02224 Real-Time Systems

An Introduction to Timed Automata using Uppaal

Michael R. Hansen

DTU Compute

Department of Applied Mathematics and Computer Science

Overview

Timed Automata: AlurDill 1994

- A *system* is a network of communication *timed automata*, put in parallel
- A *timed automaton* is a finite state machine with *clocks*
- A *clock* is a real-valued variable that is used to measure the progress of time.

Uppaal `www.uppaal.org`

- Uppaal is a tool box for *modelling* real-time systems and for *validation* and *verification* of such systems.
- *Modelling* is timed-automata based
- *Validation* is based on simulation
- *Verification* is based on *model checking*

A simple example (1)

Consider an untimed system consisting of a Lamp and a User:

- A user can switch the light on and off
- The lamp reacts appropriately on input from the user

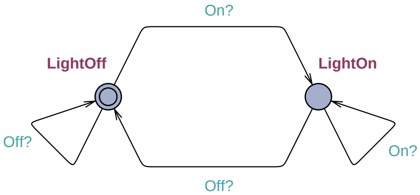


Figure: Lamp

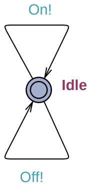
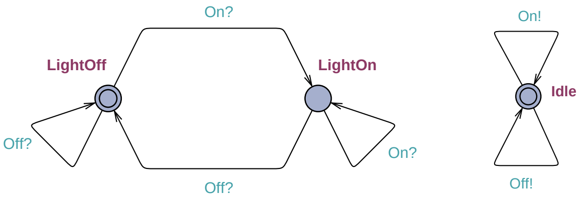


Figure: User

- LightOff, LightOn and Idle are locations
- Off and On are channels
- Off? and On? are input events
- Off! and On! are output events
- An input event *e*? may synchronize with an output event *e*!

A simple example (2)



A path in the un-timed system is a sequence of states altered by events:

$$(\mathit{LightOff}, \mathit{Idle}) \xrightarrow{on} (\mathit{LightOn}, \mathit{Idle}) \xrightarrow{off} (\mathit{LightOff}, \mathit{Idle}) \xrightarrow{on} \cdots$$

where, for example,

- $(\mathit{LightOff}, \mathit{Idle}) \xrightarrow{on} (\mathit{LightOn}, \mathit{Idle})$ is a transition of the system

because

- $\mathit{LightOff} \xrightarrow{on?} \mathit{LightOn}$ is a transition of the lamp and
- $\mathit{Idle} \xrightarrow{on!} \mathit{Idle}$ is a transition of the user

A Uppaal model comprising two lamps

- Global declarations of two channel arrays:

```
chan On[2];
chan Off[2];
```

- The Lamp template contains two formal parameters:

```
chan &On, chan &Off
```

  Notice that Channels and clocks must be reference parameters

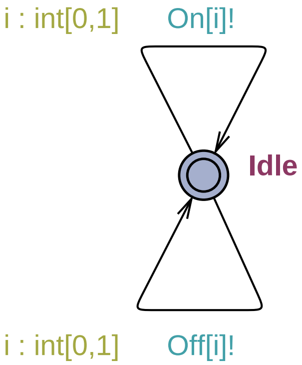
- The system declaration part contains two process assignments:

```
Lamp0 = Lamp(On[0],Off[0]);
Lamp1 = Lamp(On[1],Off[1]);
system Lamp0, Lamp1, User;
```

The Lamp automaton is as before.

A user template with a selection

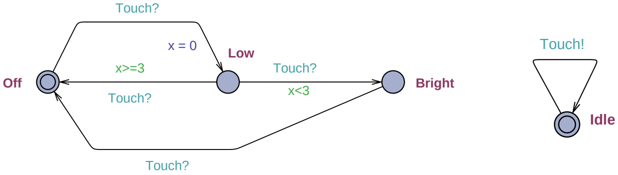
The new user automaton is:



- Both transitions contain a selection: `i: int[0,1]`
- `int[0,1]` is a type with two elements 0 and 1
- A selection denotes a family of transitions by binding `i` to the elements of the type

A simple timed system (1)

- With a single touch on the lamp, it will give a dimmed light,
- if it is touched twice quickly, it will give a bright light; otherwise it will switch off the light

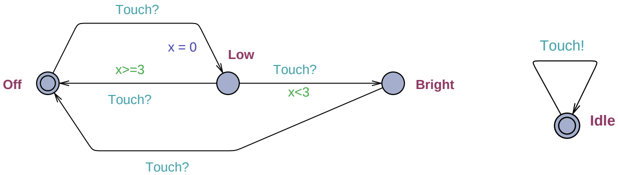


- x is a real-valued clock
- The value of a clock can be tested in guards $x < 3$ and $x \leq 3$
- A clock can be reset $x = 0$

An edge in a Uppaal template can be equipped with

- a select
- a synchronization
- a guard
- an update

A simple timed system (2)



