Time-critical reactive systems

Renato Neves and José Proença

Universidade do Minho

HASLab
HIGH-ASSURANCE SOFTWARE LABORATORY

CISTER
Research Centre in Real-Time & Embedded Computing Systems

Architecture and Calculi Course Unit

Table of Contents

Motivation

Specifying an airbag saying that in a car crash the airbag eventually inflates maybe not enough, but:

in a car crash the airbag eventually inflates within 20ms

> *Correctness in time-critical systems not only depends on the logical result of the computation, but also on the time at which the results are produced*
>
> [Baier & Katoen, 2008]

Examples of time-critical systems

Network-based traffic lights

Their lights should be activated at very specific time intervals.

Bounded retransmission protocol

Communication of large files between a remote control unit and a video/audio equipment. Correctness depends crucially on

- transmission and synchronization delays
- time-out values for times at sender and receiver

And many others...

- medical instruments
- hybrid systems (eg for controlling industrial plants)

Motivation

This suggests resorting to an automaton-based formalism with an explicit notion of clock (stopwatch) to control availability of transitions.

Timed Automata [Alur & Dill, 90]

- emphasis on decidability of the reachability problem and corresponding practically efficient algorithms
- infinite underlying timed transition systems are converted to finitely large symbolic transition systems where reachability becomes decidable (region or zone graphs)

Associated tools

- UPPAAL [Behrmann, David, Larsen, 04]
- KRONOS [Bozga, 98]

Motivation

$\boxed{\text{UPPAAL} = (\text{Uppsala University} + \text{Aalborg University})\ [1995]}$

- A toolbox for modelling, simulation and verification of real-time systems
- Systems are modelled as networks of timed automata enriched with integer variables and channel syncronisations
- Properties are specified in a subset of CTL

www.uppaal.com

Table of Contents

Timed automata

Finite-state machine equipped with a finite set of real-valued clock variables (clocks)

Clocks

- clocks can only be inspected or
- reset to zero, after which they start increasing their value implicitly as time progresses
- the value of a clock corresponds to time elapsed since its last reset
- all clocks proceed synchronously (at the same rate)

Timed automata

Definition

$$\langle L, L_0, Act, C, Tr, Inv \rangle$$

where

- L is a set of locations, and $L_0 \subseteq L$ the set of initial locations
- Act is a set of actions and C a set of clocks
- $Tr \subseteq L \times \mathcal{C}(C) \times Act \times \mathcal{P}(C) \times L$ is the transition relation

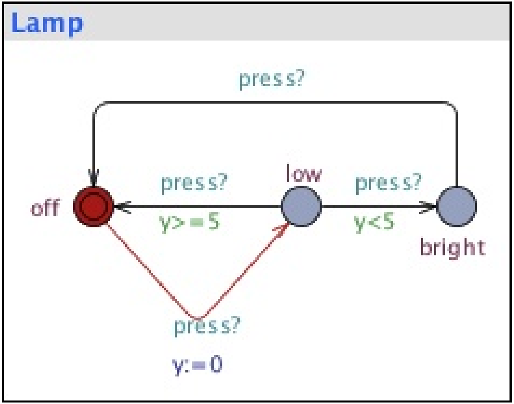
  $$\ell_1 \stackrel{g,a,U}{\longrightarrow} \ell_2$$

  denotes a transition from location ℓ_1 to ℓ_2, labelled by a, enabled if guard g is valid, which, when performed, resets the set U of clocks
- $Inv : L \longrightarrow \mathcal{C}(C)$ is the assigment of invariants to locations

where $\mathcal{C}(C)$ denotes the set of clock constraints over a set C of clock variables

Example: the lamp interrupt

(extracted from UPPAAL)



Clock constraints

$\mathcal{C}(C)$ denotes the set of clock constraints over a set C of clock variables. Each constraint is formed according to

$$g \ ::= \ x \,\Box\, n \ \mid \ x - y \,\Box\, n \ \mid \ g \wedge g \ \mid \ \mathit{true}$$

