ALBERT-LUDWIGS-
UNIVERSITÄT FREIBURG

Dr. Stefan Schwoon
Sergiy Bogomolov

April 27, 2009
Discussion: April 29, 2009
SR 00-034, Building 051

Tutorials for Model Checking
Exercise sheet 1

Exercise 1: Peterson's algorithm

Peterson's algorithm is a concurrent programming algorithm for mutual exclusion that allows two processes to share a single-use resource without conflict (i.e. two processes cannot enter the critical section simultaneously; such property is called mutual exclusion), using only shared memory for communication. Consider the following variant of Peterson's algorithm, where b_i is a boolean variable, x is an integer variable which has two values 1 and 2.

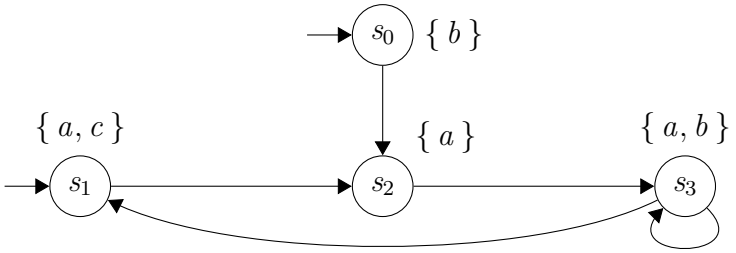
```
        while true  {

nc :      b1 := true;
req :     x := 2;
wt :      wait until(x = 1 ∨ ¬b2);
cs :      skip
          b1 := false;


        }
```

```
        while true  {

nc :      b2 := true;
req :     x := 1;
wt :      wait until(x = 2 ∨ ¬b1);
cs :      skip
          b2 := false;


        }
```

Give the transition system for the interleaving of these programs. Is mutual exclusion ensured?

Exercise 2: States satisfying LTL-formulae

We introduce the following notation. Let M be a Kripke structure and s a state of M. We say that s satisfies M (written $(M, s) \models \psi$) if $M' \models \psi$ where M' is the same Kripke structure as M but with state s as the initial state.
Consider the following transition system over the set of atomic propostions $\{ a, b, c \}$:



Give for each of the following LTL-formulae the set of states for which the formula is satisfied.

(a) $a \wedge \mathsf{X}\, b$	(b) $\mathsf{X}\, c$	(c) $\mathsf{X}\mathsf{X}\, c$
(d) $a \,\mathsf{U}\, b$	(e) $b \,\mathsf{U}\, a$	(f) $b \,\mathsf{U}\, (\mathsf{G}\, a)$
(g) $a \,\mathsf{U}\, (\mathsf{G}\, b)$	(h) $\neg(a \,\mathsf{U}\, \mathsf{G}\, b)$	(i) $(\mathsf{F}\, c) \,\mathsf{U}\, (\mathsf{G}\, a)$
(j) $\mathsf{F}\,\mathsf{G}\, a$	(k) $\mathsf{G}\,\mathsf{F}\, b$	(l) $\mathsf{G}\,\mathsf{F}\, c$

Exercise 3: Stating Properties in LTL
Consider a lift system that services N floors numbered 0 through $N-1$. Assume $door(i)$ indicates that the doors on the i-th floor are open, $lift(i)$ indicates that the lift is at floor i, and $req(i)$ indicates that the request button at floor i was pressed and is lit. In the lift cabin there are N buttons for the floors and $send(i)$ indicates that the i-th send button is lit. For solving this exercise the following notion may be useful: $\bigwedge_{i=0}^{M-1} H_i = H_0 \wedge H_1 \wedge \ldots \wedge H_M$.
State the following properties in LTL.

(a) A floor door is never open if the cabin is not present at that floor.

(b) A requested floor will be served sometime.

(c) The lift returns to floor 0 infinitely often.

(d) The lift does not move unless there is some request.

Exercise 4: LTL tautologies
Which of the following formulae are tautologies? An LTL formula is a tautology if it holds for any sequence of valuations. Give a counterexample for any formula that does not always hold.

(a) $(\mathsf{G}\,\varphi \rightarrow \mathsf{F}\,\psi) \leftrightarrow (\varphi \,\mathsf{U}\, (\psi \vee \neg\varphi))$,

(b) $(\mathsf{F}\,\mathsf{G}\,\varphi \rightarrow \mathsf{G}\,\mathsf{F}\,\psi) \leftrightarrow \mathsf{G}(\varphi \,\mathsf{U}\, (\psi \vee \neg\varphi))$,

(c) $\mathsf{F}(\varphi \wedge \psi) \leftrightarrow \mathsf{F}\,\varphi \wedge \mathsf{F}\,\psi$,

(d) $\mathsf{G}(\varphi \rightarrow \mathsf{X}\,\varphi) \rightarrow (\varphi \rightarrow \mathsf{G}\,\varphi)$,

where φ and ψ are LTL formulas.