Real-Time Systems

Lecture 11: Timed Automata

2017-12-07

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Content

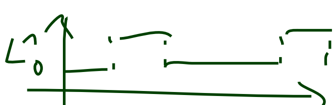
Introduction

- **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

$obs : \text{Time} \to \mathscr{D}(obs)$

$\langle obs_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_0} \langle obs_1, \nu_1 \rangle, t_1 \ldots$



- **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

(Pure) Timed Automata Syntax

Channel Names and Actions

To define timed automata formally, we need the following sets of symbols:

- A set $(a, b \in)$ Chan of **channel names** or **channels**.
- For each channel $a \in$ Chan, two **visible actions**:
  $a?$ and $a!$ denote **input** and **output** on the **channel** ($a?, a! \notin$ Chan).
- $\tau \notin$ Chan represents an **internal action**, not visible from outside.
- $(\alpha, \beta \in)$ $Act := \{a? \mid a \in \text{Chan}\} \cup \{a! \mid a \in \text{Chan}\} \cup \{\tau\}$
  is the set of **actions**.
- An **alphabet** B is a set of **channels**, i.e. $B \subseteq$ 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: $\text{Chan}_{?!} = Act$.

Example: Desktop Lamp

- $B = \{press\}$ – **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**:

$$\{press!, press?, \tau\} = B_{!?}$$

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:

$$\varphi ::= x \sim c \mid x - y \sim c \mid \varphi_1 \wedge \varphi_2$$

  where
  - $x, y \in X$,
  - $c \in \mathbb{Q}_0^+$, and
  - $\sim \in \{<, >, \leq, \geq\}$.

- Clock constraints of the form $x - y \sim c$ are called **difference constraints**.

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

$x \leq 3$ ✓

- $x \leq 3$, $x > 3$ (strictly speaking not a clock constraint: $3 \geq x$)
- $y < 2$, $y > 3$

Timed Automaton

Definition 4.3. [*Timed automaton*]
A (pure) **timed automaton** $\mathcal{A}$ is a structure

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

where

- $(\ell \in)$ L is a finite set of **locations** (or **control states**),
- $B \subseteq$ Chan is an alphabet,
- X is a finite set of clocks,
- $I : L \to \Phi(X)$ 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**.

  powerset

  Edges $(\ell, \alpha, \varphi, Y, \ell')$ from location ℓ to ℓ' are labelled with an **action** α, a **guard** φ, and a set Y of clocks that will be **reset**.
- ℓ_{ini} is the **initial location**.

Example

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

- $I : L \to \Phi(X)$,
- $E \subseteq L \times B_{?!} \times \Phi(X) \times 2^X \times L$

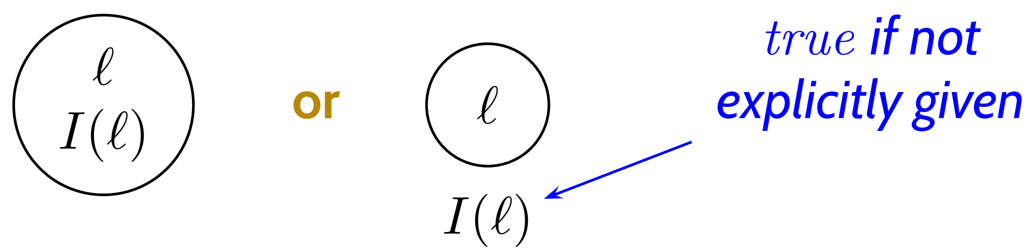
- **Locations**: $L = \{\mathit{off}, \mathit{light}, \mathit{bright}\}$
- **Alphabet**: $B = \{press\}$,
- **Clocks**: $X = \{x\}$,
- **Invariants**: $I = \{\mathit{off} \mapsto true, \mathit{light} \mapsto true, \mathit{bright} \mapsto true\}$
- **Edges**: $E = \{ \quad (\mathit{off}, press?, true,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

Graphical Representation of Timed Automata

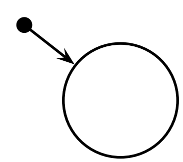
$$\mathcal{A} = (L, B, X, I, E, \ell_{ini})$$