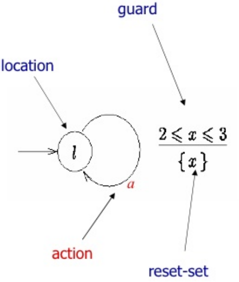
where $x, y \in C, n \in \mathbb{N}$ and $\Box \in \{<, \leq, >, \geq, =\}$
This is used in

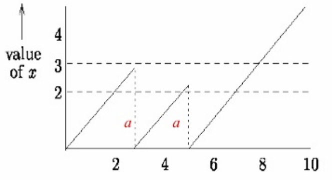
- transitions as guards (enabling conditions)
  $\boxed{\text{a transition cannot occur if its guard is invalid}}$
- locations as invariants (safety conditions)
  $\boxed{\text{a location must be left before its invariant becomes invalid}}$

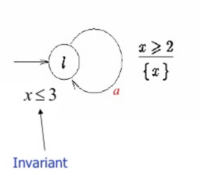
Note

Invariants are the only way to force transitions to occur

Guards, updates & invariants







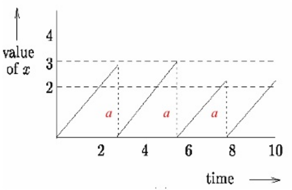


Table of Contents

Parallel composition of timed automata

- Action labels as channel identifiers
- Communication by forced handshaking over a subset of common actions
- Is defined as an automaton construction over a finite set of timed automata originating a so-called network of timed automata

Parallel composition of timed automata

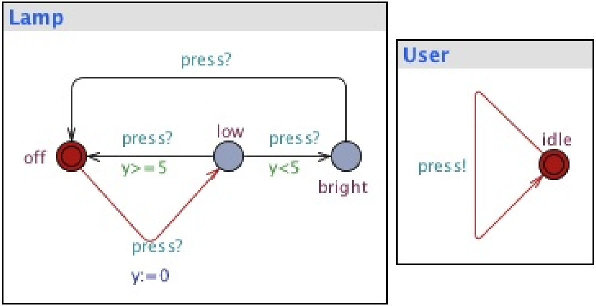
Let $H \subseteq Act_1 \cap Act_2$. The parallel composition of ta_1 and ta_2 synchronizing on H is the timed automata

$$ta_1 \parallel_H ta_2 := \langle L_1 \times L_2, L_{0,1} \times L_{0,2}, Act_{\parallel_H}, C_1 \cup C_2, Tr_{\parallel_H}, Inv_{\parallel_H} \rangle$$

where

- $Act_{\parallel_H} = ((Act_1 \cup Act_2) - H) \cup \{\tau\}$
- $Inv_{\parallel_H} \langle \ell_1, \ell_2 \rangle = Inv_1(\ell_1) \wedge Inv_2(\ell_2)$
- $Tr_{\parallel_H}$ is given by:
  - $\langle \ell_1, \ell_2 \rangle \xrightarrow{g,a,U} \langle \ell_1', \ell_2 \rangle$ if $a \notin H \wedge \ell_1 \xrightarrow{g,a,U} \ell_1'$
  - $\langle \ell_1, \ell_2 \rangle \xrightarrow{g,a,U} \langle \ell_1, \ell_2' \rangle$ if $a \notin H \wedge \ell_2 \xrightarrow{g,a,U} \ell_2'$
  - $\langle \ell_1, \ell_2 \rangle \xrightarrow{g,\tau,U} \langle \ell_1', \ell_2' \rangle$ if $a \in H \wedge \ell_1 \xrightarrow{g_1,a,U_1} \ell_1' \wedge \ell_2 \xrightarrow{g_2,a,U_2} \ell_2'$ with $g = g_1 \wedge g_2$ and $U = U_1 \cup U_2$

Example: the lamp interrupt as a closed system



UPPAAL:

- takes $H = Act_1 \cap Act_2$ (actually as complementary actions denoted by the ? and ! annotations)
- only deals with closed systems

