Prof. Dr. A. Podelski, Dr. B. Westphal
Ch. Herrera Wintersemester 2014/15

Software Design, Modeling, and Analysis in UML

`http://swt.informatik.uni-freiburg.de/teaching/WS2014-15/sdmauml`

Exercise Sheet 5

Early submission: Monday, 2015-01-12, 12:00 Regular submission: Tuesday, 2015-01-13, 8:00

Exercise 1 (15/20 Points)

Consider the UML model for a sensor system given in Figure 1. The class diagram in Figure 1(a) models *sensors* which can be assigned to a *master*.
The state-machines shown in Figure 1(c) and 1(b) model a design idea for a simple self-monitoring protocol which intuitively works as follows:

- at certain, well-defined points in time, each sensor sends an LZ ("'Lebenszeichen"', sign of being alive) message to its master its master,
- as we don't have means to model real-time in UML, we represent it as an environment signal T; intuitively, a sensor is told by its timer (which is not part of the model but part of the environment) when it is time to send the LZ,
- after sending the LZ, the sensor switches on some blinking lights to indicate that self-monitoring takes place and waits for an acknowledgement from the master,
- in the final protocol, a master sends an acknowledgement only if the LZ is sent at the right point in time,
- the condition which tells which time is "'the right time"' is not yet fixed, but is is clear that the master will be prepared for both situation, right point in time and not right point in time; this is modelled by the non-determinism in Figure 1(b).

(i) The signature corresponding to the class diagram in Figure 1(a) is called $\mathscr{S}_0$ in the definition of system configuration.

Provide the signature $\mathscr{S}$ which is constructed from $\mathscr{S}_0$ and which defines the system states used in system configurations. (2)

(ii) Provide the core state machines for Figures 1(c) and 1(b). (2)

(iii) Consider the system configuration (σ, ε) as given by Figure 2.

Analyse which behaviour is possible in the model starting from (σ, ε) and convince the reader that your analysis is exhaustive, i.e. that you really considered *all possible computation paths* starting with (σ, ε).

For simplicity, consider environment interaction only once in your analysis, namely exactly before an instance of T can be dispatched. Discuss where else environment interaction may take place and what the effect on the computation paths would be.

As ether, consider a single, global, reliable FIFO used by all instances of active classes.

In order to convince the reader, provide for each transition at least the destination system configuration, the labelling with *cons*, *Snd*, u, and point out by which of the rules (i)–(v) it is justified. (10)

Hint: choose a good way to represent the results of your analysis, i.e. a representation which is convenient to write (for you) and at the same time well readable for, e.g., the tutor. Explain your notation.

(iv) In the computation paths considered in the previous exercise, point out one example of *a step* and one example of *a run-to-completion step* of length strictly larger than 1. (1)

Exercise 2 (5/20 Points)

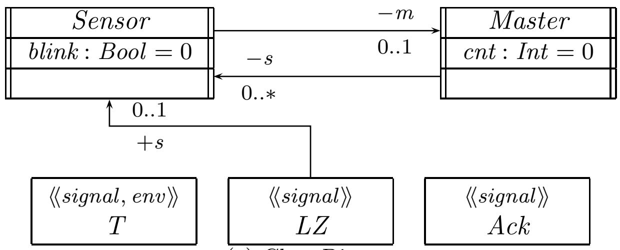
By now, we defined a satisfaction relation between system states and OCL constraints, but we didn't define how to consider OCL constraints in computation paths. That is, we didn't define what it means that the behaviour of a UML model satisfies an OCL constraint. What would you propose? Which (reasonable) options do you see?
Discuss the difference between these options using the state machine from Figure 3 and the following OCL constraints:

(i) $\mathsf{context}\ C\ \mathsf{inv} : x > 0$

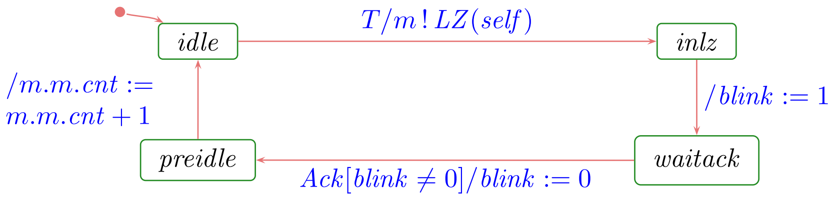
(ii) $\mathsf{context}\ C\ \mathsf{inv} : x \neq -1$

(iii) $\mathsf{context}\ C\ \mathsf{inv} : x \neq 2$

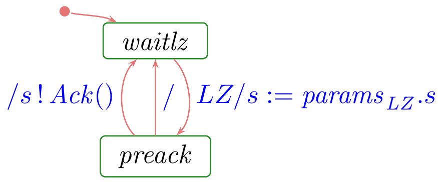
Would the model satisfy them or not? You may assume that the state machine shown in Figure 3 is the state machine of a class C whose attribute $x : Int$ is initialised to the value 27 at creation time. Further assume that E and F are environment signals, i.e. they may occur at any point in time.
Do you have a favourite option or a recommendation?



(a) Class Diagram.



(b) State machine of *Master*.



(c) State machine of *Sensor*.

Figure 1: UML model for Exercise 1.

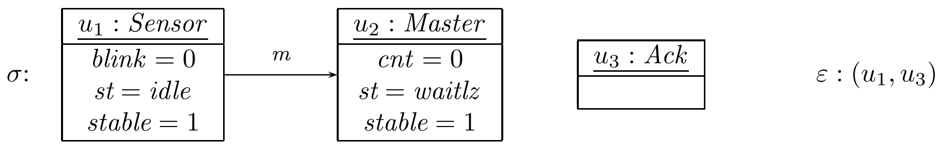


Figure 2: (Complete) system configuration for Exercise 1.

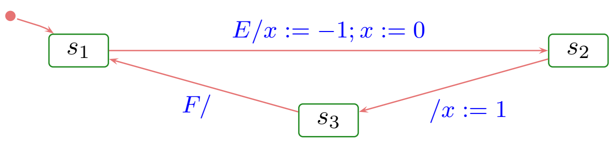


Figure 3: State machine for Exercise 2.