- $I : L \to \Phi(X)$
- $E \subseteq L \times B_{?!} \times \Phi(X) \times 2^X \times L$

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

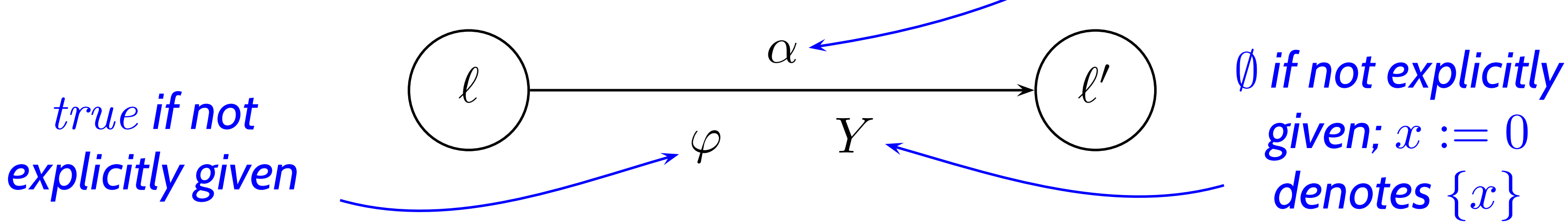


- **Initial location** ℓ_{ini}:



- **Edges**: $(\ell, \alpha, \varphi, Y, \ell') \in L \times B_{?!} \times \Phi(X) \times 2^X \times L$

