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

<table>
<thead>
<tr>
<th>State</th>
<th>Transition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>safe</td>
<td>23</td>
<td>safe</td>
</tr>
<tr>
<td></td>
<td>problem</td>
<td>alarm</td>
</tr>
<tr>
<td></td>
<td>15.6</td>
<td>alarm</td>
</tr>
<tr>
<td></td>
<td>delayed</td>
<td>failsafe</td>
</tr>
<tr>
<td></td>
<td>done</td>
<td>safe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>State</th>
<th>Transition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>problem</td>
<td></td>
<td>safe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>alarm</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repair</td>
</tr>
<tr>
<td></td>
<td></td>
<td>repairing</td>
</tr>
<tr>
<td></td>
<td></td>
<td>done</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>done</td>
</tr>
<tr>
<td>15.6</td>
<td>25</td>
<td>safe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>failsafe</td>
</tr>
<tr>
<td>15.6</td>
<td>0</td>
<td>safe</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>23</td>
<td>23</td>
<td>problem</td>
</tr>
<tr>
<td>0</td>
<td>15.6</td>
<td>repair</td>
</tr>
<tr>
<td>17.9</td>
<td>40</td>
<td>done</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>23</td>
<td>problem</td>
</tr>
<tr>
<td>3</td>
<td>17.9</td>
<td>repair</td>
</tr>
<tr>
<td>0</td>
<td>40</td>
<td>done</td>
</tr>
</tbody>
</table>
Verification

Emptiness problem

Is the language accepted by a timed automaton empty?

- basic reachability/safety properties (final states)
- basic liveness properties ($\omega$-regular conditions)

Theorem [AD90, AD94]
The emptiness problem for timed automata is decidable and PSPACE-complete.

Method: construct a finite abstraction
Verification

Emptiness problem
Is the language accepted by a timed automaton empty?

- **Problem:** the set of configurations is infinite
  ~ 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
  ~ 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
  - 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).
Verification

Emptiness problem
Is the language accepted by a timed automaton empty?

- **Problem:** the set of configurations is infinite
  \(\Rightarrow\) 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).
The region abstraction

\[ x \sim c \quad \text{with} \quad c \in \{0, 1, 2\} \]
\[ y \sim c \quad \text{with} \quad c \in \{0, 1, 2\} \]

The path \( x = 1 \quad y = 1 \) cannot be fired from.

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

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]

\[ \ldots \]
The region abstraction

clock $y$

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 $x=1$ can be fired from $y=1$ cannot be fired from

“compatibility” between regions and constraints
“compatibility” between regions and time elapsing
The region abstraction

The path $x=1$ $y=1$
- 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

\[ \sim \text{ an equivalence of finite index} \]
The region abstraction

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

\[ \sim \text{ an equivalence of finite index} \]
\[ \text{a time-abstract bisimulation} \]
Time-abstract bisimulation

This is a relation between $\bullet$ and $\bullet$ such that:
Time-abstract bisimulation

This is a relation between \( \bullet \) and \( \bullet \) such that:

\[
\forall d > 0 \exists d' > 0 \prec u (d') \quad \forall d' > 0 \exists d > 0 \prec u (d) 
\]

... and vice-versa (swap \( \bullet \) and \( \bullet \)).
This is a relation between \( \bullet \) and \( \bullet \) such that:

\[
\forall \exists a \quad (d) \quad \forall d > 0 \exists d' > 0 \quad \leftrightarrow (d') \quad \text{and vice-versa (swap \( \bullet \) and \( \bullet \)).}
\]
Time-abstract bisimulation

This is a relation between \( \bullet \) and \( \bullet \) such that:

\[
\forall d > 0 \quad \exists d' > 0 \quad \delta(d) \]

\[
\forall \quad \exists a
\]
This is a relation between $\bullet$ and $\bullet$ such that:

$$\forall \exists \forall \exists \delta(d) \delta(d')$$

$$\forall \exists \forall \exists a$$

$$\forall d > 0 \exists d' > 0$$
Time-abstract bisimulation

This is a relation between \( \circ \) and \( \cdot \) such that:

\[
\forall a \exists u \left( d > 0 \right) \exists d' > 0 \quad \text{and vice-versa (swap } \circ \text{ and } \cdot \text{).}
\]
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:
  \[
  \begin{align*}
  0 < x < 1 \\
  0 < y < 1 \\
  y < x
  \end{align*}
  \]

- 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 $\equiv$ finite bisimulation quotient
Region automaton $\equiv$ finite bisimulation quotient

Timed automaton

Region graph

Region automaton
Region automaton \equiv \text{finite bisimulation quotient}

\[ \mathcal{L} (\text{reg. aut.}) = \text{UNTIME} (\mathcal{L} (\text{timed aut.})) \]
An example [AD94]

Timed automata

- $s_0$: $x > 0, a$
  - $y := 0$

- $s_1$: $x < 1, c$
  - $y := 0$

- $s_2$: $y = 1, b$

- $s_3$: $x > 1, d$
  - $x < 1, d$

- $s_0$ to $s_1$: $y := 0$
- $s_1$ to $s_2$: $x < 1, c$
- $s_2$ to $s_3$: $x < 1, c$
- $s_3$ to $s_0$: $x > 1, d$
An example [AD94]
An example [AD94]
Timed automata

Finite bisimulation quotient

Large (but finite) automaton
(region automaton)

Timed automaton

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

- **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 automata

- **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)
Timed automata

finite bisimulation quotient

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 automata

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

- Timed games
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

\[
\begin{align*}
\text{add} & : x \leq 2 \\
\text{idle} & : x \geq 1 \\
\text{mult} & : x \geq 1
\end{align*}
\]

\[x := 0\]

(x ≤ 2)

(x ≥ 3)

