Prof. Bernd Finkbeiner, Ph.D.
Martin Zimmermann, Ph.D.
Leander Tentrup, B.Sc.

Summer term 2013
Problem Set 9
September 19, 2013

# Verification

## Problem 1: Fairness and Liveness with SPIN [6 Points]

Recall our exercise 7.3 (see below for a partial solution) where non-starvation of a channel-based mutual exclusion protocol was not provable. The reason was that one process could play unfair and take the token infinitely many times even though the other process was waiting for it. This kind of unfair behaviour can be ruled out by fairness assumptions.

In order to check liveness properties in SPIN, assertions are not powerful enough. Therefore, you should use either the builtin *ltl*-primitive or specify your properties as *never*-claims (i.e. a non-deterministic Büchi automaton).

1. Prove non-starvation for both processes under suitable (weak or strong) fairness assumptions. Hint: You should prove LTL-formulas of the form fairness assumption $\Rightarrow$ non-starvation. The fairness assumption may use the propositions enP0, enP1 execP0, execP1 below. [3 Points]

2. Prove that your fairness assumptions are not too strong. That is, prove that the system has fair traces. [3 Points]

```
mtype {TOKEN}; /* token to be passed */
  chan ch = [1] of {mtype}; /* global channel containing token */

proctype P(byte i) {
  think:  printf("MSC: P(\%d) thinking.\n", i);
  wait: ch?TOKEN;
  use:  printf("MSC: P(\%d) enters CS.\n", i);
      /* critical section */
      printf("MSC: P(\%d) leaves CS.\n", i);
  leave:  ch!TOKEN;
      goto think
}

byte pidP0, pidP1; /* process IDs of P(0) and P(1) */
init {
  atomic { ch!TOKEN; pidP0 = run P(0); pidP1 = run P(1); }
}
/* non-starvation for P(0): [] (waitP0 -> <> useP0) *
* non-starvation for P(1): [] (waitP1 -> <> useP1) */

#define useP0 (P[pidP0]@use)
#define useP1 (P[pidP1]@use)
#define waitP0 (P[pidP0]@wait)
#define waitP1 (P[pidP1]@wait)
#define enP0 (enabled(pidP0))
#define enP1 (enabled(pidP1))
```

```
#define execP0 (_last == pidP0)
#define execP1 (_last == pidP1)
```

---

The following exercises belong to the afternoon session.

## Problem 1: Bounded Model Checking [6 Points]

Use the Bounded Model Checking approach to find a counterexample of length 3 for the property $\Box(\neg x \vee \neg r_1)$ over the atomic propositions $\{x, r_1, r_2\}$ on the transition system that is given symbolically as follows:
Initial State: $(\neg x \wedge \neg r_1 \wedge \neg r_2)$
Transition function: $f_{\rightarrow}(x, r_1, r_2, x', r_1', r_2') = \Big(r_1' \leftrightarrow \big((x \wedge \neg r_1) \vee (\neg x \wedge r_1)\big)\Big) \wedge \Big(r_2' \leftrightarrow \big((\neg x \wedge r_2) \vee (x \wedge r_1)\big)\Big)$