τ if not explicitly given

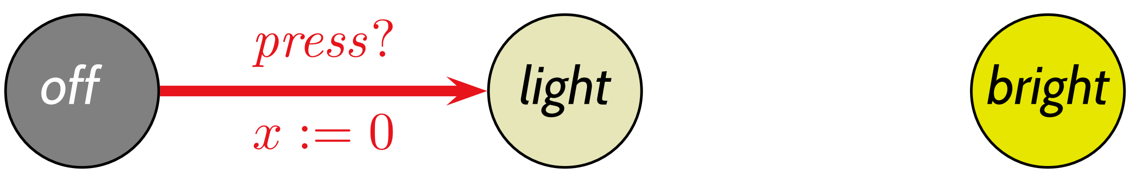


Example

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

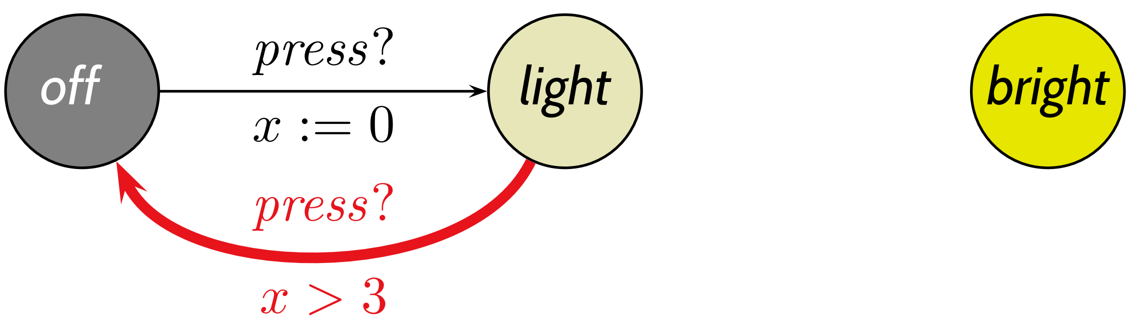
Example

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



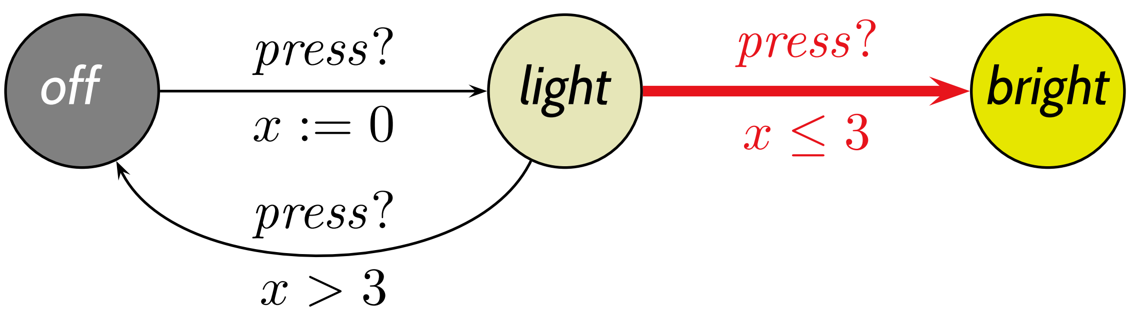
Example

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



Example

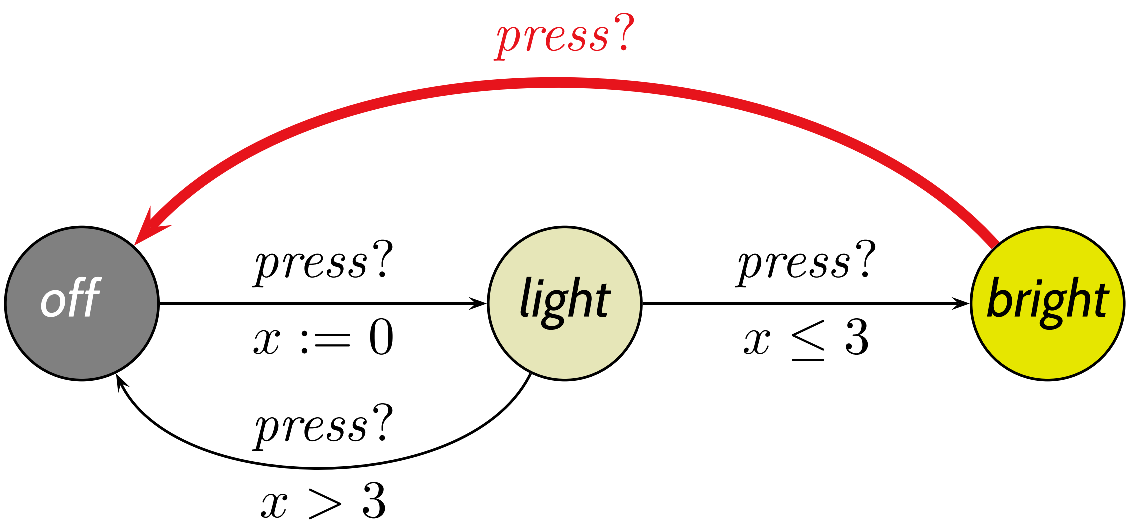
- **Locations**: $L = \{\textit{off}, \textit{light}, \textit{bright}\}$
- **Alphabet**: $B = \{press\}$,
- **Clocks**: $X = \{x\}$,
- **Invariants**: $I = \{\textit{off} \mapsto true, \textit{light} \mapsto true, \textit{bright} \mapsto true\}$
- **Edges**: $E = \{ \; (\textit{off}, press?, true, \{x\},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



