Real-Time Systems

Lecture 11: Timed Automata

2017-12-07

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany
Content

Introduction

- Observables and Evolutions
- Duration Calculus (DC)
- Semantical Correctness Proofs
- DC Decidability
- DC Implementables
- PLC-Automata

Timed Automata (TA), Uppaal
- Networks of Timed Automata
- Region/Zone-Abstraction
- TA model-checking
- Extended Timed Automata
- Undecidability Results

\[ \text{obs} : \text{Time} \rightarrow \mathcal{D}(\text{obs}) \]

\[ \langle \text{obs}_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_0} \langle \text{obs}_1, \nu_1 \rangle, t_1 \ldots \]

- Automatic Verification...
  ...whether a TA satisfies a DC formula, observer-based

- Recent Results:
  - Timed Sequence Diagrams, or Quasi-equal Clocks,
    or Automatic Code Generation, or …
Content

- Timed Automata Syntax
  - Channels, Actions, Clock Constraints
  - Pure Timed Automaton
  - Graphical Representation of TA

- Timed Automata (Operational) Semantics
  - Clock Valuations, Time Shift, Modification
  - The Labelled Transition System
    - Configurations
    - Delay transitions
    - Action transitions
  - Transition Sequences, Reachability
  - Computation Paths
  - Timelocks and Zeno behaviour
  - Runs
(Pure) Timed Automata Syntax
To define timed automata formally, we need the following sets of symbols:

- A set \((a, b \in \text{Chan})\) of **channel names** or **channels**.

- For each channel \(a \in \text{Chan}\), two **visible actions**: \(a?\) and \(a!\) denote **input** and **output** on the **channel** \((a?, a! \notin \text{Chan})\).

- \(\tau \notin \text{Chan}\) represents an **internal action**, not visible from outside.

- \((\alpha, \beta \in \text{Act})\) is the set of **actions**.

- An **alphabet** \(B\) is a set of **channels**, i.e. \(B \subseteq \text{Chan}\).

- For each alphabet \(B\), we define the corresponding **action set**

\[
B?! := \{a? \mid a \in B\} \cup \{a! \mid a \in B\} \cup \{\tau\}.
\]

- **Note**: \(\text{Chan}?! = \text{Act}\).
Example: Desktop Lamp

- \( B = \{\text{press}\} \) – **alphabet** of the desktop lamp model

- channel ‘press’ models the single button of the desktop lamp

- **Output**: press!  
  (“send a message onto channel press”)  
  - models “the button is pressed”

- **Input**: press?  
  (“receive a message from channel press”)  
  - models “button pressed is recognised”

- **Actions**:  
  \[ \{\text{press}!, \text{press}?, \tau\} = B!? \]
Simple Clock Constraints

- Let \((x, y \in X)\) be a set of clock variables (or clocks).

- The set \((\varphi \in \Phi(X))\) of (simple) clock constraints (over \(X\)) is defined by the following grammar:

\[
\varphi ::= x \sim c \mid x - y \sim c \mid \varphi_1 \land \varphi_2
\]

where

- \(x, y \in X\),
- \(c \in \mathbb{Q}_{\geq 0}\), and
- \(\sim \in \{<, >, \leq, \geq\}\).

- Clock constraints of the form \(x - y \sim c\) are called difference constraints.

Examples: Let \(X = \{x, y\}\).

\(x \leq 3\), \(x > 3\) (strictly speaking not a clock constraint: \(3 \geq x\))

\(y < 2\), \(y > 3\)
Definition 4.3. [@-timed automaton@] A (pure) timed automaton $A$ is a structure

$$A = (L, B, X, I, E, \ell_{ini})$$

where

- $(\ell \in) L$ is a finite set of locations (or control states),
- $B \subseteq \text{Chan}$ is an alphabet,
- $X$ is a finite set of clocks,
- $I : L \rightarrow \Phi(X)$ assigns to each location a clock constraint, its invariant,
- $E \subseteq L \times B_? \times \Phi(X) \times 2^X \times L$ a finite set of directed edges. Edges $(\ell, \alpha, \varphi, Y, \ell')$ from location $\ell$ to $\ell'$ are labelled with an action $\alpha$, a guard $\varphi$, and a set $Y$ of clocks that will be reset.
- $\ell_{ini}$ is the initial location.
Example

\[ A = (L, B, X, I, E, \ell_{ini}) \]

