A Probabilistic Semantics for Timed Automata

Christel Baier¹, Nathalie Bertrand², Patricia Bouyer³
Thomas Brihaye⁴, Marcus Größer¹

¹Technische Universität Dresden – Germany
²IRISA/INRIA Rennes – France
³LSV – CNRS & ENS Cachan – France
⁴Université de Mons-Hainaut – Belgium
Motivation(s)

- Timed automata, an idealized mathematical model for real-time systems
Motivation(s)

- Timed automata, an idealized mathematical model for real-time systems
  - assumes infinite precision of clocks
  - assumes instantaneous actions
  - etc...
Motivation(s)

- Timed automata, an idealized mathematical model for real-time systems
  - assumes infinite precision of clocks
  - assumes instantaneous actions
  - *etc*...

→ notion of strong robustness defined in [DDR04]
Motivation(s)

- Timed automata, an idealized mathematical model for real-time systems
  - assumes infinite precision of clocks
  - assumes instantaneous actions
  - etc...
  - notion of strong robustness defined in [DDR04]

- In a model, only few traces may violate the correctness property: they may hence not be relevant...
Motivation(s)

- Timed automata, an idealized mathematical model for real-time systems
  - assumes infinite precision of clocks
  - assumes instantaneous actions
  - etc...

  ➔ notion of strong robustness defined in [DDR04]

- In a model, only few traces may violate the correctness property: they may hence not be relevant...

  ➔ topological notion of tube acceptance in [GHJ97]
Motivation(s)

- Timed automata, an idealized mathematical model for real-time systems
  - assumes infinite precision of clocks
  - assumes instantaneous actions
  - etc...
  
  ➔ notion of strong robustness defined in [DDR04]

- In a model, only few traces may violate the correctness property: they may hence not be relevant...
  
  ➔ topological notion of tube acceptance in [GHJ97]
  
  ➔ notion of fair correctness in [VV06] based on probabilities (for untimed systems) + topological characterization
Motivation(s)

- Timed automata, an idealized mathematical model for real-time systems
  - assumes infinite precision of clocks
  - assumes instantaneous actions
  - etc...

  ➔ notion of strong robustness defined in [DDR04]

- In a model, only few traces may violate the correctness property: they may hence not be relevant...

  ➔ topological notion of tube acceptance in [GHJ97]

  ➔ notion of fair correctness in [VV06] based on probabilities (for untimed systems) + topological characterization

Aim: Use probabilities to “relax” the semantics of timed automata
Initial example

Intuition: from the initial state,
this automaton almost-surely satisfies “G green”
A maybe less intuitive example

Does it *almost-surely* satisfy “$F_{\text{red}}$”?
Our proposition

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

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

Example:

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

- \(\pi(s \xrightarrow{e_1} \ldots \xrightarrow{e_n})\): symbolic path from state \(s\) 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:

From state \(s_0\):
Our proposition

- \( \pi(s \xrightarrow{e_1} \ldots \xrightarrow{e_n}) \): 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:

From state \( s_0 \):
  - randomly choose a delay
Our proposition

- \( \pi(s \xrightarrow{e_1} \ldots \xrightarrow{e_n}) \): 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:

From state \( s_0 \):
- randomly choose a delay
- then randomly select an edge
Our proposition

- \( \pi(s \xrightarrow{e_1} \ldots \xrightarrow{e_n}) \): 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:

From state \( s_0 \):

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

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

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

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

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

- \( l(s, e_1) = \{\tau \mid s \xrightarrow{\tau, e_1} \} \) and \( \mu_s \) distrib. over \( l(s) = \bigcup_e l(s, e) \)
Our proposition

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

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

$\triangleright I(s, e_1) = \{\tau \mid s \xrightarrow{\tau,e_1}\}$ and $\mu_s$ distrib. over $I(s) = \bigcup_e I(s, e)$
Our proposition

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 $\mu_s$ distrib. over $I(s) = \bigcup_e I(s,e)$
- $p_{s+t}$ distrib. over transitions enabled in $s + t$
Our proposition

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 \( \mu_s \) distrib. over \( I(s) = \bigcup_e I(s, e) \)
- \( p_{s+t} \) distrib. over transitions enabled in \( s + t \)
- \( s \xrightarrow{t} s + t \xrightarrow{e_1} s_t \)
Our proposition

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

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

- Can be viewed as an \( n \)-dimensional integral
Our proposition

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

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

\[ \pi_C\left( s \xrightarrow{e_1} \cdots \xrightarrow{e_n} \right) = \{ s \xrightarrow{\tau_1, e_1} s_1 \cdots \xrightarrow{\tau_n, e_n} s_n \mid (\tau_1, \cdots, \tau_n) \models C \} \]
Our proposition

\[ P(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) = \int_{t \in I(s, e_1)} p_{s+t}(e_1) P(\pi(s_1 \xrightarrow{e_2} \cdots \xrightarrow{e_n})) \, d\mu_s(t) \]