–11 – 2017-12-07 – Stasyn –

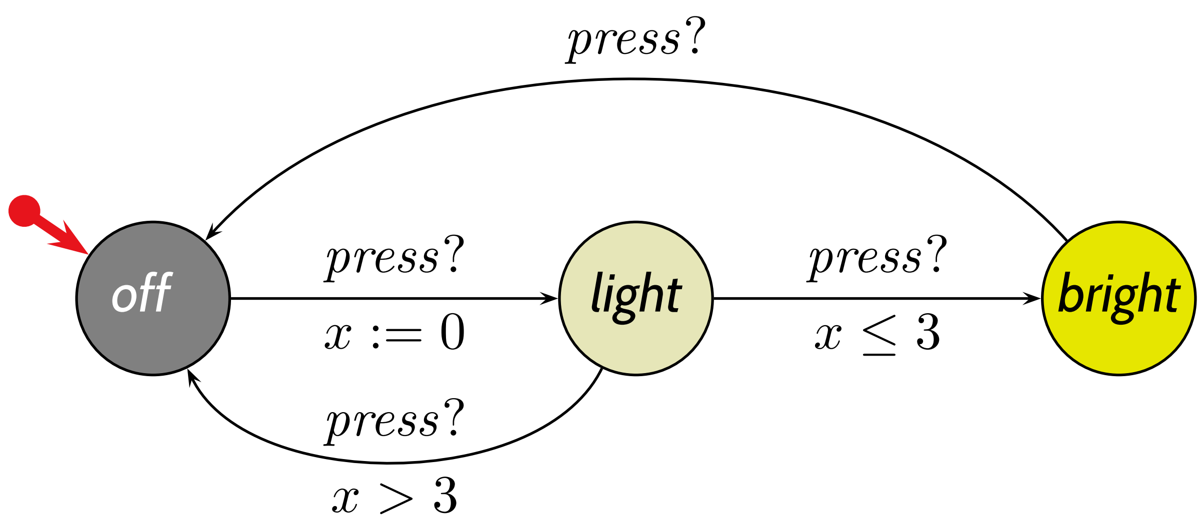
Example

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



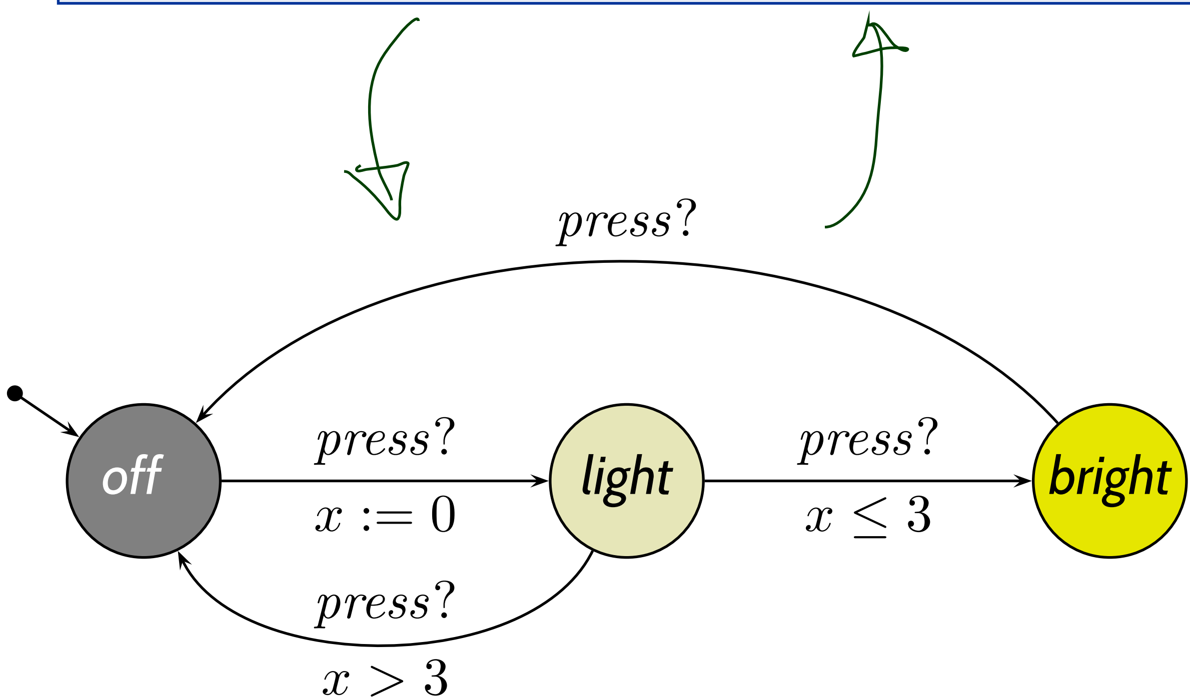
Example

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



Example

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



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 TA Operational Semantics

Clock Valuations

- Let X be a set of clocks. A **valuation ν of clocks** in X is a mapping

$$\nu : X \to \mathsf{Time}$$

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

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

  - $\nu \models x \sim c$ iff $\nu(x) \hat{\sim} \hat{c}$
  - $\nu \models x - y \sim c$ iff $\nu(x) \hat{-} \nu(y) \hat{\sim} \hat{c}$
  - $\nu \models \varphi_1 \wedge \varphi_2$ iff $\nu \models \varphi_1$ and $\nu \models \varphi_2$