- **Locations**: \( L = \{\text{off, light, bright}\} \)
- **Alphabet**: \( B = \{\text{press}\} \)
- **Clocks**: \( X = \{x\} \)
- **Invariants**: \( I = \{\text{off} \mapsto \text{true}, \text{light} \mapsto \text{true}, \text{bright} \mapsto \text{true}\} \)
- **Edges**: \( E = \{(\text{off}, \text{press}?, \text{true}, \{x\}, \text{light}), (\text{light}, \text{press}?, x > 3, \emptyset, \text{off}), (\text{light}, \text{press}?, x \leq 3, \emptyset, \text{bright}), (\text{bright}, \text{press}?, \text{true}, \emptyset, \text{off})\} \)
- **Initial Location**: \( \ell_{ini} = \text{off} \)
Graphical Representation of Timed Automata

\[ A = (L, B, X, I, E, \ell_{\text{ini}}) \]

- \( I : L \rightarrow \Phi(X) \)
- \( E \subseteq L \times B \times \Phi(X) \times 2^X \times L \)

- **Locations** (control states) \( \ell \) and their **invariants** \( I(\ell) \):

  - \( \ell \quad I(\ell) \) or \( \ell \quad I(\ell) \) if not explicitly given

- **Initial location** \( \ell_{\text{ini}} \):

  - \( \tau \) if not explicitly given

- **Edges**: \( (\ell, \alpha, \varphi, Y, \ell') \in L \times B \times \Phi(X) \times 2^X \times L \)

  - \( \ell \quad \alpha \quad \ell' \) if not explicitly given
  - \( \varphi \) if not explicitly given; \( x := 0 \) denotes \( \{x\} \)
• **Locations:** $L = \{ \text{off, light, bright} \}$

• **Alphabet:** $B = \{ \text{press} \}$,

• **Clocks:** $X = \{ x \}$,

• **Invariants:** $I = \{ \text{off} \mapsto true, \text{light} \mapsto true, \text{bright} \mapsto true \}$

• **Edges:**
  
  $E = \{ (\text{off}, \text{press?}, true, \{ x \}, \text{light}), (\text{light}, \text{press?}, x > 3, \emptyset, \text{off}),$
  
  $(\text{light}, \text{press?}, x \leq 3, \emptyset, \text{bright}), (\text{bright}, \text{press?}, true, \emptyset, \text{off}) \}$

• **Initial Location:** $\ell_{\text{ini}} = \text{off}$
• **Locations:** $L = \{off, light, bright\}$
• **Alphabet:** $B = \{press\}$,
• **Clocks:** $X = \{x\}$,
• **Invariants:** $I = \{off \mapsto true, light \mapsto true, bright \mapsto true\}$
• **Edges:** $E = \{(off, press?, true, \{x\}, light), (light, press?, x > 3, \emptyset, off),$
  $(light, press?, x \leq 3, \emptyset, bright), (bright, press?, true, \emptyset, off)\}$
• **Initial Location:** $l_{ini} = off$
Example

- **Locations**: $L = \{\text{off, light, bright}\}$
- **Alphabet**: $B = \{\text{press}\}$,
- **Clocks**: $X = \{x\}$,
- **Invariants**: $I = \{\text{off} \mapsto \text{true}, \text{light} \mapsto \text{true}, \text{bright} \mapsto \text{true}\}$
- **Edges**: $E = \{ (\text{off}, \text{press}?, \text{true}, \{x\}, \text{light}), (\text{light}, \text{press}?, x > 3, \emptyset, \text{off}), (\text{light}, \text{press}?, x \leq 3, \emptyset, \text{bright}), (\text{bright}, \text{press}?, \text{true}, \emptyset, \text{off}) \}$
- **Initial Location**: $\ell_{\text{ini}} = \text{off}$
Example

