Formal Methods in Software Development
Exercise 3 (June 9)

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

May 9, 2005

The exercise is to be submitted by **June 9** (hard deadline)

1. either as a single paper report (cover page with full name and Matrikelnummer, pages stapled) which is handed out to me in class,
2. or as a single PDF file sent to me per email.

Questions can be asked per email or in the classes before the deadline.

1 Dining Philosophers

Assume the following scenario: n philosophers sit around a pot of spaghetti with n forks placed between any pair of philosophers. A philosopher might become hungry in which case he first grabs the left fork, then grabs the right fork, then uses both forks to eat some spaghetti, then returns both forks. Apparently, if one of the forks is already in use by the corresponding neighbor, the hungry philosopher has to wait until the fork is returned.

1. Formulate this scenario with $n = 5$ as a PROMELA model and verify/falsify in SPIN that it is never the case that every philosopher is waiting for a fork. In case that this claim is falsified, give a high-level interpretation of the counterexample produced by SPIN.
2. Change the model such that one of the philosophers grabs first the right fork, then the left fork. Repeat the exercise.
3. Use the new model to verify/falsify that it is always the case that, if a philosopher gets hungry, he eventually is eating. In case that this claim is falsified, give a high-level interpretation of the SPIN counterexample.
4. Repeat the exercise under the assumption (to be formulated as part of the PLTL property to be verified) of strong fairness of getting access to a fork: if it is infinitely often the case that a philosopher has a chance of grabbing the fork (because he wants it and it is availabe), the philosopher is eventually able to grab it.

The result of this exercise consists of

1. the listings of the PROMELA models,
2. the PLTL formulas used in the verifications (including the definitions of the atomic predicates),
3. the SPIN output of each attempted verification run,
4. the counterexample runs produced by SPIN (if any) and their high-level interpretations.

2 Asynchronous Systems

Take a system of two asynchronously executing components S(ender) and R(eceiver) connected by three signal lines v(alue), r(eady) and a(cknowledge). Initially all signal lines are set to 0.

S may generate a local value $x \in \{0, 1\}$ and send this value to R by setting v to x and setting r to 1. Then S waits until R acknowledges the receipt of the value by setting a to 1. Then S sets r to 0 and waits until S sets a to 0. Now S is ready to generate and send another value.

R waits until S sets r to 1, then accepts the value of v and afterwards sets a to 1. It then waits until R sets r to 0 and then sets a to 0. Now R is ready to receive another value.

1. Formulate this system as a PROMELA model and verify/falsify, that
   (a) whenever S accepts a value, this is the current value of x,
   (b) the system runs forever.
2. Write a formal specification of the system by denoting its state space, its initial state condition, and its transition relation. Give an invariant of the system which is as strong as possible.

The result of this exercise consists of

1. the listing of the PROMELA model,
2. the PLTL formulas used in the verification (including the definitions of the atomic predicates),
3. the SPIN output of each attempted verification run,
4. the counterexample runs produced by SPIN (if any) and their high-level interpretations,
5. the formal specification and the invariant.

3 Bonus (30%)

Give a manual proof that the asynchronous system of the last exercise has the first of the stated properties.