Clock Valuations

- Let X be a set of clocks. A **valuation ν of clocks** in X is a mapping

$$\nu : X \to \mathsf{Time}$$

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

- Let φ be a clock constraint. The **satisfaction** relation between clock valuations ν and clock constraints φ, 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 $\nu(x) - \nu(y) \sim c$
  - $\nu \models \varphi_1 \wedge \varphi_2$ iff $\nu \models \varphi_1$ and $\nu \models \varphi_2$

- Two clock constraints φ_1 and φ_2 are called (**logically**) **equivalent** if and only if for all clock valuations ν, we have

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

  In that case we write $\models \varphi_1 \iff \varphi_2$.

Operations on Clock Valuations

Let ν be a valuation of clocks in X and $t \in \mathsf{Time}$.

- **Time Shift**

  We write $\nu + t$ to denote the clock valuation (for X) with

$$(\nu + t)(x) = \nu(x) + t.$$

  for all $x \in X$,

$\nu : \{x \mapsto 3.0\}$

$(\nu + 0.27)(x) = \nu(x) + 0.27$
$= 3.0 + 0.27 = 3.27$

- **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**: $t = 0$.

Operational Semantics of TA

Definition 4.4. The **operational semantics** of a timed automaton $\mathcal{A} = (L, B, X, I, E, \ell_{ini})$ is defined by the **(labelled) transition system**

$$\mathcal{T}(\mathcal{A}) = (\mathit{Conf}(\mathcal{A}), \mathsf{Time} \cup B_{?!}, \{\xrightarrow{\lambda} \mid \lambda \in \mathsf{Time} \cup B_{?!}\}, C_{ini})$$

where

- $\mathit{Conf}(\mathcal{A}) = \{\langle \ell, \nu \rangle \mid \ell \in L, \nu : X \to \mathsf{Time},\ \nu \models I(\ell)\}$
- $\mathsf{Time} \cup B_{?!}$ are the **transition labels**,
- there are **delay transition relations**

  $$\langle \ell, \nu \rangle \xrightarrow{\lambda} \langle \ell', \nu' \rangle, \quad \lambda \in \mathsf{Time} \qquad (\rightarrow \text{in a minute})$$

  and **action transition relations**

  $$\langle \ell, \nu \rangle \xrightarrow{\lambda} \langle \ell', \nu' \rangle, \quad \lambda \in B_{?!}. \qquad (\rightarrow \text{in a minute})$$

- $C_{ini} = \{\langle \ell_{ini}, \nu_0 \rangle\} \cap \mathit{Conf}(\mathcal{A})$ with $\nu_0(x) = 0$ for all $x \in X$ is the set of **initial configurations**.

Operational Semantics of TA Cont'd

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

$$\mathcal{T}(\mathcal{A}) = (\mathit{Conf}(\mathcal{A}), \mathsf{Time} \cup B_{?!}, \{\xrightarrow{\lambda} \mid \lambda \in \mathsf{Time} \cup B_{?!}\}, C_{ini})$$

- **Time** or **delay transition**:

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

$$\langle \ell, \nu \rangle \xrightarrow{t} \langle \ell, \nu + t \rangle$$

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

"Some **time** $t \in \mathsf{Time}$ **elapses** respecting invariants, location unchanged."

