Model Checking Real-Time Systems

Vincent Trélat

Technical University of Munich

December 12, 2022

Set of *time values*: $\mathbb{R}_{\geq 0}$

Timed words over $\Sigma \times \mathbb{R}_{\geq 0}$

Set of *valuations* over a set of clocks C: $\mathbb{R}_{\geq 0}^{C}$

Constraints over C: $\varphi ::= x \odot k \mid \varphi \wedge \varphi$ where $x \in C$, $k \in \mathbb{Z}$ and $\odot \in \{<, \leq, =, \geq, >\}$

Set of valuations *satisfying* φ: $[\![\varphi]\!]_C = \{v \in \mathbb{R}_{\geq 0}^{C} \mid v \models \varphi\}$

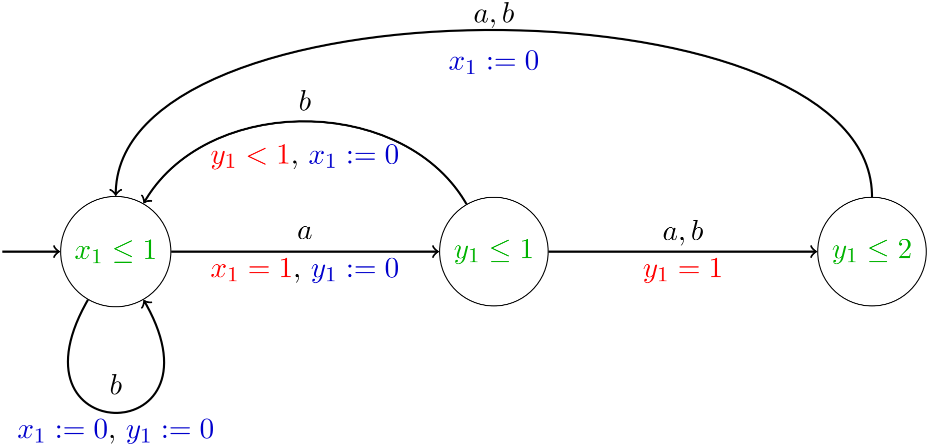
Definition 1

A *Timed Automaton* (TA) $\mathcal{A}$ is the tuple $(L, \ell_0, C, \Sigma, I, E)$ where:

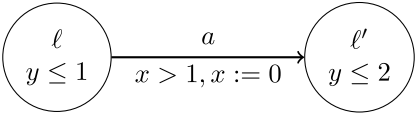
- L is a finite set of *locations* with initial location $\ell_0 \in L$
- C is a finite set of *clocks*
- Σ is a finite set of *actions*
- $I\colon L \to \Phi(C)$ is an *invariant mapping*
- $E \subseteq L \times \Phi(C) \times \Sigma \times 2^C \times L$ is a set of edges.

Edges are denoted by $\ell \xrightarrow{\varphi,a,r} \ell'$.

Example of a TA with 3 locations, 2 clocks and 2 actions (letters):

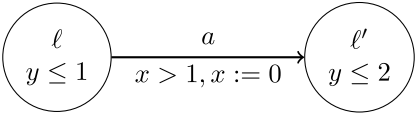


Operational Semantics



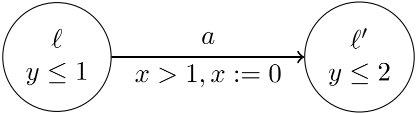
 Timed Automata

Operational Semantics



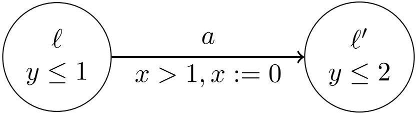
$$\begin{cases} v(x) = 1.2 \\ v(y) = 0.8 \end{cases}$$

Operational Semantics



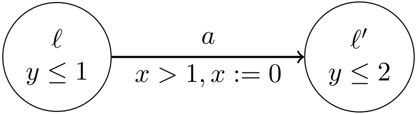
$$\begin{cases} v(x) = 1.2 \\ v(y) = 0.8 \end{cases} \qquad (\ell, v) \xrightarrow{0.5,\ a} (\ell', v')$$