A state (l, v) of a timed automaton consist of

- a location l, and
- a valuation v giving values for the clocks

A transition from one state (l, v) to another (l', v') can either be

- a discrete transition $(l, v) \xrightarrow{e} (l', v')$, where e is an event
- a time-progress $(l, v) \xrightarrow{d} (l, v')$, where $d > 0$

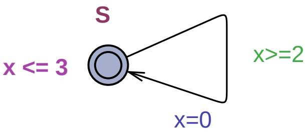
A path (ignoring the state of the user): e is Touch

$$\begin{array}{ll} & (Off, 0) \xrightarrow{2} (Off, 2) \xrightarrow{3.44} (Off, 5.44) \xrightarrow{e} (Low, 0) \xrightarrow{2.5} (Low, 2.5) \\ \xrightarrow{e} & (Bright, 2.5) \xrightarrow{3.1} (Bright, 5.6) \xrightarrow{e} (Off, 5.6) \xrightarrow{1.2} (Off, 6.8) \\ \xrightarrow{e} & (low, 0) \xrightarrow{3} (Low, 3) \xrightarrow{e} (Off, 3) \xrightarrow{3} \cdots \end{array}$$

A simple timed system with an invariant

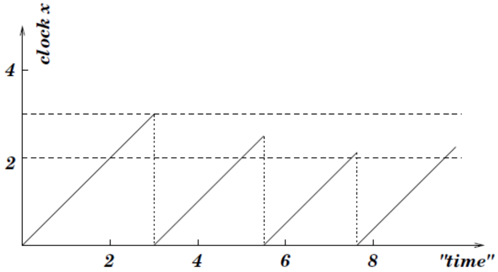
An invariant in a location is used to ensure progress

- x cannot progress beyond 3 is location *S*:



Possible values of *x*

from Uppaal Tutorial [BDL2016]



Timed Automata - Basic Definitions

- A clock constraint is a conjunction of formulas of the form:

  $$x \bowtie n \qquad \text{or} \qquad x - y \bowtie n$$

  where $x, y \in C$ are clocks, $\bowtie \in \{<, \leq =, >, \geq\}$ and n in an integer.

- $B(C)$ denotes the set of all clock constraints that can be formed from the set of clocks C

- 2^C denotes the set of all subsets of C

- $R^C_{\geq 0}$ denotes the set of functions from the set of clocks to the non-negative real numbers.

- An element $v \in R^C_{\geq 0}$ is called a clock evaluation.

- $[r \mapsto 0]v$ denotes the clock evaluation obtained from v by mapping each clock in r to 0.

Timed Automata - Syntax

A timed automaton is a tuple (L, l_0, C, A, E, I), where

- L is a finite set of locations,
- $l_0 \in L$ is the initial location,
- C is a finite set of clocks,
- A is a finite set of *actions*,
- $E \subseteq L \times A \times B(C) \times 2^C \times L$ is a set of edges, and
- $I : L \to B(C)$ assigns invariants to locations.

where an edge (l, a, g, r, l') from location l to l' is decorated with

- an action a – can be input, output or internal
- a guard g,
- a set of clocks r that is reset

Timed Automata - Semantics

The semantics of a timed automaton (L, l_0, C, A, E, I) is a labelled transition system $(S, s_0, \longrightarrow)$, where

– $S = \{(l, v) \in L \times \mathbb{R}^C_{\geq 0} \mid v$ satisfies the invariant $I(l)\}$
is the set of states,

– $s_0 = (l_0, v_0)$ is the initial state, and

– the transition relations $\longrightarrow \subseteq S \times (A \cup \mathbb{R}_{\geq 0}) \times S$ is such that

- $(l, v) \xrightarrow{d} (l, v + d)$, for $d > 0$
  if $v + d'$ satisfies the invariant $I(l)$ for every $d' : 0 \leq d' \leq d$, and
- $(l, v) \xrightarrow{a} (l', v')$
  if there is and edge $(l, a, g, r, l') \in E$ such that
  – v satisfies the guard g,
  – $v' = [r \mapsto 0]v$, and
  – v' satisfies the invariant $I(l')$ of the target location.

Paths, Deadlock, Timelock and zeno-path

A path is time divergent when time growths towards infinity; otherwise it is time convergent

A path is Zeno if it is time-convergent and it performs infinitely many discrete actions

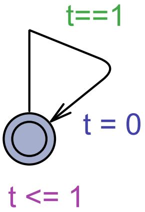
A state is a deadlock state if there are no outgoing action transitions neither from the state itself or any of its delay successors

A state contains a timelock if there is no time-divergent path starting from it.

Trick: Spotting Anomalies in Uppaal

Timelocks and Zeno paths represent typical modelling flaws:

- They should be avoided in the models.
- Add a test automaton to the system:



  where `t` is a new clock

- Verify the liveness property: `t==0 --> t==1`

Urgent channels and Urgent and committed locations

Uppaal has some special features to control timing:

- A channel may be declared as urgent: There will be no delay if an edge with a synchronization over an urgent channel can be taken
- A location may be declared as urgent: There will be no delay in an urgent location
- A location may be declared as committed: There will be no delay in a committed location, and the next transition must involve an automaton in a committed location.

We will consider these features later in the course