Embedded Systems Specification and Design

David Kendall

The essence of a formal method

Informally…

The system meets its specification

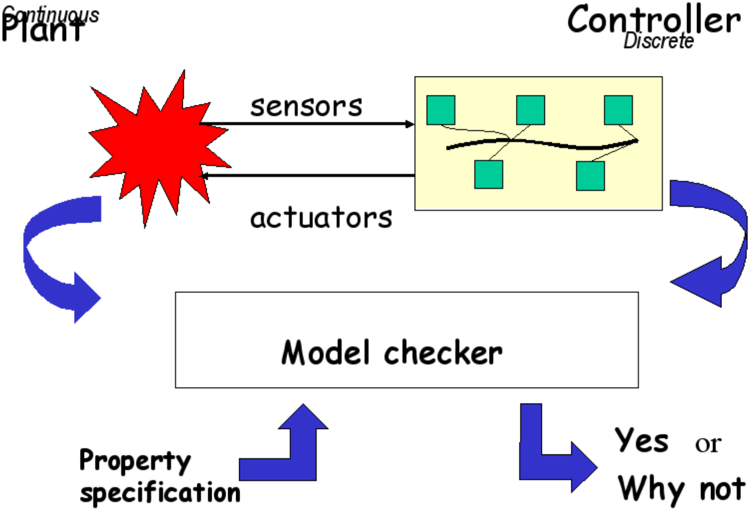
Formally…

$$\text{Sys} \models \text{Spec}$$

A formal language has well-defined

- syntax
- semantics
- rules for reasoning about the relationship between expressions

Model checking for embedded systems



Finite State Machine

- A finite state machine is a tuple $(S, s^0, A, \longrightarrow, F)$ where F is a set of accepting states
- Standard acceptance: finite executions from the start state which terminate in an acceptance state
- Buchi acceptance: infinite executions from the start state which visit an acceptance state infinitely often – needed for reactive systems
- $L(M)$ is the language of FSM M, i.e. the set of executions that M accepts under some acceptance condition

Labelled Transition System

Formally, a labelled transition system is a tuple $(S, s^0, \mathcal{L}, \longrightarrow)$ where

- S is a finite set of states $\{s_1, s_2, \ldots, s_n\}$
- s^0 is the initial state
- $\mathcal{L}$ is a set of labels $\{a, b, c, \ldots\}$
- $\longrightarrow \subseteq S \times \mathcal{L} \times S$ is the transition relation $s \xrightarrow{a} s'$

Real-time Systems

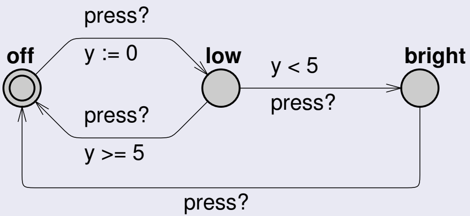
- How to include explicit reference to time in models and specifications?
- No coffee is delivered unless the coffee button is pressed – OK
- When the tea button is pressed, tea is delivered within 45s – ????

Timed Automata

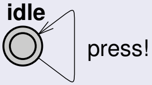
- Timed automaton – finite-state machine extended with clock variables.
- Clock variable evaluates to a real number.
- All the clocks progress synchronously (and keep perfect time!)
- Clocks can be compared against lower and upper bounds. Comparisons can be strict or non-strict
- An invariant condition is associated with each location.
- Each edge has a guard, a label and a set of clocks to reset.
- A system is modelled as a network of timed automata in parallel.

Timed Automaton Example

Lamp



User



We need a User to interact with the Lamp

Timed Automaton

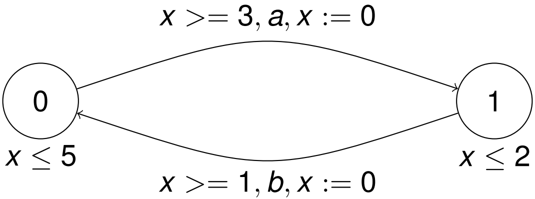
Definition (Timed Automaton)

A timed automaton is a tuple $(L, l_0, C, A, E, \mathcal{I})$, where L is a set of locations, $l_0 \in L$ is the initial location, C is the set of clocks, A is a set of actions, co-actions and the internal τ action, $E \subseteq L \times A \times B(C) \times 2^C \times L$ is a set of edges between locations with an action, a guard and a set of clocks to be reset, and $\mathcal{I} : L \to B(C)$ assigns invariants to locations.