Operational Semantics

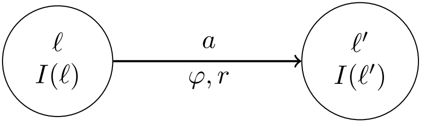


$$\begin{cases} v(x) = 1.2 \\ v(y) = 0.8 \end{cases} \qquad (\ell, v) \xrightarrow{0.5,\ a} (\ell', v') \qquad \begin{cases} v'(x) = 0 \\ v'(y) = 1.3 \end{cases}$$

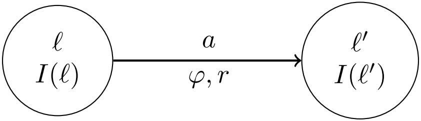
Operational Semantics



Operational Semantics



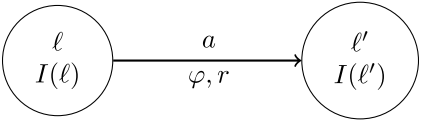
Operational Semantics



$$(\ell, v) \xrightarrow{d,\ a} (\ell', v')$$

 Timed Automata

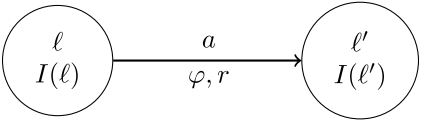
Operational Semantics



$$(\ell, v) \xrightarrow{d,\ a} (\ell', (v + d)[r])$$

 Timed Automata

Operational Semantics



$$(\ell, v) \xrightarrow{d,\ a} (\ell', (v+d)[r])$$

provided that:

$$\ell \xrightarrow{\varphi,a,r} \ell' \text{ is a transition in the TA}$$

$$\forall t \in [0, d],\ v + t \models I(\ell)$$

$$v + d \models I(\ell')$$

Operational Semantics

Definition 2

The *operational semantics* of a TA $A = (L, \ell_0, C, \Sigma, I, E)$ is the infinite-state timed transition system $[\![A]\!] = (S, s_0, \Sigma \times \mathbb{R}_{\geq 0}, T)$, where

$$S := \{(\ell, v) \in L \times \mathbb{R}^C_{\geq 0} \mid v \models I(\ell)\}, \quad s_0 := (\ell_0, \mathbf{0}_C),$$

$$T := \{(\ell, v) \xrightarrow{d,a} (\ell', (v+d)[r]) \mid d \in \mathbb{R}_{\geq 0},$$
$$\forall d' \in [0, d], v + d' \models I(\ell) \wedge \exists \ell \xrightarrow{\varphi,a,r} \ell' \in E, v + d \models \varphi\}$$

Region Equivalence

Definition 3

Two valuations $v, v' \in \mathbb{R}_{\geq 0}^{C}$ are region equivalent, i.e $v \cong_M v'$ iff:

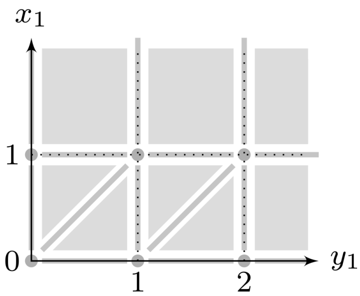
- $\forall x \in C, \lfloor v(x) \rfloor = \lfloor v'(x) \rfloor \ \vee \ v(x), v'(x) > M_x$
- $\forall x \in C, \langle v(x) \rangle = 0 \Leftrightarrow \langle v'(x) \rangle = 0 \ \vee \ v(x) \geq M_x$
- $\forall x, y \in C, \langle v(x) \rangle \leq \langle v(y) \rangle \Leftrightarrow \langle v'(x) \rangle \leq \langle v'(y) \rangle \ \vee \ v(x) > M_x \ \vee \ v(y) > M_y$

Region Equivalence

Definition 3

Two valuations $v, v' \in \mathbb{R}^{C}_{\geq 0}$ are region equivalent, i.e $v \cong_M v'$ iff:

- $\forall x \in C, \lfloor v(x) \rfloor = \lfloor v'(x) \rfloor \ \vee \ v(x), v'(x) > M_x$
- $\forall x \in C, \langle v(x) \rangle = 0 \Leftrightarrow \langle v'(x) \rangle = 0 \ \vee \ v(x) \geq M_x$
- $\forall x, y \in C, \langle v(x) \rangle \leq \langle v(y) \rangle \Leftrightarrow \langle v'(x) \rangle \leq \langle v'(y) \rangle \ \vee \ v(x) > M_x \ \vee \ v(y) > M_y$



Definition 4 (Region Automaton)

