RHEINISCH-
WESTFÄLISCHE
TECHNISCHE
HOCHSCHULE
AACHEN

LEHRSTUHL FÜR INFORMATIK 2
RWTH Aachen · D-52056 Aachen
http://www-i2.informatik.rwth-aachen.de/

Prof. Dr. Ir. J.-P. Katoen
Dr. T. Han and A. Mereacre

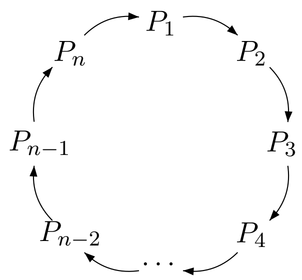
Introduction to Model Checking
Summer term 2010

– Series 2 –

Hand in on May 5 before the exercise class.

Exercise 1 (5 + 5 points)

Consider the following leader election algorithm: For $n \in \mathbb{N}$, n processes $P_1, \ldots, P_n$ are located in a ring topology where each process is connected by an unidirectional channel to its neighbour as outlined on the right.
To distinguish the processes, each process is assigned a unique identifier $id \in \{1, \ldots, n\}$. The aim is to elect the process with the highest identifier as the leader within the ring. Therefore each process executes the following algorithm:




```
send (id);                          initially set to process' id
while (true) do
    receive (m);
    if (m == id) then stop;         process is the leader
    if (m > id) then send (m);      forward identifier
od
```

a) Model the leader election protocol for n processes as a channel system.

b) Give an initial execution fragment of $TS([P_1|P_2|P_3])$ such that at least one process has executed the send-statement within the body of the while-loop.
Assume for $1 \leq i \leq 3$, that process P_i has identifier $id_i = i$.

Exercise 2 (2 + 3 + 5 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 $\text{id}_i \in \mathbb{N}$. 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.

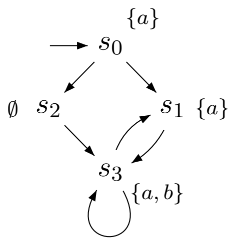
a) Informally describe how to solve the leader election problem in the above setting.

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

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

Exercise 3 **(10 points)**

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



Exercise 4 **($3 + 7$ points)**

Let TS denote a transition system with possible terminal states.

(a) Formally define a (reasonable) transformation $TS \mapsto TS^*$ such that TS^* has no terminal states but is otherwise "equivalent" to TS.

(b) Prove, that the transformation preserves trace-equivalence, i.e. show that for transition systems TS_1 and TS_2 with $Traces(TS_1) = Traces(TS_2)$, it follows $Traces(TS_1^*) = Traces(TS_2^*)$.
Remark: If TS denotes a transition system with terminal states, we define

$$Traces(TS) := \{trace(\pi) \mid \pi \in Paths(TS)\}$$