Energy consumption in timed systems

Patricia Bouyer

LSV – CNRS & ENS Cachan

November 13, 2008
Systems that need to be verified

→ include reactive, embedded systems, (communication) protocols, ...
Systems that need to be verified

→ include reactive, embedded systems, (communication) protocols, . . .

Important characteristics

They have to meet numerous quantitative constraints such as:

- timing constraints
  
  “Will the airbag open within 5ms after the car crashes?”
Systems that need to be verified

→ include reactive, embedded systems, (communication) protocols, . . .

Important characteristics

They have to meet numerous quantitative constraints such as:

- timing constraints
  
  “Will the airbag open within 5ms after the car crashes?”

- energy/cost/resource constraints
  
  “Can an autonomous robot with solar cells explore a fixed area?”
  “How should one optimize the profit in a factory?”
  “Can we schedule those tasks on two processors?”

- . . .
A rather general solution: hybrid systems

What is a hybrid system?

- a discrete control (the mode of the system)
- a continuous evolution within a mode (given by variables)

Example (The thermostat)

A simple thermostat, where \( T \) (the temperature) depends on the time:

\[
\begin{align*}
\dot{T} &= -0.5 T & (T \geq 18) \\
\dot{T} &= 2.25 - 0.5 T & (T \leq 22)
\end{align*}
\]

\( T \leq 19 \quad T \geq 21 \)
A rather general solution: hybrid systems

What is a hybrid system?

- a discrete control (the mode of the system)
- a continuous evolution within a mode (given by variables)

Example (The thermostat)

A simple thermostat, where $T$ (the temperature) depends on the time:

- Off
  $$\dot{T} = -0.5T \quad (T \geq 18)$$
- On
  $$\dot{T} = 2.25 - 0.5T \quad (T \leq 22)$$

\[\text{Henzinger 1996}\]
The thermostat example

\[ \dot{T} = -0.5T \quad (T \geq 18) \]

\[ \dot{T} = 2.25 - 0.5T \quad (T \leq 22) \]
The thermostat example

\[ \dot{T} = -0.5T \quad (T \geq 18) \]

\[ \dot{T} = 2.25 - 0.5T \quad (T \leq 22) \]

\[ T \leq 19 \]

\[ T \geq 21 \]
Going further with the thermostat

The new variable $c$ represents the cost to be paid.

Off
$\dot{T} = -0.5T$
$\dot{c} = 0$
$(T \geq 18)$

On
$\dot{T} = 2.25 - 0.5T$
$\dot{c} = 5$
$(T \leq 22)$

Is that possible to pay no more than 3\text{e} per hour to maintain the temperature between 18\textdegree C and 22\textdegree C?
Going further with the thermostat

The new variable $c$ represents the cost to be paid.

\[
\begin{align*}
\text{Off} & \quad \dot{T} = -0.5T \\
& \quad \dot{c} = 0 \\
& \quad (T \geq 18) \\
\text{On} & \quad \dot{T} = 2.25 - 0.5T \\
& \quad \dot{c} = 5 \\
& \quad (T \leq 22)
\end{align*}
\]

$T \leq 19$

$T \geq 21$

Question

Is that possible to pay no more than 3€ per hour to maintain the temperature between 18°C and 22°C?
Going further with the thermostat

The new variable $c$ represents the cost to be paid.

Off
\[ \dot{T} = -0.5T \]
\[ \dot{c} = 0 \]
\[ (T \geq 18) \]

On
\[ \dot{T} = 2.25 - 0.5T \]
\[ \dot{c} = 5 \]
\[ (T \leq 22) \]

Of course, this is a complex question, and simpler questions can be asked...
Going further with the thermostat

The variable $x$ measures the time elapsing in mode $\text{On}$.

\[ \dot{T} = -0.5T \]
\[ \dot{x} = 1 \quad (T \geq 18) \]

\[ T \leq 19 \]
\[ x := 0 \]

\[ T \geq 21 \]

\[ \dot{T} = 2.25 - 0.5T \]
\[ \dot{x} = 1 \quad (T \leq 22) \]

\[ x \geq 5, \quad T = 22 \]

Crash
Going further with the thermostat

The variable $x$ measures the time elapsing in mode On.

\[
\begin{align*}
\dot{T} &= -0.5T \\
\dot{x} &= 1 \\
&T \geq 18
\end{align*}
\]

\[
\begin{align*}
\dot{T} &= 2.25 - 0.5T \\
\dot{x} &= 1 \\
&T \leq 22
\end{align*}
\]

\[
\begin{align*}
&T \leq 19 \\
x &:= 0
\end{align*}
\]

\[
\begin{align*}
&T \geq 21
\end{align*}
\]

\[
\begin{align*}
&x \geq 5, T = 22
\end{align*}
\]

Question

Is location Crash reachable from state $(\text{Off}, T = 20, x = 0)$?
Ok...
Ok...

Easy...
Ok...

Easy...
Ok...

Easy...

Easy...
Ok... but?

Easy...

Easy...

constraint

constraint
Ok... but?

Easy...

Easy...

Hard!

constraint

constraint
Why is that hard?

<table>
<thead>
<tr>
<th>What we do</th>
<th>What we don’t do</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The hybrid system model is undecidable as soon as we use:

- differential equations of the form $\dot{x} = 0$ or $\dot{x} = 1$;
- constraints of the form $x \in [a, b]$;
- resets of the variables to 0.

There is no general algorithm (or program) to verify hybrid systems.
Why is that hard?

<table>
<thead>
<tr>
<th>What we do</th>
<th>What we don’t do</th>
</tr>
</thead>
<tbody>
<tr>
<td>- exhaustive search</td>
<td>- partial simulation</td>
</tr>
</tbody>
</table>
Why is that hard?

<table>
<thead>
<tr>
<th>What we do</th>
<th>What we don’t do</th>
</tr>
</thead>
<tbody>
<tr>
<td>- exhaustive search</td>
<td>- partial simulation</td>
</tr>
<tr>
<td>- exact and symbolic computation (no round-off errors)</td>
<td>- approximate computation</td>
</tr>
</tbody>
</table>
Why is that hard?

