Stochastic Timed Automata

Patricia Bouyer-Decitre

LSV, CNRS & ENS Cachan, France

Based on joint works with Nathalie Bertrand, Thomas Brihaye, Pierre Carlier, Quentin Menet, Christel Baier, ...
The model of timed automata
The model of timed automata

\[
\begin{align*}
\text{safe} & \xrightarrow{23} \text{safe} \\
& \quad \quad \text{problem} \xrightarrow{x:=0} \text{alarm} \\
& \quad \quad \quad \text{repair, } x \leq 15 \xrightarrow{y:=0} \text{repair} \\
& \quad \quad \quad \text{delayed, } y := 0 \xrightarrow{15 \leq x \leq 16} \text{failsafe} \\
\text{done, } 22 \leq y \leq 25 & \xrightarrow{} \text{safe}
\end{align*}
\]
An example: The task graph scheduling problem

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

\[ P_1 \text{ (fast):} \]

<table>
<thead>
<tr>
<th>time operation</th>
<th>time in picoseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>2</td>
</tr>
<tr>
<td>\times</td>
<td>3</td>
</tr>
</tbody>
</table>

\[ P_2 \text{ (slow):} \]

<table>
<thead>
<tr>
<th>time operation</th>
<th>time in picoseconds</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>5</td>
</tr>
<tr>
<td>\times</td>
<td>7</td>
</tr>
</tbody>
</table>

**Energy:**

<table>
<thead>
<tr>
<th>state</th>
<th>idle energy</th>
<th>in use energy</th>
</tr>
</thead>
<tbody>
<tr>
<td>idle</td>
<td>10 Watts</td>
<td></td>
</tr>
<tr>
<td>in use</td>
<td>90 Watts</td>
<td></td>
</tr>
</tbody>
</table>

\[
\begin{align*}
D \times (C \times (A+B)) &= 2 \times 3 = 6 \\
(A+B) &= 2 + 3 = 5 \\
(C \times D) &= 3 \times 2 = 6 \\
\end{align*}
\]

\[
B = \begin{bmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
A = \begin{bmatrix}
0 & 1 & 0 \\
0 & 0 & 1 \\
1 & 0 & 0
\end{bmatrix}
\]

**Task Graph:***

- **Sch1:**
  - \( T_1 \) 13 picoseconds, 1.37 nanojoules
  - \( T_2 \) 12 picoseconds, 1.39 nanojoules
  - \( T_3 \) 19 picoseconds, 1.32 nanojoules

- **Sch2:**
  - \( T_1 \) 13 picoseconds, 1.37 nanojoules
  - \( T_2 \) 12 picoseconds, 1.39 nanojoules
  - \( T_3 \) 19 picoseconds, 1.32 nanojoules

- **Sch3:**
  - \( T_1 \) 13 picoseconds, 1.37 nanojoules
  - \( T_2 \) 12 picoseconds, 1.39 nanojoules
  - \( T_3 \) 19 picoseconds, 1.32 nanojoules

---

Modelling the task graph scheduling problem

- **Processors**
  - $P_1$:
    - $x = 2$: $x := 0$
    - $x = 3$: $x := 0$
    - $(x \leq 2)$
    - $(x \leq 3)$
  - $P_2$:
    - $y = 5$: $x := 0$
    - $y = 7$: $x := 0$
    - $(y \leq 5)$
    - $(y \leq 7)$

- **Tasks**
  - $T_4$:
    - $t_1 \land t_2$
    - $t_4 := 1$
  - $T_5$:
    - $t_3$
    - $t_5 := 1$

A schedule is a path in the product automaton
An example [AD94]
How to model uncertainty over delays?

- Using timed games

\[
\begin{align*}
\text{add} & : x = 0 \\
\text{mult} & : x = 0
\end{align*}
\]
How to model uncertainty over delays?

- Using timed games

\[
\begin{align*}
\text{idle} & \quad \text{add} \quad \text{mult} \\
(x \leq 2) & \quad x := 0 & \quad x := 0 \\
\text{done} & \quad x \geq 1 & \quad x \geq 1 \\
\text{+} & \quad \times & \quad (x \leq 3)
\end{align*}
\]
How to model uncertainty over delays?

- Using timed games

- Using stochastic delays
How to model uncertainty over delays?

- Using timed games

```
 idle + (x ≤ 2)
\[ + \quad \text{done} \quad \text{add} \quad x := 0 \]
\[ x ≥ 1 \]

 idle ± (x ≤ 3)
\[ ± \quad \text{done} \quad \text{mult} \quad x := 0 \]
\[ x ≥ 1 \]

```

- Using stochastic delays

```
 1 1.5 2
+ (x ≤ 2)
\[ + \quad \text{done} \quad \text{add} \quad x := 0 \]
\[ x ≥ 1 \]

 1 2 3
± (x ≤ 3)
\[ ± \quad \text{done} \quad \text{mult} \quad x := 0 \]
\[ x ≥ 1 \]
```

10/43
Existing models?

Models based on timed automata

- Probabilistic timed automata [KNSS99]
  \(\sim\) only discrete probabilities over edges

- Continuous probabilistic timed automata [KNSS00]
  \(\sim\) resets of clocks are randomized, but only few results

---


[KNSS00] Kwiatkowska, Norman, Segala, Sproston. Verifying quantitative properties of continuous probabilistic timed automata (CONCUR’00).
How can we attach probabilities to delays?