+ done

idle

- 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:

```
x ≤ 1, c_1
```

```
x ≥ 1, u_3
```

```
x ≥ 2, c_4
```

```
x < 1, u_2, x := 0
```

```
x < 1, u_1
```

```
x ≤ 1, c_3
```

```
x ≤ 2
```

```
(x ≤ 2)
```

An example of a timed game

Rule of the game

- **Aim:** avoid 😞 and reach 😊
- **How do we play?** According to a strategy:

$$f : \text{history} \mapsto (\text{delay, cont. transition})$$
An example of a timed game

Rule of the game
- **Aim:** avoid 😞 and reach 😊
- **How do we play?** According to a strategy:

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

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

A (memoryless) winning strategy

- from $(\ell_0, 0)$, play $(0.5, c_1)$
  
  $\sim$ 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 \((0.5, c_1)\)
  \(\leadsto\) can be preempted by \(u_2\)
- from \((\ell_2, \ast)\), play \((1 - \ast, c_2)\)
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 \((0.5, c_1)\)
  \(\leadsto\) can be preempted by \(u_2\)
- from \((\ell_2, *)\), play \((1 - *, c_2)\)
- 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:

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

A (memoryless) winning strategy

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

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

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:

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

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:

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

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.


[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 \(\text{EXPTIME}\)-complete. Furthermore memoryless and “region-based” strategies are sufficient.

\[ \leadsto \text{classical regions are sufficient for solving such problems} \\
\text{(one only needs to compute the so-called attractor)} \]

---

[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

\[ x \leq 2 \]

\[ x \leq 1, c_1 \]

\[ x < 1, u_2, x := 0 \]

\[ x < 1, u_1 \]

\[ x \geq 2, c_4 \]

\[ x \geq 1, u_3 \]

\[ x \leq 1, c_3 \]

\[ c_2 \]

\[ x \leq 1, c_1 \]
Back to the example: computing winning states

Timed games

\[
\ell_0 : x \leq 2 \\
\ell_1 : x \leq 1, c_1 \\
\ell_2 : x < 1, u_1 \\
\ell_3 : c_2
\]

\[
\ell_0 : x \geq 1, u_3 \\
\ell_1 : x \geq 2, c_4 \\
\ell_2 : x \leq 1, c_3 \\
\ell_3 : x < 1, u_2, x := 0
\]
Back to the example: computing winning states

Timed games
Back to the example: computing winning states

Back to the example: computing winning states

Timed games

Back to the example: computing winning states

Back to the example: computing winning states

Back to the example: computing winning states

Timed games
Back to the example: computing winning states

\[ \ell_0 \]

\[ (x \leq 2) \]

\[ x \geq 1, u_3 \]

\[ x \leq 1, c_1 \]

\[ x \leq 1, c_3 \]

\[ x \leq 1, c_1 \]

\[ x \leq 1, c_3 \]

\[ x < 1, u_2, x := 0 \]

\[ x < 1, u_1 \]

\[ x \geq 1, c_4 \]

\[ x \geq 2, c_4 \]

\[ c_2 \]

\[ \ell_1 \]

\[ \ell_2 \]

\[ \ell_3 \]

\[ \ell_0 \]

\[ 0 \quad 1 \quad 2 \quad 3 \]

\[ \ell_1 \]

\[ 0 \quad 1 \quad 2 \quad 3 \]

\[ \ell_2 \]

\[ 0 \quad 1 \quad 2 \quad 3 \]

\[ \ell_3 \]

\[ 0 \quad 1 \quad 2 \quad 3 \]
Back to the example: computing winning states

Timed games

\(x \leq 2\)

\(x \geq 1, u_3\)

\(x \leq 1, c_1\)

\(x < 1, u_1\)

\(x < 1, u_2, x := 0\)

\(x \geq 1, c_1\)

\(x \geq 2, c_4\)

\(x \leq 1, c_3\)

\(c_2\)

\(0, 1, 2, 3\)

\(\ell_0\)

\(\ell_1\)

\(\ell_2\)

\(\ell_3\)
Back to the example: computing winning states

Timed games
Back to the example: computing winning states

Timed games

$$x < 1, u_2, x := 0$$

$$x < 1, u_1$$

$$x \leq 1, c_1$$

$$x \leq 1, c_3$$

$$x \geq 1, u_3$$

$$x \geq 2, c_4$$

$$c_2$$
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

\[
\text{send} \quad x := 0 \quad \xrightarrow{\text{lost}} \quad x \leq 2 \quad \xrightarrow{\text{delivered}}
\]
Why add stochastic features? And how?
- to model probabilistic behaviours

Example of losses when sending messages

\[
\begin{align*}
\text{send} & \quad x := 0 \\
& \rightarrow \\
\text{lost} & \quad 0.1 \quad x \leq 2 \\
& \quad 0.9 \\
\text{delivered} & \quad \leadsto
\end{align*}
\]

\sim the probabilistic timed automata model used e.g. in PRISM and UPPAAL-PRO

[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

\[ x := 0 \quad \begin{array}{c}
\text{send} \\ x \leq 2 \\
\text{lost} \\
\text{delivered}
\end{array} \]

\[ \leadsto \text{the probabilistic timed automata model} \]

used e.g. in PRISM and UPPAAL-PRO

[Baier et al. 2008, Bouyer & Forejt 2009]

- 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**

![Diagram showing send, lost, and delivered states with probabilities]

- to model uncertainty on delays

**Example of a processor in the taskgraph example**

![Diagram showing states and transitions for a processor]

[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

\[
\begin{align*}
\text{send: } & x:=0 \\
\text{lost: } & 0.1 \quad x \leq 2 \\
\text{delivered: } & 0.9
\end{align*}
\]

\[\leadsto \text{the probabilistic timed automata model used e.g. in PRISM and UPPAAL-PRO [KNSS02]}\]

