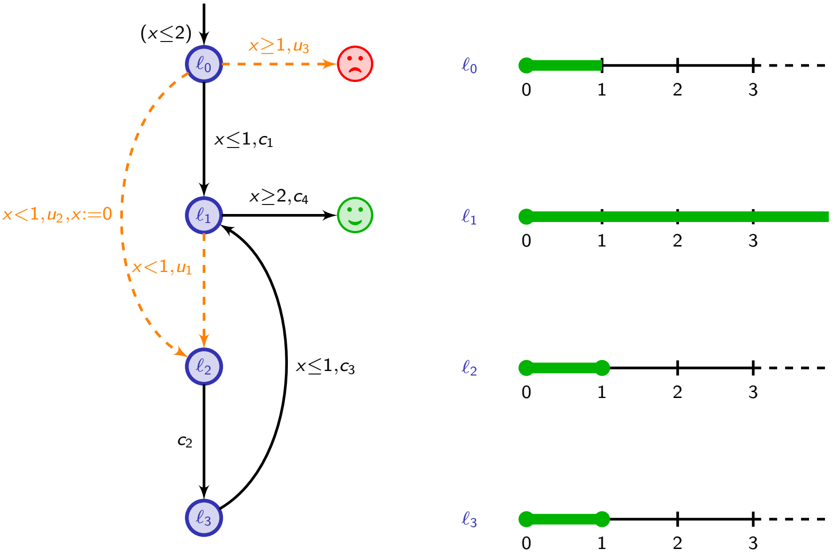
Back to the example: computing winning states



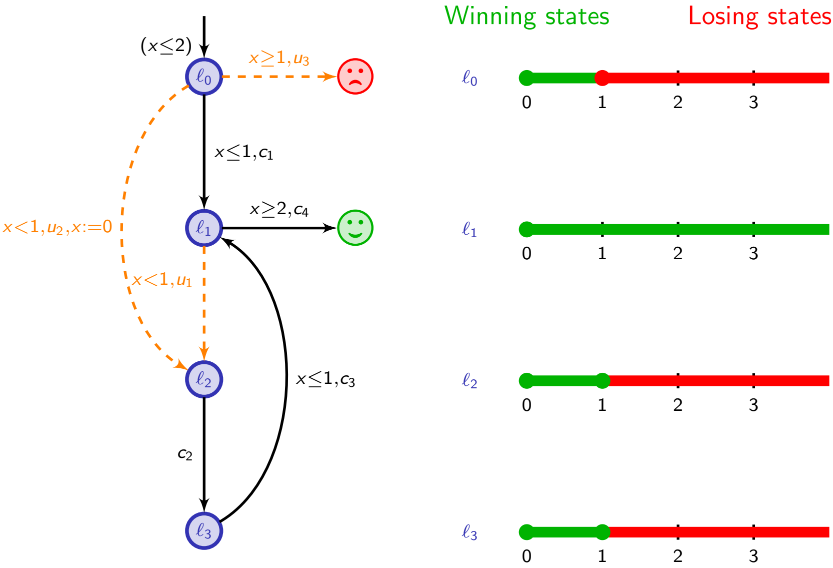
Back to the example: computing winning states



Back to the example: computing winning states



Back to the example: computing winning states



Outline

1. Timed automata

2. Timed games

3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games

4. Conclusion

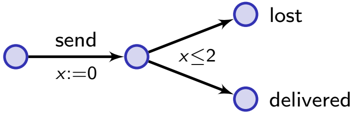
Why add stochastic features? And how?

- to model probabilistic behaviours

Why add stochastic features? And how?

- to model probabilistic behaviours

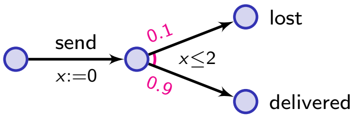
Example of losses when sending messages



Why add stochastic features? And how?

- to model probabilistic behaviours

Example of losses when sending messages



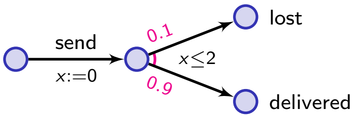
$\rightsquigarrow$ the probabilistic timed automata model used e.g. in PRISM and UPPAAL-PRO [KNSS02]

[KNSS02] Automatic verification of real-time systems with discrete probability distributions *(TCS)*.

Why add stochastic features? And how?

- to model probabilistic behaviours

Example of losses when sending messages



$\rightsquigarrow$ the probabilistic timed automata model used e.g. in PRISM and UPPAAL-PRO [KNSS02]

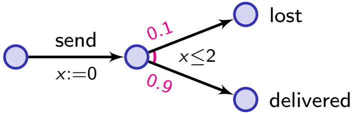
- to model uncertainty on delays

[KNSS02] Automatic verification of real-time systems with discrete probability distributions *(TCS)*.

Why add stochastic features? And how?

- to model probabilistic behaviours

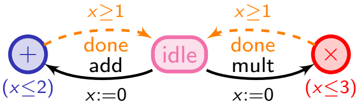
Example of losses when sending messages



$\rightsquigarrow$ the probabilistic timed automata model used e.g. in PRISM and UPPAAL-PRO [KNSS02]

- to model uncertainty on delays

Example of a processor in the taskgraph example

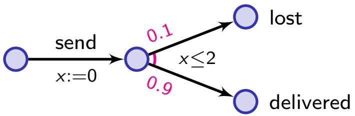


[KNSS02] Automatic verification of real-time systems with discrete probability distributions *(TCS)*.

Why add stochastic features? And how?

- to model probabilistic behaviours

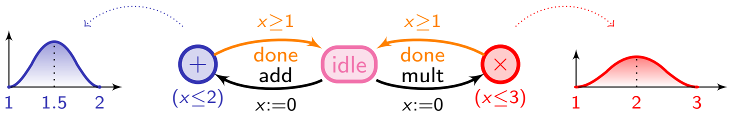
Example of losses when sending messages



$\rightsquigarrow$ the probabilistic timed automata model used e.g. in PRISM and UPPAAL-PRO [KNSS02]

- to model uncertainty on delays

Example of a processor in the taskgraph example



$\rightsquigarrow$ the stochastic timed automata model [BBB+08,BF09]

[KNSS02] Automatic verification of real-time systems with discrete probability distributions *(TCS)*.
[BBB+08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[BF09] Bouyer, Forejt. Reachability in stochastic timed games *(ICALP'09)*.

Outline

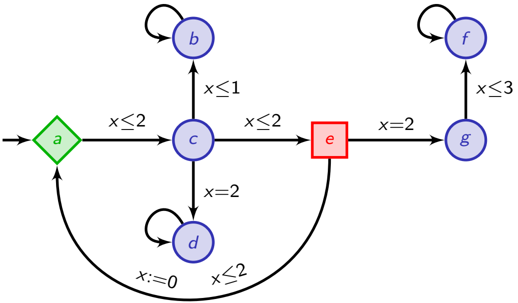
1. Timed automata

2. Timed games

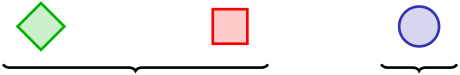
3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games

4. Conclusion

Stochastic timed game: an example



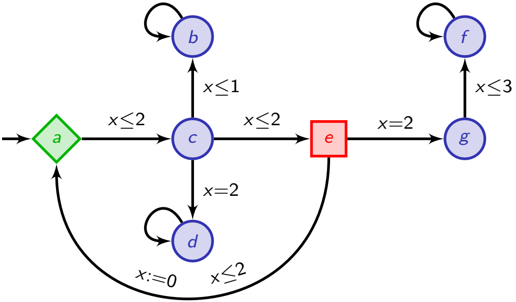
- Timed graph with vertices partitioned among three players:



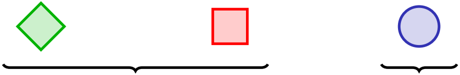
classical players playing "turn-based"

the Nature stochastic player

Stochastic timed game: an example



- Timed graph with vertices partitioned among three players:

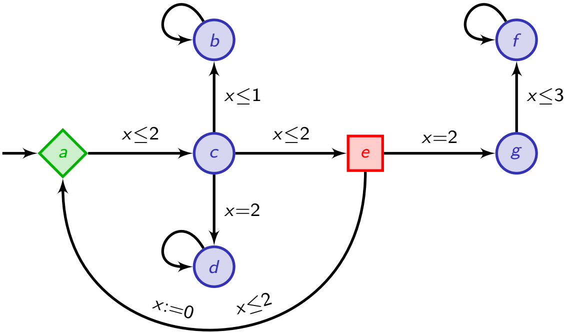


classical players
playing "turn-based"

the Nature
stochastic player

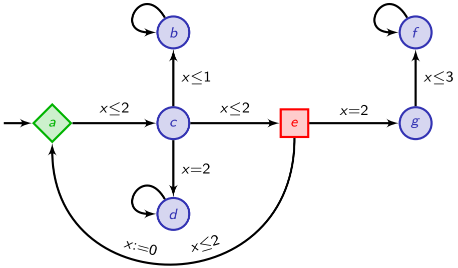
- There are prescribed probability distributions from vertices.

How is this game played?

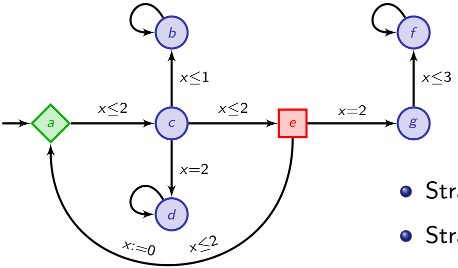


- Players ◇ and □ play according to standard strategies
- Player ○ plays according to the prescribed probability distributions:
  - choose a delay according to some distribution
  - choose an action according to some discrete distribution

Play, an example

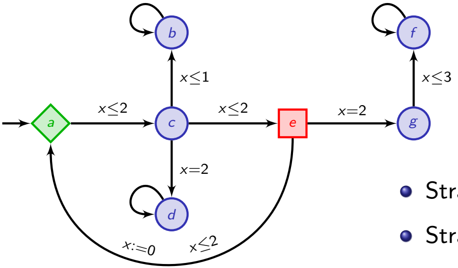


Play, an example



- Strategy for ◇: go to c when $x = 1$
- Strategy for □ : go to g when $x = 2$

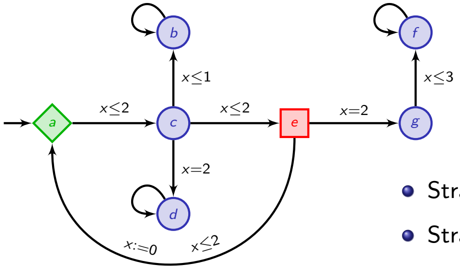
Play, an example



- Strategy for ◇: go to c when $x = 1$
- Strategy for □ : go to g when $x = 2$

- From the game and the strategies we obtain a Markov chain:

Play, an example

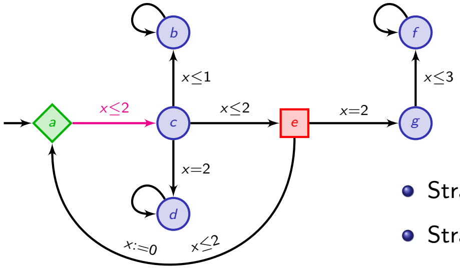


- Strategy for $\Diamond$: go to c when $x = 1$
- Strategy for $\square$: go to g when $x = 2$

- From the game and the strategies we obtain a Markov chain:

$(a,0)$

Play, an example

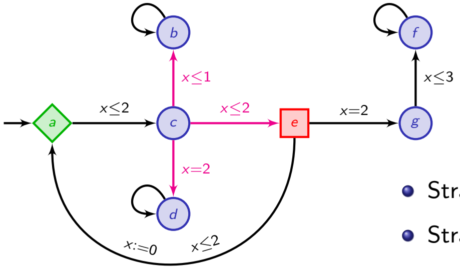


- Strategy for : go to c when $x = 1$
- Strategy for : go to g when $x = 2$

- From the game and the strategies we obtain a Markov chain:

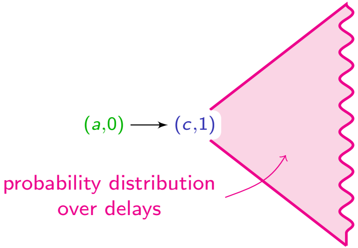
$$(a,0) \longrightarrow (c,1)$$

Play, an example

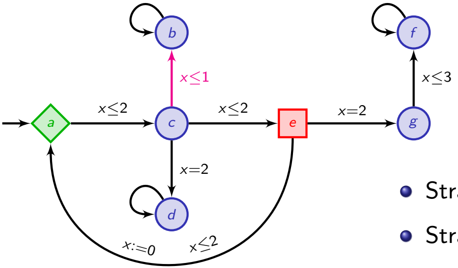


- Strategy for : go to c when $x = 1$
- Strategy for : go to g when $x = 2$

- From the game and the strategies we obtain a Markov chain:

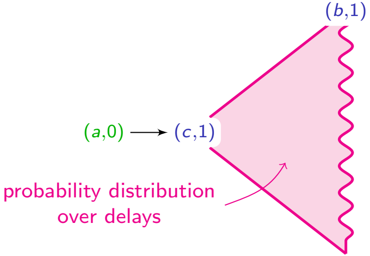


Play, an example

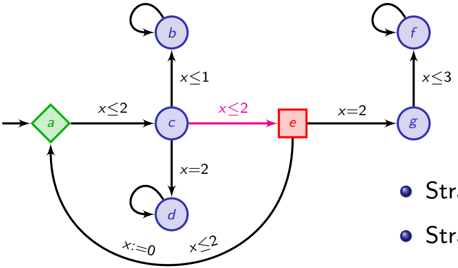


- Strategy for : go to c when $x = 1$
- Strategy for : go to g when $x = 2$

- From the game and the strategies we obtain a Markov chain:

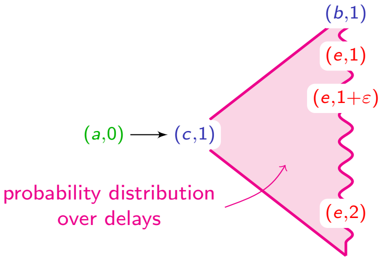


Play, an example

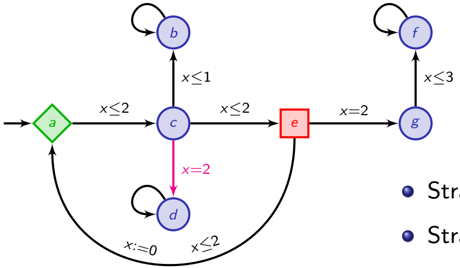


- Strategy for ◇: go to c when $x = 1$
- Strategy for □ : go to g when $x = 2$

- From the game and the strategies we obtain a Markov chain:

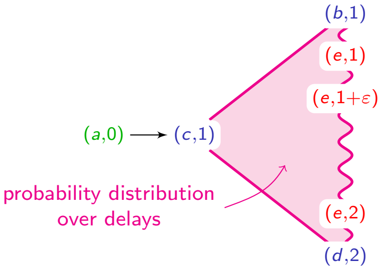


Play, an example

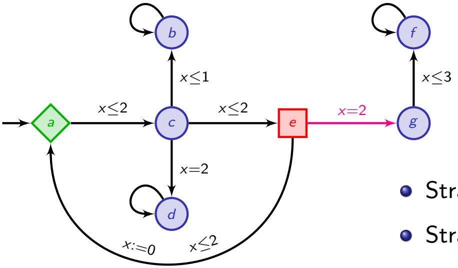


- Strategy for ◇: go to c when $x = 1$
- Strategy for □ : go to g when $x = 2$

- From the game and the strategies we obtain a Markov chain:

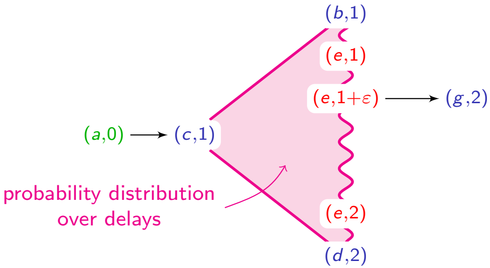


Play, an example

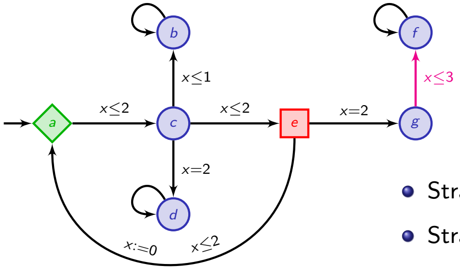


- Strategy for ◇: go to c when $x = 1$
- Strategy for □ : go to g when $x = 2$

- From the game and the strategies we obtain a Markov chain:

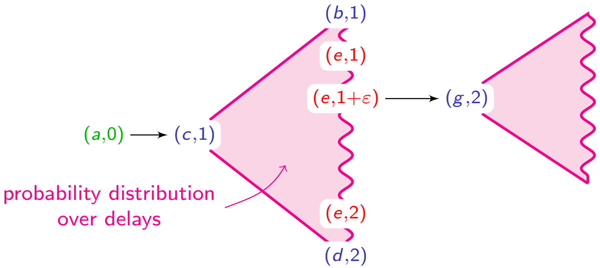


Play, an example

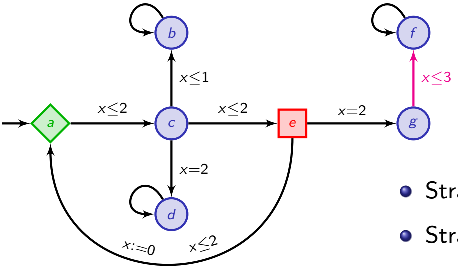


- Strategy for ◇: go to c when $x = 1$
- Strategy for □ : go to g when $x = 2$

- From the game and the strategies we obtain a Markov chain:

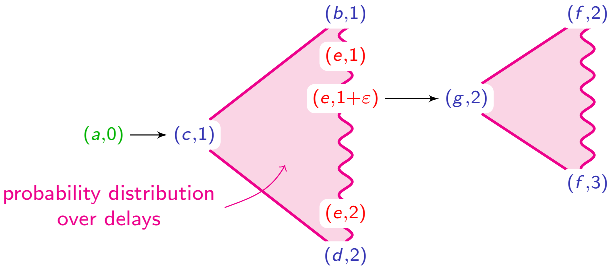


Play, an example



- Strategy for ◇: go to c when $x=1$
- Strategy for □ : go to g when $x=2$

- From the game and the strategies we obtain a Markov chain:



Outline

1. Timed automata

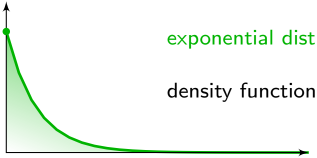
2. Timed games

3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games

4. Conclusion

How can we attach probabilities to delays?

- The example of continuous-time Markov chains

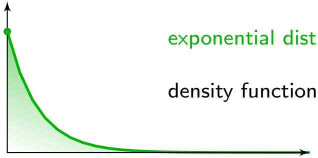


exponential distribution

density function $t \mapsto \begin{cases} \lambda \cdot \exp(-\lambda t) & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$

How can we attach probabilities to delays?

- The example of continuous-time Markov chains



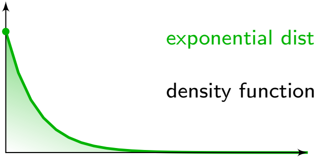
exponential distribution

density function $t \mapsto \begin{cases} \lambda \cdot \exp(-\lambda t) & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$

$\rightsquigarrow$ this is ok if delays are in $[0, +\infty)$

How can we attach probabilities to delays?

- The example of continuous-time Markov chains



exponential distribution

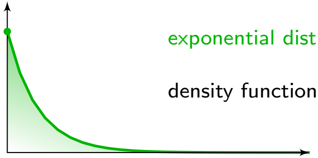
density function $t \mapsto \begin{cases} \lambda \cdot \exp(-\lambda t) & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$

$\leadsto$ this is ok if delays are in $[0, +\infty)$

- But what if bounded intervals?

How can we attach probabilities to delays?

- The example of continuous-time Markov chains

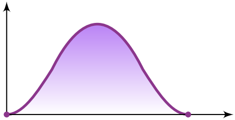


exponential distribution

density function $t \mapsto \begin{cases} \lambda \cdot \exp(-\lambda t) & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$

$\rightsquigarrow$ this is ok if delays are in $[0, +\infty)$

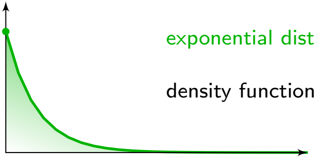
- But what if bounded intervals?



truncated normal distribution

How can we attach probabilities to delays?

- The example of continuous-time Markov chains

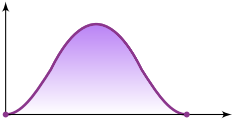


exponential distribution

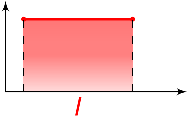
density function $t \mapsto \begin{cases} \lambda \cdot \exp(-\lambda t) & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$

$\rightsquigarrow$ this is ok if delays are in $[0, +\infty)$

- But what if bounded intervals?



truncated normal distribution



uniform distribution
density function $t \mapsto \begin{cases} \frac{1}{|I|} & \text{if } t \geq 0 \\ 0 & \text{otherwise} \end{cases}$

How does the semantics formalize?

- We will explain it more formally when all vertices belong to player $\bigcirc$. Those are called $\frac{1}{2}$-player games.

How does the semantics formalize?

- We will explain it more formally when all vertices belong to player ◯. Those are called $\frac{1}{2}$-player games.

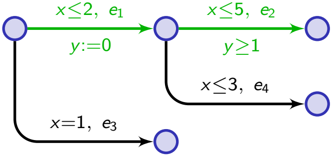
- We will then extend it using standard strategies for the two other players, which need however satisfy some measurability assumption

The $\frac{1}{2}$-player game model

- $\pi(s \xrightarrow{e_1} \dots \xrightarrow{e_n})$: symbolic path from s firing edges $e_1, \dots, e_n$

The $\frac{1}{2}$-player game model

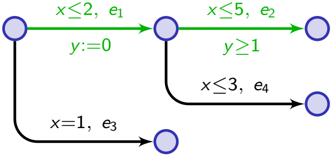
- $\pi(s \xrightarrow{e_1} \dots \xrightarrow{e_n})$: symbolic path from s firing edges $e_1, \dots, e_n$
- Example:



$$\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2}) = \{s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \mid \tau_1 \leq 2,\ \tau_1 + \tau_2 \leq 5,\ \tau_2 \geq 1\}$$

The $\frac{1}{2}$-player game model

- $\pi(s \xrightarrow{e_1} \dots \xrightarrow{e_n})$: symbolic path from s firing edges $e_1, \dots, e_n$
- Example:



$$\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2}) = \{s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \mid \tau_1 \leq 2,\ \tau_1 + \tau_2 \leq 5,\ \tau_2 \geq 1\}$$

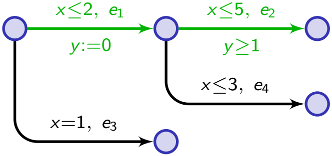
- Idea: compute the probability of a symbolic path

From state s:

s

The $\frac{1}{2}$-player game model

- $\pi(s \xrightarrow{e_1} \dots \xrightarrow{e_n})$: symbolic path from s firing edges $e_1, \dots, e_n$
- Example:

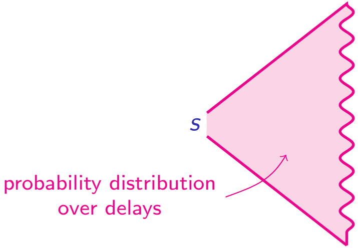


$$\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2}) = \{s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \mid \tau_1 \leq 2,\ \tau_1 + \tau_2 \leq 5,\ \tau_2 \geq 1\}$$

- Idea: compute the probability of a symbolic path

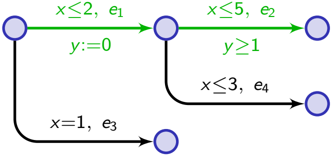
From state s:

- randomly choose a delay



The $\frac{1}{2}$-player game model

- $\pi(s \xrightarrow{e_1} \ldots \xrightarrow{e_n})$: symbolic path from s firing edges $e_1, \ldots, e_n$
- Example:

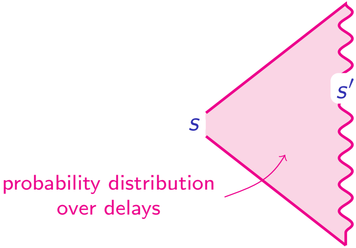


$$\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2}) = \{s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \mid \tau_1 \leq 2,\ \tau_1 + \tau_2 \leq 5,\ \tau_2 \geq 1\}$$

- Idea: compute the probability of a symbolic path

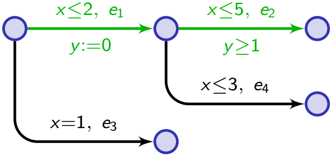
From state s:

- randomly choose a delay



The $\frac{1}{2}$-player game model

- $\pi(s \xrightarrow{e_1} \dots \xrightarrow{e_n})$: symbolic path from s firing edges $e_1, \dots, e_n$
- Example:

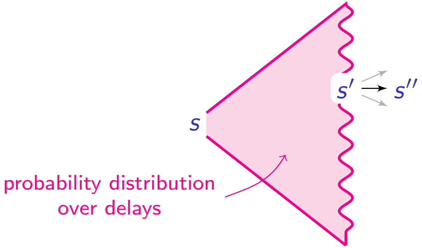


$$\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2}) = \{s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \mid \tau_1 \leq 2,\ \tau_1 + \tau_2 \leq 5,\ \tau_2 \geq 1\}$$