<table>
<thead>
<tr>
<th>What we do</th>
<th>What we don’t do</th>
</tr>
</thead>
<tbody>
<tr>
<td>- exhaustive search</td>
<td>- partial simulation</td>
</tr>
<tr>
<td>- exact and symbolic computation (no round-off errors)</td>
<td>- approximate computation</td>
</tr>
<tr>
<td>- fully-automated methods (for large classes of systems)</td>
<td>- <em>ad-hoc</em> methods</td>
</tr>
</tbody>
</table>

**Theorem [Henzinger 1996]**

The hybrid system model is undecidable as soon as we use:

- differential equations of the form \( \dot{x} = 0 \) or \( \dot{x} = 1 \);
- constraints of the form \( x \in [a, b] \);
- resets of the variables to 0.

There is no general algorithm (or program) to verify hybrid systems.
### Why is that hard?

<table>
<thead>
<tr>
<th>What we do</th>
<th>What we don’t do</th>
</tr>
</thead>
<tbody>
<tr>
<td>- exhaustive search</td>
<td>- partial simulation</td>
</tr>
<tr>
<td>- exact and symbolic computation</td>
<td>- approximate computation</td>
</tr>
<tr>
<td>(no round-off errors)</td>
<td>- <em>ad-hoc</em> methods</td>
</tr>
<tr>
<td>- fully-automated methods</td>
<td></td>
</tr>
<tr>
<td>(for large classes of systems)</td>
<td></td>
</tr>
</tbody>
</table>

**Theorem [Henzinger 1996]**

The hybrid system model is **undecidable** as soon as we use:

- differential equations of the form $\dot{x} = 0$ or $\dot{x} = 1$;
- constraints of the form $x \in [a, b]$;
- resets of the variables to 0.

$\leadsto$ There is no general algorithm (or program) to verify hybrid systems.
What is undecidability? The Post correspondence problem

An example

- baa
  - a
- aa
  - ab
- bb
  - bba
What is undecidability? The Post correspondence problem

An example

Theorem

PCP is undecidable.

There is no general algorithm (or program) to solve PCP.
What is undecidability? The Post correspondence problem

An example

<table>
<thead>
<tr>
<th>baa</th>
<th>aa</th>
<th>bb</th>
</tr>
</thead>
<tbody>
<tr>
<td>a</td>
<td>ab</td>
<td>bba</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bb</th>
<th>aa</th>
<th>bb</th>
</tr>
</thead>
<tbody>
<tr>
<td>bba</td>
<td>ab</td>
<td>bba</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>bba</th>
<th>bb</th>
<th>a</th>
</tr>
</thead>
<tbody>
<tr>
<td>bba</td>
<td>bb</td>
<td>a</td>
</tr>
</tbody>
</table>

Theorem [Post 1946]

PCP is **undecidable**.

\[ \sim \text{There is no general algorithm (or program) to solve PCP.} \]
Understanding further PCP

Another example

- bab
  - a
- bb
  - ba
- a
  - abb

There is no solution!

The PCP@home contest has length 781.

http://www.theory.informatik.uni-kassel.de/~stamer/pcp/
Understanding further PCP

Another example

There is no solution!

http://www.theory.informatik.uni-kassel.de/~stamer/pcp/
Understanding further PCP

Another example

There is no solution!

The PCP@home contest

http://www.theory.informatik.uni-kassel.de/~stamer/pcp/
Understanding further PCP

Another example

There is no solution!

The PCP@home contest

The shortest solution for

has length 781.

http://www.theory.informatik.uni-kassel.de/~stamer/pcp/
Further undecidability

Hilbert’s tenth problem

Given a multivariate polynomial \( P(X_1, \ldots, X_n) \in \mathbb{Q}[X_1, \ldots, X_n] \), do there exist integers \((a_1, \ldots, a_n) \in \mathbb{Z}^n\) such that \( P(a_1, \ldots, a_n) = 0 \).
Further undecidability

Hilbert’s tenth problem

Given a multivariate polynomial \( P(X_1, \ldots, X_n) \in \mathbb{Q}[X_1, \ldots, X_n] \), do there exist integers \((a_1, \ldots, a_n) \in \mathbb{Z}^n\) such that \( P(a_1, \ldots, a_n) = 0 \).

Theorem [Matiyasevich 1970]

Hilbert’s tenth problem is undecidable.
Undecidability can be understood as follows

Reduction from tenth Hilbert’s problem
Given a multivariate polynomial $P$, one can construct a hybrid system $H_P$ such that $H_P$ is safe iff $P$ has an integral solution.
Undecidability can be understood as follows

**Reduction from tenth Hilbert’s problem**

Given a multivariate polynomial $P$, one can construct a hybrid system $H_P$ such that $H_P$ is safe iff $P$ has an integral solution.

**Reduction from PCP**

Given a finite set of tiles $S$ for PCP, one can construct a hybrid system $H_S$ such that $H_S$ is safe iff PCP has a solution with those tiles.
Undecidability can be understood as follows

<table>
<thead>
<tr>
<th>Reduction from tenth Hilbert’s problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given a multivariate polynomial $P$, one can construct a hybrid system $H_P$ such that $H_P$ is safe iff $P$ has an integral solution.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reduction from PCP</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given a finite set of tiles $S$ for PCP, one can construct a hybrid system $H_S$ such that $H_S$ is safe iff PCP has a solution with those tiles.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Reduction from your favorite difficult problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>Given any instance $I$ of a difficult problem, one can construct a hybrid system $H_I$ such that $H_I$ is safe iff there is a solution to $I$.</td>
</tr>
</tbody>
</table>
What our work consists in

system:

⇒

property:

G(request → Fgrant)

model-checking algorithm

design classes of models such that:

we will be able to analyze them automatically (and efficiently);

they will be powerful enough to represent numerous systems.

design efficient model-checking algorithms
What our work consists in

system:

⇒

property:

G (request → F grant)
What our work consists in

system: $G (\text{request} \rightarrow \text{F grant})$

property: Design classes of models such that:
we will be able to analyze them automatically (and efficiently);
they will be powerful enough to represent numerous systems.

develop efficient model-checking algorithms

model-checking algorithm

$G (\text{request} \rightarrow \text{F grant})$
What our work consists in

- **System:**
  - A system where requests can be granted or denied.

- **Property:**
  - A property \( G \) that checks if a request results in a specific outcome (granting or not granting).