- **Action** or **discrete transition**:

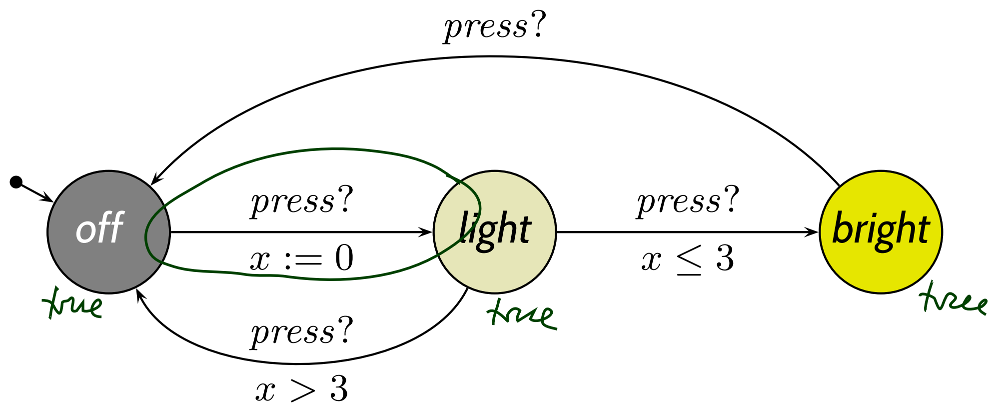
$$\langle \ell, \nu \rangle \xrightarrow{\alpha} \langle \ell', \nu' \rangle$$

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

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

"An action occurs, location may change, some clocks may be reset, **time does not elapse**."

Example



- **Configurations**:

$$Conf(\mathcal{A}) = \{\langle \textsf{off}, \nu\rangle, \langle \textsf{light}, \nu\rangle, \langle \textsf{light}, \nu\rangle \mid \nu : X \to \textsf{Time}\}$$

- **Initial Configurations**:

$$\{\langle \textsf{off}, \nu_0\rangle\} \cap Conf(\mathcal{A}) = \{\langle \textsf{off}, \{x \mapsto 0\}\rangle\}$$

$\{\langle \textsf{off}, x=0\rangle\}$

- **Delay Transition**:

$$\langle \textsf{off}, \{x \mapsto 0\}\rangle \xrightarrow{27} \langle \textsf{off}, \{x \mapsto 27\}\rangle$$

- **Action Transition**:

$$\langle \textsf{off}, \{x \mapsto 27\}\rangle \xrightarrow{press?} \langle \textsf{light}, \{x \mapsto 0\}\rangle$$ ✓

Transition Sequences

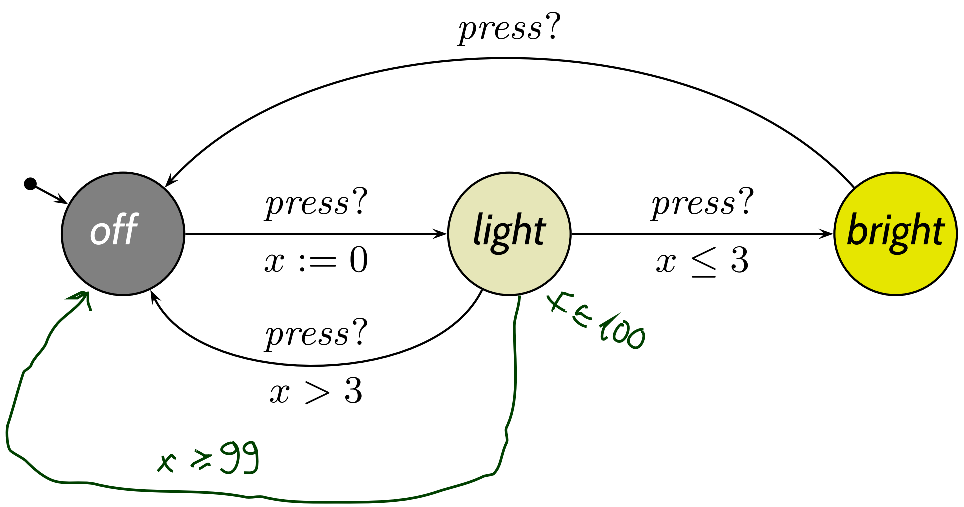
- A **transition sequence** of $\mathcal{A}$ is any finite or infinite sequence of the form

