

Problems and Exercises
“Nichtsequentielle Systeme und nebenläufige
Prozesse”, SS05
Part 4

Prof. Helmut Veith
Dipl.-Ing. Christian Schallhart
Dr. Stefan Katzenbeisser

SPIN

Use SPIN to solve the following exercises. SPIN can be obtained from <http://spinroot.com>

1. *Unreliable Channel*. Model an unreliable channel, i.e., a channel which might loose packets. Use this channel transmit packets from a sender to a receiver process. Use SPIN to Simulate the system.
2. *Unreliable Channel: Proof*. Prove that the channel is unreliable, i.e., that not every packet that is sent, will arrive at the receiver.
3. *Mutual Exclusion*. Consider the following algorithm, well published (Comm. of the ACM, Vol. 9, No. 1, p. 45) in pseudo-Algol:

```
1 Boolean array b(0;1) integer k, i, j,
2 comment process i, with i either 0 or 1 and j = 1-i;
3 C0: b(i) := false;
4 C1: if k != i then begin
5 C2:   if not (b(j)) then go to C2;
6     else k := i; go to C1 end;
7   else critical section;
8     b(i) := true;
9     remainder of program;
10    go to C0;
11    end
```

Model it in Promela and prove or disprove its correctness!

4. *Dining Philosophers*. Model the Dining Philosophers Problem in Promela with 5 philosophers. Prove that they could starve. Modify the model such that each philosopher will eventually eat. Prove this property.
5. * *Token Ring*. A token ring consists of m independent processors which are arranged in a cycle, where each processor is connected to its left and right neighbors. The processes of the token ring use a token (represented by a message in channel) to synchronize each other. After each processing step, the token is passed on to one of its neighbors.
 - Implement the token ring for $m = 4$, where the token is passed to one of the neighbors nondeterministically. Simulate the token ring in the interactive environment.
 - Use SPIN to check whether it is guaranteed that at most one processor gets access to the critical section at the same time.
 - Use SPIN to check whether a deadlock can occur.
 - Use SPIN to check whether at least one process enters the critical section infinitely often, i.e., whether progress is achieved.
 - Repeat the above steps for a model where the token is passed deterministically to the left.
 - *Optional*: Find the maximal m for which you can verify the model on your machine.
6. * *Token Ring: Fairness*. Use the two models from the last exercise. Use SPIN to produce a `never` claim which states that each process gets access to the critical section infinitely often, i.e., whether each process is able to make progress. Check this condition.