- **Locations:** \( L = \{\text{off}, \text{light}, \text{bright}\} \)
- **Alphabet:** \( B = \{\text{press}\}, \)
- **Clocks:** \( X = \{x\}, \)
- **Invariants:** \( I = \{\text{off} \mapsto true, \text{light} \mapsto true, \text{bright} \mapsto true\} \)
- **Edges:** \( E = \{ (\text{off}, \text{press?}, \text{true}, \{x\}, \text{light}), (\text{light}, \text{press?}, x > 3, \emptyset, \text{off}), (\text{light}, \text{press?}, x \leq 3, \emptyset, \text{bright}), (\text{bright}, \text{press?}, \text{true}, \emptyset, \text{off})\} \)
- **Initial Location:** \( l_{\text{ini}} = \text{off} \)

![Diagram](image-url)
- **Locations:** $L = \{\text{off, light, bright}\}$
- **Alphabet:** $B = \{\text{press}\}$,
- **Clocks:** $X = \{x\}$,
- **Invariants:** $I = \{\text{off} \mapsto \text{true}, \text{light} \mapsto \text{true}, \text{bright} \mapsto \text{true}\}$
- **Edges:** $E = \{(\text{off, press?}, \text{true, } \{x\}, \text{light}), (\text{light, press?}, x > 3, \emptyset, \text{off}),$
  $(\text{light, press?}, x \leq 3, \emptyset, \text{bright}), (\text{bright, press?}, \text{true}, \emptyset, \text{off})\}$
- **Initial Location:** $l_{\text{ini}} = \text{off}$
Example

- **Locations**: $L = \{\text{off, light, bright}\}$
- **Alphabet**: $B = \{\text{press}\}$,
- **Clocks**: $X = \{x\}$,
- **Invariants**: $I = \{\text{off} \mapsto true, \text{light} \mapsto true, \text{bright} \mapsto true\}$
- **Edges**: $E = \{(\text{off}, \text{press}?, \text{true}, \{x\}, \text{light}), (\text{light}, \text{press}?, x > 3, \emptyset, \text{off}), (\text{light}, \text{press}?, x \leq 3, \emptyset, \text{bright}), (\text{bright}, \text{press}?, \text{true}, \emptyset, \text{off})\}$
- **Initial Location**: $\ell_{\text{ini}} = \text{off}$
- **Locations**: $L = \{\text{off, light, bright}\}$
- **Alphabet**: $B = \{\text{press}\}$,
- **Clocks**: $X = \{x\}$,
- **Invariants**: $I = \{\text{off} \mapsto \text{true}, \text{light} \mapsto \text{true}, \text{bright} \mapsto \text{true}\}$
- **Edges**: $E = \{(\text{off, press?}, \text{true}, \{x\}, \text{light}), (\text{light, press?}, x > 3, \emptyset, \text{off}), (\text{light, press?}, x \leq 3, \emptyset, \text{bright}), (\text{bright, press?}, \text{true}, \emptyset, \text{off})\}$
- **Initial Location**: $\ell_{\text{ini}} = \text{off}$
Timed Automata Syntax
- Channels, Actions, Clock Constraints
- Pure Timed Automaton
- Graphical Representation of TA

Timed Automata (Operational) Semantics
- Clock Valuations, Time Shift, Modification
- The Labelled Transition System
  - Configurations
  - Delay transitions
  - Action transitions
- Transition Sequences, Reachability
- Computation Paths
- Timelocks and Zeno behaviour
- Runs
Pure TA Operational Semantics
Clock Valuations

- Let $X$ be a set of clocks. A **valuation** $\nu$ of clocks in $X$ is a mapping
  \[ \nu : X \rightarrow \text{Time} \]
  assigning each clock $x \in X$ the **current time** $\nu(x)$.