- **Model-checking algorithm:**
  - Design classes of models such that we can analyze them automatically (and efficiently).

- **yes/no:**
  - Determine if the given property holds for the system.
What our work consists in

- Design classes of models such that:
  - we will be able to analyze them automatically (and efficiently);
  - they will be powerful enough to represent numerous systems.
What our work consists in

- Design classes of models such that:
  - we will be able to analyze them automatically (and efficiently);
  - they will be powerful enough to represent numerous systems.
- Design efficient model-checking algorithms
Timed automata [Alur, Dill 1990]

A timed automaton: a hybrid system with only clocks, i.e. variables whose derivative is always 1 ($\dot{x} = 1$) and that can be reset to 0.
Timed automata [Alur, Dill 1990]

A timed automaton: a hybrid system with only clocks, i.e. variables whose derivative is always 1 ($\dot{x} = 1$) and that can be reset to 0.

\[\ell_0 \xrightarrow{x \leq 2, c, y := 0} \ell_1 (y = 0) \xrightarrow{u} \ell_2 \xrightarrow{x = 2, c} \ell_3 \xrightarrow{u} \ell_4 \]
Timed automata [Alur, Dill 1990]

A timed automaton: a hybrid system with only clocks, i.e. variables whose derivative is always 1 ($\dot{x} = 1$) and that can be reset to 0.
Timed automata [Alur, Dill 1990]

A timed automaton: a hybrid system with only clocks, i.e. variables whose derivative is always 1 ($\dot{x} = 1$) and that can be reset to 0.

\[
x \leq 2, c, y := 0 \\
(x=0) \quad (y=0) \\
\]

\[
\ell_0 \xrightarrow{1.3} \ell_0 \\
x \quad 0 \\
y \quad 0
\]

\[
\ell_0 \xrightarrow{1.3} \ell_0 \\
x \quad 0 \\
y \quad 0
\]
Timed automata [Alur, Dill 1990]

A timed automaton: a hybrid system with only clocks, i.e. variables whose derivative is always 1 ($\dot{x} = 1$) and that can be reset to 0.

\[
\begin{align*}
\ell_0 & \xrightarrow{x \leq 2, c, y := 0} \ell_1 \\
\ell_0 & \xrightarrow{c} \ell_1 \\
\ell_1 & \xrightarrow{u} \ell_2 \\
\ell_1 & \xrightarrow{u} \ell_3 \\
\ell_2 & \xrightarrow{x = 2, c} \ell_3 \\
\ell_3 & \xrightarrow{x = 2, c} \text{smiley}
\end{align*}
\]

<table>
<thead>
<tr>
<th></th>
<th>$\ell_0$</th>
<th>$\ell_0$</th>
<th>$\ell_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>0</td>
<td>1.3</td>
<td>1.3</td>
</tr>
<tr>
<td>$y$</td>
<td>0</td>
<td>1.3</td>
<td>0</td>
</tr>
</tbody>
</table>
Timed automata [Alur, Dill 1990]

A timed automaton: a hybrid system with only clocks, i.e. variables whose derivative is always 1 ($\dot{x} = 1$) and that can be reset to 0.
Timed automata [Alur, Dill 1990]

A timed automaton: a hybrid system with only clocks, i.e. variables whose derivative is always 1 ($\dot{x} = 1$) and that can be reset to 0.
**Timed automata [Alur, Dill 1990]**

A timed automaton: a hybrid system with only clocks, i.e. variables whose derivative is always 1 ($\dot{x} = 1$) and that can be reset to 0.

\[
\begin{align*}
\ell_0 \xrightarrow{x \leq 2, c, y := 0} \ell_1 \\
\ell_1 \xrightarrow{u} \ell_2 \\
\ell_1 \xrightarrow{u} \ell_3 \\
\ell_2 \xrightarrow{x = 2, c} \ell_3 \\
\ell_3 \xrightarrow{x = 2, c} \text{happy}
\end{align*}
\]

\[
\begin{align*}
\ell_0 & \xrightarrow{1.3} \ell_0 & \xrightarrow{c} \ell_1 & \xrightarrow{u} \ell_3 & \xrightarrow{0.7} \ell_3 & \xrightarrow{c} \text{happy} \\
x & 0 & 1.3 & 1.3 & 1.3 & 2 \\
y & 0 & 1.3 & 0 & 0 & 0.7
\end{align*}
\]
Timed automata [Alur, Dill 1990]

A timed automaton: a hybrid system with only clocks, i.e. variables whose derivative is always 1 ($\dot{x} = 1$) and that can be reset to 0.

Questions

- Is that possible to reach location 🤠?
- How long will that take to reach location 🤠?
A second example

\[ x := 0 \]
\[ y := 0 \]
\[ 15 \leq x \leq 16 \]
\[ y := 0 \]
\[ 2 \leq y \land x \leq 56 \]

\[
\text{problem, } x := 0 \quad \text{repair, } x \leq 15 \quad \text{repair, } x \leq 15 \\
\text{done, } 22 \leq y \leq 25 \\
\text{delayed, } y := 0 \\
\text{repair, } y := 0 \\
\text{failsafe, } 2 \leq y \land x \leq 56 \]

\[
\text{safe} \quad \text{alarm} \quad \text{repairing} \quad \text{failsafe} \\
\]
A second example

- **safe**
  - $x = 0$
  - $y = 0$

- **alarm**
  - $y = 0$
  - $15 \leq x \leq 16$

- **repair**
  - $x \leq 15$
  - $y = 0$

- **failsafe**
  - $2 \leq y \land x \leq 56$
  - $y = 0$

- **done**
  - $22 \leq y \leq 25$

Transition arrows:
- From **safe** to **problem**
- From **problem** to **alarm**
- From **alarm** to **failsafe**
- From **failsafe** to **repair**
- From **repair** to **done**
A second example

\[
\begin{align*}
&\text{safe} \quad 23 \quad \text{safe} \\
X & \quad 0 \quad \rightarrow \quad 23 \\
Y & \quad 0 \quad \rightarrow \quad 23
\end{align*}
\]
A second example

\[
\begin{align*}
\text{safe} & \rightarrow \text{problem, } x:=0 & \text{repairing} & \rightarrow \text{repair, } x \leq 15 \\
\text{alarm} & \rightarrow \text{repair, } y:=0 & & \text{failed, } y:=0 \\
& & & \rightarrow \text{delayed, } y:=0 \\
\text{healthy} & \rightarrow \text{done, } 22 \leq y \leq 25 & & \rightarrow \text{safe} \\
\end{align*}
\]
A second example

\[ x := 0, x \leq 15 \]
\[ y := 0, 15 \leq x \leq 16 \]
\[ y := 0, delayed \]
\[ y := 0, failsafe \]

\[ x, y \]
\[ 0, 22 \leq y \leq 25 \]
\[ 0, done \]
\[ 0, repair, 15 \leq x \leq 56 \]
\[ 0, repair \]
\[ 2, alarm \]
\[ 2, problem \]
\[ 2, safe \]