We are following the notation used in Behrmann,G., David,A., and Larsen,K. A Tutorial on Uppaal, Department of Computer Science, Aalborg University, Denmark, 2004

Anatomy of a timed automaton

- Name the parts…
- Location, Edge, Label, Guard, Reset, Invariant



Some notation for clocks

- Assume C is a set of clock variables
- $B(C)$ is the set of conjunctions over simple conditions of the form $x \bowtie c$ or $x - y \bowtie c$, where $x, y \in C$, $c \in N$ and $\bowtie \in \{<, \leq, =, \geq, >\}$.
- Clock valuation
  - A clock valuation u gives the current value of each clock in C, i.e. $u(x)$ gives the value of clock x
  - $u : C \to \mathbb{R}_{\geq 0}$
- Clock reset
  - $[r \mapsto 0]u$ is a clock valuation in which all the clocks in r have the value 0, and all other clocks have the same value as given by u

Some notation for clocks

- Time passes
  - $u + d$ is the clock valuation in which all clocks have their value in u increased by d time units
  - formally $u + d$ maps each clock $x \in C$ to the value $u(x) + d$
- Constraint satisfaction
  - Assume $g \in B(C)$ is some constraint
  - $u \in g$ is the notation that means g is true for clock valuation u.
- Zero valuation
  - u_0 is a clock valuation in which all clocks have the value 0

Exercise

- Assume $C = \{x, y, z\}$, $u = \{x \mapsto 1, y \mapsto 2, z \mapsto 3\}$ and $r = \{x, z\}$.
- Write out the following valuations:
  1. $u + 3$
  2. $[r \mapsto 0]u$
  3. $[r \mapsto 0]u + 10$
- Give clock valuations u such that
  1. $u \in x \leq 5 \wedge z \geq 2$
  2. $u \in y \leq 5 \wedge z \geq 2$

Timed Transition System

- A timed transition system is just a labelled transition system where the labels include the real numbers $\mathbb{R}_{\geq 0}$
- Formally, a TTS is a tuple $(S, s^0, \mathcal{L}, \longrightarrow)$ where
  - S is the set of states
  - s^0 is the initial state
  - $\mathcal{L} = \mathbb{R}_{\geq 0} \cup A$ is the set of labels (assuming A is some set of discrete actions disjoint from $\mathbb{R}$)
  - $\longrightarrow \subseteq S \times L \times S$ is the transition relation, e.g. $s \xrightarrow{3.5} s'$, $s \xrightarrow{a} s'$.

Deriving a TTS from a TA

- The TTS derived from a TA $(L, l_0, C, A, E, \mathcal{I})$ is constructed as follows
  - The set of states is the set of all possible combinations of locations and clock valuations (l, u) where $u \in \mathcal{I}(l)$, i.e. the invariant is satisfied
  - The initial state is given by (l_0, u_0), i.e. the initial location with the zero clock valuation
  - The set of labels is $\mathbb{R}_{\geq 0} \cup A$
  - The transition relation $\longrightarrow$ is given by the rules TA.1 and TA.2, following

Timed Automaton Semantics

Definition (TA semantics)

Let $(L, l_0, C, A, E, \mathcal{I})$ be a timed automaton. The semantics is defined as a labelled transition system $(S, s^0, \longrightarrow)$, where $S \subseteq L \times \mathbb{R}^C$ is the set of states, $s^0 = (l_0, u_0)$ is the initial state, and $\longrightarrow \in S \times (\mathbb{R}_{\geq 0} \cup A) \times S$ is the transition relation such that:

- TA.1 $(l, u) \xrightarrow{a} (l', u')$ if there exists $e = (l, a, g, r, l') \in E$ s.t. $u \in g$, $u' = [r \mapsto 0]u$, and $u' \in \mathcal{I}(l')$, and
- TA.2 $(l, u) \xrightarrow{d} (l, u + d)$ if $\forall\, d' : 0 \leq d' \leq d \Rightarrow u + d' \in \mathcal{I}(l)$.

Properties of Time

- Deterministic whenever $s \xrightarrow{d} s'$ and $s \xrightarrow{d} s''$ then $s' = s''$, i.e. its not possible to reach different states simply by passage of time
- Dense for any two time points $d1 < d2$, there exists a third point d such that $d1 < d < d2$
- Non-Zeno there is no bound on the progress of time, i.e. for any real value c, the time can progress beyond c.

Two-phase system

- Systems are modelled as two-phase systems
- In one phase, time passes
- In the other phase, the state changes
- The phases alternate forever
  Time passes, state changes, time passes, state changes, time passes, state changes, ...
- State changes are instantaneous (take no time)
- Time-consuming activities modelled by distinct start and end actions, e.g. `start_transmission`, `end_transmission`

Exercise

Consider just the Lamp automaton from an earlier slide

1. Write down the components of the tuple for the TA
2. Use the rules TA.1 and TA.2 to derive a (finite prefix of a) possible execution of the automaton

Deadlock and Time-deadlock

- A state s in the TTS of a TA is a deadlocked state if there is no time delay d and action a such that $s \xrightarrow{d} s'' \xrightarrow{a} s'$
- A state s in the TTS of a TA is a time-deadlocked state if every execution from s is a Zeno execution. A Zeno execution is an infinite trace of a system in which the progress of time is bounded by some upper limit $c \in \mathbb{R}_{\geq 0}$
- Give examples of TA with deadlocked and time-deadlocked states.

Uppaal

- www.uppaal.org
- Integrated environment for modelling, simulation and verification of real-time systems
- Developed by universities of Uppsala and Aalborg
- Systems modelled as collection of extended TA
- Applications: real-time controllers, communication protocols etc
- Gearbox controller: Mecel AB and Uppsala