- to model uncertainty on delays

Example of a processor in the taskgraph example

\[\leadsto \text{the stochastic timed automata model [BBB+08,BF09]}\]
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 $\blacksquare$ and $\Box$ play according to standard strategies.
Player $\bigcirc$ 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

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

- \( a, 0 \)\n- \( c, 1 \)\n- \( e, 1 + \frac{1}{u} \)\n- \( g, 2 \)\n- \( f, 2 \)
Play, an example

- Strategy for \( \diamond \): go to \( c \) when \( x = 1 \)
- Strategy for \( \square \): go to \( g \) when \( x = 2 \)
Play, an example

- Strategy for $\diamond$: go to $c$ when $x = 1$
- Strategy for $\Box$: 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:

\[(a,0) \rightarrow (c,1)\]
Play, an example

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

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

The graph shows transitions between states with probability distributions over delays.
Play, an example

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

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

Probability distribution over delays
Play, an example

- Strategy for $\blackdiamond$: 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:

probability distribution over delays
Play, an example

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

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

probability distribution over delays
Play, an example

- Strategy for green diamond: go to \( c \) when \( x = 1 \)
- Strategy for red box: go to \( g \) when \( x = 2 \)

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

\[
\begin{align*}
(a,0) &\rightarrow (c,1) \\
(b,1) &\rightarrow (e,1) \\
(e,1+\varepsilon) &\rightarrow (g,2) \\
(e,1) &\rightarrow (e,2) \\
(d,2) &\rightarrow (d,2)
\end{align*}
\]
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:

- probability distribution over delays

$$
\begin{align*}
(a,0) & \rightarrow (c,1) \\
(b,1) & \rightarrow (e,1) \\
(e,1+\varepsilon) & \rightarrow (g,2) \\
(d,2) & \rightarrow (e,2)
\end{align*}
$$
Play, an example

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

- Strategy for \( \diamondsuit \): go to \( c \) when \( x = 1 \)
- Strategy for \( \square \): go to \( g \) when \( x = 2 \)

A hint into stochastic timed games
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

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

  \( \sim \) this is ok if delays are in \([0, +\infty)\)
How can we attach probabilities to delays?

- The example of continuous-time Markov chains

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

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

- But what if bounded intervals?

A hint into stochastic timed games
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

  \[
  \text{exponential distribution}
  \]

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

  \[\sim \text{ this is ok if delays are in } [0, +\infty)\]

- But what if bounded intervals?

  \[
  \text{truncated normal distribution}
  \]

  \[
  \text{density function } t \mapsto \begin{cases} 
  \frac{1}{\sqrt{\pi} \cdot \sigma} \exp\left(-\frac{t^2}{\sigma^2}\right) & \text{if } t \geq 0 \\
  0 & \text{otherwise}
  \end{cases}
  \]
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 \( \bigcirc \). 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

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

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

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

![Diagram showing symbolic path with transitions $x \leq 2, e_1$, $x \leq 5, e_2$, $y := 0$, $y \geq 1$, $x = 1, e_3$, $x \leq 3, e_4$]

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

- Idea: compute the probability of a symbolic path

From state $s$: $s$
A hint into stochastic timed games

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:

  ![Diagram](image)

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

- **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:

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

- 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:

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

- 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

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

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

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

$\mathbb{P}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \int_{t \in I(s,e_1)} p_{s+t}(e_1) \mathbb{P}(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})) \, d\mu_s(t)$
The $\frac{1}{2}$-player game model

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

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

- $I(s, e_1) = \{ \tau \mid s \xrightarrow{\tau,e_1} \}$ and $\mu_s$ distribution over $I(s) = \bigcup_e I(s, e)$
The $\frac{1}{2}$-player game model

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

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

- $I(s,e_1) = \{\tau \mid s \xrightarrow{\tau,e_1}\}$ and $\mu_s$ distribution over $I(s) = \bigcup_e I(s,e)$
The $\frac{1}{2}$-player game model

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

$$\mathbb{P}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \int_{t \in I(s,e_1)} \mu_{s}(t) d \mu_{s}(t)$$

- $l(s, e_1) = \{\tau \mid s \xrightarrow{\tau,e_1}\}$ and $\mu_{s}$ distribution over $l(s) = \bigcup_{e} l(s, e)$

- $p_{s+t}$ distribution over transitions enabled in $s + t$
  (given by weights on transitions)
The $\frac{1}{2}$-player game model

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

$$P(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \int_{t \in I(s, e_1)} p_s(t) P(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})) d\mu_s(t)$$

- $l(s, e_1) = \{\tau \mid s \xrightarrow{\tau, e_1}\}$ and $\mu_s$ distribution over $l(s) = \bigcup_e l(s, e)$
- $p_s(t)$ 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

$$\mathbb{P}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) ) = \int_{t \in I(s,e_1)} p_{s+t}(e_1) \mathbb{P}(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n}) ) \, d\mu_s(t)$$
The $\frac{1}{2}$-player game model

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

- Can be viewed as an $n$-dimensional integral
The $\frac{1}{2}$-player game model

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

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

$$\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} ) = \{ s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models C\}$$
The $\frac{1}{2}$-player game model

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

- Can be viewed as an $n$-dimensional integral
- Easy extension to constrained symbolic paths
  $$
  \pi_c\left(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} \right) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models C\}
  $$
- Definition over sets of infinite runs:
The $\frac{1}{2}$-player game model

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

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

- Easy extension to constrained symbolic paths
  $$\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1,e_1}s_1 \cdots \xrightarrow{\tau_n,e_n}s_n \mid (\tau_1, \cdots, \tau_n) \models C\}$$