- Idea: compute the probability of a symbolic path

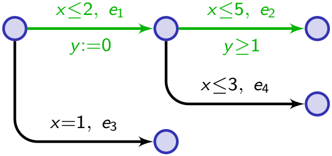
From state s:

- randomly choose a delay
- then randomly select an edge



The $\frac{1}{2}$-player game model

- $\pi(s \xrightarrow{e_1} \dots \xrightarrow{e_n})$: symbolic path from s firing edges $e_1, \dots, e_n$
- Example:

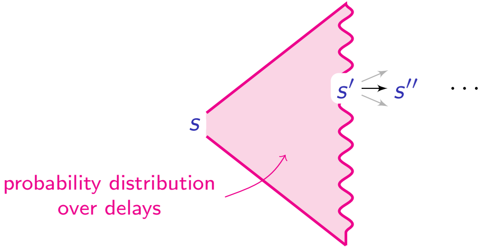


$$\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2}) = \{s_0 \xrightarrow{\tau_1, e_1} s_1 \xrightarrow{\tau_2, e_2} s_2 \mid \tau_1 \leq 2,\ \tau_1 + \tau_2 \leq 5,\ \tau_2 \geq 1\}$$

- Idea: compute the probability of a symbolic path

From state s:

- randomly choose a delay
- then randomly select an edge
- then continue



The $\frac{1}{2}$-player game model

symbolic path: $\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n\}$

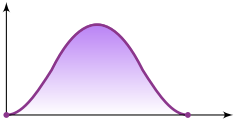
$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

The $\frac{1}{2}$-player game model

symbolic path: $\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n\}$

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

- $I(s,e_1) = \{\tau \mid s \xrightarrow{\tau, e_1}\}$ and μ_s distribution over $I(s) = \bigcup_e I(s,e)$

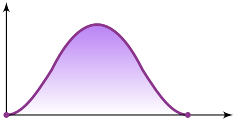


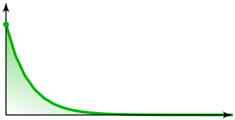
The $\frac{1}{2}$-player game model

symbolic path: $\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n\}$

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

- $I(s, e_1) = \{\tau \mid s \xrightarrow{\tau, e_1}\}$ and μ_s distribution over $I(s) = \bigcup_e I(s, e)$





The $\frac{1}{2}$-player game model

symbolic path: $\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n\}$

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

- $I(s, e_1) = \{\tau \mid s \xrightarrow{\tau, e_1}\}$ and μ_s distribution over $I(s) = \bigcup_e I(s, e)$
- p_{s+t} distribution over transitions enabled in $s + t$ (given by weights on transitions)

The $\frac{1}{2}$-player game model

symbolic path: $\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n\}$

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

- $I(s, e_1) = \{\tau \mid s \xrightarrow{\tau, e_1}\}$ and μ_s distribution over $I(s) = \bigcup_e I(s, e)$
- p_{s+t} distribution over transitions enabled in $s + t$ (given by weights on transitions)
- $s \xrightarrow{t} s + t \xrightarrow{e_1} s_t$

The $\frac{1}{2}$-player game model

$$\mathbb{P}\Big(\pi\big(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\big)\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi\big(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n}\big)\Big)\, \mathrm{d}\mu_s(t)$$

The $\frac{1}{2}$-player game model

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

- Can be viewed as an *n*-dimensional integral

The $\frac{1}{2}$-player game model

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

- Can be viewed as an n-dimensional integral
- Easy extension to constrained symbolic paths

$$\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models \mathcal{C}\}$$

The $\frac{1}{2}$-player game model

$$\mathbb{P}\Big(\pi\big(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\big)\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi\big(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n}\big)\Big)\, \mathrm{d}\mu_s(t)$$

- Can be viewed as an n-dimensional integral
- Easy extension to constrained symbolic paths

$$\pi_{\mathcal{C}}\big(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\big) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models \mathcal{C}\}$$

- Definition over sets of infinite runs:

The $\frac{1}{2}$-player game model

$$\mathbb{P}\Big(\pi\big(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\big)\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi\big(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n}\big)\Big)\, \mathrm{d}\mu_s(t)$$

- Can be viewed as an n-dimensional integral
- Easy extension to constrained symbolic paths

$$\pi_{\mathcal{C}}\big(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\big) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models \mathcal{C}\}$$

- Definition over sets of infinite runs:
  - $\mathsf{Cyl}(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{\varrho \cdot \varrho' \mid \varrho \in \pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\}$

The $\frac{1}{2}$-player game model

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

- Can be viewed as an n-dimensional integral
- Easy extension to constrained symbolic paths

$$\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models \mathcal{C}\}$$

- Definition over sets of infinite runs:
  - $\mathsf{Cyl}(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{\varrho \cdot \varrho' \mid \varrho \in \pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\}$
  - $\mathbb{P}\Big(\mathsf{Cyl}(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))\Big) = \mathbb{P}\Big(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big)$

The $\frac{1}{2}$-player game model

$$\mathbb{P}\Big(\pi\big(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\big)\Big) = \int_{t\in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi\big(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n}\big)\Big)\, \mathrm{d}\mu_s(t)$$

- Can be viewed as an n-dimensional integral
- Easy extension to constrained symbolic paths

$$\pi_{\mathcal{C}}\big(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\big) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models \mathcal{C}\}$$

- Definition over sets of infinite runs:
  - $\mathsf{Cyl}(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{\varrho \cdot \varrho' \mid \varrho \in \pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\}$
  - $\mathbb{P}\Big(\mathsf{Cyl}(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))\Big) = \mathbb{P}\Big(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big)$
  - unique extension of $\mathbb{P}$ to the generated σ-algebra

The $\frac{1}{2}$-player game model

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

- Can be viewed as an n-dimensional integral
- Easy extension to constrained symbolic paths

$$\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models \mathcal{C}\}$$

- Definition over sets of infinite runs:
  - $\mathsf{Cyl}(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{\varrho \cdot \varrho' \mid \varrho \in \pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\}$
  - $\mathbb{P}\Big(\mathsf{Cyl}(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))\Big) = \mathbb{P}\Big(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big)$
  - unique extension of $\mathbb{P}$ to the generated σ-algebra
- Property: $\mathbb{P}$ is a probability measure over sets of infinite runs

The $\frac{1}{2}$-player game model

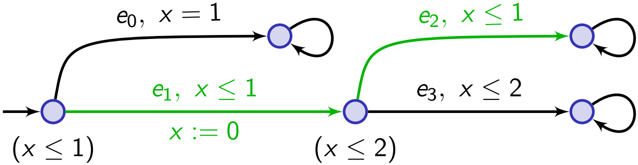
$$\mathbb{P}\Big(\pi\big(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\big)\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\,\mathbb{P}\Big(\pi\big(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n}\big)\Big)\,\mathrm{d}\mu_s(t)$$

- Can be viewed as an n-dimensional integral
- Easy extension to constrained symbolic paths

$$\pi_{\mathcal{C}}\big(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\big) = \{s \xrightarrow{\tau_1,e_1} s_1 \cdots \xrightarrow{\tau_n,e_n} s_n \mid (\tau_1,\cdots,\tau_n) \models \mathcal{C}\}$$

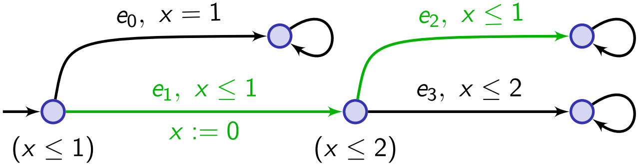
- Definition over sets of infinite runs:
  - $\mathsf{Cyl}(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{\varrho \cdot \varrho' \mid \varrho \in \pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\}$
  - $\mathbb{P}\big(\mathsf{Cyl}(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))\big) = \mathbb{P}\big(\pi_{\mathcal{C}}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\big)$
  - unique extension of $\mathbb{P}$ to the generated σ-algebra
- Property: $\mathbb{P}$ is a probability measure over sets of infinite runs
- Example:
  - $\mathsf{Zeno}(s) \;=\; \bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1,\cdots,e_n) \in E^n} \mathsf{Cyl}(\pi_{\Sigma_i \tau_i \leq M}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))$

An example of computation (with uniform distributions)



The probability of the symbolic path $\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2})$ is $\frac{1}{4}$.

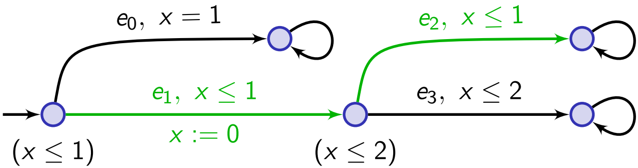
An example of computation (with uniform distributions)



The probability of the symbolic path $\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2})$ is $\frac{1}{4}$.

$$\mathbb{P}\Big(\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2})\Big) = \int_0^1 \mathbb{P}\Big(\pi(s_1 \xrightarrow{e_2})\Big)\mathrm{d}\mu_{s_0}(t) + \int_1^1 \frac{\mathbb{P}\Big(\pi(s_1 \xrightarrow{e_2})\Big)}{2}\mathrm{d}\mu_{s_0}(t)$$

An example of computation (with uniform distributions)

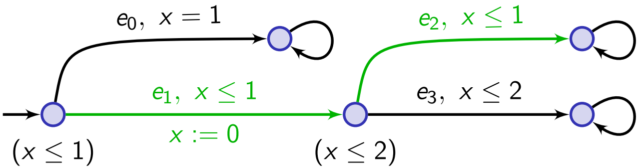


The probability of the symbolic path $\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2})$ is $\frac{1}{4}$.

$$\mathbb{P}\Big(\pi\big(s_0 \xrightarrow{e_1}\xrightarrow{e_2}\big)\Big) = \int_0^1 \mathbb{P}\Big(\pi\big(s_1 \xrightarrow{e_2}\big)\Big)\mathrm{d}\mu_{s_0}(t) + \int_1^1 \frac{\mathbb{P}\Big(\pi\big(s_1 \xrightarrow{e_2}\big)\Big)}{2}\mathrm{d}\mu_{s_0}(t)$$

$$= \int_0^1 \int_0^1 \left(\frac{\mathbb{P}\Big(\pi\big(s_2\big)\Big)}{2}\mathrm{d}\mu_{s_1}(u)\right) \mathrm{d}\mu_{s_0}(t)$$

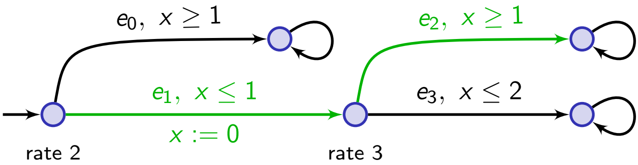
An example of computation (with uniform distributions)



The probability of the symbolic path $\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2})$ is $\frac{1}{4}$.

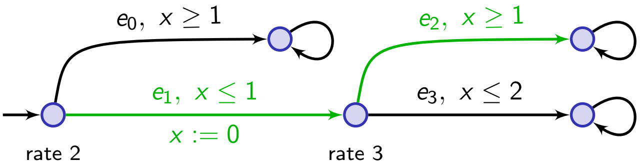
$$
\begin{aligned}
\mathbb{P}\Big(\pi\big(s_0 \xrightarrow{e_1}\xrightarrow{e_2}\big)\Big) &= \int_0^1 \mathbb{P}\Big(\pi\big(s_1 \xrightarrow{e_2}\big)\Big)\mathrm{d}\mu_{s_0}(t) + \int_1^1 \frac{\mathbb{P}\Big(\pi\big(s_1 \xrightarrow{e_2}\big)\Big)}{2}\mathrm{d}\mu_{s_0}(t) \\
&= \int_0^1 \int_0^1 \left(\frac{\mathbb{P}\big(\pi(s_2)\big)}{2}\mathrm{d}\mu_{s_1}(u)\right) \mathrm{d}\mu_{s_0}(t) \\
&= \int_0^1 \int_0^1 \left(\frac{1}{2}\,\frac{\mathrm{d}u}{2}\right) \mathrm{d}t \quad = \frac{1}{4}
\end{aligned}
$$

An example of computation (with exponential distrib.)



The probability of the symbolic path $\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2})$ is $e^{-3} - e^{-5} \approx 0.043$

An example of computation (with exponential distrib.)



The probability of the symbolic path $\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2})$ is $e^{-3} - e^{-5} \approx 0.043$

$$\begin{aligned}
\mathbb{P}\Big(\pi(s_0 \xrightarrow{e_1}\xrightarrow{e_2}\ )\Big) &= \int_0^1 \mathbb{P}\Big(\pi(s_1 \xrightarrow{e_2}\ )\Big)\mathrm{d}\mu_{s_0}(t) = \int_0^1 \mathbb{P}\Big(\pi(s_1 \xrightarrow{e_2}\ )\Big) 2\exp(-2t)\mathrm{d}t \\
&= \int_0^1 \left(\int_1^{+\infty} 3\exp(-3u)\mathrm{d}u\right) 2\exp(-2t)\mathrm{d}t \\
&= [-\exp(-2t)]_{t=0}^1 \cdot [-\exp(-3u)]_{u=1}^{+\infty} \\
&= (1 - e^{-2}) \cdot e^{-3} = e^{-3} - e^{-5}
\end{aligned}$$

Some remarks

- This defines a purely stochastic process ($\frac{1}{2}$-player game).