<table>
<thead>
<tr>
<th>safe</th>
<th>23</th>
<th>safe</th>
<th>problem</th>
<th>alarm</th>
<th>15.6</th>
<th>alarm</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>0</td>
<td>23</td>
<td>0</td>
<td>15.6</td>
<td>15.6</td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>0</td>
<td>23</td>
<td>23</td>
<td>38.6</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A second example

- **safe** → **problem**, \( x := 0 \)
- **problem** → **alarm**, \( y := 0 \)
- **repair**, \( x \leq 15 \)
- **delayed**, \( y := 0 \)
- **done**, \( 22 \leq y \leq 25 \)
- **repairing** → **failsafe**
- **failsafe** → **safe**

<table>
<thead>
<tr>
<th>State</th>
<th>Transition</th>
<th>Value 1</th>
<th>Value 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>safe</td>
<td>23</td>
<td>safe</td>
<td>23</td>
</tr>
<tr>
<td>safe</td>
<td>problem</td>
<td>0</td>
<td>alarm</td>
</tr>
<tr>
<td>alarm</td>
<td>15.6</td>
<td>alarm</td>
<td>15.6</td>
</tr>
<tr>
<td>alarm</td>
<td>delayed</td>
<td>failsafe</td>
<td>15.6</td>
</tr>
<tr>
<td>failsafe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>15.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>15.6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
A second example

- **Safe**
  - X: 0
  - Y: 0

- **Problem**, \( x := 0 \)
  - X: 23
  - Y: 0

- **Alarm**
  - X: 0
  - Y: 23

- **Repairing**, \( y := 0 \)
  - X: 15.6
  - Y: 38.6

- **Failsafe**, \( 2 \leq y \land x \leq 56 \)
  - X: 15.6
  - Y: 0

- **Delayed**, \( y := 0 \)
  - X: 15.6
  - Y: 17.9

- **Safe**, \( x := 0 \)
  - X: 2.3
  - Y: 0
A second example

- **Problem**: \( x := 0 \)
- **Repair**: \( y := 0 \), \( 15 \leq x \leq 16 \)
- **Delayed**: \( y := 0 \)

- **Safe** to **Problem**: \( 22 \leq y \leq 25 \)
- **Done** to **FailSafe**: \( 2 \leq y \land x \leq 56 \)

**Table**

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>23</td>
<td>23</td>
</tr>
<tr>
<td>0</td>
<td>23</td>
</tr>
<tr>
<td>15.6</td>
<td>38.6</td>
</tr>
<tr>
<td>15.6</td>
<td>0</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Y</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
</tr>
<tr>
<td>2.3</td>
</tr>
<tr>
<td>0</td>
</tr>
</tbody>
</table>

**Diagram**

- **Safe** to **Alarm**: \( x := 0 \)
- **Alarm** to **FailSafe**: \( y := 0 \), \( 15 \leq x \leq 16 \)
- **FailSafe** to **Repairing**: \( x := 0 \)

**Transition**

- **Safe** to **Problem**: \( 23 \)
- **Problem** to **Alarm**: \( 0 \)
- **Alarm** to **Repairing**: \( 15.6 \)
- **Repairing** to **Done**: \( 17.9 \)
- **Done** to **FailSafe**: \( 15.6 \)
A second example
A second example

\[
\begin{align*}
\text{safe} & \xrightarrow{23} \text{safe} & \text{problem, } x := 0 & \xrightarrow{} \text{alarm} & \xrightarrow{15.6} \text{alarm} & \xrightarrow{\text{delayed}} \text{failsafe} \\
X & 0 & 23 & 0 & 15.6 & 15.6 \\
Y & 0 & 23 & 23 & 38.6 & 0 \\
\text{failsafe} & \xrightarrow{2.3} \text{failsafe} & \xrightarrow{} \text{repairing} & \xrightarrow{22.1} \text{repairing} & \xrightarrow{\text{done}} \text{safe} \\
\ldots & 15.6 & 17.9 & 17.9 & 40 & 40 \\
\ldots & 0 & 2.3 & 0 & 22.1 & 22.1
\end{align*}
\]
A third example: B&O collision detection protocol
A fourth example

\[ \ell_0 \quad x_3 \leq 3 \quad x_1, x_3 := 0 \]

\[ \ell_1 \quad x_2 = 3 \quad x_2 := 0 \]

\[ \ell_2 \]

\[ x_2 = 2, x_2 := 0 \]

\[ x_1 = 2, x_1 := 0 \]

\[ \ell_3 \]

\[ \ell_6 \quad x_1 = 3 \quad x_1 := 0 \]

\[ x_4 < x_3 + 2 \]

\[ \ell_7 \quad x_2 > x_1 + 2 \]

Error

\[ x_4 < x_3 + 2 \]

\[ x_1 := 0 \]

\[ x_2 := 0 \]

\[ \ell_4 \]

\[ x_2 = 2 \]

\[ x_1 := 0 \]

\[ \ell_5 \]

\[ x_2 := 0 \]

\[ \ell_0 \quad x_3 \leq 3 \quad x_1, x_3 := 0 \]

\[ x_2 = 3 \quad x_2 := 0 \]

\[ x_2 = 2, x_2 := 0 \]

\[ x_1 = 2, x_1 := 0 \]
A fundamental result

**Theorem [Alur & Dill 1990]**

There is a general algorithm (or program) to check whether a timed automaton is safe or not.
A fundamental result