- Definition over sets of infinite runs:
  - $\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{\varphi \cdot \varphi' \mid \varphi \in \pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\}$
The $\frac{1}{2}$-player game model

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

- Can be viewed as an $n$-dimensional integral
- Easy extension to constrained symbolic paths
  $$\pi_C\left( s \xrightarrow{e_1} \cdots \xrightarrow{e_n} \right) = \{ s \xrightarrow{\tau_1,e_1} s_1 \cdots \xrightarrow{\tau_n,e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models C \}$$
- Definition over sets of infinite runs:
  - $\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{ \rho \cdot \rho' \mid \rho \in \pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) \}$
  - $\mathbb{P}(\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))) = \mathbb{P}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))$
The $\frac{1}{2}$-player game model

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

- Can be viewed as an $n$-dimensional integral
- Easy extension to constrained symbolic paths
  $$\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} ) = \{ s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models C \}$$
- Definition over sets of infinite runs:
  - $\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} )) = \{ \varrho \cdot \varrho' \mid \varrho \in \pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} ) \}$
  - $\mathbb{P}(\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} ))) = \mathbb{P}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} ))$
  - unique extension of $\mathbb{P}$ to the generated $\sigma$-algebra
The $\frac{1}{2}$-player game model

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

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

- Easy extension to constrained symbolic paths

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

- Definition over sets of infinite runs:
  - $\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} )) = \{o \cdot o' \mid o \in \pi_c(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} )\}$
  - $\mathbb{P}(\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} ))) = \mathbb{P}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n} ))$
  - unique extension of $\mathbb{P}$ to the generated $\sigma$-algebra

- Property: $\mathbb{P}$ is a probability measure over sets of infinite runs
The $\frac{1}{2}$-player game model

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

- Can be viewed as an $n$-dimensional integral
- Easy extension to constrained symbolic paths
  $$\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \{s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models C\}$$
- Definition over sets of infinite runs:
  - $\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{q \cdot q' \mid q \in \pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\}$
  - $\mathbb{P}(\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))) = \mathbb{P}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))$
  - unique extension of $\mathbb{P}$ to the generated $\sigma$-algebra
- Property: $\mathbb{P}$ is a probability measure over sets of infinite runs
- Example:
  $$\text{Zeno}(s) = \bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1, \cdots, e_n) \in E^n} \text{Cyl}(\pi_{\sum_i \tau_i \leq M}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))$$
An example of computation (with uniform distributions)

The probability of the symbolic path $\pi(s_0 \xrightarrow{e_1} e_2)$ is $\frac{1}{4}$. 
An example of computation (with uniform distributions)

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

$$\mathbb{P}\left( \pi(s_0 \xrightarrow{e_1} e_2) \right) = \int_0^1 \mathbb{P}\left( \pi(s_1 \xrightarrow{e_2}) \right) \, d\mu_{s_0}(t) + \int_1^1 \frac{\mathbb{P}\left( \pi(s_1 \xrightarrow{e_2}) \right)}{2} \, d\mu_{s_0}(t)$$
An example of computation (with uniform distributions)

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

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

$$= \int_0^1 \int_0^1 \left( \frac{P(\pi(s_2))}{2} d\mu_{s_1}(u) \right) d\mu_{s_0}(t)$$
An example of computation (with uniform distributions)

A hint into stochastic timed games

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

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

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

$$= \int_0^1 \int_0^1 \left(\frac{1 \cdot du}{2} \cdot \frac{du}{2}\right) dt = \frac{1}{4}$$
An example of computation (with exponential distrib.)

The probability of the symbolic path $\pi(s_0 \xrightarrow{e_1} e_2)$ is $e^{-3} - e^{-5} \approx 0.043$
An example of computation (with exponential distrib.)

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

$$\mathbb{P}\left(\pi(s_0 \xrightarrow{e_1} e_2)\right) = \int_0^1 \mathbb{P}\left(\pi(s_1 \xrightarrow{e_2})\right) d\mu_{s_0}(t) = \int_0^1 \mathbb{P}\left(\pi(s_1 \xrightarrow{e_2})\right) e^{-2t} dt$$

$$= \int_0^1 \left( \int_1^{+\infty} 3 e^{-3u} du \right) e^{-2t} dt$$

$$= \left[ - e^{-2t} \right]_{t=0}^1 \cdot \left[ - e^{-3u} \right]_{u=1}^{+\infty}$$

$$= (1 - e^{-2}) \cdot e^{-3} = e^{-3} - e^{-5}$$
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:

\[
\mathbb{P}\left(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\right) = \int_{t \in l(s, e_1)} p_{s+t}(e_1) \mathbb{P}\left(\pi(s_t \xrightarrow{e_2} \cdots \xrightarrow{e_n})\right) \, d\mu_s(t)
\]

mass distribution given by the strategy

if $s$ is a player vertex
A hint into stochastic timed games

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:

\[
P(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \int_{t \in I(s, e_1)} p_{s+t}(e_1) P(\pi(s \xrightarrow{e_2} \cdots \xrightarrow{e_n})) \, d\mu_s(t)
\]

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 $\varphi$, a rational threshold $\triangleright r$, is there a strategy $f_\diamond$ for player $\Diamond$ s.t. for all strategies $f_\square$ of player $\Box$, $\mathbb{P}(G_{f_\diamond}, f_\square \models \varphi) \triangleright r$?
Reachability problem – Example

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.

- Uniform distribution over delays
- Uniform distribution over edges
Reachability problem – Example

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

- Uniform distribution over delays
- Uniform distribution over edges
Reachability problem – Example

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

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

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

- Yes: it is the case when \(\Diamond\) 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: ♣️ ○
- $\frac{1}{2}$-player games: ○

(“Markov decision process”)

(“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 $\ominus u_1D714$-regular condition, or some LTL property, or some more restrictive condition like a reachability condition.
## What kind of games will we play?

