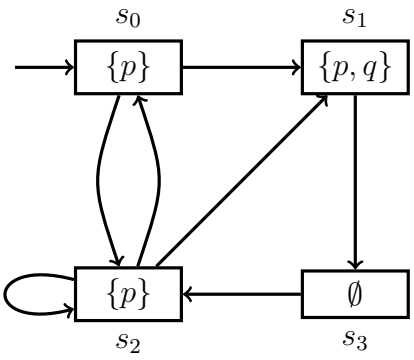
Technische Universität München (I7) **Summer Semester 2020**
J. Křetínský / P. Ashok / K. Grover / T. Meggendorfer **25.06.2019**

Model Checking – Exercise sheet 8

Exercise 8.1

Create a `NuSMV` model for the following Kripke structure over $AP = \{p, q\}$:



Use `NuSMV` to model check each of the following formulas. Explain in word if the formula holds, or give a counterexample otherwise.

(a) $\mathbf{EG}\ p$,

(b) $\mathbf{AG}(p \rightarrow \mathbf{AX}\ p)$,

(c) $\mathbf{EX}(\neg q \wedge (\neg p\ \mathbf{EU}\ q))$.

Exercise 8.2

Recall the "Dining Philosophers" example from the lectures. Assume that there are 5 philosopher $\{0, 1, 2, 3, 4\}$ sitting at a dining table and there are five forks $\{0, 1, 2, 3, 4\}$. For a philosopher to be able to eat, he/she needs both left and right forks. Consider the following pseudo code which describes the behavior of the ith philosopher:

while true **do**:
q_0: wait($fork[i] = 10$);
q_1: $fork[i] := i$;
q_2: wait($fork[i + 1] = 10$); // it is $(i + 1 \mod 5)$ actually
q_3: $fork[i + 1] := i$;
q_4: Eat(i);
q_5: $fork[i + 1] := 10$;
q_6: $fork[i] := 10$;
od

Model this in `NuSMV` and check:

(a) if the *ith* philosopher can eat eventually.

(b) if it satisfies that no philosopher is starved i.e 'all of them get to eat infinitely often'. If it doesn't then what is the reason and how can you fix it.

Exercise 8.3

(Taken from *Principles of Model Checking*)
In the original mutual exclusion protocol by Dijkstra in 1965 (another Dutch mathematician), it is assumed that there are $n \geq 2$ processes, and global variables b, c : array $[1...n]$ of Boolean and an integer k. Initially all elements of b and of c have the value true and the value of k belongs to $1, 2, \ldots, n$. The ith process may be represented as follows:

var j : integer;
while true **do**
begin $b[i]$:= false;
    L_i : **if** $k \neq i$ **then**
        begin $c[i]$:= true;
            if $b[k]$ **then** $k := i$;
            goto L_i
        end;
        else begin c[i] := false;
            for $j = 1$ to n **do**
                if $(j \neq i \wedge \neg(c[j]))$ **then goto** L_i
            end
            $\langle$ critical section $\rangle$;
    $c[i]$:= true;
    $b[i]$:= true;
end

Questions:

(a) Model this algorithm in `NuSMV`.

(b) Check the mutual exclusion property (at most one process can be in its critical section at any point in time) by specifying the property in CTL. Try to check this property for $n = 2$ through $n = 5$ by increasing the number of processes gradually.

(c) Check the individual starvation property: if a process wants to enter its critical section, it is eventually able to do so.