Formal Methods in Software Development
Exercise 9 (January 23)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

December 13, 2011

The result is to be submitted by the deadline stated above *via the Moodle interface* of the course as a *.zip or .tgz* file which contains

- a PDF file with
  - a cover page with the title of the course, your name, Matrikelnummer, and email-address,
  - for each exercise, a section with the number and name of the exercise and the content of the exercise,
- the file with the Promela model used in the exercise.

9a: Modeling an Elevator (40)

Given a natural number $N \geq 0$, an *N-floor elevator* operates as follows:

1. The elevator is always at one of the floors $0 \ldots N$ (we ignore the times when the elevator is between floors) and it is always in one of the states "stopped", "moving up", or "moving down". Initially the elevator is stopped at floor 0.
2. Inside the cabin, the elevator has a button b_i for each floor i; each button is either "on" (its light is on) or "off" (its light is off). At any time, any button b_i that is "off" may become "on" (a person has pressed the button which indicates a request to move to floor i). Initially no button is "on".
3. On each floor i, there is a button f_i that may be "on" or "off". At any time, a button f_i that is "off" may become "on" (a person has pressed the button which indicates a request for the elevator). Initially no button is on.
4. If no button is pressed, the elevator starts to move to floor 0, and then waits for a button b_i or f_i to be pressed. As soon as this happens, the elevator starts to move to floor i.
5. If the elevator moves and reaches a floor i, it stops at that floor if button b_i or f_i is "on" (the buttons become "off" again). Then it behaves as follows:
   a) If some button b_j or f_j is pressed in the direction in which the elevator was moving before the stop, the elevator continues its movement in the same direction.
   b) If there is no button b_j or f_j pressed in the direction in which the elevator was moving but one such button is pressed in the opposite direction, the elevator starts to move in the new direction.
6. If the elevator is moving to a floor for which there is no request, the elevator is continuing to move towards the direction of the current request (if there is any).

Model this system formally by defining its state space, initial state condition, and transition relation in the syntax specified in the lecture:

$$\begin{aligned} &\mathit{State} := \ldots \\ &I(\ldots) :\Leftrightarrow \ldots \\ &R(\ldots, \ldots) :\Leftrightarrow \ldots \end{aligned}$$

In our model, the individual transitions are

- pressing a button,
- changing the status of the elevator, and/or
- moving the elevator from one floor to another.

Do not overlook that you have to record in the system the direction of the movement of the elevator before it stopped.

Define auxiliary predicates for the definition of R which shall be as readable as possible. Also use comments to indicate your informal intentions.

Show (the initial part of) a run of the system (a sequence of states) for $N = 3$ with three buttons pressed until the elevator is again stopped at floor 0.

9b: Modeling an Elevator System (20)

Model a system with $E \geq 1$ elevators of the kind similar to the one shown above. Each elevator e (with $0 \leq e < E$) has its own set of floor buttons b_i^e, but all elevators share a common set of request buttons f_i (if a request button is pressed, any elevator may stop to handle the request).

Model this system by defining its state space, initial state condition, and transition relation using the interleaving model of concurrency.

Redefine the transition relation defined in the previous assignment to a transition relation $R_e(\ldots,\ldots)$ for elevator e such that it may serve as a building block of the transition relation of the elevator system:

$$R(\ldots,\ldots) :\Leftrightarrow \ldots \vee (\exists e : 0 \leq e < E \wedge R_e(\ldots,\ldots))$$

However, notice that the definition of the transition relation has to be appropriately changed to make sense.

9c: Modeling the Elevator in Promela (40)

Formalize the model of Exercise 9a for $N = 3$ in Promela; give the source code of the model and a screenshot of (part of) an execution trace in Spin.

In the model, use a single `proctype Elevator()` that models the elevator and start the corresponding process by `run Elevator()`.

Use global arrays b and e to model the states of the buttons and global scalar variables to model the status of the elevator.

The core of the `Elevator()` process is a forever running loop

```
do
  :: c1 -> a1;
  :: c2 -> a2;
  ...
od;
```

where every transition i is modeled by a pair of an (optional) state condition c_i (constraining the state in which the transition is enabled) and action a_i (describing the effect of the transition). If multiple conditions hold, some of the enabled transitions is nondeterministically chosen for execution; if no condition holds, the system is deadlocked (which must not be the case in our model).

Define the process in close analogy to the model of Exercise 9a. However, since Promela expressions do not support quantified formulas as state conditions, you have to expand these to corresponding finite conjunctions or disjunctions.

Hints for the Formalization

Let $\mathbb{B} := \{\mathsf{true}, \mathsf{false}\}$ and $\mathbb{N}_N := \{n \in \mathbb{N} : n < N\}$.

- A single button b can be modelled as a value $b \in \mathbb{B}$.
- A sequence of $N+1$ buttons can be modelled as a function (array) $b : \mathbb{N}_{N+1} \to \mathbb{B}$, i.e. b_i $(= b(i))$ is the status of button i.
- E sequences of $N+1$ buttons can be modelled as a function (array) $b : (\mathbb{N}_E \times \mathbb{N}_{N+1}) \to \mathbb{B}$, i.e. b_i^e $(= b(e, i))$ is the status of button i in sequence e.
- For given function (array) a, $a[i \mapsto e]$ represents a copy of a except that index i is mapped to value e.