- Let $\varphi$ be a clock constraint. The **satisfaction** relation between clock valuations $\nu$ and clock constraints $\varphi$, denoted by $\nu \models \varphi$, is defined inductively:
  - $\nu \models x \sim c$ iff $\nu(x) \wedge c$
  - $\nu \models x - y \sim c$ iff $\nu(x) \wedge \ nu(y) \wedge c$
  - $\nu \models \varphi_1 \land \varphi_2$ iff $\nu \models \varphi_1$ and $\nu \models \varphi_2$
Let $X$ be a set of clocks. A **valuation** $\nu$ of clocks in $X$ is a mapping

$$\nu : X \rightarrow \text{Time}$$

assigning each clock $x \in X$ the **current time** $\nu(x)$.

Let $\varphi$ be a clock constraint. The **satisfaction** relation between clock valuations $\nu$ and clock constraints $\varphi$, denoted by $\nu \models \varphi$, is defined inductively:

- $\nu \models x \sim c$ iff $\nu(x) \sim c$
- $\nu \models x - y \sim c$ iff $\nu(x) - \nu(y) \sim c$
- $\nu \models \varphi_1 \land \varphi_2$ iff $\nu \models \varphi_1$ and $\nu \models \varphi_2$

Two clock constraints $\varphi_1$ and $\varphi_2$ are called **(logically) equivalent** if and only if for all clock valuations $\nu$, we have

$$\nu \models \varphi_1 \text{ if and only if } \nu \models \varphi_2.$$  

In that case we write $\models \varphi_1 \iff \varphi_2$. 
Let $\nu$ be a valuation of clocks in $X$ and $t \in \text{Time}$.

- **Time Shift**
  We write $\nu + t$ to denote the clock valuation (for $X$) with
  \[
  (\nu + t)(x) = \nu(x) + t.
  \]
  for all $x \in X$,

- **Modification / Update**
  Let $Y \subseteq X$ be a set of clocks.
  We write $\nu[Y := t]$ to denote the clock valuation with
  \[
  (\nu[Y := t])(x) = \begin{cases} 
  t & \text{if } x \in Y \\
  \nu(x) & \text{otherwise}
  \end{cases}
  \]
  Special case **reset**: $t = 0$. 

\[
\nu : \mathcal{X} \mapsto 3.0$
\[
(\nu + 0.2\tau)(\mathcal{x}) = \nu(\mathcal{x}) + 0.2\tau = 3.0 + 0.2\tau = 3.2\tau
\]
Definition 4.4. The operational semantics of a timed automaton $A = (L, B, X, I, E, \ell_{ini})$ is defined by the (labelled) transition system

$$\mathcal{T}(A) = (Conf(A), Time \cup B?! , \{ \xrightarrow{\lambda} | \lambda \in Time \cup B?! \}, C_{ini})$$

where

- $Conf(A) = \{ \langle \ell, \nu \rangle | \ell \in L, \nu : X \to Time, \nu \models I(\ell) \}$
- $Time \cup B?!$ are the transition labels,
- there are delay transition relations
  $$\langle \ell, \nu \rangle \xrightarrow{\lambda} \langle \ell', \nu' \rangle, \quad \lambda \in Time$$  ($\to$ in a minute)

  and action transition relations
  $$\langle \ell, \nu \rangle \xrightarrow{\lambda} \langle \ell', \nu' \rangle, \quad \lambda \in B?!.$$  ($\to$ in a minute)

- $C_{ini} = \{ \langle \ell_{ini}, \nu_0 \rangle \} \cap Conf(A)$ with $\nu_0(x) = 0$ for all $x \in X$

  is the set of initial configurations.
Operational Semantics of TA Cont’d

\[ A = (L, B, X, I, E, \ell_{ini}) \]

\[ \mathcal{T}(A) = (\text{Conf}(A), \text{Time} \cup B?!, \{ \lambda \mapsto | \lambda \in \text{Time} \cup B?! \}, C_{ini}) \]

- **Time or delay transition:**

  \((\langle \ell, \nu \rangle, \langle \ell, \nu + t \rangle) \in \xrightarrow{\xi} \)

  if and only if \(\forall t' \in [0, t] : \nu + t' \models I(\ell)\).