**Theorem [Alur & Dill 1990]**

There is a general algorithm (or program) to check whether a timed automaton is safe or not.
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[ \ell_0 + 5 \xrightarrow{x \leq 2, c, y:=0} \ell_1 \]
\[ \ell_1 \xrightarrow{(y=0)} \ell_2 + 10 \]
\[ \ell_2 \xrightarrow{u} \ell_3 + 1 \xrightarrow{x=2, c} \text{smiley} \]
\[ \ell_3 \xrightarrow{0.7} \ell_3 \xrightarrow{c} \text{smiley} \]

Cost:

\begin{align*}
x & \quad \ell_0 & 1.3 & \ell_0 & c & \ell_1 & u & \ell_3 & 0.7 & \ell_3 & c & \text{smiley} \\
y & \quad 0 & 1.3 & 0 & 0 & 0.7 & \end{align*}
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[
\begin{aligned}
\ell_0 & \xrightarrow{5} \ell_0 \\
\ell_1 & \xrightarrow{u} \ell_2 \\
\ell_3 & \xrightarrow{u} \ell_3 \\
\end{aligned}
\]

\[
\begin{aligned}
x & \leq 2, c, y := 0 \\
(y=0) & \\
\end{aligned}
\]

\[
\begin{aligned}
x & = 2, c \\
x & = 2, c \\
\end{aligned}
\]

\[
\begin{aligned}
x & = 2, c \\
x & = 2, c \\
\end{aligned}
\]

\[
\begin{aligned}
x & \leq 2, c, y := 0 \\
(y=0) & \\
\end{aligned}
\]

\[
\begin{aligned}
\ell_0 & \xrightarrow{1.3} \ell_0 \\
\ell_0 & \xrightarrow{c} \ell_1 \\
\ell_1 & \xrightarrow{u} \ell_3 \\
\ell_3 & \xrightarrow{0.7} \ell_3 \\
\ell_3 & \xrightarrow{c} \text{😊} \\
\end{aligned}
\]

\[
\begin{aligned}
x & = 0, 1.3, 1.3, 1.3, 1.3, 2 \\
y & = 0, 1.3, 0, 0, 0, 0.7 \\
\end{aligned}
\]

\[
\begin{aligned}
cost & : 6.5 \\
\end{aligned}
\]
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[
\ell_0 + 5 \quad \xrightarrow{x \leq 2,c,y:=0} \quad \ell_1 (y=0) \quad \xrightarrow{u} \quad \ell_2 \quad +10 \quad \xrightarrow{x=2,c} \quad \ell_3 \quad +1 \quad \xrightarrow{x=2,c} \quad \text{Smiley}
\]

\[
\begin{array}{c|cccccccc}
& \ell_0 & \ell_0 & \ell_1 & \ell_3 & \ell_3 & \ell_3 & \ell_3 \\
\hline
x & 0 & 1.3 & 1.3 & 1.3 & 2 \\
y & 0 & 1.3 & 0 & 0 & 0.7 \\
\end{array}
\]

cost : 6.5 + 0
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[
\ell_0 + 5 \xrightarrow{x \leq 2, c, y := 0} \ell_1 \xrightarrow{(y = 0)} \ell_2 \xrightarrow{u \text{ or } u} \ell_3 \xrightarrow{x = 2, c} +1 \xrightarrow{x = 2, c} \ell_0 \xrightarrow{c} +1 \xrightarrow{c} \ell_3 \xrightarrow{c} \ell_3 \xrightarrow{c} \ell_3 \xrightarrow{c} \ell_3 \xrightarrow{c} +1
\]

<table>
<thead>
<tr>
<th></th>
<th>$\ell_0$</th>
<th>$\ell_0$</th>
<th>$\ell_1$</th>
<th>$\ell_1$</th>
<th>$\ell_3$</th>
<th>$\ell_3$</th>
<th>$\ell_3$</th>
<th>$\ell_3$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>0</td>
<td>1.3</td>
<td>1.3</td>
<td>1.3</td>
<td>1.3</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$y$</td>
<td>0</td>
<td>1.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cost: $6.5 + 0 + 0$
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[
\ell_0 + 5 \xrightarrow{x \leq 2, c, y:=0} \ell_1 \xrightarrow{(y=0)} \ell_2 \xrightarrow{u} \ell_3 \xrightarrow{x=2, c} +1
\]

Cost:

\[
\begin{align*}
\ell_0 & \quad 1.3 & \ell_0 & \quad c & \ell_1 & \quad u & \ell_3 & \quad 0.7 & \ell_3 & \quad c & \text{\smiley} \\
x & 0 & 1.3 & 1.3 & 1.3 & 1.3 & 2 & 0.7 \\
y & 0 & 1.3 & 0 & 0 & 0 & 0.7 \\
\text{cost} & 6.5 & + & 0 & + & 0 & + & 0.7
\end{align*}
\]
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[
\ell_0 + 5 \xrightarrow{x \leq 2, c, y := 0} \ell_1 (y = 0) \xrightarrow{y = 0} \ell_2 \xrightarrow{u} \ell_3 \xrightarrow{c} \ell_4 \xrightarrow{x = 2, c} +1
\]

<table>
<thead>
<tr>
<th></th>
<th>\ell_0</th>
<th>\ell_0</th>
<th>c</th>
<th>\ell_1</th>
<th>u</th>
<th>\ell_3</th>
<th>0.7</th>
<th>\ell_3</th>
<th>c</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>0</td>
<td>1.3</td>
<td>c</td>
<td>1.3</td>
<td>u</td>
<td>1.3</td>
<td>2</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>y</td>
<td>0</td>
<td>1.3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.7</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Cost:
\[
6.5 + 0 + 0 + 0.7 + 7
\]
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[ \ell_0 + 5 \xrightarrow{x \leq 2, c, y := 0} \ell_1 \] (y=0)

\[ \ell_1 \xrightarrow{u} \ell_2 \]

\[ \ell_2 \xrightarrow{x = 2, c} \ell_3 \]

\[ \ell_3 \xrightarrow{c} +1 \]

