Formal Methods in Software Development
Exercise 10 (January 31)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

January 19, 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

1. a PDF file with
   - a cover page with the course title, your name, Matrikelnummer, and email address,
   - the deliverables requested in the description of the exercise,
2. the file with the Promela model used in the exercise.
3. the files with the LTL properties (Button “Save As” in the LTL Property Manager).

Exercise 10: Model Checking in Spin

Consider a system with $N + 1 \geq 1$ processes numbered $0, \ldots, N$ linked by a ring of buffered channels such that every process can send a message to its successor in the ring (respectively every process can receive a message from its predecessor). Each of the processes $1, \ldots, N$ maintains a bit, process 0 would like to know whether all the bits are 1 or not.

In more detail, the system proceeds in an infinite number of rounds as follows:

- Process 0 starts a round by sending a message with value 1 to process 1.
- Every process $1, \ldots, N$ receives a message from its predecessor and generates a random value for its own bit. If the bit is 0, it sends a message with value 0 to its successor, otherwise it forwards the received value.
- When process 0 receives a message from process N, the round is completed.

Your task is as follows:

1. Develop a Promela model for the system with a global bit array of length N to hold the bits of the N processes. Deliver the source of the model.
2. Validate the model by simulation; the model must not run into a deadlock. Deliver a screenshot of (the final part) of some simulation (suspended after some hundreds of steps).
3. Formulate in PLTL the properties formulated below. Deliver the formulas (including the definitions of the atomic predicates) and indicate whether the formulated property is a safety property, a fairness property, or a combination of both (justify your opinion).
4. Model check each formula for $N = 2$ (using appropriate fairness constraints, if necessary, and optimization options, if necessary). Deliver the options you have chosen, your reason for choosing the options, the output of Spin for the verification, and your interpretation:
   - Does the property hold or not?
   - Does this indicate an error in your model/specification or not (i.e. is the answer to be expected or not)? Why?

   If the property does not hold, give a screenshot of a (final part) of the counterexample run and give an interpretation of this run.

The formulas to be formulated and model-checked are the following:

1. Always, if process 0 receives a message, the message holds value 1 if and only if all the processes $1, \ldots, N$ currently hold bit 1.
2. Process 0 always receives value 0.
3. Process 0 eventually receives value 1.
4. Process 0 infinitely often receives a message.
5. Always, after sending a message, process 0 does not send another message until it receives a message (which will be eventually the case).