$\mathcal{R}_{\cong_M}(A) = (S, s_0, \Sigma, T)$ is the *region automaton* of A, where:

- $S := (L \times \mathbb{R}^C_{\geq 0})_{/\cong_M}$, $s_0 := [(\ell_0, \mathbf{0}_C)]_{/\cong_M}$
- $T := \{[\ell, v]_{\cong_M} \xrightarrow{a} [\ell', v']_{\cong_M} \mid \exists d \in \mathbb{R}_{\geq 0}, (\ell, v) \xrightarrow{d,a} (\ell', v')\}$

Definition 4 (Region Automaton)

$\mathcal{R}_{\cong_M}(A) = (S, s_0, \Sigma, T)$ is the *region automaton* of A, where:

- $S := (L \times \mathbb{R}^C_{\geq 0})_{/\cong_M}$, $s_0 := [(\ell_0, \mathbf{0}_C)]_{/\cong_M}$
- $T := \{[\ell, v]_{\cong_M} \xrightarrow{a} [\ell', v']_{\cong_M} \mid \exists d \in \mathbb{R}_{\geq 0}, (\ell, v) \xrightarrow{d,a} (\ell', v')\}$

- $|S|$ is exponential in the number of clocks and in the maximal constants of the timed automaton
- Is there a way to reduce the number of states?

Definition 5 (Zone)

A set of valuations $Z \subseteq \mathbb{R}_{\geq 0}^{C}$ is a zone iff: $\exists \varphi \in \Phi_d(C), Z = [\![\varphi]\!]_C$
In this case, we define:

- the *delay* of Z: $Z^{\uparrow} \triangleq \{v + d \mid V \in Z \wedge d \in \mathbb{R}_{\geq 0}\}$
- the *reset* of Z: $Z[r] \triangleq \{v[r] \mid v \in Z\}$ for $r \subseteq C$

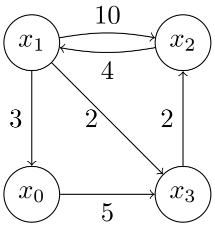
Definition 6 (Zone automaton)

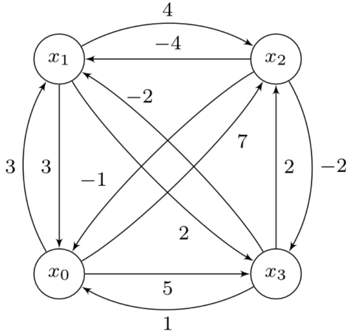
The *zone automaton* $[\![A]\!]_Z$ of A is the tuple $(S, s_0, \Sigma \cup \{\delta\}, T)$, where:

$$S := \{(\ell, Z) \mid \ell \in L, Z \in \mathbb{R}_{\geq 0}^{C} \text{ is a zone}\}, \quad s_0 := (\ell_0, [\![\mathbf{0}_C]\!]_C)$$
$$T := \{(\ell, Z) \overset{\delta}{\rightsquigarrow} (\ell, Z^{\uparrow} \cap [\![I(\ell)]\!]_C)\} \cup$$
$$\{(\ell, Z) \overset{a}{\rightsquigarrow} (\ell', (Z \cap [\![\varphi]\!]_C)[r] \cap [\![I(\ell')]\!]_C) \mid \ell \xrightarrow{\varphi, a, r} \ell' \in E\}$$

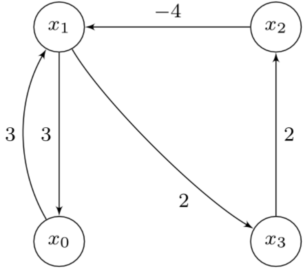
$\rightarrow$ Normalisation ($\simeq$ quotienting), shortest-path-closed DBM extrapolation

$$Z = \begin{cases} x_1 & \leq 3 \\ x_1 - x_2 & \leq 10 \\ x_1 - x_2 & \geq 4 \\ x_1 - x_3 & \leq 2 \\ x_3 - x_2 & \leq 2 \\ x_3 & \geq -5 \end{cases}$$





Shortest-path closure



Shortest-path reduction

Decidable extensions:

- *diagonal constraints*: $x - y \odot k$
- *updatable* TA: clocks can be reset to any natural number ($x := k \in \mathbb{Z}$) or can be synchronized with another clock ($x := y$)
- *urgency* constraints: some locations must be left immediately

Undecidable extensions:

- *linear* constraints: $\lambda x + \mu y \leq k$
- updates such as $x := x + k$ or $x :> k$ (non-determinism)
- *stopwatch* automata
- *hybrid* automata
- ...