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

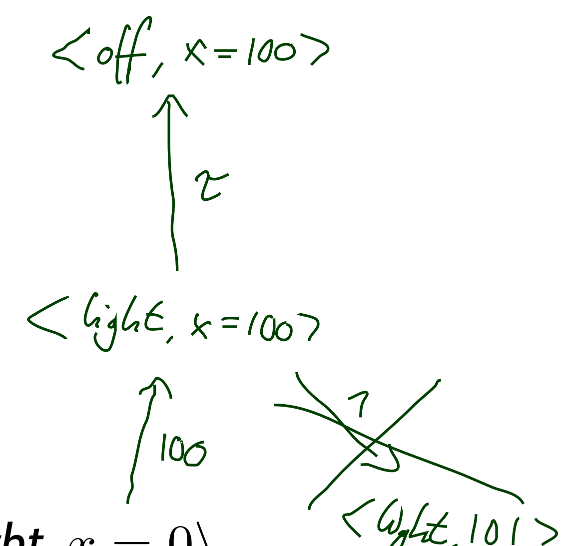
with

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

Example



$$\langle \textsf{off}, x = 0 \rangle \xrightarrow{2.5} \langle \textsf{off}, x = 2.5 \rangle$$
$$\xrightarrow{1.7} \langle \textsf{off}, x = 4.2 \rangle$$
$$\xrightarrow{press?} \langle \textsf{light}, x = 0 \rangle$$
$$\xrightarrow{2.1} \langle \textsf{light}, x = 2.1 \rangle$$
$$\xrightarrow{press?} \langle \textsf{bright}, x = 2.1 \rangle$$
$$\xrightarrow{10} \langle \textsf{bright}, x = 12.1 \rangle$$
$$\xrightarrow{press?} \langle \textsf{off}, x = 12.1 \rangle$$
$$\xrightarrow{press?} \langle \textsf{light}, x = 0 \rangle \xrightarrow{0} \langle \textsf{light}, x = 0 \rangle$$



Reachability

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

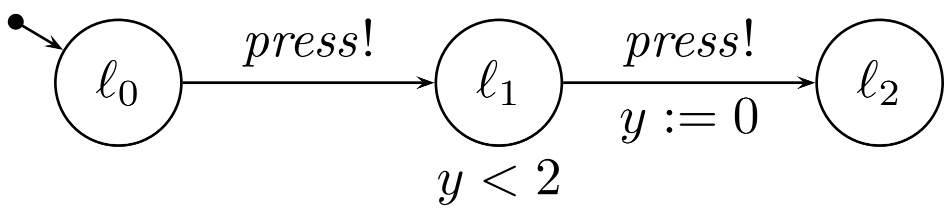
$$\langle \ell_0, \nu_0 \rangle \xrightarrow{\lambda_1} \langle \ell_1, \nu_1 \rangle \xrightarrow{\lambda_2} \langle \ell_2, \nu_2 \rangle \xrightarrow{\lambda_3} \dots \xrightarrow{\lambda_n} \langle \ell_n, \nu_n \rangle = \langle \ell, \nu \rangle$$

- A **location** ℓ is called **reachable** if and only if **any** configuration $\langle \ell, \nu \rangle$ is reachable, i.e. there exists a valuation ν such that $\langle \ell, \nu \rangle$ is reachable.

Location Invariants

Recall: $Conf(\mathcal{A}) = \{\langle \ell, \nu \rangle \mid \ell \in L, \nu : X \to \mathsf{Time},\ \nu \models I(\ell)\}$

Example:



- **Configurations**:
  - $Conf(\mathcal{A}) = \{\langle \ell_0, \nu \rangle, \langle \ell_2, \nu \rangle \mid \nu : \{y\} \to \mathsf{Time}\} \cup \{\langle \ell_1, \nu \rangle \mid \nu : \{y\} \to [0, 2[\}$

