RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

LEHRSTUHL FÜR INFORMATIK II
RWTH Aachen · D-52056 Aachen · GERMANY
http://www-i2.informatik.rwth-aachen.de/

Prof. Dr. Ir. J.-P. Katoen
M. Neuhäußer

Introduction to Model Checking
Summer term 2007

– Series 2 –

Hand in on April 20 before the exercise class.

Exercise 1 (1 + 1 + 2 points)

Consider a system consisting of n processes $P_0, \ldots, P_{n-1}$ and a central moderator M in a fully connected network. Each process P_i (for $0 \leq i < n$) executes the same algorithm and stores a unique identifier $\mathrm{id}_i \in \mathbb{N}$. Further, we assume that n is known a priori.

In order to elect a leader, the system is supposed to determine the process with the highest *id* and communicate it to every process.

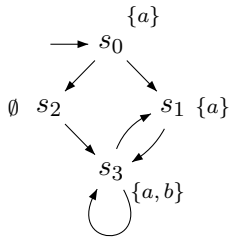
a) Informally describe how to solve the leader election problem in the above setting.

b) Write nanoPromela models for the algorithm of the process and the moderator. Add comments!

c) Formally derive the program graphs for a process and the moderator.

Exercise 2 (1 points)

Consider the transition system given below. Formally define its traces!



Exercise 3 (3 + 2 points)

Consider the set AP of atomic propositions defined by $AP = \{x = 0, x > 1\}$ and consider a non-terminating sequential computer program P that manipulates the variable x over the domain $\mathbb{N}$.

a) Formulate the following informally stated properties as LT properties:

- false and true
- initially x is equal to zero
- initially x differs from zero
- x exceeds one only finitely many times
- the value of x alternates between zero and one
- initially x is equal to zero, but at some point x exceeds one

b) Determine for each LT property whether it is a safety or a liveness property. Justify your answers!

Exercise 4 **(2 + 1 + 4 points)**

Consider the following generalization of Peterson's mutual exclusion algorithm that is aimed for an arbitrary number $n \geq 2$ of processes. The basic concept of the algorithm is that each process passes through n "levels" before acquiring access to the critical section.
The concurrent processes share two bounded integer arrays:

- $y[0..n-1]$ with $y[k] \in \{1, \ldots, n\}$. $y[j] = i$ means that process i has the lowest priority at level j.
- $p[1..n]$ with $p[k] \in \{0, \ldots, n-1\}$. $p[i] = j$ expresses that process i is currently at level j.

Each process starts at level 0; before entering the critical section, it has to pass through levels 1 to $n-1$. Process i waits at level j until either all other processes are at lower levels (i.e. $p[k] < j$ for all $k \neq i$) or another process enters level j, thereby enabling process i to move to the next level (i.e. $y[j] \neq i$). The behaviour of process i is given by the following algorithm:

while true **do**
  ... non-critical section ...
  for $j := 1$ **to** $n-1$ **do**
    $p[i] := j$;
    $y[j] := i$;
    wait until $(y[j] \neq i) \vee (\bigwedge_{k \neq i} p[k] < j)$
  od
  ... critical section ...
  $p[i] := 0$;
od

a) Formally define the program graph for process i and outline it.

b) Determine the cardinality of the set of states of the parallel composition $TS(P_1 ||| P_2 ||| \cdots ||| P_n)$.

c) Prove, that this algorithm indeed ensures mutual exclusion for n processes.