Some remarks

- This defines a purely stochastic process ($\frac{1}{2}$-player game).
- Continuous-time Markov chains $=$ timed automata with a single "useless" clock which is reset on all transitions. The distributions on delays are exponential distributions with a rate per location.

Some remarks

- This defines a purely stochastic process ($\frac{1}{2}$-player game).
- Continuous-time Markov chains = timed automata with a single "useless" clock which is reset on all transitions. The distributions on delays are exponential distributions with a rate per location.
- The semantics can be extended in a natural way to several players:

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

mass distribution given by the strategy
if s is a player $\Diamond$ vertex

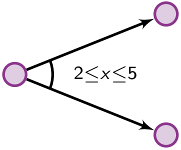
Some remarks

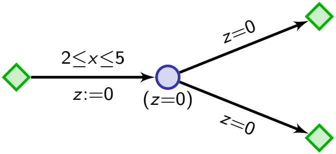
- This defines a purely stochastic process ($\frac{1}{2}$-player game).
- Continuous-time Markov chains = timed automata with a single "useless" clock which is reset on all transitions. The distributions on delays are exponential distributions with a rate per location.
- The semantics can be extended in a natural way to several players:

$$\mathbb{P}\Big(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\Big) = \int_{t \in I(s,e_1)} p_{s+t}(e_1)\, \mathbb{P}\Big(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\Big)\, \mathrm{d}\mu_s(t)$$

mass distribution given by the strategy
if s is a player ◇ vertex

- Probabilistic timed automata = a subclass of the $1\frac{1}{2}$-player games





The synthesis problem

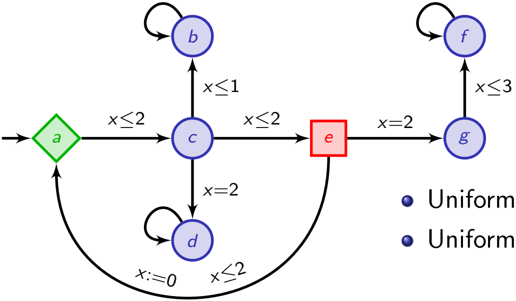
Problem statement

Given a game G, a (linear-time) property φ, a rational threshold $\bowtie r$,

is there a strategy $f_{\diamond}$ for player $\diamond$ s.t.

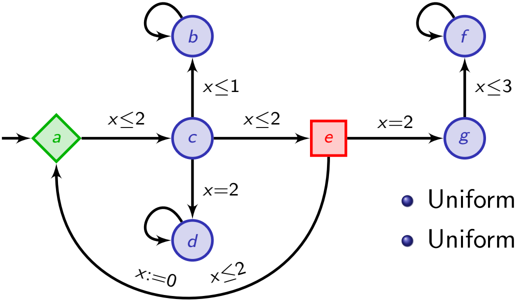
for all strategies $f_{\square}$ of player $\square$, $\mathbb{P}\Big(G_{f_{\diamond},f_{\square}} \models \varphi\Big) \bowtie r$?

Reachability problem – Example



- Uniform distribution over delays
- Uniform distribution over edges

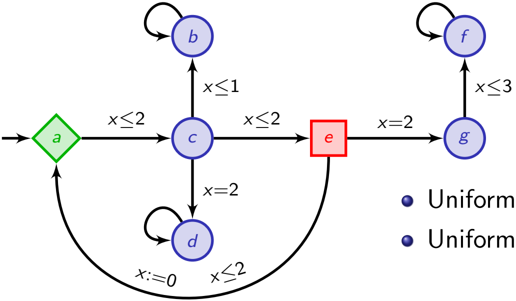
Reachability problem – Example



- Uniform distribution over delays
- Uniform distribution over edges

- Are vertices $\{b, f\}$ reachable with probability 1 from $(a, 0)$?

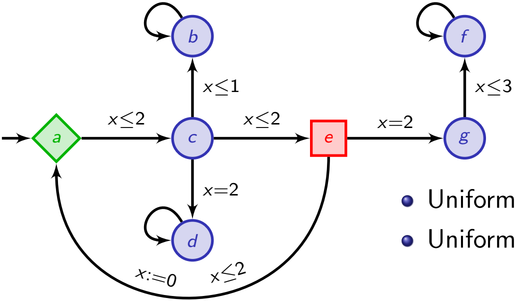
Reachability problem – Example



- Uniform distribution over delays
- Uniform distribution over edges

- Are vertices $\{b, f\}$ reachable with probability 1 from $(a, 0)$?
  - Yes: it is the case when  always chooses to move when $x = 0.5$.

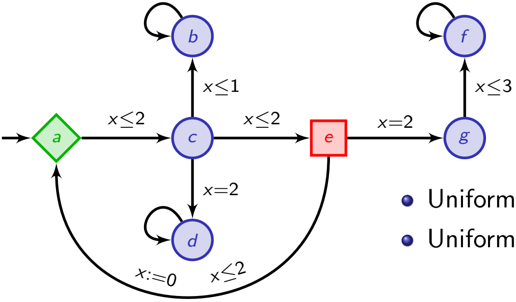
Reachability problem – Example



- Uniform distribution over delays
- Uniform distribution over edges

- Are vertices $\{b, f\}$ reachable with probability 1 from $(a, 0)$?
  - Yes: it is the case when  always chooses to move when $x = 0.5$.
- Is the vertex b reachable with probability at least $\frac{2}{3}$?

Reachability problem – Example



- Uniform distribution over delays
- Uniform distribution over edges

- Are vertices $\{b, f\}$ reachable with probability 1 from $(a, 0)$?
  - Yes: it is the case when  always chooses to move when $x = 0.5$.
- Is the vertex b reachable with probability at least $\frac{2}{3}$?
  - No.

What kind of games will we play?

Number of players

- $2\frac{1}{2}$-player games:
- $1\frac{1}{2}$-player games: ("Markov decision process")
- $\frac{1}{2}$-player games: ("Markov chain")

What kind of games will we play?

Number of players

- $2\frac{1}{2}$-player games:
- $1\frac{1}{2}$-player games: ("Markov decision process")
- $\frac{1}{2}$-player games: ("Markov chain")

Kind of questions

- qualitative questions (threshold is either 0 or 1)
- quantitative questions (threshold is a rational number in $(0,1)$)

What kind of games will we play?

Number of players

- $2\frac{1}{2}$-player games:
- $1\frac{1}{2}$-player games: ("Markov decision process")
- $\frac{1}{2}$-player games: ("Markov chain")

Kind of questions

- qualitative questions (threshold is either 0 or 1)
- quantitative questions (threshold is a rational number in $(0, 1)$)

Winning objective

The winning objective will be an ω-regular condition, or some LTL property, or some more restrictive condition like a reachability condition.

Outline

1. Timed automata

2. Timed games

3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games

4. Conclusion

Rough summary of the results

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable[1] [BBBM08]
	n clocks	decidable?[2]	?
$1\frac{1}{2}$-player game	1 clock	decidable[3] [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable[4] [BF09]

under some assumptions...

1. reactive automata $I(s) = \mathbb{R}_+$, exponential distributions and resets on every cycle
2. reactive automata $I(s) = \mathbb{R}_+$ and exponential distributions
3. reachability properties
4. even for reachability properties, exponential (or uniform) distributions

[BBB+08] 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)*.
[BF09] Bouyer, Forejt. Reachability in stochastic timed games *(ICALP'09)*.

Rough summary of the results

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable [1] [BBBM08]
	n clocks	decidable?[2]	?
$1\frac{1}{2}$-player game	1 clock	decidable[3] [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable [4] [BF09]

under some assumptions...

[1] reactive automata $I(s) = \mathbb{R}_+$, exponential distributions and resets on every cycle

[2] reactive automata $I(s) = \mathbb{R}_+$ and exponential distributions

[3] reachability properties

[4] even for reachability properties, exponential (or uniform) distributions

[BBB+08] 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)*.
[BF09] Bouyer, Forejt. Reachability in stochastic timed games *(ICALP'09)*.

Outline

1. Timed automata

2. Timed games

3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games

4. Conclusion

Summary of the results

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable [BBBM08]
	n clocks	decidable?	?
$1\frac{1}{2}$-player game	1 clock	decidable [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable [BF09]

[BBB+08] 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)*.
[BF09] Bouyer, Forejt. Reachability in stochastic timed games *(ICALP'09)*.

Almost-sure model-checking

The qualitative synthesis problem reduces to the so-called "almost-sure model-checking problem"

$$s \mathrel{|\!\!\approx} \varphi \quad \overset{\text{def}}{\Leftrightarrow} \quad \mathbb{P}\Big(\{\varrho \in \mathsf{Runs}(s) \mid \varrho \models \varphi\}\Big) = 1$$

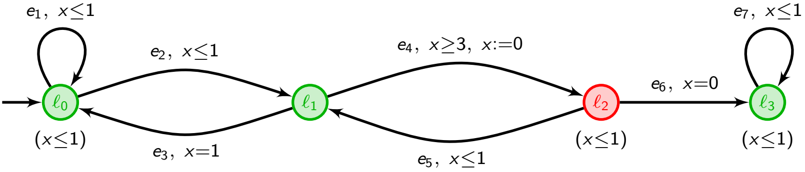
Almost-sure model-checking

The qualitative synthesis problem reduces to the so-called "almost-sure model-checking problem"

$$s \mathrel{|\!\!\approx} \varphi \quad \overset{\text{def}}{\Leftrightarrow} \quad \mathbb{P}\Big(\{\varrho \in \mathsf{Runs}(s) \mid \varrho \models \varphi\}\Big) = 1$$

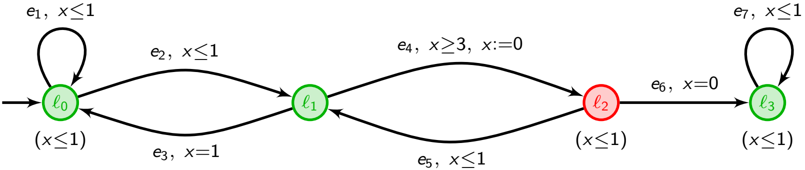
There are only ◯ vertices, but we will use extra colors to represent atomic propositions.

An example



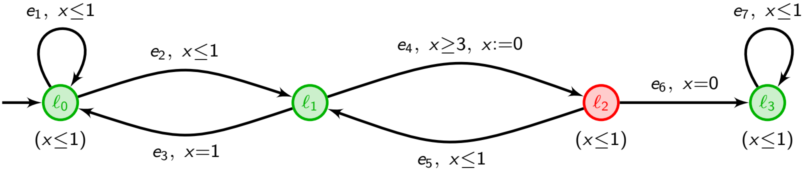
$e_1, x \leq 1$
$e_2, x \leq 1$
$e_4, x \geq 3, x := 0$
$e_7, x \leq 1$
$e_6, x = 0$
ℓ_0
ℓ_1
ℓ_2
ℓ_3
$(x \leq 1)$
$(x \leq 1)$
$(x \leq 1)$
$e_3, x = 1$
$e_5, x \leq 1$

An example



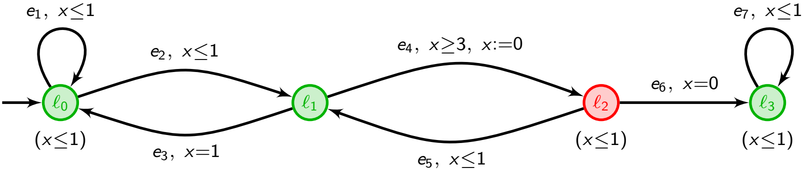
$$\mathcal{A} \not\models \mathbf{G}\,(\text{green} \Rightarrow \mathbf{F}\,\text{red})$$

An example



$\mathcal{A} \not\models \mathbf{G}\,(\text{green} \Rightarrow \mathbf{F}\,\text{red})$ but $\mathbb{P}\Big(\mathcal{A} \models \mathbf{G}\,(\text{green} \Rightarrow \mathbf{F}\,\text{red})\Big) = 1$

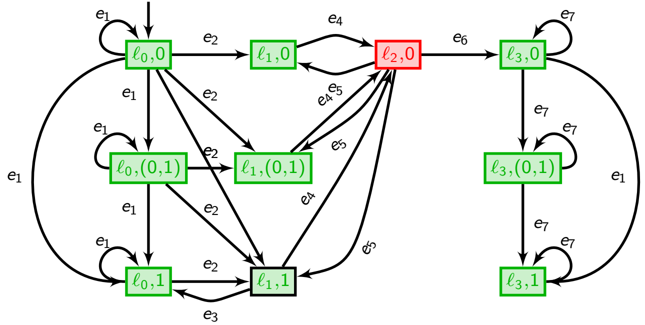
An example



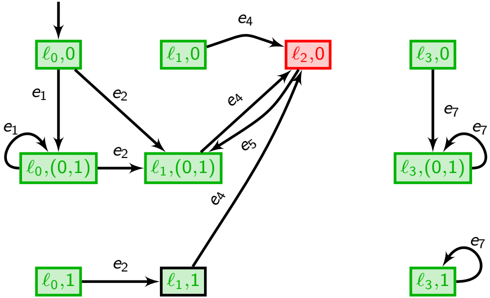
$\mathcal{A} \not\models \mathbf{G}\,(\text{green} \Rightarrow \mathbf{F}\,\text{red})$ but $\mathbb{P}\Big(\mathcal{A} \models \mathbf{G}\,(\text{green} \Rightarrow \mathbf{F}\,\text{red})\Big) = 1$

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

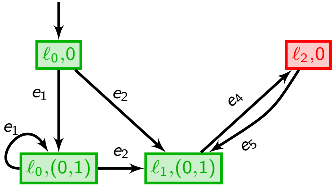
The classical region automaton



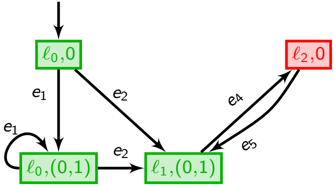
The pruned region automaton



The pruned region automaton

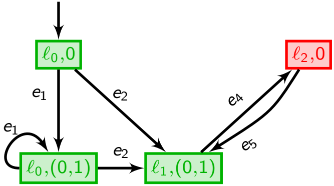


The pruned region automaton



... 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,

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

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

Result

Theorem [BBB+08]

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

- of LTL is PSPACE-Complete
- of ω-regular properties is NLOGSPACE-Complete

[BBB+08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.

