dr inż. Paweł Głuchowski (Wrocław University of Science and Technology, Faculty of Information and Communication Technology)

Information Systems Analysis

Laboratories no. 12

version 4.2

Subject: Temporal logic and timed automata – construction and verification of synchronized timed UPPAAL automata (part 1).

Exercise 1. (5 pts)

To do: Make a system of two automata: *Time* i *Alarm*, modeling the operation of an alarm clock so that their operation is as follows:

The *Time* automaton models the hour and minute:

- the current time is shown by the current state of the automaton: *G0*, *G1*, …, *G11*;
- the current minute is shown by the current value of the *minute* variable of type *clock*, with the values *0–60*;
- the hour change is caused by the minute change from *59* to *60*;
- a minute equal to *60* is reset to *0*;
- the hour and minute are changed endlessly and independently of the *Alarm* automaton.

The *Alarm* automaton models an audible signal that lasts from *7:00* to *8:00*:

- the absence and presence of the signal are shown by the states of the automaton: *SILENCE* and *SIGNAL*;
- the state change is only caused by the automaton *Time* by synchronization by a channel or channels.

The *Time* automaton immediately causes synchronization with the *Alarm* automaton when it changes the hour to *7:00* to set it to the *SIGNAL* state, and when it changes the hour to *8:00* to set it to the *SILENCE* state.

Notice: Use *chan* or *urgent chan* channels to synchronize the automata to maintain the correct moment of the synchronization.

Global declarations (*Declarations*) can only contain channel declarations.

Define numeric variables with the range of their possible values.

Exercise 2. (5 pts)

To do: Verify correctness of operation of the model made in exercise 1 using logic formulas, in particular:

- is an hour exactly 60 minutes (for an exemplary hour), i.e. not less and not more,
- does the alarm last from *7:00* to *8:00* and only then.

Use each of the following types of formulas: reachability, liveness, and safety.

Notice: For each formula, write:

- its form in the UPPAAL language,
- its form in the CTL language,
- its verbal description (i.e. how you understand its operation),
- the result of its verification.

Formulas expressing the correct operation of the model should be verified positively (in green), and formulas expressing incorrect operation of the model – negatively (in red).

Aid to the exercises

Safety

Safety defines the truth of something from now on throughout the future with the temporal operator G ([] in UPPAAL). In CTL logic it is most often described as necessary or unavoidable with the path operator A; less often it is referred to as merely possible, or potential, with the path operator E.

Examples of practical applications of such a formula in the verification of UPPAAL automata:

- $A[]automaton.variable >= 7$ – is it sure that the value of *automaton.variable* is and will always be $\geqslant 15$? (we expect the result *true*);
- $A[]automaton.state\, imply\, automaton.variable == 11$ – is it sure that *automaton.state* results in *automaton.variable = 11*, regardless of the moment of *automaton.state*? (we expect the result *true*).

Reachability

Reachability defines the truth of something now or at some future moment with the temporal operator F (<> in UPPAAL). In CTL logic it is most often described as merely possible, or potential, with the path operator E; less often it is referred to as necessary or unavoidable with the path operator A.

Examples of practical applications of such a formula in the verification of UPPAAL automata:

- $A <> automaton.variable = 15$ – is it sure that *automaton.variable* will ever be *15*? (we expect the result *true*);
- $E <> automaton.state$ – is it not possible that *automaton.state* will be reached sometime? (we expect the result *false*).

Checking the reachability of a formula with an implication, writing the rule *"from something something follows"*, is usually a mistake, because it does not prove the following at any moment in time when the cause occurs.

Liveness

Liveness defines the truth and necessity (inevitability) of the *"cause–effect"* relationship: if the cause occurs now or at some future moment, its effect is certain to occur at the same or some later moment in time. This is expressed by the formula $AG(cause \Rightarrow AF(effect))$ ($cause -- > effect$ in UPPAAL).

Examples of practical applications of such a formula in the verification of UPPAAL automata:

- $automaton.state1 -- > automaton.state2$ – is it sure that reaching sometime *automaton.state1* will eventually lead to *automaton.state2*? (we expect the result *true*).

Recomended websites

- *System model verification (lecture)*
- *UPPAAL timed automata (lecture)*
- *System model verification in UPPAAL (lecture)*
- *UPPAAL 4.0: Small Tutorial*
- *UPPAAL Web Help*