- 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:
Our proposition

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

- 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:
  - \(\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{\varrho \cdot \varrho' \mid \varrho \in \pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\}\)
Our proposition

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

- 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:
  - \( \text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{ \varphi \cdot \varphi' \mid \varphi \in \pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) \} \)
  - \( P(\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))) = P(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) \)
Our proposition

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

- 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:
  - \( \text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{ \varrho \cdot \varrho' \mid \varrho \in \pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) \} \)
  - \( P(\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))) = P(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) \)
  - unique extension of \( P \) to the generated \( \sigma \)-algebra
Our proposition

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

- 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:
  - \( \text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{\varrho \cdot \varrho' \mid \varrho \in \pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})\} \)
  - \( P(\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))) = P(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) \)
  - unique extension of \( P \) to the generated \( \sigma \)-algebra
- Property: \( P \) is a probability measure over sets of infinite runs
Our proposition

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

- 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:
  - \( \text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) = \{ \varrho \cdot \varrho' \mid \varrho \in \pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}) \} \)
  - \( \mathbb{P}(\text{Cyl}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))) = \mathbb{P}(\pi_C(s \xrightarrow{e_1} \cdots \xrightarrow{e_n})) \)
  - unique extension of \( \mathbb{P} \) to the generated \( \sigma \)-algebra
- Property: \( \mathbb{P} \) is a probability measure over sets of infinite runs
- Example:
  \[
  \text{Zeno}(s) = \bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1, \cdots, e_n) \in E^n} \text{Cyl}(\pi_{\Sigma_i \tau_i \leq M}(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))
  \]
An example of computation (with uniform distributions)

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

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

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

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

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

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

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

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

\[
= \int_0^1 \int_0^1 \left( \frac{1}{2} \frac{du}{2} \right) \, dt = \frac{1}{4}
\]
Back to the first example

\begin{equation*}
  x \leq 1 \\
  e_1, \quad x \leq 1 \\
  x \leq 1 \\
  e_2, \quad x \leq 1 \\
  e_3, \quad x = 1
\end{equation*}
Back to the first example

\[ \mathbb{P}(\pi(s_0 \xrightarrow{e_1} e_2)) = 1 \]
Back to the first example

\[
\begin{align*}
\mathbb{P}(s_0 \xrightarrow{e_1} e_2) &= 1 \\
\mathbb{P}(s_0 \xrightarrow{e_1} e_3) &= 0
\end{align*}
\]
Back to the first example

\[ x \leq 1 \]

\[ e_1, \ x \leq 1 \]

\[ x \leq 1 \]

\[ e_2, \ x \leq 1 \]

\[ e_3, \ x = 1 \]

- \[ \mathbb{P}(\pi(s_0 \xrightarrow{e_1} e_2)) = 1 \]
- \[ \mathbb{P}(\pi(s_0 \xrightarrow{e_1} e_3)) = 0 \]
- \[ \mathbb{P}(G \text{ green}) = 1 \]
Back to the second example

\[ x \leq 1 \]

\[ e_1, \ x \leq 1 \]

\[ x \leq 1 \]

\[ e_2, \ x = 0 \]

\[ e_3, \ x = 1 \]
Back to the second example

\[ P(\pi(s_0 \xrightarrow{e_1} e_2)) = 0 \]
Back to the second example

\[ x \leq 1 \]

- \[ \mathbb{P}(\pi(s_0 \xrightarrow{e_1} e_2)) = 0 \]
- \[ \mathbb{P}(\pi(s_0 \xrightarrow{e_1} e_3)) = 1 \]
Back to the second example

\[ x \leq 1, \quad e_1, \ x \leq 1 \]

\[ x \leq 1 \]

\[ e_2, \ x = 0 \]

\[ e_3, \ x = 1 \]

\[ \mathbb{P}(\pi(s_0 \xrightarrow{e_1} e_2)) = 0 \]

\[ \mathbb{P}(\pi(s_0 \xrightarrow{e_1} e_3)) = 1 \]

\[ \mathbb{P}(F \text{ red}) = 1 \]
Almost-sure model-checking

If \( \varphi \) is an LTL formula,

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

(This definition extends naturally to CTL\(^\star\) specifications...)
Almost-sure model-checking

If \( \varphi \) is an LTL formula,

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

(This definition extends naturally to CTL* specifications...)

Almost-sure model-checking

If $\varphi$ is an LTL formula,

$$s \models \varphi \iff \mathbb{P}(\{\rho \in \text{Runs}(s) \mid \rho \models \varphi\}) = 1$$

(This definition extends naturally to CTL$^*$ specifications...)