Exercise: worker, hammer, nail

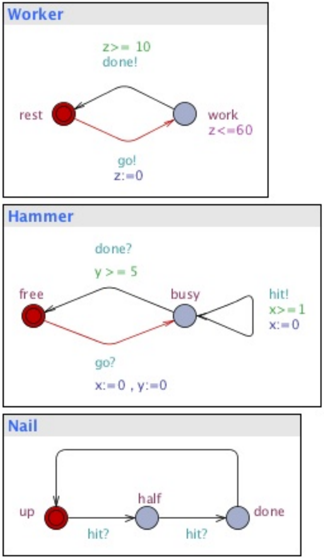


Table of Contents

Timed Labelled Transition Systems

Syntax	Semantics
How to write	*How to execute*
Timed Automaton	TLTS (Timed LTS)

Timed Labelled Transition Systems

Syntax	Semantics
How to write	*How to execute*
Timed Automaton	TLTS (Timed LTS)

Timed LTS

Introduce delay transitions to capture the passage of time within a LTS:

$s \xrightarrow{a} s'$ for $a \in Act$, are ordinary transitions due to action occurrence

$s \xrightarrow{d} s'$ for $d \in \mathcal{R}^+$, are delay transitions

subject to a number of constraints, eg,

Dealing with time in system models

Timed LTS

- time additivity

$$(s \xrightarrow{d} s' \wedge 0 \leq d' \leq d) \Rightarrow s \xrightarrow{d'} s'' \xrightarrow{d-d'} s' \text{ for some state } s''$$

- delay transitions are deterministic

$$(s \xrightarrow{d} s' \wedge s \xrightarrow{d} s'') \Rightarrow s' = s''$$

Semantics of Timed Automata

Semantics of TA:

Every TA *ta* defines a TLTS

$$\mathcal{T}(ta)$$

whose states are pairs

$$\langle \text{location}, \text{clock valuation} \rangle$$

with infinitely, even uncountably many states

Clock valuations

Definition

A clock valuation η for a set of clocks C is a function

$$\eta : C \longrightarrow \mathcal{R}_0^+$$

assigning to each clock $x \in C$ its current value $\eta\, x$.

Satisfaction of clock constraints

$$\eta \models x \,\square\, n \;\Leftrightarrow\; \eta\, x \,\square\, n$$

$$\eta \models x - y \,\square\, n \;\Leftrightarrow\; (\eta\, x - \eta\, y) \,\square\, n$$

$$\eta \models g_1 \wedge g_2 \;\Leftrightarrow\; \eta \models g_1 \wedge \eta \models g_2$$

Operations on clock valuations

Delay

For each $d \in \mathcal{R}_0^+$, valuation $\eta + d$ is given by

$$(\eta + d)\,x \;=\; \eta\,x \;+\; d$$

Reset

For each $R \subseteq C$, valuation $\eta[R]$ is given by

$$\begin{cases} \eta[R]\,x \;=\; \eta\,x & \Leftarrow x \notin R \\ \eta[R]\,x \;=\; 0 & \Leftarrow x \in R \end{cases}$$

From ta to $\mathcal{T}(ta)$

Let $ta = \langle L, L_0, Act, C, Tr, Inv \rangle$

$$\mathcal{T}(ta) \;=\; \langle S, S_0 \subseteq S, N, T \rangle$$

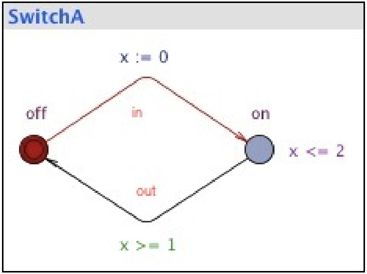
where

- $S = \{\langle l, \eta \rangle \in L \times (\mathcal{R}_0^+)^C \mid \eta \models Inv(l)\}$
- $S_0 = \{\langle \ell_0, \eta \rangle \mid \ell_0 \in L_0 \;\wedge\; \eta\, x = 0 \text{ for all } x \in C\}$
- $N = Act + \mathcal{R}_0^+$ (ie, transitions can be labelled by actions or delays)
- $T \subseteq S \times N \times S$ is given by:

$$\langle l, \eta \rangle \xrightarrow{a} \langle l', \eta' \rangle \quad \Leftarrow \quad \exists_{l \xrightarrow{g,a,U} l' \in Tr} \;\; \eta \models g \;\wedge\; \eta' = \eta[U] \;\wedge\; \eta' \models Inv(l')$$

$$\langle l, \eta \rangle \xrightarrow{d} \langle l, \eta + d \rangle \quad \Leftarrow \quad \exists_{d \in \mathcal{R}_0^+} \;\; \eta + d \models Inv(l)$$

Example: the simple switch

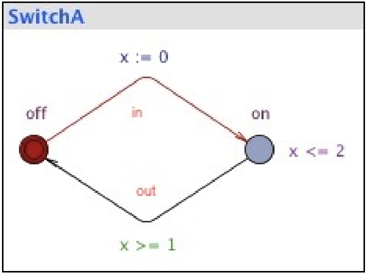


$\mathcal{T}$(SwitchA)

$$S = \{\langle off, t\rangle \mid t \in \mathcal{R}_0^+\} \cup \{\langle on, t\rangle \mid 0 \leq t \leq 2\}$$

where t is a shorthand for η such that $\eta\, x = t$

Example: the simple switch



$\mathcal{T}$(SwitchA)

$$\langle \mathit{off}, t \rangle \xrightarrow{d} \langle \mathit{off}, t + d \rangle \text{ for all } t, d \geq 0$$

$$\langle \mathit{off}, t \rangle \xrightarrow{in} \langle on, 0 \rangle \text{ for all } t \geq 0$$

$$\langle on, t \rangle \xrightarrow{d} \langle on, t + d \rangle \text{ for all } t, d \geq 0 \text{ and } t + d \leq 2$$

$$\langle on, t \rangle \xrightarrow{out} \langle \mathit{off}, t \rangle \text{ for all } 1 \leq t \leq 2$$

Note

- The elapse of time in timed automata only takes place in locations:
- ... actions take place instantaneously
- Thus, several actions may take place at a single time unit

Behaviours

- Paths in $\mathcal{T}(ta)$ are discrete representations of continuous-time behaviours in ta
- ... *i.e.* they indicate the states immediately before and after the execution of an action
- However, as interval delays may be realised in uncountably many different ways, different paths may represent the same behaviour

Behaviours

- Paths in $\mathcal{T}(ta)$ are discrete representations of continuous-time behaviours in ta
- ... *i.e.* they indicate the states immediately before and after the execution of an action
- However, as interval delays may be realised in uncountably many different ways, different paths may represent the same behaviour
- ... but not all paths correspond to valid (realistic) behaviours:

undesirable paths:

- time-convergent paths
- timelock paths
- zeno paths

Table of Contents

Traces

Definition

A timed trace over a timed LTS is a (finite or infinite) sequence $\langle t_1, a_1\rangle, \langle t_2, a_2\rangle, \cdots$ in $\mathcal{R}_0^+ \times Act$ such that there exists a path

$$\langle \ell_0, \eta_0\rangle \xrightarrow{d_1} \langle \ell_0, \eta_1\rangle \xrightarrow{a_1} \langle \ell_1, \eta_2\rangle \xrightarrow{d_2} \langle \ell_1, \eta_3\rangle \xrightarrow{a_2} \cdots$$

such that

$$t_i = t_{i-1} + d_i$$

with $t_0 = 0$ and, for all clock x, $\eta_0\, x = 0$.