Result

Theorem [BBB+08]

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

- of LTL is PSPACE-Complete
- of ω-regular properties is NLOGSPACE-Complete

- Complexity:

[BBB+08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.

Result

Theorem [BBB+08]

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

- of LTL is PSPACE-Complete
- of ω-regular properties is NLOGSPACE-Complete

- Complexity:
  - size of single-clock region automata $=$ polynomial [LMS04]

[BBB+08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[LMS04] Laroussinie, Markey, Schnoebelen. Model checking timed automata with one or two clocks *(CONCUR'04)*.

Result

Theorem [BBB+08]

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

- of LTL is PSPACE-Complete
- of ω-regular properties is NLOGSPACE-Complete

- Complexity:
  - size of single-clock region automata = polynomial [LMS04]
  - apply result of [CSS03] to the finite Markov chain

[BBB+08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[LMS04] Laroussinie, Markey, Schnoebelen. Model checking timed automata with one or two clocks *(CONCUR'04)*.
[CSS03] Couvreur, Saheb, Sutre. An optimal automata approach to LTL model checking of probabilistic systems *(LPAR'03)*.

Result

Theorem [BBB+08]

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

- of LTL is PSPACE-Complete
- of ω-regular properties is NLOGSPACE-Complete

- Complexity:
  - size of single-clock region automata = polynomial [LMS04]
  - apply result of [CSS03] to the finite Markov chain
- Correctness: the proof is rather involved

[BBB+08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[LMS04] Laroussinie, Markey, Schnoebelen. Model checking timed automata with one or two clocks *(CONCUR'04)*.
[CSS03] Couvreur, Saheb, Sutre. An optimal automata approach to LTL model checking of probabilistic systems *(LPAR'03)*.

Result

Theorem [BBB+08]

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

- of LTL is PSPACE-Complete
- of ω-regular properties is NLOGSPACE-Complete

- Complexity:
  - size of single-clock region automata = polynomial [LMS04]
  - apply result of [CSS03] to the finite Markov chain
- Correctness: the proof is rather involved
  - requires the definition of a topology over the set of paths

[BBB+08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[LMS04] Laroussinie, Markey, Schnoebelen. Model checking timed automata with one or two clocks *(CONCUR'04)*.
[CSS03] Couvreur, Saheb, Sutre. An optimal automata approach to LTL model checking of probabilistic systems *(LPAR'03)*.

Result

Theorem [BBB+08]

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

- of LTL is PSPACE-Complete
- of ω-regular properties is NLOGSPACE-Complete

- Complexity:
  - size of single-clock region automata = polynomial [LMS04]
  - apply result of [CSS03] to the finite Markov chain
- Correctness: the proof is rather involved
  - requires the definition of a topology over the set of paths
  - notions of largeness (for proba 1) and meagerness (for proba 0)

[BBB+08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[LMS04] Laroussinie, Markey, Schnoebelen. Model checking timed automata with one or two clocks *(CONCUR'04)*.
[CSS03] Couvreur, Saheb, Sutre. An optimal automata approach to LTL model checking of probabilistic systems *(LPAR'03)*.

Result

Theorem [BBB+08]

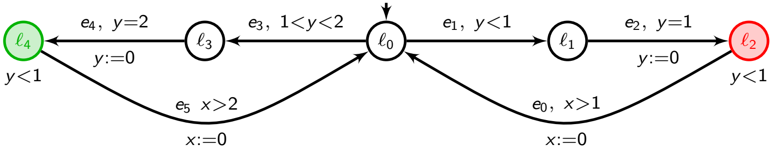
For single-clock timed automata, the almost-sure model-checking

- of LTL is PSPACE-Complete
- of ω-regular properties is NLOGSPACE-Complete

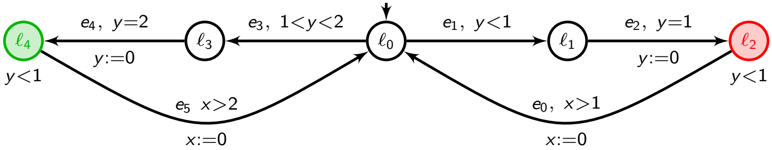
- Complexity:
  - size of single-clock region automata = polynomial [LMS04]
  - apply result of [CSS03] to the finite Markov chain
- Correctness: the proof is rather involved
  - requires the definition of a topology over the set of paths
  - notions of largeness (for proba 1) and meagerness (for proba 0)
  - link between probabilities and topology thanks to the topological games called Banach-Mazur games

[BBB+08] Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata *(LICS'08)*.
[LMS04] Laroussinie, Markey, Schnoebelen. Model checking timed automata with one or two clocks *(CONCUR'04)*.
[CSS03] Couvreur, Saheb, Sutre. An optimal automata approach to LTL model checking of probabilistic systems *(LPAR'03)*.

An example with two clocks

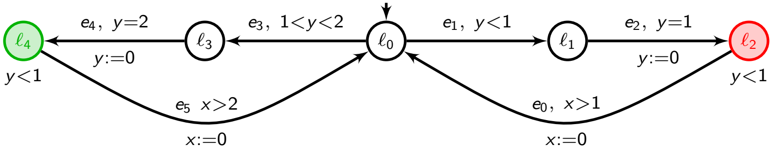


An example with two clocks



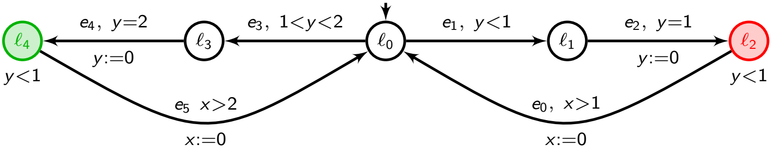
- If the previous algorithm was correct, $\mathcal{A} \mathrel{|\!\!\approx} \mathbf{G}\,\mathbf{F}$ red $\wedge$ $\mathbf{G}\,\mathbf{F}$ green

An example with two clocks



- If the previous algorithm was correct, $\mathcal{A} \mathrel{|}\!\!\approx \mathbf{G}\,\mathbf{F}$ red $\wedge$ $\mathbf{G}\,\mathbf{F}$ green
- However, we can prove that $\mathbb{P}\Big(\mathbf{G}\,\neg\text{red}\Big) > 0$

An example with two clocks



- If the previous algorithm was correct, $\mathcal{A} \not\approx \mathbf{G}\,\mathbf{F}$ red $\wedge$ $\mathbf{G}\,\mathbf{F}$ green
- However, we can prove that $\mathbb{P}\Big(\mathbf{G}\,\neg\text{red}\Big) > 0$
- There is a *strange* convergence phenomenon: along an execution, if $\delta_i > 0$ is the delay in locations ℓ_2 or ℓ_4, then we have that $\sum_i \delta_i \leq 1$

A note on Zeno behaviours

- The set of Zeno behaviours is measurable:

$$\mathrm{Zeno}(s) \;=\; \bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1, \cdots, e_n) \in E^n} \mathrm{Cyl}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))$$

A note on Zeno behaviours

- The set of Zeno behaviours is measurable:

$$\mathsf{Zeno}(s) = \bigcup_{M\in\mathbb{N}} \bigcap_{n\in\mathbb{N}} \bigcup_{(e_1,\cdots,e_n)\in E^n} \mathsf{Cyl}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))$$

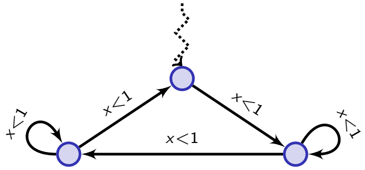
- In single-clock timed automata, we can decide in NLOGSPACE whether $\mathbb{P}\Big(\mathsf{Zeno}(s)\Big) = 0$:

A note on Zeno behaviours

- The set of Zeno behaviours is measurable:

$$\mathsf{Zeno}(s) = \bigcup_{M\in\mathbb{N}} \bigcap_{n\in\mathbb{N}} \bigcup_{(e_1,\cdots,e_n)\in E^n} \mathsf{Cyl}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))$$

- In single-clock timed automata, we can decide in NLOGSPACE whether $\mathbb{P}\Big(\mathsf{Zeno}(s)\Big) = 0$:
  - check whether there is a purely Zeno BSCC in $MC(\mathcal{A})$

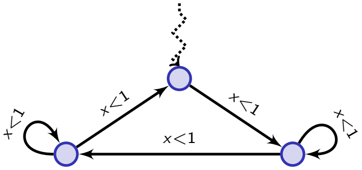


A note on Zeno behaviours

- The set of Zeno behaviours is measurable:

$$\mathrm{Zeno}(s) = \bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1, \cdots, e_n) \in E^n} \mathrm{Cyl}\left(\pi\left(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}\right)\right)$$

- In single-clock timed automata, we can decide in NLOGSPACE whether $\mathbb{P}\Big(\mathrm{Zeno}(s)\Big) = 0$:
  - check whether there is a purely Zeno BSCC in $MC(\mathcal{A})$



  - an interesting notion of non-Zeno timed automata

Outline

1. Timed automata
2. Timed games
3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games
4. Conclusion

Summary of the results

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable [BBBM08]
	n clocks	decidable?	?
$1\frac{1}{2}$-player game	1 clock	decidable [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable [BF09]

[BBB+08] 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)*.
[BF09] Bouyer, Forejt. Reachability in stochastic timed games *(ICALP'09)*.

Undecidability

Theorem [BF09]

The reachability problem for stochastic timed games ($2\frac{1}{2}$ players) is undecidable.

[BF09] Bouyer, Forejt. Reachability in stochastic timed games *(ICALP'09)*.

Undecidability

Theorem [BF09]

The reachability problem for stochastic timed games ($2\frac{1}{2}$ players) is undecidable.

- Holds for uniform and exponential distributions on delays.
- Holds for any quantitative question; we give hints for proba $= \frac{1}{2}$
- Proof by reduction from halting problem of two-counter machine to the reachability with probability precisely $\frac{1}{2}$:

[BF09] Bouyer, Forejt. Reachability in stochastic timed games *(ICALP'09)*.

Undecidability

Theorem [BF09]

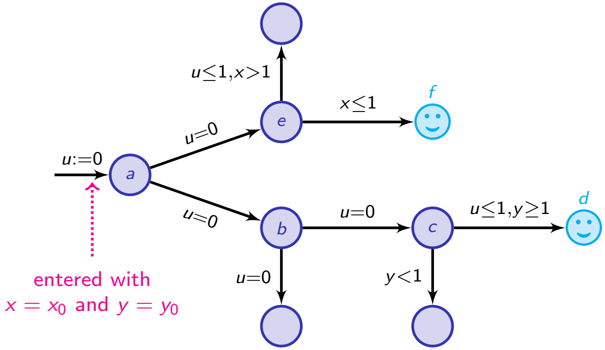
The reachability problem for stochastic timed games ($2\frac{1}{2}$ players) is undecidable.

