Verification Techniques 2010, Homework 2

The homework til **Tuesday, Feb. 9** is to hand in solutions to exercises A and B.

A. Solve the old Farmer, Wolf, Sheep, and Cabbage problem:

> A farmer must ship a wolf, a sheep, and a cabbage across a river. He has a boat which takes one of them. But he must not leave the wolf alone with the sheep, nor the sheep alone with the cabbage on either shore if he is not there himself to watch them.

Make a **beautiful** Promal model, by which SPIN can find a best solution to this problem. This means that your model should **not** give any hint about what is the solution. You model should have three clear parts:

- one part which just allows shipping, without checking that "bad pairs" are left alone,
- one part which aborts the search of SPIN if"bad pairs" are left alone,
- a condition, which SPIN interprets as an error, thereby generating an error trace, which is the final solution.

B. Below is, in pseudocode, an algorithm, due to Burns, for solving the mutual exclution problem between an arbitrary number of processes. The processes are number from 1 to N. Below is the pseudocode executed by process i. Each process i has a variable $\mathit{flag}(i)$, which can be written by process i, and read by any process.

```
     enter the trying section
L    flag(i) := 0
     for j from 1 upto i − 1 do
          if flag(j) = 1 then goto L
     flag(i) := 1
     for j from 1 upto i − 1 do
          if flag(j) = 1 then goto L
M    for j from i + 1 upto N do
          if flag(j) = 1 then goto M
     *** critical region ***
     flag(i) := 0
     goto beginning
```

Your task is to model this algorithm in Promela, and to use SPIN to investigate whether the algorithm satisfies the following properties:

- mutual exclusion, i.e., that at most one process can be in its critical region at any time
- freedom of starvation, i.e., that whenever a process has reached L, it will at some later point reach M. This property might be satisfied differently for different process indices i.

- Check whether the answers to these questions change if we remove the last statement (goto beginning), meaning that processes enter the critical section at most once.

Your solution should contain beautiful Promela models, describing how you added checks, and results of SPIN runs. If the property is satisfied, use a not too small value of N. If the property is not satisfied, show a shortest error-trace leading to a bad state or a bad loop with as few processes as possible.