### Number of players

- **2₁²-player games:**
  - “Markov decision process”

- **1₁²-player games:**
  - “Markov chain”

- **1₂-player games:**

### 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:
  - Markov decision process
- $1\frac{1}{2}$-player games:
  - Markov chain
- $\frac{1}{2}$-player games:

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 $\omega$-regular condition, or some LTL property, or some more restrictive condition like a reachability condition.
A hint into stochastic timed games

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Qualitative</th>
<th>Quantitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>decidable [BBB+08]</td>
</tr>
<tr>
<td></td>
<td>$n$ clocks</td>
<td>decidable?$^2$</td>
</tr>
<tr>
<td>$1\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>decidable$^3$ [BF09]</td>
</tr>
<tr>
<td></td>
<td>$n$ clocks</td>
<td>?</td>
</tr>
<tr>
<td>$2\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>$n$ clocks</td>
<td>?</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Qualitative</th>
<th>Quantitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>decidable $^1$ [BBB+08]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>decidable $^1$ [BBBM08]</td>
</tr>
<tr>
<td>$n$ clocks</td>
<td>decidable? $^2$</td>
<td>?</td>
</tr>
<tr>
<td>$1\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>decidable $^3$ [BF09]</td>
</tr>
<tr>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>$n$ clocks</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>$2\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>$n$ clocks</td>
<td>?</td>
<td>undecidable $^4$ [BF09]</td>
</tr>
</tbody>
</table>

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

---

$^1$ Baier, Bertrand, Bouyer, Brihaye, Größer. Almost-sure model checking of infinite paths in one-clock timed automata ($LICS’08$).
$^2$ Bertrand, Bouyer, Brihaye, Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics ($QEST’08$).
$^3$ 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

<table>
<thead>
<tr>
<th>Model</th>
<th>Qualitative</th>
<th>Quantitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{1}{2})-player game</td>
<td>1 clock</td>
<td>(\text{decidable} [\text{BBB+08}])</td>
</tr>
<tr>
<td></td>
<td>(n \text{ clocks})</td>
<td>(\text{decidable?})</td>
</tr>
<tr>
<td>(1\frac{1}{2})-player game</td>
<td>1 clock</td>
<td>(\text{decidable} [\text{BF09}])</td>
</tr>
<tr>
<td></td>
<td>(n \text{ clocks})</td>
<td>?</td>
</tr>
<tr>
<td>(2\frac{1}{2})-player game</td>
<td>1 clock</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>(n \text{ clocks})</td>
<td>?</td>
</tr>
</tbody>
</table>

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


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

\[ s \models \varphi \iff \mathbb{P}(\{ \sigma \in \text{Runs}(s) \mid \sigma \models \varphi \}) = 1 \]
Almost-sure model-checking

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

\[ s \models \varphi \iff P \left( \{ \rho \in \text{Runs}(s) \mid \rho \models \varphi \} \right) = 1 \]

There are only \( \bigcirc \) vertices, but we will use extra colors to represent atomic propositions.
An example
An example

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

\[
A \not\models G(\text{green }\Rightarrow \text{ F red}) \quad \text{but} \quad \mathbb{P}(A \models G(\text{green }\Rightarrow \text{ F red})) = 1
\]
An example

\[ A \not\models G(\text{green } \Rightarrow \text{ F red}) \quad \text{but} \quad \mathbb{P}(A \models G(\text{green } \Rightarrow \text{ F red})) = 1 \]

Indeed, almost surely, paths are of the form \( e_1^* e_2 (e_4 e_5)^\omega \)
The classical region automaton
The pruned region automaton

A hint into stochastic timed games
The pruned region automaton
The pruned region automaton

... viewed as a finite Markov chain $MC(A)$
The pruned region automaton

... viewed as a finite Markov chain \( MC(A) \)

**Proposition**

For single-clock timed automata,

\[
P(A \models \varphi) = 1 \quad \text{iff} \quad P(MC(A) \models \varphi) = 1
\]

(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 $\omega$-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 $\omega$-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).
Theorem [BBB+08]

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

- of LTL is PSPACE-Complete
- of \( \omega \)-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 $\omega$-regular properties is NLOGSPACE-Complete

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

- Correctness: the proof is rather involved
Result

Theorem [BBB+08]

For single-clock timed automata, the almost-sure model-checking
- of LTL is PSPACE-Complete
- of $\omega$-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 $\omega$-regular properties is NLOGSPACE-Complete

Complexity:
- size of single-clock region automata $\equiv$ 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

If the previous algorithm was correct, \[ \mu \in \Delta^4 \exists C \mid \approx \]
However, we can prove that \[ \mathbb{P} \in G \neg \mathit{red} \neq 0 \]
There is a strange convergence phenomenon: along an execution, if \( \mu_1 \) is the delay in locations \( \ell_2 \) or \( \ell_4 \), then we have that \( P \mu_1 \leq 1 - \frac{43}{60} \)
An example with two clocks

If the previous algorithm was correct, \( \mathcal{A} \models G F \text{ red} \land G F \text{ green} \)
An example with two clocks

If the previous algorithm was correct, $\mathcal{A} \models GF \text{ red} \land GF \text{ green}$

However, we can prove that $P(G \neg\text{red}) > 0$
An example with two clocks

- If the previous algorithm was correct, $\mathcal{A} \models GF \text{ red} \land GF \text{ green}$
- However, we can prove that $P(G \neg\text{red}) > 0$
- There is a strange convergence phenomenon: along an execution, if $\delta_i > 0$ is the delay in locations $\ell_2$ or $\ell_4$, then we have that $\sum_i \delta_i \leq 1$
A note on Zeno behaviours