- Holds for uniform and exponential distributions on delays.
- Holds for any quantitative question; we give hints for proba $= \frac{1}{2}$
- Proof by reduction from halting problem of two-counter machine to the reachability with probability precisely $\frac{1}{2}$:
  - $\diamond$ simulates a computation of the two-counter machine and encodes counter values in clock values
  - $\diamond$ stores counter values c_1 and c_2 as $\frac{1}{2^{c_1} 3^{c_2}}$
  - $\square$ will check that $\diamond$ is not cheating using the power of the probabilities

[BF09] Bouyer, Forejt. Reachability in stochastic timed games *(ICALP'09)*.

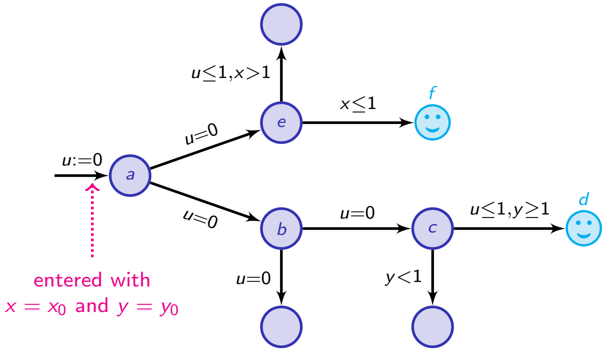
Undecidability – Comparing counter values

- Check clock y stores $(c_1 + 1, c_2)$, assuming that x stores (c_1, c_2)



Undecidability – Comparing counter values

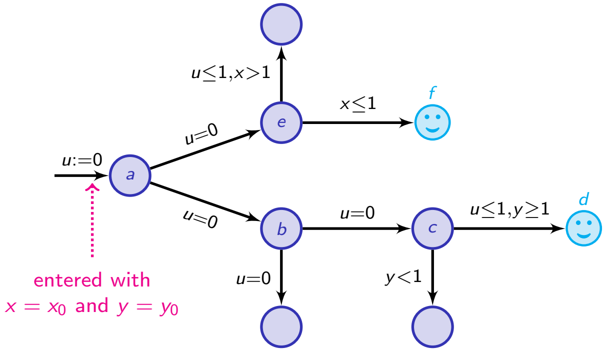
- Check clock y stores $(c_1 + 1, c_2)$, assuming that x stores (c_1, c_2)



- vertex d is reached with probability:

Undecidability – Comparing counter values

- Check clock y stores $(c_1 + 1, c_2)$, assuming that x stores (c_1, c_2)

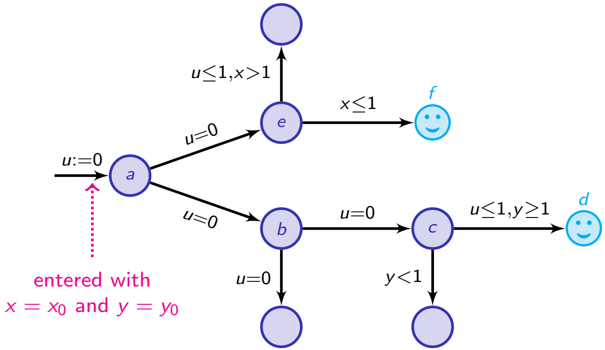


- vertex d is reached with probability:

$$\frac{1}{2} \cdot \frac{1}{2} \cdot \int_{t=1-y_0}^{1} \mathrm{d}t$$

Undecidability – Comparing counter values

- Check clock y stores $(c_1 + 1, c_2)$, assuming that x stores (c_1, c_2)

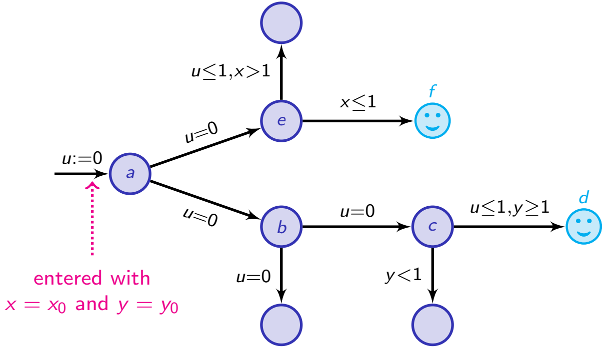


- vertex d is reached with probability:

$$\frac{1}{2} \cdot \frac{1}{2} \cdot \int_{t=1-y_0}^{1} \mathrm{d}t = \frac{1}{4} \cdot y_0$$

Undecidability – Comparing counter values

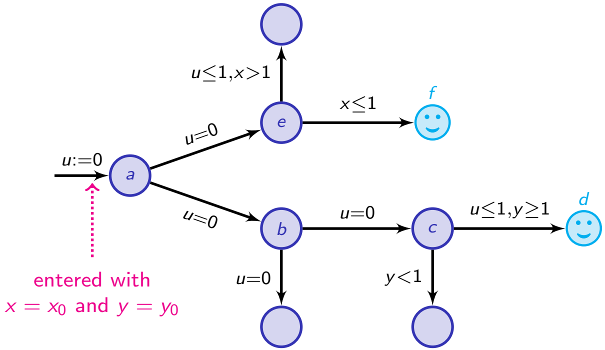
- Check clock y stores $(c_1 + 1, c_2)$, assuming that x stores (c_1, c_2)



- vertex d is reached with probability $\frac{1}{4} \cdot y_0$

Undecidability – Comparing counter values

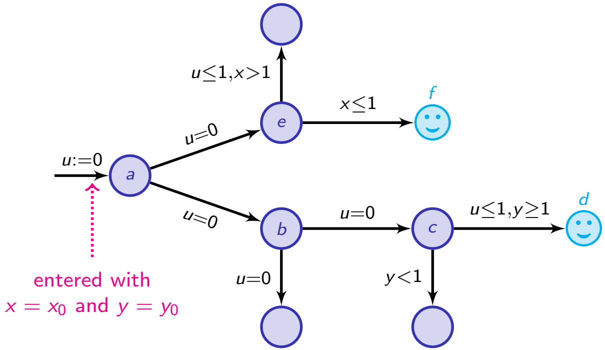
- Check clock y stores (c_1+1, c_2), assuming that x stores (c_1, c_2)



- vertex d is reached with probability $\frac{1}{4} \cdot y_0$
  vertex f with prob. $\frac{1}{2} \cdot (1 - x_0)$

Undecidability – Comparing counter values

- Check clock y stores $(c_1 + 1, c_2)$, assuming that x stores (c_1, c_2)

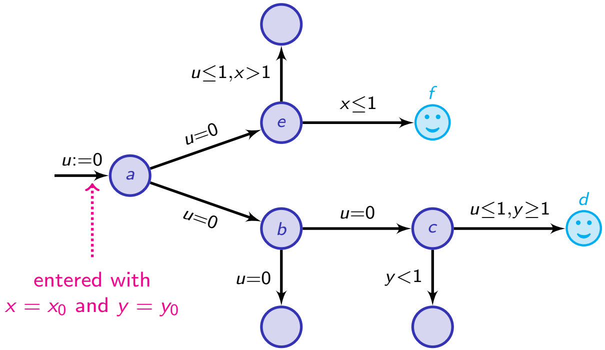


- vertex d is reached with probability $\frac{1}{4} \cdot y_0$
  vertex f with prob. $\frac{1}{2} \cdot (1 - x_0)$
- vertices d, f are reached with probability $\frac{1}{2}$ iff

$$\frac{1}{2} = \frac{1}{4} \cdot y_0 + \frac{1}{2} \cdot (1 - x_0)$$

Undecidability – Comparing counter values

- Check clock y stores $(c_1 + 1, c_2)$, assuming that x stores (c_1, c_2)

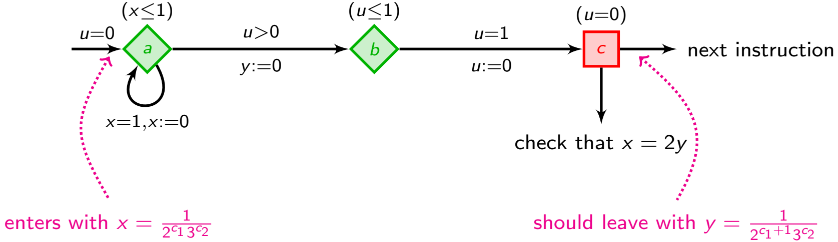


- vertex d is reached with probability $\frac{1}{4} \cdot y_0$
  vertex f with prob. $\frac{1}{2} \cdot (1 - x_0)$
- vertices d, f are reached with probability $\frac{1}{2}$ iff

$$\frac{1}{2} = \frac{1}{4} \cdot y_0 + \frac{1}{2} \cdot (1 - x_0) \qquad \text{iff} \qquad x_0 = 2y_0$$

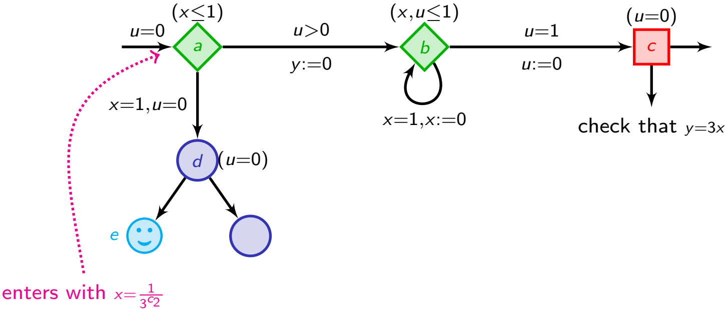
Undecidability – Incrementation

How do we properly increment the first counter?



Undecidability – Zero test

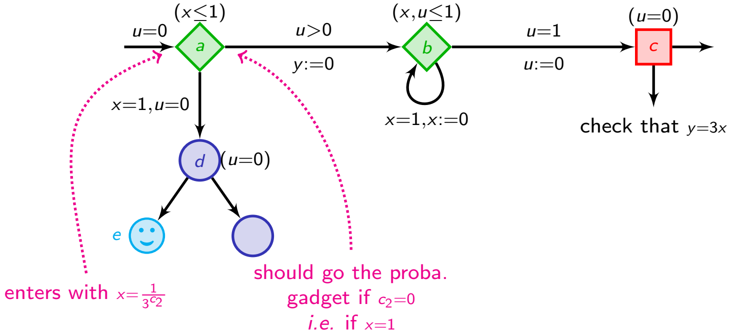
How do we check that c_1 is zero?



Player  has a strategy to reach  with proba. $\frac{1}{2}$ iff c_1 is initially zero.

Undecidability – Zero test

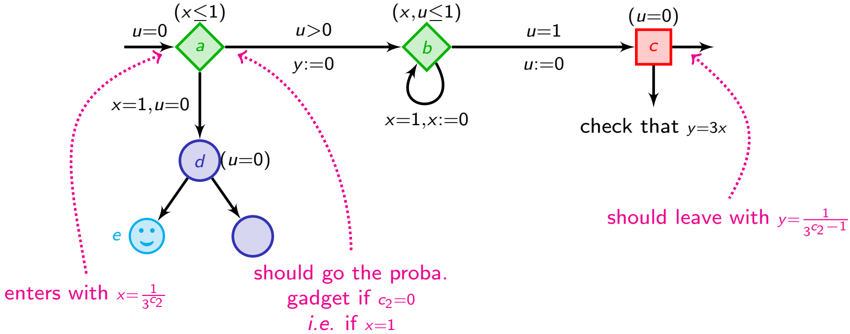
How do we check that c_1 is zero?



Player  has a strategy to reach  with proba. $\frac{1}{2}$ iff c_1 is initially zero.

Undecidability – Zero test

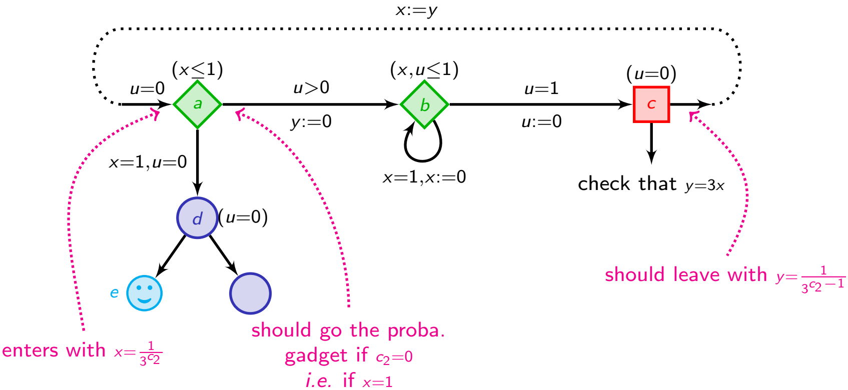
How do we check that c_1 is zero?



Player  has a strategy to reach  with proba. $\frac{1}{2}$ iff c_1 is initially zero.

Undecidability – Zero test

How do we check that c_1 is zero?



Player ◇ has a strategy to reach ☺ with proba. $\frac{1}{2}$ iff c_1 is initially zero.