Cost:

\[
\begin{array}{ccccccccc}
 & \ell_0 & 1.3 & \ell_0 & c & \ell_1 & u & \ell_3 & 0.7 & \ell_3 & c & \text{Smiley} \\
\hline
x & 0 & 1.3 & 1.3 & 1.3 & 1.3 & 2 & 0.7 & \text{Smiley} \\
y & 0 & 1.3 & 0 & 0 & 0.7 & 7 & \text{Smiley} \\
\end{array}
\]

\[
\text{cost : } 6.5 + 0 + 0 + 0.7 + 7 = 14.2
\]
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[ \ell_0 + 5 \xrightarrow{x \leq 2, c, y:=0} \ell_1 + 10 \xrightarrow{u} \ell_2 + 10 \xrightarrow{x=2, c} \ell_3 + 1 \]

\[ (y=0) \]

\[ +1 \]

\[ +7 \]

Question

What is the optimal cost for reaching 😊?
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[ \ell_0 + 5 \xrightarrow{x \leq 2, c, y := 0} \ell_1 \]
\[ (y = 0) \]
\[ \ell_1 \xrightarrow{u} \ell_2 \]
\[ +10 \]
\[ \ell_2 \xrightarrow{x = 2, c} \ell_3 \]
\[ +1 \]
\[ \ell_3 \xrightarrow{x = 2, c} \text{smiley} \]

Question

What is the optimal cost for reaching \( \text{smiley} \)?

\[ 5t + 10(2 - t) + 1 \]
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[ \ell_0 + 5 \xrightarrow{x \leq 2, c, y := 0} \ell_1 \xrightarrow{(y = 0)} \ell_3 \xrightarrow{+1} \ell_2 \xrightarrow{+10} \text{happy face} \]

Question

What is the optimal cost for reaching ☺?

\[ 5t + 10(2 - t) + 1 , \ 5t + (2 - t) + 7 \]
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

Question

What is the optimal cost for reaching 😊?

\[
\min \left( 5t + 10(2 - t) + 1 , \ 5t + (2 - t) + 7 \right)
\]
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

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

**Question**

What is the optimal cost for reaching 😊?
Timed automata with costs (or energy information)

[Alur et al, Larsen et al 2001]

\[
\begin{align*}
\ell_0 & \xrightarrow{x \leq 2, c, y := 0} \ell_1 \\
\ell_1 & \xrightarrow{u} \ell_2 \quad \xrightarrow{u} \ell_3 \\
\ell_2 & \xrightarrow{x = 2, c} \rightarrow \\
\ell_3 & \xrightarrow{x = 2, c} \rightarrow \\
\end{align*}
\]

Question

What is the optimal cost for reaching \( \smiley \)?

\[
\inf_{0 \leq t \leq 2} \min (5t + 10(2 - t) + 1, 5t + (2 - t) + 7) = 9
\]

\( \leadsto \) **strategy:** leave immediately \( \ell_0 \), go to \( \ell_3 \), and wait there 2 t.u.
A further example with negative costs

Globally ($x \leq 1$)

\[ -3 \ell_0 + 6 \ell_1 - 6 \ell_2 = 1 \]

\[ x := 0 \quad \text{and} \quad x = 1 \]
A further example with negative costs

Globally \((x \leq 1)\)

\[
\ell_0 \rightarrow +6 \rightarrow -6
\]

\[
\ell_0 \rightarrow x:=0 \rightarrow \ell_1 \rightarrow x=1 \rightarrow \ell_2
\]

Safe bounds problems

- **Lower-bound problem**: can we stay above 0?
A further example with negative costs

Globally \((x \leq 1)\)

Lower-bound problem: can we stay above 0?
A further example with negative costs

Globally ($x \leq 1$)

Safe bounds problems
- **Lower-bound problem:** can we stay above 0?
A further example with negative costs

\[\ell_0 - 3 \xleftarrow{x := 0} \ell_1 + 6 \xrightarrow{x = 1} \ell_2 - 6\]

Globally \((x \leq 1)\)

Safe bounds problems

- Lower-bound problem: can we stay above 0?
A further example with negative costs

Globally \((x \leq 1)\)

\[
\ell_0 \xrightarrow{x:=0} \ell_1 \xrightarrow{x=1} \ell_2
\]

\[
-3 - 3x + 6 - 6x = 1
\]

\[
x := 0
\]

\[
x = 1
\]

Safe bounds problems

- Lower-bound problem: can we stay above 0?
A further example with negative costs

Globally ($x \leq 1$)

Safe bounds problems

- Lower-bound problem: can we stay above 0?
A further example with negative costs

Globally \((x \leq 1)\)

Safe bounds problems

- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?
A further example with negative costs

Globally \( (x \leq 1) \)

\[
\ell_0 - 3 \rightarrow l_1 + 6 \rightarrow l_2 - 6
\]

\( x := 0 \rightarrow x = 1 \)

Safe bounds problems

- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?
A further example with negative costs

Globally \((x \leq 1)\)

\[\ell_0 -3 \rightarrow \ell_1 +6 \rightarrow \ell_2 -6\]

\[x := 0 \quad x = 1\]

Safe bounds problems

- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?
A further example with negative costs

Globally ($x \leq 1$)

Safe bounds problems

- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?
A further example with negative costs

Globally \((x \leq 1)\)

\[\ell_0 \to +6 \to -6\]

\[x:=0 \to x=1\]

Safe bounds problems

- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?
A further example with negative costs

Globally \((x \leq 1)\)

\[
\ell_0 \xrightarrow{-3} \ell_1 \xrightarrow{+6} \ell_2 \xrightarrow{-6}
\]

\[x:=0 \quad x=1\]

Safe bounds problems

- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?
A further example with negative costs

Globally \((x \leq 1)\)

Safe bounds problems
- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?
A further example with negative costs

Globally \((x \leq 1)\)

\[-3 \rightarrow +6 \rightarrow -6\]

\(x := 0 \quad x = 1\)

Safe bounds problems

- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?
A further example with negative costs

Globally \((x \leq 1)\)

\[\begin{align*}
\ell_0 & \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad \quad 
\end{align*}\]

Safe bounds problems

- Lower-bound problem
- Lower-upper-bound problem: can we stay within bounds?
A further example with negative costs

Globally \((x \leq 1)\)

\[
\ell_0 \rightarrow \ell_1 \rightarrow \ell_2
\]

\[
\ell_0: -3 \quad \ell_1: +6 \quad \ell_2: -6
\]

\[
x := 0 \quad x = 1
\]

