MVS – Esercitazione 1 del 6 aprile 2011

Esercizio 1

Consider the following mutual exclusion algorithm that uses the shared variables y_1 and y_2 (which are initially both 0):

Process P_1:	Process P_2:
while true do	**while true do**
... non-critical section ...	... non-critical section ...
$y_1 := y_2 + 1$	$y_2 := y_1 + 1$
wait until $(y_2 = 0) \vee (y_1 \leq y_2)$	**wait until** $(y_1 = 0) \vee (y_2 < y_1)$
... critical section ...	... critical section ...
$y_1 := 0$	$y_2 := 0$
... non-critical section ...	... non-critical section ...
od	**od**

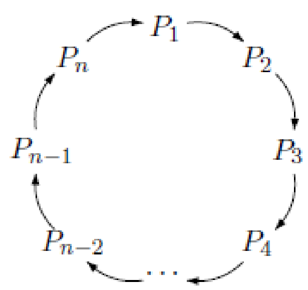
Questions:

a) Give the program graph representation of both processes.
(A pictorial representation suffices)

b) Give the reachable part of the transition system of $P_1 ||| P_2$ where $y_1 \leq 2$ and $y_2 \leq 2$.

Esercizio 2

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:



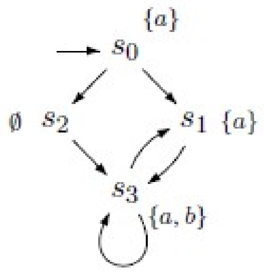

```
send (id);                          initially set to process' id
while (true) do
    receive (m);
    if (m == id) then stop;         process is the leader
    if (m > id) then send (m);      forward identifier
od
```

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 $id_i = i$.

Esercizio 3

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



Esercizio 4

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

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

Esercizio 5

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 $\mathbb{N}$. 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

Determine which of these LT properties are safety properties.

Esercizio 6

Let P and P' be liveness properties over AP. Prove or disprove the following claims:

a) $P \cup P'$ is a liveness property and

b) $P \cap P'$ is a liveness property.

Answer the same question for safety properties.