We want to decide the almost-sure model-checking...
(This is a qualitative question)
An example
An example

\[ \ell_0, x \leq 1 \]

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

\[ \ell_1, e_3, x = 1 \]

\[ \ell_2, e_4, x \geq 3, x := 0 \]

\[ \ell_2, e_5, x \leq 1 \]

\[ \ell_3, e_6, x = 0 \]

\[ e_7, x \leq 1 \]

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

\[ \ell_0, x \leq 1 \]
\[ \ell_1, x \leq 1 \]
\[ \ell_2, x \geq 3, x = 0 \]
\[ \ell_3, x \leq 1 \]

\[ A \not\models G(\text{green} \Rightarrow F \text{ red}) \]
\[ \text{but} \]
\[ A \models G(\text{green} \Rightarrow F \text{ red}) \]
An example

\[ \ell_0, x \leq 1 \]
\[ \ell_1, x \leq 1 \]
\[ \ell_2, x \leq 1 \]
\[ \ell_3, x \leq 1 \]

\[ e_1, x \leq 1 \]
\[ e_2, x \leq 1 \]
\[ e_3, x = 1 \]
\[ e_4, x \geq 3, x = 0 \]
\[ e_5, x \leq 1 \]
\[ e_6, x = 0 \]
\[ e_7, x \leq 1 \]

\[ \mathcal{A} \not\models \mathbf{G}(\text{green } \Rightarrow \text{ F red}) \quad \text{but} \quad \mathcal{A} \models \mathbf{G}(\text{green } \Rightarrow \text{ F red}) \]

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

... viewed as a finite Markov chain

For single-clock timed automata, $A \mid \approx \phi$ iff $\mathbb{P}(MC(A) \mid = \phi) = 1$
The pruned region automaton

Theorem
For single-clock timed automata, $A | \approx \phi$ iff $P(MC(A) | = \phi) = 1$
The pruned region automaton

\[
\ell_0,0 \\
\ell_0,(0,1) \\
\ell_1,(0,1) \\
\ell_2,0
\]

\[
e_1 \\
e_2 \\
\Delta \\
e_5
\]

Theorem
For single-clock timed automata, \( A | \approx \varphi \) if \( P(MC(A)|=\varphi) = 1 \)

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

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

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

**Theorem**

For single-clock timed automata,

$$A \models \varphi \iff P(MC(A) \models \varphi) = 1$$
Theorem

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

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

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

Complexity:
Result

Theorem

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]
Theorem

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
Result

**Theorem**

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
**Theorem**

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
Theorem

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
  - notions of largeness (for proba 1) and meagerness (for proba 0)
Theorem

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
An example with two clocks

If the previous algorithm was correct, $A \mid \approx GF_{\text{red}} \land GF_{\text{green}}$.

However, we can prove that $P \in G \neg red > 0$.

There is a strange convergence phenomenon: along an execution, if $\delta_i > 0$ is the delay in location $\ell_4$, then we have that $P_{t+\delta_i} \leq 1$. 

$y < 1$
An example with two clocks

If the previous algorithm was correct, \( \mathcal{A} \models GF \text{ red} \land GF \text{ green} \)
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 $\mathbb{P}(G \neg\text{red}) > 0$
An example with two clocks

- If the previous algorithm was correct, $\mathcal{A} \models G \neg F \text{ red} \land G \neg F \text{ green}$
- 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 location $\ell_4$, then we have that $\sum_i \delta_i \leq 1$
A note on Zeno behaviours

- The set of Zeno behaviours is measurable:

\[
\text{Zeno}(s) = \bigcup_{M \in \mathbb{N}} \bigcap_{n \in \mathbb{N}} \bigcup_{(e_1, \ldots, e_n) \in E^n} \text{Cyl}(\pi(s \xrightarrow{e_1} \cdots \xrightarrow{e_n}))
\]
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 \( \mathbb{P}(\text{Zeno}(s)) = 0 \):
  - check whether there is a purely Zeno BSCC in \( MC(A) \)
A note on Zeno behaviours

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

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

- an interesting notion of non-Zeno timed automata

\[
\begin{align*}
x \leq 1, \ x := 0
\end{align*}
\]
Related works

- Other “probabilistic and timed” (automata-)based models

- Real-time probabilistic systems [ACD91, ACD92]
- Dense-time Markov chains [BHHK03]
- Labelled Markov processes over a continuum [DGJP03,04]
- Strong relation with robustness
- Robust timed automata [GHJ97, HR00]
- Robust model-checking [Puri98, DDR04, DDMR04, ALM05, BMR06, BMR08]

cf Pierre-Alain Reynier’s talk tomorrow
Related works

- Other “probabilistic and timed” (automata-)based models
  - probabilistic timed automata à la PRISM [KNSS02]