- The set of Zeno behaviours is measurable:

\[ Zeno(s) = \bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1, \ldots, e_n) \in E^n} \text{Cyl}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) \]
A note on Zeno behaviours

- The set of Zeno behaviours is measurable:
  \[ \text{Zeno}(s) = \bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1, \ldots, e_n) \in E^n} \text{Cyl}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) \]

- In single-clock timed automata, we can decide in NLOGSPACE whether \( \mathbb{P}(\text{Zeno}(s)) = 0 \):
A note on Zeno behaviours

- The set of Zeno behaviours is measurable:

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

- In single-clock timed automata, we can decide in NLOGSPACE whether \( P(\text{Zeno}(s)) = 0 \):
  - check whether there is a purely Zeno BSCC in \( MC(A) \)
A note on Zeno behaviours

- The set of Zeno behaviours is measurable:
  \[ Zeno(s) = \bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1, \ldots, e_n) \in E^n} \text{Cyl}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) \]

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

- an interesting notion of non-Zeno timed automata
  \[ x \leq 1, \ x := 0 \]
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

<table>
<thead>
<tr>
<th>Model</th>
<th>Qualitative</th>
<th>Quantitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/2-player game</td>
<td>1 clock</td>
<td>decidable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[BBB+08]</td>
</tr>
<tr>
<td></td>
<td>n clocks</td>
<td>decidable?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>1 1/2-player game</td>
<td>1 clock</td>
<td>decidable</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[BF09]</td>
</tr>
<tr>
<td></td>
<td>n clocks</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>2 1/2-player game</td>
<td>1 clock</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>n clocks</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td></td>
<td>undecided</td>
</tr>
<tr>
<td></td>
<td></td>
<td>[BF09]</td>
</tr>
</tbody>
</table>

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


Undecidability

Theorem [BF09]

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

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 $\frac{1}{2}$.

Undecidability

Theorem [BF09]

The reachability problem for stochastic timed games (2½ players) is undecidable.

- Holds for uniform and exponential distributions on delays.
- Holds for any quantitative question; we give hints for proba = ½.
- Proof by reduction from halting problem of two-counter machine to the reachability with probability precisely ½:
  - ♦ simulates a computation of the two-counter machine and encodes counter values in clock values
  - ♦ stores counter values $c_1$ and $c_2$ as $\frac{1}{2^{c_1}3^{c_2}}$
  - ✘ will check that ♦ is not cheating using the power of the probabilities

Undecidability – Comparing counter values

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

![Diagram showing states and transitions involving clock values $x$ and $y$ with conditions for reaching states $a$, $b$, $c$, and $d$.]

- Entered with $x = x_0$ and $y = y_0$
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:
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:

$$\frac{1}{2} \cdot \frac{1}{2} \cdot \int_{t=1-y_0}^{1} dt$$
Undecidability – Comparing counter values

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

```
\begin{align*}
\text{entered with } &\left\{ x = x_0 \text{ and } y = y_0 \right\} \\
\text{vertex } &d \text{ is reached with probability: } \\
\frac{1}{2} \cdot \frac{1}{2} &\cdot \int_{t=1-y_0}^{1} \frac{1}{t^2} \cdot \frac{1}{4} \cdot y_0
\end{align*}
```
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

$$\frac{1}{2} = \frac{1}{4} \cdot y_0 + \frac{1}{2} \cdot (1 - x_0) \quad \text{iff} \quad x_0 = 2y_0$$
How do we properly increment the first counter?

enters with $x = \frac{1}{2c_1 3c_2}$

should leave with $y = \frac{1}{2c_1 + 13c_2}$
Undecidability – Zero test

How do we check that \( c_1 \) is zero?

Player \( \Diamond \) has a strategy to reach \( \bigcirc \) with proba. \( \frac{1}{2} \) iff \( c_1 \) is initially zero.
Undecidability – Zero test

How do we check that $c_1$ is zero?

Player $\Diamond$ has a strategy to reach $\smiley$ with proba. $\frac{1}{2}$ iff $c_1$ is initially zero.
Undecidability – Zero test

How do we check that $c_1$ is zero?

Player $\blacklozenge$ has a strategy to reach $\smiley$ 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

<table>
<thead>
<tr>
<th>Model</th>
<th>Qualitative</th>
<th>Quantitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{1}{2} )-player game</td>
<td>1 clock [decidable] [BBB+08]</td>
<td>[decidable] [BBBM08]</td>
</tr>
<tr>
<td></td>
<td>( n ) clocks</td>
<td>decidable?</td>
</tr>
<tr>
<td>( 1 \frac{1}{2} )-player game</td>
<td>1 clock [decidable] [BF09]</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>( n ) clocks</td>
<td>?</td>
</tr>
<tr>
<td>( 2 \frac{1}{2} )-player game</td>
<td>1 clock</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>( n ) clocks</td>
<td>?</td>
</tr>
</tbody>
</table>

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


Towards quantitative analysis

- The abstraction $MC(A)$ is no more correct.
Towards quantitative analysis

- The abstraction $MC(A)$ 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.
  - ~ hard to solve in general, even for simple distributions
Towards quantitative analysis

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

- The abstraction $MC(A)$ 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 $M_C(A)$ is no more correct.
- Can be reduced to solving a system of differential equations.
  - $\sim$ 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.
  - 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 $p^-_\varepsilon$ and $p^+\varepsilon$ such that:

$$
\begin{align*}
  p^-_\varepsilon & \leq \mathbb{P}(s_0 \models \varphi) \leq p^-_\varepsilon + \varepsilon \\
  p^+\varepsilon - \varepsilon & \leq \mathbb{P}(s_0 \models \varphi) \leq p^+\varepsilon
\end{align*}
$$
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 $p^-_\varepsilon$ and $p^+_\varepsilon$ such that:

  \[
  \begin{cases}
  p^-_\varepsilon \leq \mathbb{P}(s_0 \models \varphi) \leq p^-_\varepsilon + \varepsilon \\
  p^+_\varepsilon - \varepsilon \leq \mathbb{P}(s_0 \models \varphi) \leq p^+_\varepsilon
  \end{cases}
  \]