Outline

1. Timed automata

2. Timed games

3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games

4. Conclusion

Summary of the results

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable [BBBM08]
	n clocks	decidable?	?
$1\frac{1}{2}$-player game	1 clock	decidable [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable [BF09]

[BBB+08] 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)*.
[BF09] Bouyer, Forejt. Reachability in stochastic timed games *(ICALP'09)*.

Towards quantitative analysis

- The abstraction $MC(\mathcal{A})$ is no more correct.

Towards quantitative analysis

- The abstraction $MC(\mathcal{A})$ is no more correct.
- Can be reduced to solving a system of differential equations.

Towards quantitative analysis

- The abstraction $MC(\mathcal{A})$ is no more correct.
- Can be reduced to solving a system of differential equations.
  $\rightsquigarrow$ hard to solve in general, even for simple distributions

Towards quantitative analysis

- The abstraction $MC(\mathcal{A})$ is no more correct.
- Can be reduced to solving a system of differential equations.
  $\rightsquigarrow$ hard to solve in general, even for simple distributions
- We will describe a restricted framework in which:

Towards quantitative analysis

- The abstraction $MC(\mathcal{A})$ is no more correct.
- Can be reduced to solving a system of differential equations.
  $\leadsto$ hard to solve in general, even for simple distributions
- We will describe a restricted framework in which:
  - we will compute a closed-form expression for the probability

Towards quantitative analysis

- The abstraction $MC(\mathcal{A})$ is no more correct.
- Can be reduced to solving a system of differential equations.
  $\rightsquigarrow$ hard to solve in general, even for simple distributions
- We will describe a restricted framework in which:
  - we will compute a closed-form expression for the probability
  - we will be able to approximate the probability

Towards quantitative analysis

- The abstraction $MC(\mathcal{A})$ is no more correct.
- Can be reduced to solving a system of differential equations.
  $\leadsto$ hard to solve in general, even for simple distributions
- We will describe a restricted framework in which:
  - we will compute a closed-form expression for the probability
  - we will be able to approximate the probability, *i.e.*, for every $\varepsilon > 0$, we will compute two rationals p_ε^- and p_ε^+ such that:

$$\begin{cases} p_\varepsilon^- \leq \mathbb{P}\big(s_0 \models \varphi\big) \leq p_\varepsilon^- + \varepsilon \\ p_\varepsilon^+ - \varepsilon \leq \mathbb{P}\big(s_0 \models \varphi\big) \leq p_\varepsilon^+ \end{cases}$$

Towards quantitative analysis

- The abstraction $MC(\mathcal{A})$ is no more correct.
- Can be reduced to solving a system of differential equations.
  $\rightsquigarrow$ hard to solve in general, even for simple distributions
- We will describe a restricted framework in which:
  - we will compute a closed-form expression for the probability
  - we will be able to approximate the probability, *i.e.*, for every $\varepsilon > 0$, we will compute two rationals p_ε^- and p_ε^+ such that:

$$\begin{cases} p_\varepsilon^- \leq \mathbb{P}\Big(s_0 \models \varphi\Big) \leq p_\varepsilon^- + \varepsilon \\ p_\varepsilon^+ - \varepsilon \leq \mathbb{P}\Big(s_0 \models \varphi\Big) \leq p_\varepsilon^+ \end{cases}$$

  - we will be able to decide the threshold problem

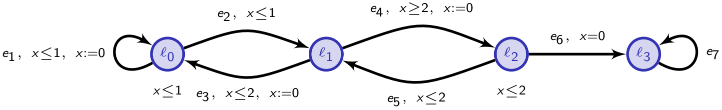
Towards quantitative analysis

- The abstraction $MC(\mathcal{A})$ is no more correct.
- Can be reduced to solving a system of differential equations.
  $\rightsquigarrow$ hard to solve in general, even for simple distributions
- We will describe a restricted framework in which:
  - we will compute a closed-form expression for the probability
  - we will be able to approximate the probability, *i.e.*, for every $\varepsilon > 0$, we will compute two rationals p_ε^- and p_ε^+ such that:

$$\begin{cases} p_\varepsilon^- \leq \mathbb{P}\big(s_0 \models \varphi\big) \leq p_\varepsilon^- + \varepsilon \\ p_\varepsilon^+ - \varepsilon \leq \mathbb{P}\big(s_0 \models \varphi\big) \leq p_\varepsilon^+ \end{cases}$$

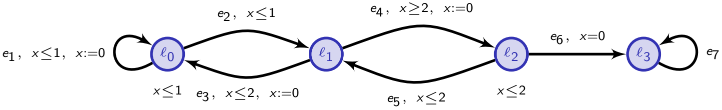
  - we will be able to decide the threshold problem:
    "Given $\mathcal{A}$, φ, $c \in \mathbb{Q}$, and $\sim \in \{<, \leq, =, \geq, >\}$,
    does $\mathbb{P}\big(s_0 \models \varphi\big) \sim c$ in $\mathcal{A}$?"

An example



+ distributions $\mu_s : t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$
μ_s uniform distribution when $I(s)$ is bounded
+ uniform weights on transitions

An example

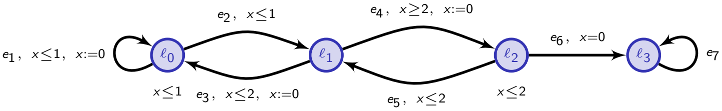


+ distributions $\mu_s\colon t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$
μ_s uniform distribution when $I(s)$ is bounded

+ uniform weights on transitions

We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:

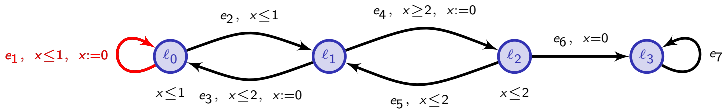
An example



+ distributions $\quad \mu_s : t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$
μ_s uniform distribution when $I(s)$ is bounded
+ uniform weights on transitions

We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:

An example



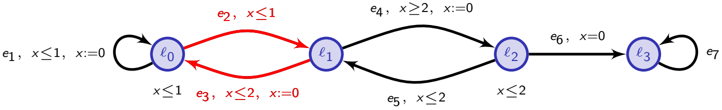
\+ distributions $\mu_s : t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$

μ_s uniform distribution when $I(s)$ is bounded

\+ uniform weights on transitions

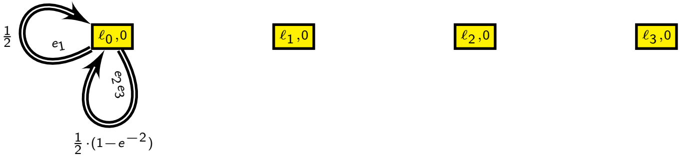
We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:

An example

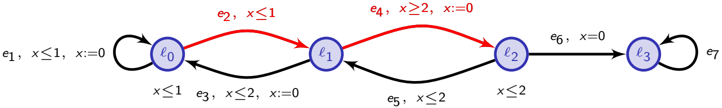


+ distributions $\mu_s : t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$
μ_s uniform distribution when $I(s)$ is bounded
+ uniform weights on transitions

We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:

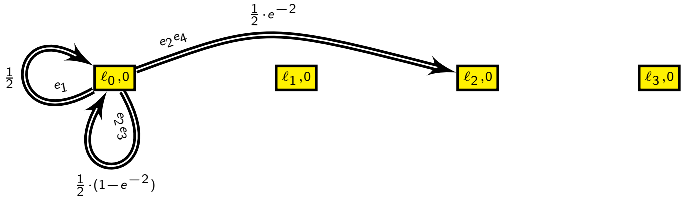


An example

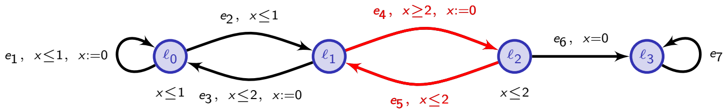


+ distributions $\mu_s : t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$
μ_s uniform distribution when $I(s)$ is bounded
+ uniform weights on transitions

We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:

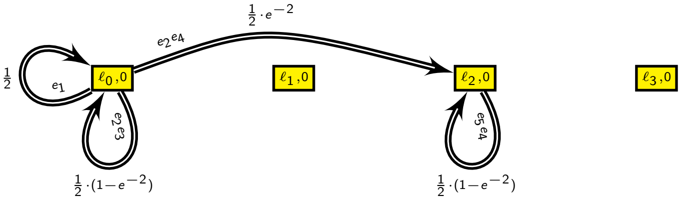


An example

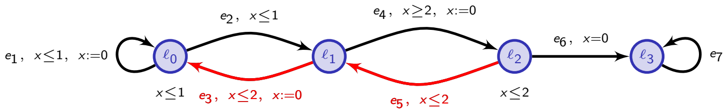


$+$ distributions $\mu_s : t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$
μ_s uniform distribution when $I(s)$ is bounded
$+$ uniform weights on transitions

We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:



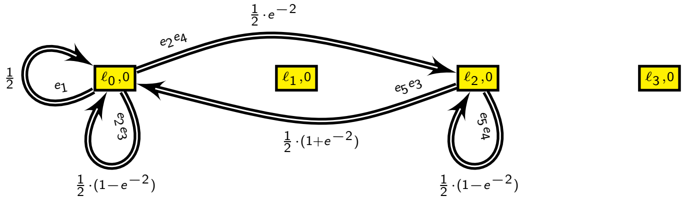
An example



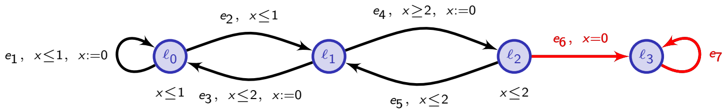
+ distributions $\quad \mu_s: t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$
μ_s uniform distribution when $I(s)$ is bounded

+ uniform weights on transitions

We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:



An example

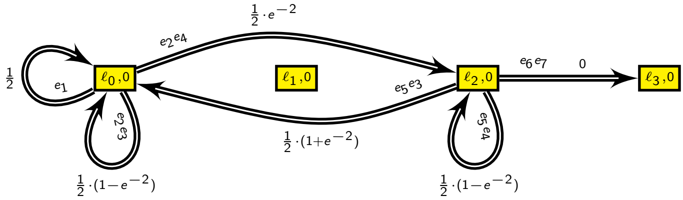


$+$ distributions $\quad \mu_s : t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$

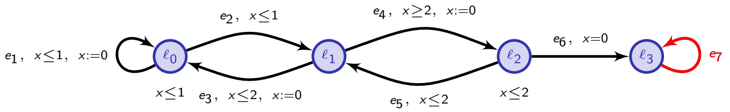
μ_s uniform distribution when $I(s)$ is bounded

$+$ uniform weights on transitions

We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:

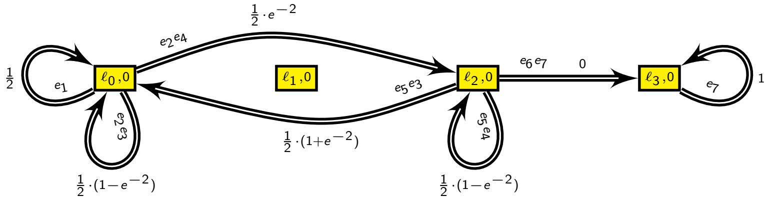


An example

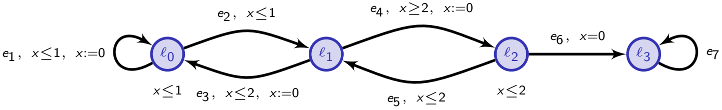


+ distributions $\mu_s : t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$
μ_s uniform distribution when $I(s)$ is bounded
+ uniform weights on transitions

We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:

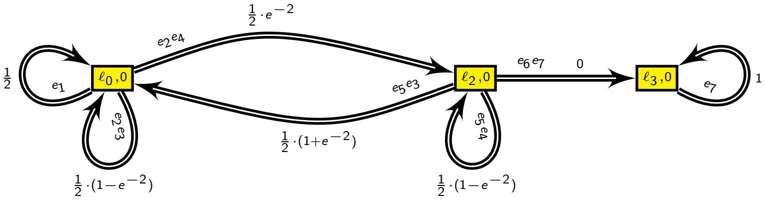


An example



+ distributions $\mu_s : t \mapsto e^{-t}$ when $I(s) = \mathbb{R}_+$
μ_s uniform distribution when $I(s)$ is bounded
+ uniform weights on transitions

We construct a finite Markov chain $MC'(\mathcal{A})$ with macro-edges:



Correctness of the abstraction

Theorem

Under some hypotheses, for single-clock automaton $\mathcal{A}$ and property φ,

$$\mathbb{P}_{\mathcal{A}}(s_0 \models \varphi) = \mathbb{P}_{MC'(\mathcal{A})}(s_0 \models \Diamond F_\varphi)$$

for some well-chosen set F_φ.

Correctness of the abstraction

Theorem

Under some hypotheses, for single-clock automaton $\mathcal{A}$ and property φ,

$$\mathbb{P}_{\mathcal{A}}(s_0 \models \varphi) = \mathbb{P}_{MC'(\mathcal{A})}(s_0 \models \Diamond F_\varphi)$$

for some well-chosen set F_φ.