NB: our model generalizes dense-time Markov chains [BHHK03]

- Labelled Markov processes over a continuum [DGJP03,04]

- Strong relation with robustness
  - robust timed automata [GHJ97,HR00]
  - robust model-checking [Puri98,DDR04,DDMR04,ALM05,BMR06,BMR08]

cf Pierre-Alain Reynier’s talk tomorrow
Related works

- Other “probabilistic and timed” (automata-)based models
  - probabilistic timed automata à la PRISM \[\text{[KNSS02]}\]
  - real-time probabilistic systems \[\text{[ACD91, ACD92]}\]
  - dense-time Markov chains \[\text{[BHHK03]}\]
  - Labelled Markov processes over a continuum \[\text{[DGJP03,04]}\]
  - Strong relation with robustness
  - robust timed automata \[\text{[GHJ97, HR00]}\]
  - robust model-checking \[\text{[Puri98, DDR04, DDMR04, ALM05, BMR06, BMR08]}\]

Note: Our model generalizes dense-time Markov chains.
Related works

- Other “probabilistic and timed” (automata-)based models
  - probabilistic timed automata à la PRISM \cite{KNSS02}
  - real-time probabilistic systems \cite{ACD91,ACD92}
  - dense-time Markov chains \cite{BHHK03}

NB: our model generalizes dense-time Markov chains

- Labelled Markov processes over a continuum \cite{DGJP03,04}
- Strong relation with robustness
  - robust timed automata \cite{GHJ97,HR00}
  - robust model-checking \cite{Puri98,DDR04,DDMR04,ALM05,BMR06,BMR08}

cf Pierre-Alain Reynier’s talk tomorrow
Related works

- Other “probabilistic and timed” (automata-)based models
  - probabilistic timed automata à la PRISM [KNSS02]
  - real-time probabilistic systems [ACD91, ACD92]
  - dense-time Markov chains [BHHK03]

NB: our model generalizes dense-time Markov chains
Related works

- Other “probabilistic and timed” (automata-)based models
  - probabilistic timed automata \textit{à la} PRISM \cite{KNSS02}
  - real-time probabilistic systems \cite{ACD91,ACD92}
  - dense-time Markov chains \cite{BHHK03}

  NB: our model generalizes dense-time Markov chains

- Labelled Markov processes over a continuum \cite{DGJP03,04}

\text{cf} Pierre-Alain Reynier's talk tomorrow
Related works

- Other “probabilistic and timed” (automata-)based models
  - probabilistic timed automata à la PRISM \[KNSS02\]
  - real-time probabilistic systems \[ACD91, ACD92\]
  - dense-time Markov chains \[BHHK03\]

  NB: our model generalizes dense-time Markov chains

- Labelled Markov processes over a continuum \[DGJP03,04\]

- Strong relation with robustness
Related works

- Other “probabilistic and timed” (automata-)based models
  - probabilistic timed automata à la PRISM [KNSS02]
  - real-time probabilistic systems [ACD91, ACD92]
  - dense-time Markov chains [BHHK03]

  NB: our model generalizes dense-time Markov chains

- Labelled Markov processes over a continuum [DGJP03, 04]

- Strong relation with robustness
  - robust timed automata [GHJ97, HR00]
  - robust model-checking [Puri98, DDR04, DDMR04, ALM05, BMR06, BMR08]

  *cf* Pierre-Alain Reynier’s talk tomorrow
Conclusions

- a probabilistic semantics for timed automata which removes “unlikely” (sequences of) events
- qualitative model-checking has a topological interpretation
- algorithm for qualitative LTL model-checking
Conclusions

▷ a probabilistic semantics for timed automata which removes “unlikely” (sequences of) events
▷ qualitative model-checking has a topological interpretation
▷ algorithm for qualitative LTL model-checking
▷ remark: extends to hybrid systems with finite bisimulation quotient
Conclusions

- a probabilistic semantics for timed automata which removes “unlikely” (sequences of) events
- qualitative model-checking has a topological interpretation
- algorithm for qualitative LTL model-checking
- remark: extends to hybrid systems with finite bisimulation quotient

Ongoing works

- quantitative analysis
- games
Conclusions

▶ a probabilistic semantics for timed automata which removes “unlikely” (sequences of) events
▶ qualitative model-checking has a topological interpretation
▶ algorithm for qualitative LTL model-checking
▶ remark: extends to hybrid systems with finite bisimulation quotient

Ongoing works

▶ quantitative analysis
▶ games

Further works

▶ efficient zone-based algorithm
▶ apply to relevant examples
▶ add non-determinism (à la MDP)
▶ handle several clocks
▶ timed properties