Decidable extensions:

- *diagonal constraints*: $x - y \odot k$
- *updatable* TA: clocks can be reset to any natural number ($x := k \in \mathbb{Z}$) or can be synchronized with another clock ($x := y$)
- *urgency* constraints: some locations must be left immediately

Undecidable extensions:

- *linear* constraints: $\lambda x + \mu y \leq k$
- updates such as $x := x + k$ or $x :> k$ (non-determinism)
- *stopwatch* automata
- *hybrid* automata
- pretty much everything else you can think of

Definition 7 (Metric Temporal Logic)

Given a set of atomic propositions P, the formulas of MTL are defined for any time interval I with the *time-constrained until* operator $\mathbf{U}_I$ as follows:

$$\varphi ::= p \in P \mid \neg\varphi \mid \varphi \land \varphi \mid \varphi \mathbf{U}_I \varphi$$

The *constrained always* $\Box_I$ and *constrained eventually* $\Diamond_I$ operators can be defined with $\mathbf{U}_I$ in a similar way as in LTL, namely:

$$\Diamond_I \varphi \triangleq \top \mathbf{U}_I \varphi$$
$$\Box_I \varphi \triangleq \neg \Diamond_I \neg \varphi$$

Definition 7 (Metric Temporal Logic)

Given a set of atomic propositions P, the formulas of MTL are defined for any time interval I with the *time-constrained until* operator $\mathbf{U}_I$ as follows:

$$\varphi ::= p \in P \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathbf{U}_I \varphi$$

The *constrained always* $\Box_I$ and *constrained eventually* $\Diamond_I$ operators can be defined with $\mathbf{U}_I$ in a similar way as in LTL, namely:

$$\Diamond_I \varphi \triangleq \top \mathbf{U}_I \varphi$$

$$\Box_I \varphi \triangleq \neg \Diamond_I \neg \varphi$$

What is $\mathbf{U}_I$?!

Definition 7 (Metric Temporal Logic)

Given a set of atomic propositions P, the formulas of MTL are defined for any time interval I with the *time-constrained until* operator $\mathbf{U}_I$ as follows:

$$\varphi ::= p \in P \mid \neg\varphi \mid \varphi \wedge \varphi \mid \varphi \mathbf{U}_I \varphi$$

The *constrained always* $\Box_I$ and *constrained eventually* $\Diamond_I$ operators can be defined with $\mathbf{U}_I$ in a similar way as in LTL, namely:

$$\Diamond_I \varphi \triangleq \top \mathbf{U}_I \varphi$$

$$\Box_I \varphi \triangleq \neg \Diamond_I \neg \varphi$$

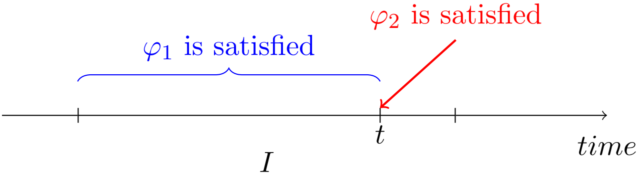
$\boxed{\text{What is } \mathbf{U}_I\text{?!}}$ $\rightarrow$ give its semantics

Continuous semantics:

$$f \models \varphi_1 \mathbf{U}_I \varphi_2 \quad \text{iff} \quad \exists t \in I, f^t \models \varphi_2 \wedge \forall u \in (0,t), f^u \models \varphi_1$$

Continuous semantics:

$$f \models \varphi_1 \mathbf{U}_I \varphi_2 \quad \text{iff} \quad \exists t \in I, f^t \models \varphi_2 \land \forall u \in (0,t), f^u \models \varphi_1$$



Pointwise semantics: Let $w = ((a_i, t_i))_{i \in \mathbb{N}}$ be a timed word over 2^P.

$$w, i \models \varphi_1 \mathbf{U}_I \varphi_2 \quad \text{iff} \quad \exists i < j < |w|, \begin{cases} w, j \models \varphi_2 & \wedge \\ t_j - t_i \in I & \wedge \\ \forall i < k < j,\ w, k \models \varphi_1 \end{cases}$$

<Insert figure>

References