  “Some time \(t \in \text{Time}\) elapses respecting invariants, location unchanged.”

- **Action or discrete transition:**

  \(\langle \ell, \nu \rangle \xrightarrow{\alpha} \langle \ell', \nu' \rangle\)

  if and only if there is \((\ell, \alpha, \varphi, Y, \ell') \in E\) such that

  \(\nu \models \varphi, \quad \nu' = \nu[Y := 0], \quad \text{and} \quad \nu' \models I(\ell').\)

  “An action occurs, location may change, some clocks may be reset, time does not elapse.”
Example

- **Configurations:**

  \[ \text{Conf}(A) = \{ \langle \text{off}, \nu \rangle, \langle \text{light}, \nu \rangle, \langle \text{light}, \nu \rangle \mid \nu : X \rightarrow \text{Time} \} \]

- **Initial Configurations:**

  \[ \{ \langle \text{off}, \nu_0 \rangle \} \cap \text{Conf}(A) = \{ \langle \text{off}, \exists x \mapsto 0 \rangle \} \]

- **Delay Transition:**

  \[ \langle \text{off}, \{ x \mapsto 0 \} \rangle \xrightarrow{27} \langle \text{off}, \{ x \mapsto 27 \} \rangle \]

- **Action Transition:**

  \[ \langle \text{off}, \{ x \mapsto 27 \} \rangle \xrightarrow{\text{press?}} \langle \text{light}, \{ x \mapsto 0 \} \rangle \]
A transition sequence of $A$ is any finite or infinite sequence of the form

\[
\langle \ell_0, \nu_0 \rangle \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle \xrightarrow{\lambda_3} \ldots
\]

with

- $\langle \ell_0, \nu_0 \rangle \in C_{ini}$,
- for all $i \in \mathbb{N}$, there is $\xrightarrow{\lambda_{i+1}}$ in $\mathcal{T}(A)$ with $\langle \ell_i, \nu_i \rangle \xrightarrow{\lambda_{i+1}} \langle \ell_{i+1}, \nu_{i+1} \rangle$
Example

\[ \langle \text{off}, x = 0 \rangle \xrightarrow{2.5} \langle \text{off}, x = 2.5 \rangle \]
\[ \quad \xrightarrow{1.7} \langle \text{off}, x = 4.2 \rangle \]
\[ \xrightarrow{\text{press?}} \langle \text{light}, x = 0 \rangle \]
\[ \quad \xrightarrow{2.1} \langle \text{light}, x = 2.1 \rangle \]
\[ \xrightarrow{\text{press?}} \langle \text{bright}, x = 2.1 \rangle \]
\[ \quad \xrightarrow{10} \langle \text{bright}, x = 12.1 \rangle \]
\[ \xrightarrow{\text{press?}} \langle \text{off}, x = 12.1 \rangle \]
\[ \xrightarrow{\text{press?}} \langle \text{light}, x = 0 \rangle \]

\[ \langle \text{light}, x = 0 \rangle \xrightarrow{0} \langle \text{light}, x = 0 \rangle \]
Reachability

- A configuration $\langle \ell, \nu \rangle$ is called reachable (in $A$) if and only if there is a transition sequence of the form

$$\langle \ell_0, \nu_0 \rangle \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle \xrightarrow{\lambda_3} \ldots \xrightarrow{\lambda_n} \langle \ell_n, \nu_n \rangle = \langle \ell, \nu \rangle$$