  - 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.
  \[ \leadsto \text{hard to solve in general, even for simple distributions} \]

- 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 $p_{\varepsilon}^-$ and $p_{\varepsilon}^+$ such that:
    \[
    \begin{align*}
    p_{\varepsilon}^- & \leq P(s_0 \models \varphi) \leq p_{\varepsilon}^- + \varepsilon \\
    p_{\varepsilon}^+ - \varepsilon & \leq P(s_0 \models \varphi) \leq p_{\varepsilon}^+
    \end{align*}
    \]
  - we will be able to decide the threshold problem:
    “Given $\mathcal{A}$, $\varphi$, $c \in \mathbb{Q}$, and $\sim \in \{<,\leq,=,\geq,>\}$, does $P(s_0 \models \varphi) \sim c$ in $\mathcal{A}$?”
An example

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

- $e_0, x \leq 1$, $x:=0$
- $e_1, x \leq 1$, $x:=0$
- $e_2, x \leq 1$
- $e_3, x \leq 2$, $x:=0$
- $e_4, x \geq 2$, $x:=0$
- $e_5, x \leq 2$
- $e_6, x \geq 2$, $x:=0$
- $e_7, x \geq 2$

± distributions

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

We construct a finite Markov chain $MC'(A)$ with macro-edges:
An example

\[ \begin{align*}
\ell_0, \ x \leq 1, \ x := 0 & \quad e_2, \ x \leq 1 \\
\ell_1 & \quad e_4, \ x \geq 2, \ x := 0 \\
\ell_2, \ x \leq 2 & \quad e_5, \ x \leq 2 \\
\ell_3, \ x \leq 2 & \quad e_6, \ x := 0 \\
e_7 &
\end{align*} \]

\[ \begin{align*}
+ \text{ distributions} & \quad \mu_s : t \mapsto e^{-t} \text{ when } I(s) = \mathbb{R}_+ \\
& \quad \mu_s \text{ uniform distribution when } I(s) \text{ is bounded} \\
& \quad \text{uniform weights on transitions}
\end{align*} \]

We construct a finite Markov chain \( MC'(A) \) with macro-edges:
An example

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

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

We construct a finite Markov chain $MC'(A)$ with macro-edges:
An example

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

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

\[ e_1, \ x \leq 1, \ x := 0 \]

\[ e_2, \ x \leq 1 \]

\[ e_3, \ x \leq 2, \ x := 0 \]

\[ e_4, \ x \geq 2, \ x := 0 \]

\[ e_5, \ x \leq 2 \]

\[ e_6, \ x = 0 \]

\[ e_7 \]

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

\( \mu_s \) 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

\[ \begin{align*}
\ell_0, \ x \leq 1, \ x := 0, \\
\ell_1, \ x \leq 1, \ x := 0, \\
\ell_2, \ x \leq 1, \ x := 0, \\
\ell_3, \ x := 0
\end{align*} \]

\[ \begin{align*}
e_2, \ x \leq 1, \\
e_4, \ x \geq 2, \ x := 0, \\
e_6, \ x := 0
\end{align*} \]

\[ \begin{align*}
e_3, \ x \leq 2, \ x := 0, \\
e_5, \ x \leq 2, \\
e_7
\end{align*} \]

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

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

\[ \begin{align*}
\frac{1}{2} \cdot e^{-2} \\
\frac{1}{2} \cdot (1 + e^{-2}) \\
\frac{1}{2} \cdot (1 - e^{-2})
\end{align*} \]
An example

\[ e_1, \ x \leq 1, \ x := 0 \]
\[ e_2, \ x \leq 1 \]
\[ e_3, \ x \leq 2, \ x := 0 \]
\[ e_4, \ x \geq 2, \ x := 0 \]
\[ e_5, \ x \leq 2 \]
\[ e_6, \ x = 0 \]
\[ e_7 \]

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

We construct a finite Markov chain $MC'(A)$ with macro-edges:
An example

\[ \ell_0, \ x \leq 1, \ x:=0 \]

\[ \ell_1, \ x \leq 1, \ e_2, \ x \leq 1 \]

\[ \ell_2, \ x \leq 2, \ e_5, \ x \leq 2 \]

\[ \ell_3, \ x = 0 \]

\[ e_6, \ x = 0 \]

\[ e_7 \]

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

\[ \mu_s \text{ uniform distribution when } I(s) \text{ is bounded} \]

+ uniform weights on transitions

We construct a finite Markov chain \( MC'(A) \) with macro-edges:
An example

We construct a finite Markov chain $MC'(A)$ with macro-edges:
Correctness of the abstraction

Theorem

Under some hypotheses, for single-clock automaton $A$ and property $\varphi$,

$$P_A(s_0 \models \varphi) = P_{MC}(s_0 \models \Diamond F_\varphi)$$

for some well-chosen set $F_\varphi$. 
Correctness of the abstraction

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

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

for some well-chosen set $F_{\varphi}$.

**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 $\varphi$,

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

for some well-chosen set $F_\varphi$.