Safe bounds problems

- **Lower-bound problem**
- **Lower-upper-bound problem**: can we stay within bounds?
A further example with negative costs

Globally \((x \leq 1)\)

\[-3 \quad +6 \quad -6\]

\(\ell_0 \rightarrow \ell_1 \rightarrow \ell_2\)

\(x := 0 \quad x = 1\)

Safe bounds problems

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

lost!
A further example with negative costs

Globally \((x \leq 1)\)

\[
\ell_0 - 3 \rightarrow \ell_1 + 6 \rightarrow \ell_2 - 6
\]

\[
x = 0 \rightarrow x = 1
\]

Safe bounds problems
- Lower-bound problem
- Lower-upper-bound problem
- **Lower-weak-upper-bound problem**: can we “weakly” stay within bounds?
Games over timed automata

[Asarin, Maler, Pnueli, Sifakis 1998]
Games over timed automata

[Asarin, Maler, Pnueli, Sifakis 1998]

Question
Can we reach our goal whatever does the adversary?
A further example

\[ \ell_0, \ell_1, \ell_2, \ell_3 \]

\[ (x \leq 2) \rightarrow x \geq 1 \]

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

\[ x < 1 \]

\[ x \geq 2 \]

\[ x \leq 1 \]
Games over timed automata with costs

\[ \ell_0 + 5 \xrightarrow{x \leq 2, c, y := 0} \ell_1 \]

\[ \ell_1 \quad \text{(y=0)} \]

\[ \ell_1 \xrightarrow{u} \ell_2 \]

\[ \ell_2 \xrightarrow{x = 2, c + 1} \]

\[ \ell_2 \xrightarrow{x = 2, c} \ell_3 + 1 \]

\[ \ell_3 + 1 \xrightarrow{u} \]

\[ \ell_3 + 7 \]

\[ \ell_0 + 5 \]

\[ \ell_1 \quad \text{(y=0)} \]

\[ \ell_2 \xrightarrow{x = 2, c} \ell_3 + 1 \]

\[ \ell_3 + 7 \]

\[ \text{Question: What is the optimal cost we can ensure from } \ell_0? \]

\[ \inf_{0 \leq t \leq 2} \max \left( 5t + 10(2 - t) + 1, 5t + (2 - t) + 7 \right) = 14 + 13/3 \]

\[ \Rightarrow \text{strategy: wait in } \ell_0, \text{ and when } t = 4/3, \text{ go to } \ell_1 \]
Games over timed automata with costs

Question
What is the optimal cost we can ensure from $\ell_0$?
Games over timed automata with costs

Question

What is the optimal cost we can ensure from $\ell_0$?

$$5t + 10(2 - t) + 1$$
Games over timed automata with costs

\[ \ell_0 + 5 \xrightarrow{x \leq 2, c, y:=0} \ell_1 \]

\[ (y=0) \]

\[ \ell_1 \xrightarrow{u} \ell_2 \xrightarrow{+10} +1 \]

\[ \ell_2 \xrightarrow{x=2, c} +1 \]

\[ \ell_3 \xrightarrow{x=2, c} +7 \]

Question

What is the optimal cost we can ensure from \( \ell_0 \)?

\[ 5t + 10(2 - t) + 1 , \ 5t + (2 - t) + 7 \]
Games over timed automata with costs

Question

What is the optimal cost we can ensure from $\ell_0$?

$$\max \left( 5t + 10(2 - t) + 1, 5t + (2 - t) + 7 \right)$$
Games over timed automata with costs

Question

What is the optimal cost we can ensure from $\ell_0$?

$$\inf_{0 \leq t \leq 2} \max \left( 5t + 10(2 - t) + 1, 5t + (2 - t) + 7 \right) = 14 + \frac{1}{3}$$
Games over timed automata with costs

Question
What is the optimal cost we can ensure from \( \ell_0 \)?

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

\( \leadsto \) **strategy:** wait in \( \ell_0 \), and when \( t = \frac{4}{3} \), go to \( \ell_1 \)
What can we model with those features?

- Timed automata: systems with constraints on delays between events, on durations of tasks, etc.
  
  "Is any message delivered in no more than 5 minutes?"
What can we model with those features?

- **Timed automata**: systems with constraints on delays between events, on durations of tasks, *etc*.
  
  "*Is any message delivered in no more than 5 minutes?*"

- no real energy constraints can be expressed
What can we model with those features?

- **Timed automata**: systems with constraints on delays between events, on durations of tasks, *etc.*
  
  “*Is any message delivered in no more than 5 minutes?*”

  - no real energy constraints can be expressed

- **Timed automata with costs**: energy consumption, resources
  
  \[\leadsto \text{observe the quality of the system}\]
What can we model with those features?

- **Timed automata**: systems with constraints on delays between events, on durations of tasks, *etc.*
  
  "*Is any message delivered in no more than 5 minutes?*

  - no real energy constraints can be expressed

- **Timed automata with costs**: energy consumption, resources
  
  → observe the quality of the system

  - Optimization questions: minimal energy consumption, mean-cost optimization, minimization of the resources
    
    "*Can we minimize the power consumption w.r.t. the production?*"
What can we model with those features?

- **Timed automata:** systems with constraints on delays between events, on durations of tasks, *etc.*
  
  "Is any message delivered in no more than 5 minutes?"

- no real energy constraints can be expressed

- **Timed automata with costs:** energy consumption, resources
  
  ↦ observe the quality of the system

- **Optimization questions:** minimal energy consumption, mean-cost optimization, minimization of the resources
  
  "Can we minimize the power consumption w.r.t. the production?"

- **Safe bounds constraints:** check whether a system can stay alive with some amount of energy (that can possibly be regained)
  
  ex: laptop battery, autonomous robot
What can we model with those features?

- **Timed automata**: systems with constraints on delays between events, on durations of tasks, *etc.*
  
  *Is any message delivered in no more than 5 minutes?*

  - no real energy constraints can be expressed

- **Timed automata with costs**: energy consumption, resources
  
  → observe the quality of the system

- **Optimization questions**: minimal energy consumption, mean-cost optimization, minimization of the resources
  
  *Can we minimize the power consumption w.r.t. the production?*

- **Safe bounds constraints**: check whether a system can stay alive with some amount of energy (that can possibly be regained)
  
  *ex:* laptop battery, autonomous robot

- **Games over timed automata**: interaction with an environment (*open systems*)
What can we model with those features?