- A location $\ell$ is called reachable if and only if any configuration $\langle \ell, \nu \rangle$ is reachable, i.e. there exists a valuation $\nu$ such that $\langle \ell, \nu \rangle$ is reachable.
Recall: \( \text{Conf}(A) = \{ \langle \ell, \nu \rangle \mid \ell \in L, \nu : X \to \text{Time}, \nu \models I(\ell) \} \)

Example:

- Configurations:
  - \( \text{Conf}(A) = \{ \langle \ell_0, \nu \rangle, \langle \ell_2, \nu \rangle \mid \nu : \{y\} \to \text{Time} \} \cup \{ \langle \ell_1, \nu \rangle \mid \nu : \{y\} \to [0, 2] \} \)
  - \( \langle \ell_1, y \mapsto 1.01 \rangle \) is a configuration,
  - \( \langle \ell_1, y \mapsto 27 \rangle \) is not a configuration,
  - \( \langle \ell_0, y \mapsto 0 \rangle \xrightarrow{0.707} \langle \ell_0, y \mapsto 0.707 \rangle \xrightarrow{\text{press!}} \langle \ell_1, y \mapsto 0.707 \rangle \) is a transition sequence
  - \( \langle \ell_0, y \mapsto 0 \rangle \xrightarrow{27} \langle \ell_0, y \mapsto 27 \rangle \) is a transition sequence
  - \( \langle \ell_0, y \mapsto 0 \rangle \xrightarrow{27} \langle \ell_0, y \mapsto 27 \rangle \xrightarrow{\text{press!}} \langle \ell_1, y \mapsto 27 \rangle \) is not a transition sequence
Two Approaches to Exclude “Bad” Configurations

- The approach taken for TA:
  - Rule out bad configurations in the step from $\mathcal{A}$ to $\mathcal{T}(\mathcal{A})$.
  - “Bad” configurations are not even configurations!

- Recall Definition 4.4:
  - $Conf(\mathcal{A}) = \{ \langle \ell, \nu \rangle \mid \ell \in L, \nu : X \rightarrow Time, \nu \models I(\ell) \}$
  - $C_{ini} = \{ \langle \ell_{ini}, \nu_0 \rangle \} \cap Conf(\mathcal{A})$

- The approach not taken for TA:
  - consider every $\langle \ell, \nu \rangle$ to be a configuration, i.e. have
    $$Conf(\mathcal{A}) = \{ \langle \ell, \nu \rangle \mid \ell \in L, \nu : X \rightarrow Time \}$$
  - “bad” configurations not in transition relation with others, i.e. have, e.g.,
    $$\langle \ell, \nu \rangle \xrightarrow{t} \langle \ell, \nu + t \rangle$$
    if and only if $\forall t' \in [0, t] : \nu + t' \models I(\ell)$ and $\nu + t' \models I(\ell')$. 
Content

- Timed Automata Syntax
  - Channels, Actions, Clock Constraints
  - Pure Timed Automaton
  - Graphical Representation of TA

- Timed Automata (Operational) Semantics
  - Clock Valuations, Time Shift, Modification
  - The Labelled Transition System
    - Configurations
    - Delay transitions
    - Action transitions
    - Transition Sequences, Reachability
  - Computation Paths
  - Timelocks and Zeno behaviour
  - Runs
Computation Path, Run
Time Stamped Configurations

- \langle \ell, \nu \rangle, t is called **time-stamped configuration**

- **Time-stamped delay transition:**

  \[ \langle \ell, \nu \rangle, t \xrightarrow{t'} \langle \ell, \nu + t' \rangle, t + t' \quad \text{iff } t' \in \text{Time and } \langle \ell, \nu \rangle \xrightarrow{t'} \langle \ell, \nu + t' \rangle. \]

- **Time-stamped action transition:**

  \[ \langle \ell, \nu \rangle, t \xrightarrow{\alpha} \langle \ell', \nu' \rangle, t \quad \text{iff } \alpha \in B \text{ and } \langle \ell, \nu \rangle \xrightarrow{\alpha} \langle \ell', \nu' \rangle. \]
A sequence of time-stamped configurations

\[ \xi = \langle \ell_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle, t_1 \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle, t_2 \xrightarrow{\lambda_3} \ldots \]

is called a computation path (or path) of \( A \) starting in \( \langle \ell_0, \nu_0 \rangle, t_0 \) if and only if it is either infinite or maximally finite (wrt. the time stamped transition relations).

