Formal Methods in Software Development
Exercise 8 (January 28)

Wolfgang Schreiner
Wolfgang.Schreiner@risc.uni-linz.ac.at

December 13, 2007

The result is to be submitted by the deadline stated above via the Moodle interface 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,
  - the PROMELA model of the system,
  - the formulation of each property in Spin's version of PLTL (including the definitions of the atomic predicates),
  - for each property, the output of the Spin model checker,
  - for each violated property, (a significant part of) a screenshot of the sequence chart for a counterexample run.
- files (.promela and .pltl) with the PROMELA model of the system and the Spin formulations of the properties that were model checked.

8(all): Verifying a Manager/Worker System

Take the PROMELA model of a manager/worker system developed in Exercise 7 (with suitable modifications) and formulate the following properties in Spin's version of PLTL:

1. If the system terminates, all tasks have been completed.
2. The system terminates.
3. For every task submitted to a worker, eventually a result is returned to the manager.
4. Every task is eventually being executed by some worker.
5. Every worker eventually receives a task for execution.

6. Every task runs exactly through the states *open* → *under computation* → *completed* (with no return to a previous state).

In the formulation of these properties, you may replace every universally quantified statement ("every task does ...") by a statement on a single element ("task 0 does ..."). Give the PROMELA model to which the properties refer and, for each property, the corresponding PLTL formula and the definitions of the atomic predicates referenced in the formula.

Use the Spin model checker to determine the truth of each property in a system with $P = 3$ managers and $N = 10$ tasks (if the model checking should take too long, you may also reduce P and/or N); give the Spin output for each property. If a property is violated by the system, determine a counterexample run and give (a significant part of) a screenshot of the corresponding sequence chart.