- **Timed automata**: systems with constraints on delays between events, on durations of tasks, *etc.*
  
  "*Is any message delivered in no more than 5 minutes?*"
  
  - no real energy constraints can be expressed

- **Timed automata with costs**: energy consumption, resources
  
  ↦ observe the quality of the system

  - **Optimization questions**: minimal energy consumption, mean-cost optimization, minimization of the resources
    
    "*Can we minimize the power consumption w.r.t. the production?*"

  - **Safe bounds constraints**: check whether a system can stay alive with some amount of energy (that can possibly be regained)
    
    ex: laptop battery, autonomous robot

- **Games over timed automata**: interaction with an environment (*open systems*)

- **Games over timed automata with costs**: when the above-mentioned features are combined
Which are the results?

A taste of the results

Adding cost (observer) variables to timed automata incredibly increases the difficulty of the problems

- Many problems become undecidable
- (Proofs of) algorithms become pretty much complex for restrictive decidable cases
Which are the results?

A taste of the results

😊 Adding cost (observer) variables to timed automata incredibly increases the difficulty of the problems
  - Many problems become undecidable
  - (Proofs of) algorithms become pretty much complex for restrictive decidable cases

😊 In several cases, algorithms have been developed, and case studies have been handled.

Tools that we use

- Automata theory
- Fixpoint computation
- Game reasoning
- Abstractions
- Linear programming
- etc.
Which are the results?

A taste of the results

😊 Adding cost (observer) variables to timed automata incredibly increases the difficulty of the problems
   - Many problems become undecidable
   - (Proofs of) algorithms become pretty much complex for restrictive decidable cases

😊 In several cases, algorithms have been developed, and case studies have been handled.

Tools that we use

- Automata theory
- Fixpoint computation
- Game reasoning
- Abstractions
- Linear programming
- *etc.*
How does this theory apply?

### Various tools are being developed

<table>
<thead>
<tr>
<th>Category</th>
<th>Tool(s)</th>
<th>Since</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid systems</td>
<td>HyTech</td>
<td>1995</td>
</tr>
<tr>
<td>Timed automata</td>
<td>Uppaal</td>
<td>1995</td>
</tr>
<tr>
<td>Timed automata with costs</td>
<td>Uppaal Cora</td>
<td>2001</td>
</tr>
<tr>
<td>Games on timed automata</td>
<td>Uppaal Tiga</td>
<td>2005</td>
</tr>
</tbody>
</table>

Case studies (a selection):

- **Purely timed systems**: An audio/video protocol (Bang & Olufsen) 1997
- Verification of SPSMALL (STMicroelectronics) 2008
- **Games on timed automata**: A climate controller in a pig stable (Skov A/S) 2007
- **Timed automata with costs**: Optimization questions (EU project Ametist) 2004
- Safe bounds constraints (EU project Quasimodo) 2004
- A pump system (Hydac Electronic GmbH) theory not yet understood
How does this theory apply?

Various tools are being developed

<table>
<thead>
<tr>
<th>Category</th>
<th>Tool</th>
<th>Since</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hybrid systems</td>
<td>HyTech</td>
<td>since 1995</td>
</tr>
<tr>
<td>Timed automata</td>
<td>Uppaal</td>
<td>since 1995</td>
</tr>
<tr>
<td>Timed automata with costs</td>
<td>Uppaal Cora</td>
<td>since 2001</td>
</tr>
<tr>
<td>Games on timed automata</td>
<td>Uppaal Tiga</td>
<td>since 2005</td>
</tr>
</tbody>
</table>

Case studies (a selection)

<table>
<thead>
<tr>
<th>Category</th>
<th>Example</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>Purely timed systems</td>
<td>An audio/video protocol (Bang &amp; Olufsen)</td>
<td>1997</td>
</tr>
<tr>
<td></td>
<td>Verification of SPSMALL (STMicroelectronics)</td>
<td>2008</td>
</tr>
</tbody>
</table>
How does this theory apply?

Various tools are being developed

- **Hybrid systems:** HyTech
  - since 1995
- **Timed automata:** Uppaal
  - since 1995
- **Timed automata with costs:** Uppaal Cora
  - since 2001
- **Games on timed automata:** Uppaal Tiga
  - since 2005

Case studies (a selection)

- **Purely timed systems:**
  - An audio/video protocol (Bang & Olufsen) 1997
  - Verification of SPSMALL (STMicroelectronics) 2008

- **Games on timed automata:**
  - A climate controller in a pig stable (Skov A/S) 2007

- Optimization questions (EU project Ametist)
  - A lacquer production planning problem (AXXOM) 2004

- Safe bounds constraints (EU project Quasimodo)
  - A pump system (Hydac Electronic GmbH)
How does this theory apply?

Various tools are being developed

- **Hybrid systems**: HyTech since 1995
- **Timed automata**: Uppaal since 1995
- **Timed automata with costs**: Uppaal Cora since 2001
- **Games on timed automata**: Uppaal Tiga since 2005

Case studies (a selection)

- **Purely timed systems**:
  - An audio/video protocol (Bang & Olufsen) 1997
  - Verification of SPSMALL (STMicroelectronics) 2008

- **Games on timed automata**:
  - A climate controller in a pig stable (Skov A/S) 2007

- **Timed automata with costs**:
  - *Optimization questions* (EU project Ametist) 2004
    - A lacquer production planning problem (AXXOM)
How does this theory apply?

Various tools are being developed
- Hybrid systems: HyTech since 1995
- Timed automata: Uppaal since 1995
- Timed automata with costs: Uppaal Cora since 2001
- Games on timed automata: Uppaal Tiga since 2005

Case studies (a selection)
- Purely timed systems:
  - An audio/video protocol (Bang & Olufsen) 1997
  - Verification of SPSMALL (STMicroelectronics) 2008
- Games on timed automata:
  - A climate controller in a pig stable (Skov A/S) 2007
- Timed automata with costs:
  - Optimization questions (EU project Ametist) 2004
    A lacquer production planning problem (AXXOM)
  - Safe bounds constraints (EU project Quasimodo) 2004
    A pump system (Hydac Electronic GmbH) theory not yet understood