Final Exam, Reactive Systems 2002

Saturday, March 23, 09.00 - 14.00. Post Scriptum..

For assistance, you may bring and consult any notes, course material, books, etc. that you like. In Swedish:

Alla hjälpmedel tillåtna

This exam consists of 4 problems. The problems are formulated in English. Start solution of each problem on a new sheet of paper. Write your name on each sheet. You may write in English or in Swedish, whichever you prefer.

Before starting to work on any problem, read first through the entire exam.

Problem 1. (8 p) Consider the following action system.

$$
\begin{array}{l}
\textbf{ActionSystem } \textit{Program 1} \\
\quad \textbf{variables} \\
\qquad x\ , y\ :\ \text{integer} \\
\quad \textbf{initially} \\
\qquad x = y = 0 \\
\quad \textbf{actions} \\
\qquad x \quad := \quad x+1 \\
\quad [\!] \\
\qquad x > y \quad \longrightarrow \quad x \quad := \quad x-1 \\
\quad [\!] \\
\qquad y < x \quad \longrightarrow \quad y \quad := \quad y+1 \\
\textbf{end}
\end{array}
$$

Assume that there are three weak fairness constraints: one for each of the three actions. Which of the following properties hold for the program:

a) $\Diamond\ (y \geq 4)$

b) $\Box\ (x \geq y)$

c) $\Box\ ((y = 7 \implies \Diamond\ (x = 7 \lor y = 8))$

d) $\Box \Diamond\ (x = y)$

e) $\Box \Diamond\ (x > y)$

Give *short* motivations for your answers.

Problem 2. (8 p) Mr. Toggle has constructed a Promela model with two boolean variables, p and q. The purpose of the program is that

> when p is true then q should change its value in the next transition;
> when p if false, then q should remain unchanged;
> furthermore, q should change its value infinitely many times.

Your problems are

a) State the above correctness property in temporal logic.

b) Construct a never-claim which is added to the Promela model, so that SPIN can check that the program satisfies this correctness property.

Problem 3. (14 p) The following action system models an algorithm for guaranteeing mutual exclusion among any number (N) of processes. Each processes has an index, ranging from 1 to N. Mutual exclusion is administered by an arbiter. The processes and arbiter communicate via two shared integer variables x and y. Each process has a control state, the value of which is either *idle* or *crit*. We represent the control states of all processes by an array PC with indices ranging from 1 to N, where $PC[i]$ is the control state of process i. The arbiter has two control states: *free* and *granted*, representing the state of the resource. The control state of the arbiter is represented in the variable AC. Here is the algorithm

ActionSystem *Mutex*
variables

$$
\begin{array}{lcl}
x\ , y & : & \text{integer} \\
PC & : & \text{array } [1..N] \text{ of } \{idle, crit\} \\
AC & : & \{\mathit{free}, granted\}
\end{array}
$$

initially

$$x = y = 0 \ \wedge\ AC = \mathit{free} \ \wedge\ \forall i : 1 \le i \le N : PC[i] = idle$$

actions

$$
\begin{array}{l}
AC = \mathit{free} \wedge y \neq 0 \ \longrightarrow\ AC, x \ := \ granted, y \\
[\!] \\
AC = granted \wedge x = 0 \ \longrightarrow\ AC, y \ := \ \mathit{free}, 0 \\
[\!] \\
\langle [\!]\, i \ :\ 1 \le i \le N \ ::\ PC[i] = idle \ \longrightarrow\ y \ := \ i \rangle \\
[\!] \\
\langle [\!]\, i \ :\ 1 \le i \le N \ ::\ PC[i] = idle \wedge x = i \ \longrightarrow\ PC[i] \ := \ crit \rangle \\
[\!] \\
\langle [\!]\, i \ :\ 1 \le i \le N \ ::\ PC[i] = crit \ \longrightarrow\ PC[i], x \ := \ idle, 0 \rangle
\end{array}
$$

end

Your problem is

a) To formally state the property of mutual exclusion for this algorithm.

b) To prove this property, using the techniques given in the course.

and to answer the following quesitons

c) Let us assume weak fairness for each action, where by action we mean either the first action, the second action, or one instance (given by a specific value of N) of one of the last three actions. Thus, in total, there are $3N + 2$ fairness sets. Does the algorithm guarantee absense of starvation for each process (i.e., that each process will eventually enter the critical section)?

d) The same question as c), with strong fairness instead of weak fairness.

Problem 4. (6 p) Mr. Promel tried to model the algorithm in the previous problem in Promela, for the case that $N = 2$, as follows:

```
short x,y;

proctype Arbiter()
{
free:    if
         :: (y == 1) -> x = 1
         :: (y == 2) -> skip
         fi;
         if
         :: (y == 2) -> x = 2
         :: (y != 2) -> skip
         fi;
granted: (x == 0) -> y = 0; goto free
}

proctype Process (short i)
{
idle:    do
         :: y = i
         :: x == i -> break
         od;
crit:    x = 0; goto idle
}

init
{
        x = 0;y=0;
        atomic {
                run Arbiter();
                run Process(1);
                run Process(2);
        }
}
```

However, the algorithm does not work satisfactorily. Explain what is the problem, and how it should be fixed.

Problem 5. (4 p) Suppose that p_1, p_2, q_1 and q_2 are state formulas (or boolean variables). Assume that a system satisfies: $p_1 \mathcal{W} q_1$ and $p_2 \mathcal{W} p_2$. Which of the following can be inferred from this information?

a) the system satisfies $(p_1 \wedge p_2) \mathcal{W} (q_1 \wedge q_2)$

b) the system satisfies $(p_1 \vee p_2) \mathcal{W} (q_1 \vee q_2)$