- **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 $\ell$, 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 $\ell$, the distribution over delays has density
    $t \mapsto \lambda_\ell \cdot e^{-\lambda_\ell \cdot t}$ for some $\lambda_\ell \in \mathbb{Q}_+$
  - more general than continuous-time Markov chains
Computing the probability

- We assume furthermore that:
  - for every state $s$, $l(s) = \mathbb{R}_+$
    (the timed automaton is ‘reactive’)
  - in every location $\ell$, the distribution over delays has density
    $t \mapsto \lambda_\ell \cdot e^{-\lambda_\ell \cdot t}$ for some $\lambda_\ell \in \mathbb{Q}_+$
      $\leadsto$ more general than continuous-time Markov chains

Proposition

Under those hypotheses, $P(s_0 \models \varphi)$ can be expressed as $f(e^{-r})$ 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 \( \ell \), the distribution over delays has density
    \[ t \mapsto \lambda_{\ell} \cdot e^{-\lambda_{\ell} \cdot t} \]
    for some \( \lambda_{\ell} \in \mathbb{Q}_+ \)
    \[ \leadsto \text{more general than continuous-time Markov chains} \]

Proposition

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

\[ \leadsto \text{Note: the hypothesis “reset all bounded cycles” is necessary to get this form.} \]
Approximating the probability

\[ \mathbb{P}(s_0 \models \varphi) = f(e^{-r}) \]
Approximating the probability

\[ \mathbb{P}(s_0 \models \varphi) = f(e^{-r}) \]

- 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

\[ \mathbb{P}(s_0 \models \varphi) = f(e^{-r}) \]

- 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

\[ \mathbb{P}(s_0 \models \varphi) = f(e^{-r}) \]

- We can compute sequences \((a_i)\) and \((b_i)\) 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

\[ \mathbb{P}(s_0 \models \varphi) = f(e^{-r}) \]

- 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 \(f = \frac{P}{Q}\), 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 \((\alpha, \beta)\) (that we can compute)
Approximating the probability

\[ P(s_0 \models \varphi) = f(e^{-r}) \]

- 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 \((\alpha, \beta)\) (that we can compute)

If the sign of \(R'\) is constant over \((\alpha', \beta')\) (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

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

- 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 $(\alpha, \beta)$ (that we can compute)
    - If the sign of $R'$ is constant over $(\alpha', \beta')$ (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 $f(e^{-r})$
Deciding the threshold problem

**Theorem [BBBM08]**

Under the previous hypotheses, the threshold problem is decidable.

Deciding the threshold problem

Theorem [BBBM08]

Under the previous hypotheses, the threshold problem is decidable.

- Check whether $c = f(e^{-r})$
Deciding the threshold problem

**Theorem** [BBBM08]

Under the previous hypotheses, the threshold problem is decidable.

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

---

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

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

<table>
<thead>
<tr>
<th>Model</th>
<th>Qualitative</th>
<th>Quantitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\frac{1}{2})-player game</td>
<td>1 clock</td>
<td>decidable ([\text{BBB}+08])</td>
</tr>
<tr>
<td></td>
<td>(n) clocks</td>
<td>decidable(^2)</td>
</tr>
<tr>
<td>(1\frac{1}{2})-player game</td>
<td>1 clock</td>
<td>decidable(^3) ([\text{BF09}])</td>
</tr>
<tr>
<td></td>
<td>(n) clocks</td>
<td>?</td>
</tr>
<tr>
<td>2(\frac{1}{2})-player game</td>
<td>1 clock</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>(n) clocks</td>
<td>?</td>
</tr>
</tbody>
</table>
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!
  - even for simple untimed objectives, few is known
  - nothing about more involved quantitative objectives (e.g. expected time, timed properties, ...)

<table>
<thead>
<tr>
<th>Model</th>
<th>Qualitative</th>
<th>Quantitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>decidable [BBB+08]</td>
</tr>
<tr>
<td></td>
<td>$n$ clocks</td>
<td>decidable$^2$</td>
</tr>
<tr>
<td>$1\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>decidable$^3$ [BF09]</td>
</tr>
<tr>
<td></td>
<td>$n$ clocks</td>
<td>?</td>
</tr>
<tr>
<td>$2\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>?</td>
</tr>
<tr>
<td></td>
<td>$n$ clocks</td>
<td>?</td>
</tr>
</tbody>
</table>
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!
  - even for simple untimed objectives, few is known
  - nothing about more involved quantitative objectives (e.g. expected time, timed properties, ...)
  - what about approximate probabilities?

<table>
<thead>
<tr>
<th>Model</th>
<th>Qualitative</th>
<th>Quantitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \frac{1}{2} )-player game</td>
<td>1 clock</td>
<td>decidable \cite{BBB+2009}</td>
</tr>
<tr>
<td>( n ) clocks</td>
<td>decidable\textsuperscript{2}</td>
<td>?</td>
</tr>
<tr>
<td>1( \frac{1}{2} )-player game</td>
<td>1 clock</td>
<td>decidable\textsuperscript{3} \cite{BF2009}</td>
</tr>
<tr>
<td>( n ) clocks</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>2( \frac{1}{2} )-player game</td>
<td>1 clock</td>
<td>?</td>
</tr>
<tr>
<td>( n ) clocks</td>
<td>?</td>
<td>undecidable\textsuperscript{4} \cite{BF2009}</td>
</tr>
</tbody>
</table>
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!**
  - 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

<table>
<thead>
<tr>
<th>Model</th>
<th>Qualitative</th>
<th>Quantitative</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>decidable \cite{BBB+08}</td>
</tr>
<tr>
<td>$n$ clocks</td>
<td>decidable$^2$</td>
<td>?</td>
</tr>
<tr>
<td>$1\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>decidable$^3$ \cite{BF09}</td>
</tr>
<tr>
<td>$n$ clocks</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>$2\frac{1}{2}$-player game</td>
<td>1 clock</td>
<td>?</td>
</tr>
<tr>
<td>$n$ clocks</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>
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!**
  - 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