A computation path (or path) of \( A \) is a computation path starting in \( \langle \ell_0, \nu_0 \rangle, 0 \) with \( \langle \ell_0, \nu_0 \rangle \in C_{ini} \).
Timelocks and Zeno Behaviour

- Configuration \( \langle \ell, \nu \rangle \) is called timelock iff no delay transitions with \( t > 0 \) from \( \langle \ell, \nu \rangle \)

Examples:

- \( \langle \ell, x = 0 \rangle, 0 \xrightarrow{\frac{2}{3}} \langle \ell, x = 2 \rangle, 2 \)

- \( \langle \ell', x = 0 \rangle, 0 \xrightarrow{3} \langle \ell', x = 3 \rangle, 3 \xrightarrow{a?} \langle \ell', x = 3 \rangle, 3 \xrightarrow{a?} \ldots \)

Zeno behaviour:

- \( \langle \ell, x = 0 \rangle, 0 \xrightarrow{\frac{1}{2}} \langle \ell, x = \frac{1}{2} \rangle, \frac{1}{2} \xrightarrow{\frac{1}{4}} \langle \ell, x = \frac{3}{4} \rangle, \frac{3}{4} \ldots \xrightarrow{\frac{1}{2^n}} \langle \ell, x = \frac{2^n-1}{2^n} \rangle, \frac{2^n-1}{2^n} \ldots \)

- \( \langle \ell, x = 0 \rangle, 0 \xrightarrow{0.1} \langle \ell, x = 0.1 \rangle, 0.1 \xrightarrow{0.01} \langle \ell, x = 0.11 \rangle, 0.11 \xrightarrow{0.001} \langle \ell, x = 0.111 \rangle, 0.111 \ldots \)
Definition 4.9. An infinite sequence

\[ t_0, t_1, t_2, \ldots \]

of values \( t_i \in \text{Time} \) for \( i \in \mathbb{N}_0 \) is called real-time sequence if and only if it has the following properties:

- **Monotonicity:**
  \[ \forall i \in \mathbb{N}_0 : t_i \leq t_{i+1} \]

- **Non-Zeno behaviour (or unboundedness (or progress)):**
  \[ \forall t \in \text{Time} \exists i \in \mathbb{N}_0 : t < t_i \]
Definition 4.10. A run of $A$ starting in $\langle \ell_0, \nu_0 \rangle, t_0$ is an infinite computation path

$$\xi = \langle \ell_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle, t_1 \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle, t_2 \xrightarrow{\lambda_3} \ldots$$

of $A$ where $\langle t_i \rangle_{i \in \mathbb{N}_0}$ is a real-time sequence.

We call $\xi$ a run of $A$ if and only if $\xi$ is a computation path of $A$.

Example:

$$\ell$$

$x \leq 2$
Content

- Timed Automata Syntax
  - Channels, Actions, Clock Constraints
  - Pure Timed Automaton
  - Graphical Representation of TA

- Timed Automata (Operational) Semantics
  - Clock Valuations, Time Shift, Modification
  - The Labelled Transition System
    - Configurations
    - Delay transitions
    - Action transitions
  - Transition Sequences, Reachability
  - Computation Paths
  - Timelocks and Zeno behaviour
  - Runs
A timed automaton is basically a finite automaton with actions, guards, invariants, and resets of clocks.

The (operational) semantics of TA is a labelled transition system with delay transitions (where locations do not change), and action transitions (where time does not elapse).

We distinguish:
- Transition Sequences: without timestamps
- Computation Paths: with timestamps,
- Runs: timestamps form a real-time sequence.

The reachability problem is an important decision problem for timed automata.
References
References