- **Hypotheses:**
  - if $s = (\ell, \alpha)$ and $s' = (\ell, \alpha')$ with $\alpha, \alpha' > M$, $\mu_s = \mu_{s'}$
  - every bounded cycle resets the clock

Correctness of the abstraction

Theorem

Under some hypotheses, for single-clock automaton $\mathcal{A}$ and property φ,

$$\mathbb{P}_{\mathcal{A}}(s_0 \models \varphi) = \mathbb{P}_{MC'(\mathcal{A})}(s_0 \models \Diamond F_\varphi)$$

for some well-chosen set F_φ.

- **Hypotheses:**
  - if $s = (\ell, \alpha)$ and $s' = (\ell, \alpha')$ with $\alpha, \alpha' > M$, $\mu_s = \mu_{s'}$
  - every bounded cycle resets the clock
- **Limits of the abstraction:** there may be no closed form for the values labelling the edges of $MC'(\mathcal{A})$.

Computing the probability

- We assume furthermore that:
  - for every state s, $I(s) = \mathbb{R}_+$
    (the timed automaton is 'reactive')

Computing the probability

- We assume furthermore that:
  - for every state s, $I(s) = \mathbb{R}_+$
    (the timed automaton is 'reactive')
  - in every location ℓ, the distribution over delays has density
    $t \mapsto \lambda_\ell \cdot e^{-\lambda_\ell \cdot t}$ for some $\lambda_\ell \in \mathbb{Q}_+$

Computing the probability

- We assume furthermore that:
  - for every state s, $I(s) = \mathbb{R}_+$
    (the timed automaton is 'reactive')
  - in every location ℓ, the distribution over delays has density $t \mapsto \lambda_\ell \cdot e^{-\lambda_\ell \cdot t}$ for some $\lambda_\ell \in \mathbb{Q}_+$

$\rightsquigarrow$ more general than continuous-time Markov chains

Computing the probability

- We assume furthermore that:
  - for every state s, $I(s) = \mathbb{R}_+$
    (the timed automaton is 'reactive')
  - in every location ℓ, the distribution over delays has density $t \mapsto \lambda_\ell \cdot e^{-\lambda_\ell \cdot t}$ for some $\lambda_\ell \in \mathbb{Q}_+$

$\rightsquigarrow$ more general than continuous-time Markov chains

Proposition

Under those hypotheses, $\mathbb{P}\Big(s_0 \models \varphi\Big)$ can be expressed as $f\left(e^{-r}\right)$ where r is a rational number, and $f \in \mathbb{Q}(X)$ is a rational function.

Computing the probability

- We assume furthermore that:
  - for every state s, $I(s) = \mathbb{R}_+$ (the timed automaton is 'reactive')
  - in every location ℓ, the distribution over delays has density $t \mapsto \lambda_\ell \cdot e^{-\lambda_\ell \cdot t}$ for some $\lambda_\ell \in \mathbb{Q}_+$

$\rightsquigarrow$ more general than continuous-time Markov chains

Proposition

Under those hypotheses, $\mathbb{P}\Big(s_0 \models \varphi\Big)$ can be expressed as $f\,(e^{-r})$ where r is a rational number, and $f \in \mathbb{Q}(X)$ is a rational function.

$\rightsquigarrow$ Note: the hypothesis "reset all bounded cycles" is necessary to get this form.

Approximating the probability

$$\mathbb{P}\Big(s_0 \models \varphi\Big) = f\left(e^{-r}\right)$$

Approximating the probability

$$\mathbb{P}\Big(s_0 \models \varphi\Big) = f\left(e^{-r}\right)$$

- We can compute sequences $(a_i)_i$ and $(b_i)_i$ with
  - $\lim_i a_i = \lim_i b_i = e^{-r}$
  - $a_i \leq a_{i+1} \leq e^{-r} \leq b_{i+1} \leq b_i$

Approximating the probability

$$\mathbb{P}\Big(s_0 \models \varphi\Big) = f\left(e^{-r}\right)$$

- We can compute sequences $(a_i)_i$ and $(b_i)_i$ with
  - $\lim_i a_i = \lim_i b_i = e^{-r}$
  - $a_i \leq a_{i+1} \leq e^{-r} \leq b_{i+1} \leq b_i$
- As e^{-r} is transcendental, we can compute an interval $(\alpha, \beta) \ni e^{-r}$ over which f is monotonic:

Approximating the probability

$$\mathbb{P}\Big(s_0 \models \varphi\Big) = f\left(e^{-r}\right)$$

- We can compute sequences $(a_i)_i$ and $(b_i)_i$ with
  - $\lim_i a_i = \lim_i b_i = e^{-r}$
  - $a_i \leq a_{i+1} \leq e^{-r} \leq b_{i+1} \leq b_i$
- As e^{-r} is transcendental, we can compute an interval $(\alpha, \beta) \ni e^{-r}$ over which f is monotonic:
  - writing $f = P/Q$, we have that $f' = (P'Q - PQ')/Q^2$

Approximating the probability

$$\mathbb{P}\Big(s_0 \models \varphi\Big) = f\left(e^{-r}\right)$$

- We can compute sequences $(a_i)_i$ and $(b_i)_i$ with
  - $\lim_i a_i = \lim_i b_i = e^{-r}$
  - $a_i \leq a_{i+1} \leq e^{-r} \leq b_{i+1} \leq b_i$
- As e^{-r} is transcendental, we can compute an interval $(\alpha, \beta) \ni e^{-r}$ over which f is monotonic:
  - writing $f = P/Q$, we have that $f' = (P'Q - PQ')/Q^2$
  - by induction on the degree of $R = P'Q - PQ'$, we prove that the sign of R is constant over (α, β) (that we can compute)

Approximating the probability

$$\mathbb{P}\Big(s_0 \models \varphi\Big) = f\left(e^{-r}\right)$$

- We can compute sequences $(a_i)_i$ and $(b_i)_i$ with
  - $\lim_i a_i = \lim_i b_i = e^{-r}$
  - $a_i \leq a_{i+1} \leq e^{-r} \leq b_{i+1} \leq b_i$
- As e^{-r} is transcendental, we can compute an interval $(\alpha, \beta) \ni e^{-r}$ over which f is monotonic:
  - writing $f = P/Q$, we have that $f' = (P'Q - PQ')/Q^2$
  - by induction on the degree of $R = P'Q - PQ'$, we prove that the sign of R is constant over (α, β) (that we can compute)

    If the sign of R' is constant over (α', β') (containing e^{-r}), the sign of R will be constant over $(\alpha, \beta) = (a_j, b_j) \subseteq (\alpha', \beta')$ if $R(a_j) \cdot R(b_j) > 0$.

Approximating the probability

$$\mathbb{P}\Big(s_0 \models \varphi\Big) = f\left(e^{-r}\right)$$

- We can compute sequences $(a_i)_i$ and $(b_i)_i$ with
  - $\lim_i a_i = \lim_i b_i = e^{-r}$
  - $a_i \leq a_{i+1} \leq e^{-r} \leq b_{i+1} \leq b_i$
- As e^{-r} is transcendental, we can compute an interval $(\alpha, \beta) \ni e^{-r}$ over which f is monotonic:
  - writing $f = P/Q$, we have that $f' = (P'Q - PQ')/Q^2$
  - by induction on the degree of $R = P'Q - PQ'$, we prove that the sign of R is constant over (α, β) (that we can compute)

    If the sign of R' is constant over (α', β') (containing e^{-r}), the sign of R will be constant over $(\alpha, \beta) = (a_j, b_j) \subseteq (\alpha', \beta')$ if $R(a_j) \cdot R(b_j) > 0$.
- When $(a_N, b_N) \subseteq (\alpha, \beta)$, the two sequences $(f(a_i))_{i \geq N}$ and $(f(b_i))_{i \geq N}$ are monotonic and converge to $f\left(e^{-r}\right)$

Deciding the threshold problem

Theorem [BBBM08]

Under the previous hypotheses, the threshold problem is decidable.

[BBBM08] Bertrand, Bouyer, Brihaye, Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics *(QEST'08)*.

Deciding the threshold problem

Theorem [BBBM08]

Under the previous hypotheses, the threshold problem is decidable.

- Check whether $c = f(e^{-r})$

[BBBM08] Bertrand, Bouyer, Brihaye, Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics *(QEST'08)*.

Deciding the threshold problem

Theorem [BBBM08]

Under the previous hypotheses, the threshold problem is decidable.

- Check whether $c = f(e^{-r})$
- If not:

[BBBM08] Bertrand, Bouyer, Brihaye, Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics *(QEST'08)*.

Deciding the threshold problem

Theorem [BBBM08]

Under the previous hypotheses, the threshold problem is decidable.

- Check whether $c = f(e^{-r})$
- If not:
  - use the approximation scheme for a sequence $(\varepsilon_n)_n$ that converges to 0

[BBBM08] Bertrand, Bouyer, Brihaye, Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics *(QEST'08)*.

Deciding the threshold problem

Theorem [BBBM08]

Under the previous hypotheses, the threshold problem is decidable.

- Check whether $c = f(e^{-r})$
- If not:
  - use the approximation scheme for a sequence $(\varepsilon_n)_n$ that converges to 0
  - stop when the under- and the over-approximations are on the same side of the threshold c

[BBBM08] Bertrand, Bouyer, Brihaye, Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics *(QEST'08)*.

Outline

1. Timed automata

2. Timed games

3. A hint into stochastic timed games
   - Some informal description
   - A more formal view of the semantics
   - Summary of the results
   - Qualitative analysis of $\frac{1}{2}$-player games
   - Quantitative analysis of $2\frac{1}{2}$-player games
   - Quantitative analysis of $\frac{1}{2}$-player games

4. Conclusion

Conclusion and perspectives

- We have presented a general model for stochastic timed games:
  - timing constraints
  - probabilistic features
  - non-determinism and interaction

Conclusion and perspectives

- We have presented a general model for stochastic timed games:
  - timing constraints
  - probabilistic features
  - non-determinism and interaction
- Not much has been done so far!

Conclusion and perspectives

- We have presented a general model for stochastic timed games:
  - timing constraints
  - probabilistic features
  - non-determinism and interaction

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable[1] [BBBM08]
	n clocks	decidable?[2]	?
$1\frac{1}{2}$-player game	1 clock	decidable[3] [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable[4] [BF09]

- Not much has been done so far!
  - even for simple untimed objectives, few is known

Conclusion and perspectives

- We have presented a general model for stochastic timed games:
  - timing constraints
  - probabilistic features
  - non-determinism and interaction

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable[1] [BBBM08]
	n clocks	decidable?[2]	?
$1\frac{1}{2}$-player game	1 clock	decidable[3] [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable[4] [BF09]

- Not much has been done so far!
  - even for simple untimed objectives, few is known
  - nothing about more involved quantitative objectives (e.g. expected time, timed properties, ...)

Conclusion and perspectives

- We have presented a general model for stochastic timed games:
  - timing constraints
  - probabilistic features
  - non-determinism and interaction

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable[1] [BBBM08]
	n clocks	decidable?[2]	?
$1\frac{1}{2}$-player game	1 clock	decidable[3] [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable[4] [BF09]

- Not much has been done so far!
  - even for simple untimed objectives, few is known
  - nothing about more involved quantitative objectives (e.g. expected time, timed properties, ...)
  - what about approximate probabilities?

Conclusion and perspectives

- We have presented a general model for stochastic timed games:
  - timing constraints
  - probabilistic features
  - non-determinism and interaction

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable[1] [BBBM08]
	n clocks	decidable?[2]	?
$1\frac{1}{2}$-player game	1 clock	decidable[3] [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable[4] [BF09]

- Not much has been done so far!
  - even for simple untimed objectives, few is known
  - nothing about more involved quantitative objectives (e.g. expected time, timed properties, ...)
  - what about approximate probabilities?
  - compositionality problems

Conclusion and perspectives

- We have presented a general model for stochastic timed games:
  - timing constraints
  - probabilistic features
  - non-determinism and interaction

Model		Qualitative	Quantitative
$\frac{1}{2}$-player game	1 clock	decidable [BBB+08]	decidable[1] [BBBM08]
	n clocks	decidable?[2]	?
$1\frac{1}{2}$-player game	1 clock	decidable[3] [BF09]	?
	n clocks	?	?
$2\frac{1}{2}$-player game	1 clock	?	?
	n clocks	?	undecidable[4] [BF09]

- Not much has been done so far!
  - even for simple untimed objectives, few is known
  - nothing about more involved quantitative objectives (e.g. expected time, timed properties, ...)
  - what about approximate probabilities?
  - compositionality problems

- Probabilistic timed automata (PRISM and UPPAAL-PRO model)
  - the questions considered in this presentation can be "trivially" answered (because they reduce to similar questions on discrete-time Markov decision processes)
  - quantitative objectives should be investigated