- $\langle \ell_1, y \mapsto 1.01 \rangle$ **is a** configuration,
- $\langle \ell_1, y \mapsto 27 \rangle$ **is not a** configuration,
- $\langle \ell_0, y \mapsto 0 \rangle \xrightarrow{0.707} \langle \ell_0, y \mapsto 0.707 \rangle \xrightarrow{press!} \langle \ell_1, y \mapsto 0.707 \rangle$ **is a** transition sequence
- $\langle \ell_0, y \mapsto 0 \rangle \xrightarrow{27} \langle \ell_0, y \mapsto 27 \rangle$ **is a** transition sequence
- $\langle \ell_0, y \mapsto 0 \rangle \xrightarrow{27} \langle \ell_0, y \mapsto 27 \rangle \xrightarrow{press!} \langle \ell_1, y \mapsto 27 \rangle$ **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**:
    - $Conf(\mathcal{A}) = \{\langle \ell, \nu \rangle \mid \ell \in L, \nu : X \to \mathsf{Time}, \nu \models I(\ell)\}$
    - $C_{ini} = \{\langle \ell_{ini}, \nu_0 \rangle\} \cap Conf(\mathcal{A})$
- **The approach not taken for TA:**
  - consider every $\langle \ell, \nu \rangle$ to be a configuration, i.e. have

    $$Conf(\mathcal{A}) = \{\langle \ell, \nu \rangle \mid \ell \in L, \nu : X \to \mathsf{Time} \,\cancel{, \nu \models I(\ell)}\}$$

  - "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**:

$$\langle \ell, \nu \rangle, t \xrightarrow{t'} \langle \ell, \nu + t' \rangle, t + t' \qquad \text{iff } t' \in \mathsf{Time} \text{ and } \langle \ell, \nu \rangle \xrightarrow{t'} \langle \ell, \nu + t' \rangle.$$

- **Time-stamped action transition**:

$$\langle \ell, \nu \rangle, t \xrightarrow{\alpha} \langle \ell', \nu' \rangle, t \qquad \text{iff } \alpha \in B_{?!} \text{ and } \langle \ell, \nu \rangle \xrightarrow{\alpha} \langle \ell', \nu' \rangle.$$

Computation Paths

- A sequence of time-stamped configurations

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

  is called

  - **computation path** (or path) **of $\mathcal{A}$**
  - **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 $\mathcal{A}$** 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:

  - $\langle \ell, x = 0 \rangle, 0 \xrightarrow{2} \langle \ell, x = 2 \rangle, 2$
  - $\langle \ell', x = 0 \rangle, 0 \xrightarrow{3} \langle \ell', x = 3 \rangle, 3 \xrightarrow{a?} \langle \ell', x = 3 \rangle, 3 \xrightarrow{a?} \dots$

- **Zeno** behaviour:

  - $\langle \ell, x = 0 \rangle, 0 \xrightarrow{\frac{1}{2}} \langle \ell, x = \frac{1}{2} \rangle, \frac{1}{2} \xrightarrow{\frac{1}{4}} \langle \ell, x = \frac{3}{4} \rangle, \frac{3}{4} \dots \xrightarrow{\frac{1}{2^n}} \langle \ell, x = \frac{2^n - 1}{2^n} \rangle, \frac{2^n - 1}{2^n} \dots$
  - $\langle \ell, x = 0 \rangle, 0 \xrightarrow{0.1} \langle \ell, x = 0.1 \rangle, 0.1 \xrightarrow{0.01} \langle \ell, x = 0.11 \rangle, 0.11 \xrightarrow{0.001} \langle \ell, x = 0.111 \rangle, 0.111 \dots$

Real-Time Sequence

Definition 4.9. An infinite sequence

$$t_0, t_1, t_2, \ldots$$

of values $t_i \in \mathsf{Time}$ 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**)):

$$\forall\, t \in \mathsf{Time}\; \exists\, i \in \mathbb{N}_0 : t < t_i$$

Run

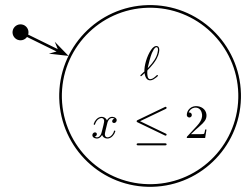
Definition 4.10. A **run of $\mathcal{A}$ starting in $\langle \ell_0, \nu_0 \rangle, t_0$** is an **infinite computation path**

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

of $\mathcal{A}$ where $(t_i)_{i \in \mathbb{N}_0}$ is a **real-time sequence**.

We call ξ a **run of $\mathcal{A}$** if and only if ξ is a **computation path** of $\mathcal{A}$.

Example:



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**

Tell Them What You've Told Them...

- 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

Olderog, E.-R. and Dierks, H. (2008). Real-Time Systems - Formal Specification and Automatic Verification. Cambridge University Press.