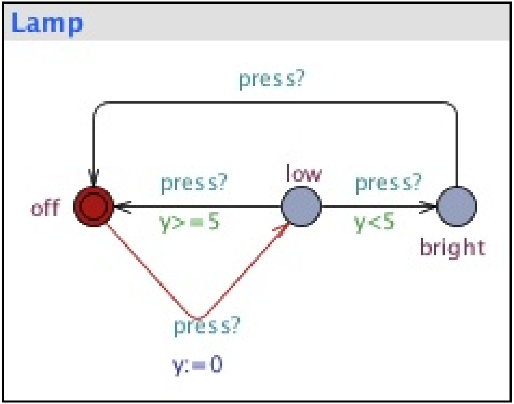
Intuitively, each t_i is an absolute time value acting as a time-stamp.

Warning

All results from now on are given over an arbitrary timed LTS; they naturally apply to $\mathcal{T}(ta)$ for any timed automata ta.

Traces

Write possible traces



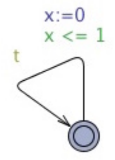
Traces

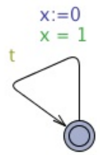
Given a timed trace tc, the corresponding untimed trace is $(\pi_2)^\omega\ tc$.

Definition

- two states s_1 and s_2 of a timed LTS are timed-language equivalent if the set of finite timed traces of s_1 and s_2 coincide;
- ... similar definition for untimed-language equivalent ...

Example





are not timed-language equivalent

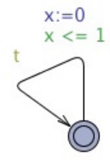
Traces

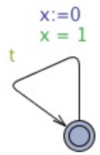
Given a timed trace tc, the corresponding untimed trace is $(\pi_2)^\omega\ tc$.

Definition

- two states s_1 and s_2 of a timed LTS are timed-language equivalent if the set of finite timed traces of s_1 and s_2 coincide;
- ... similar definition for untimed-language equivalent ...

Example





are not timed-language equivalent

$\langle (0, t) \rangle$ is not a trace of the TLTS generated by the second system.

Bisimulation

Timed bisimulation (between states of timed LTS)

A relation R is a timed simulation iff whenever $s_1 R s_2$, for any action a and delay d,

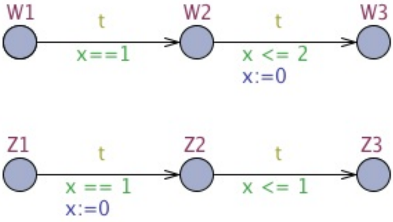
$$s_1 \xrightarrow{a} s_1' \Rightarrow \text{there is a transition} \quad s_2 \xrightarrow{a} s_2' \wedge s_1' R s_2'$$

$$s_1 \xrightarrow{d} s_1' \Rightarrow \text{there is a transition} \quad s_2 \xrightarrow{d} s_2' \wedge s_1' R s_2'$$

And a timed bisimulation if its converse is also a timed simulation.

Bisimulation

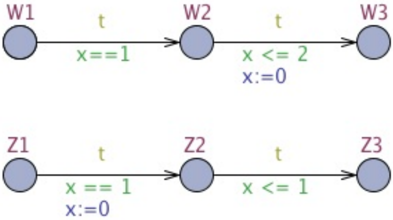
Example



W1 bisimilar to Z1?

Bisimulation

Example



W1 bisimilar to Z1?

$$\langle\langle W1, \{x \mapsto 0\}\rangle, \langle Z1, \{x \mapsto 0\}\rangle\rangle \in R$$

where

$$\begin{array}{rcllll} R & = & \{\langle\langle W1, \{x \mapsto d\}\rangle & , \langle Z1, \{x \mapsto d\}\rangle\rangle & \mid d \in \mathcal{R}_0^+\} & \cup \\ & & \{\langle\langle W2, \{x \mapsto d+1\}\rangle & , \langle Z2, \{x \mapsto d\}\rangle\rangle & \mid d \in \mathcal{R}_0^+\} & \cup \\ & & \{\langle\langle W3, \{x \mapsto d\}\rangle & , \langle Z3, \{x \mapsto e\}\rangle\rangle & \mid d, e \in \mathcal{R}_0^+\} & \end{array}$$