
Softwaretechnik/Software Engineering

<http://swt.informatik.uni-freiburg.de/teaching/SS2020/swtv1>

Exercise Sheet 4

Early submission: Monday, 2020-07-06, 14:00

Regular submission: Tuesday, 2020-07-07, 14:00

Exercise 1 – Computation Graph / Transition Graph (8/20 Points)

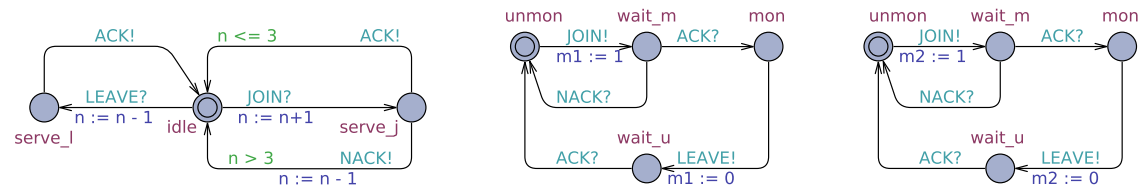


Figure 1: Network of Communicating Finite Automata

Provide the reachable part of the transition graph of the CFA model shown in Figure 1.

Hint: Make sure to clearly indicate the initial configuration(s).

Exercise 2 – Basic Behavioural Model Analysis (7/20 Points)

The file `sensormaster.xml` provides a model of the sensor/master system from Exercise Sheet 4. Recall the requirement from Exercise Sheet 4, that masters only need to monitor a bounded number of sensors.

- (i) Use the UPPAAL¹ simulator to create one computation path which shows that the model is able to exhibit the simple scenario that one sensor joins a master and then leaves again. (1)

Hint: Include a screenshot showing the final state of the computation path and the sequence diagram into your submitted document; and submit the .xtr file of the simulation as well so that your tutor can reproduce your result.

- (ii) a) Use the verifier to check whether, for some master, it is possible to monitor the allowed maximum number of sensors (as given by global constant/parameter `max_sns`). (2)

What is the result? How should this result be interpreted?

Hint: That is, what can we conclude from the result of the above check about the modelled design idea?

- b) Use the verifier to check whether it holds that for each master, the value of `n` never exceeds the assumed maximum number of sensors. Some queries have been proposed for the check and are included in `sensormaster.xml`.

What are the results? How should these results be interpreted?

Hint: Submit a copy of `sensormaster.xml` with your 1.(ii).a) query filled in; make sure that your tutor understands which of the many files you submit is to be considered for this task.

¹See Appendix A for instructions.

- (iii) The author of the model has expressed doubts about the model's correctness. One invariant that needs to hold for the model to be correct (according to the requirements department) is that if the value of variable `m` in any sensor is a proper master's id, then the value of `n` of this master must not be 0.

Explain how the outcome of checking the corresponding query in `sensormaster.xml` (unfortunately) confirms these doubts. (1)

- (iv) Explain the reason for the incorrectness of the model according to Task (iii) and suggest an as small-as-possible change that preserves the outcomes of the Tasks (i) and (ii), but changes the outcome of the previous check. (1)

Hint: Also submit the changed model in a separate file.

Hint: If you do not spot any reasons for the incorrectness, ask your tutor for clues (together with a clear description of the state of your investigation and your hypotheses so far).

Exercise 3 – Cross-Connections

(5/20 Points)

It is claimed that Task 2.(ii) is conceptionally related to Exercise 3 on Exercise Sheet 3 ('Writing OCL Formulae') as there are similarities and differences. Both, Task 2.(ii) and that Exercise 3 refer to models of a wireless fire alarm system, for a start. Yet Exercise 3 was part of an exercise sheet on the topic 'structural software modelling' and this exercise sheet addresses 'behavioural software modelling'.

- (i) Does the model from Task 2.(ii) have a structural aspect? Where is it? How is system structure represented in the model to be considered for Task 2.(ii)? (2)
- (ii) Which task of that Exercise 3 is Task 2.(iib) most closely related to, in your opinion, and what could be a purpose of Exercise 3 that is completed by means of Task 2.(iib). (2)
- (iii) Discuss Task 2.(iia) similarly to the previous task in this exercise, now in relation to the whole Exercise Sheet 3. (1)

Hint: If you are unsure about, e.g., how to understand the questions above or how to answer, then do not hesitate to contact your tutor: Just briefly explain your state of thoughts or doubts, and your tutor will be happy to provide further hints.

A Tool Availability

UPPAAL is available on the Linux machines in the computer pool.

To run UPPAAL, use the following command:

```
/usr/local/ufrb/uppaal/uppaal-4.1.19/uppaal
```

Note: if you *have access to other versions of* UPPAAL then the one named above *and need to use one of those*, please make sure that the version named above is able to read the model and trace files that you submit, since this is the version that the tutors will use. The (otherwise remarkably) stable UPPAAL tool faced some regressions regarding file formats lately (4.1.22, for example, is known to be fine).