- The example of continuous-time Markov chains

**Exponential distribution**

Density function: $t \mapsto \begin{cases} 
\lambda \cdot \exp(-\lambda t) & \text{if } t \geq 0 \\
0 & \text{otherwise}
\end{cases}$
How can we attach probabilities to delays?

- The example of continuous-time Markov chains

Exponential distribution

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

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

- The example of continuous-time Markov chains

  exponential distribution

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

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

- But what if bounded intervals?

  truncated normal distribution

  \( \frac{1}{\beta_1} |I| \text{if } t \geq 0 \)

  \( \text{otherwise} \)
How can we attach probabilities to delays?

- The example of continuous-time Markov chains

  \[
  \text{exponential distribution}
  \]

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

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

- But what if bounded intervals?

  \[
  \text{truncated normal distribution}
  \]
How can we attach probabilities to delays?

- The example of continuous-time Markov chains

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

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

- But what if bounded intervals?

\[
\text{truncated normal distribution} \\
\text{uniform distribution} \\
\text{density function } t \mapsto \begin{cases} 
\frac{1}{|I|} & \text{if } t \geq 0 \\
0 & \text{otherwise}
\end{cases}
\]
How does a STA look like?

- **Safe**
  - Problem, $x := 0$
  - Done, $22 \leq y \leq 25$

- **Alarm**
  - Repair, $x \leq 15$
  - Delayed, $y := 0$
  - $15 \leq x \leq 16$

- **Repairing**
  - Repair
  - $2 \leq y \land x \leq 56$
  - $y := 0$

- **Failsafe**
  - $2 \leq 56 - x$
  - $y := 0$
Some remarks

- This defines a purely stochastic process
Some remarks

- This defines a purely stochastic process

- **Continuous-time Markov chains** = STA 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

- **Continuous-time Markov chains** = STA with a single “useless” clock which is reset on all transitions. The distributions on delays are exponential distributions with a rate per location

- Finite-state **generalized semi-Markov processes** (residual-lifetime semantics) are STAs (if no fixed-delay events)
Some remarks

- This defines a purely stochastic process

- **Continuous-time Markov chains** = STA with a single “useless” clock which is reset on all transitions. The distributions on delays are exponential distributions with a rate per location

- Finite-state **generalized semi-Markov processes** (residual-lifetime semantics) are STAs (if no fixed-delay events)

- Allows to express richer timing constraints
Almost-sure model-checking

We are interested in (automatic) model-checking algorithms!

- **Qualitative model-checking**: decide whether

  \[ P(\{ \varrho \in \text{Runs}(s) \mid \varrho \models \varphi \}) = 1 \]

  We write \( s \approx \varphi \) whenever it is the case.
  This is the almost-sure model-checking problem.

- **Quantitative model-checking**: compute (or approximate) the value

  \[ P(\{ \varrho \in \text{Runs}(s) \mid \varrho \models \varphi \}) \]
An example

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

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

\[ e_3, x = 1 \]

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

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

\[ \ell_0,0 \xrightarrow{e_1} \ell_1,0 \xrightarrow{e_2} \ell_2,0 \xrightarrow{e_4} \ell_3,0 \xrightarrow{e_6} \]
\[ \ell_0,(0,1) \xrightarrow{e_1} \ell_1,(0,1) \xrightarrow{e_2} \ell_3,(0,1) \xrightarrow{e_7} \]
\[ \ell_0,1 \xrightarrow{e_2} \ell_1,1 \xrightarrow{e_3} \ell_3,1 \]

\[ e_1 \]

It holds as well that:
[green] \( P(\text{MC}(A) | = G \Rightarrow F) = 1 \)

\[ \text{When is that the case that } P(A | = \phi) = 1 \text{ iff } P(\text{MC}(A) | = \phi) = 1? \]
The pruned region automaton

\[ \ell_0,0 \xrightarrow{e_1} \ell_0,(0,1) \xrightarrow{e_2} \ell_1,(0,1) \xrightarrow{e_6} \ell_2,0 \]

\[ \ell_1,0 \xrightarrow{e_2} \ell_1,(0,1) \xrightarrow{e_6} \ell_2,0 \]

\[ \ell_3,0 \xrightarrow{e_7} \ell_3,(0,1) \]

\[ \ell_0,1 \xrightarrow{e_2} \ell_1,1 \]

\[ \ell_3,1 \]

It holds as well that:

\[ \mathbb{P}(MC(A) \mid = G (\text{green} \Rightarrow F \text{red})) = 1 \]

When is that the case that \( \mathbb{P}(A \mid = \phi) = 1 \) iff \( \mathbb{P}(MC(A) \mid = \phi) = 1 \)?
The pruned region automaton
The pruned region automaton

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

It holds as well that:

$$\mathbb{P}(MC(A) \models G(\text{green} \Rightarrow F \text{ red})) = 1$$
The pruned region automaton

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

It holds as well that:

$$P(MC(\mathcal{A}) \models G(\text{green} \Rightarrow F \text{ red})) = 1$$

When is that the case that

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