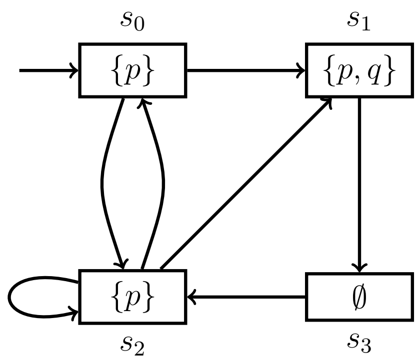
Technische Universität München (I7) **Summer Semester 2018**
J. Esparza / M. Blondin / P. Ashok **28.06.2018**

Model Checking – Exercise sheet 10

Exercise 10.1
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) $\mathbf{AX}\ \mathbf{AF}\ \mathbf{EG}\ p$,

(c) $p\ \mathbf{AU}\ q$,

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

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

Exercise 10.2
Model the following stack system in `NuSMV`:

> The stack system consists of three input interfaces: `push`, `pop`, `in_val`; and one output interface: `out_val`. The values of `push` and `pop` can be either `true` or `false`, while `in_val` and `out_val` can take any number between 0 and 9.
>
> When `push` is `true`, the system takes the input from `in_val` and pushes it onto its internal stack. When `pop` is `true`, the system removes the value on the top of the stack and outputs it via `out_val`. It is forbidden to call `push` and `pop` at the same time. The size of the stack is 5, i.e. the stack is full if there are 5 pushes without a pop. When the stack is full, it ignores `push` and `in_val`. Similarly, the system ignores `pop` when the stack is empty. The value of `out_val` is undefined if the stack is empty or `pop` is `false`.

Write the following properties in CTL and use `NuSMV` to model check the formulas:

(a) The stack cannot be empty and full at the same time.

(b) There exists a path along which the stack is eventually always full.

(c) From any given point of time, there always exists a path in which the stack will be full.

(d) The stack cannot be empty after a push.

(e) The internal stack is correctly updated after a push or pop.

(f) Whenever the stack is full, there exists a path in which the stack stays full forever or it remains full until a pop.

(g) For every push, there exists a path that pops the value without pushing another value.

(h) After every pop, `out_val` holds the correct value.

Exercise 10.3

Let $\mathcal{K} = (S, \rightarrow, r, AP, \nu)$ be a Kripke structure. For every $X \subseteq S$, $i \in \mathbb{N}$ and CTL formulas φ and ψ, let

$$\begin{aligned} \xi^0_{\varphi,\psi}(X) &= X, \\ \xi^{i+1}_{\varphi,\psi}(X) &= [\![\psi]\!] \cup \left([\![\varphi]\!] \cap \mathrm{pre}(\xi^i_{\varphi,\psi}(X))\right). \end{aligned}$$

(a) Show that if $[\![\varphi]\!] \subseteq [\![\varphi']\!]$, $[\![\psi]\!] \subseteq [\![\psi']\!]$ and $X \subseteq X'$, then $\xi^i_{\varphi,\psi}(X) \subseteq \xi^i_{\varphi',\psi'}(X')$ for every $i \in \mathbb{N}$.

(b) Show that if $(\varphi \Rightarrow \varphi') \land (\psi \Rightarrow \psi')$, then $(\varphi\ \mathbf{EU}\ \psi) \Rightarrow (\varphi'\ \mathbf{EU}\ \psi')$, $\mathbf{EF}\varphi \Rightarrow \mathbf{EF}\varphi'$ and $\mathbf{AG}\varphi \Rightarrow \mathbf{AG}\varphi'$.

Solution 10.1

```
MODULE main
VAR
  state : {s0, s1, s2, s3};
ASSIGN
  init(state) := s0;
  next(state) :=
    case
      state = s0 : {s1, s2};
      state = s1 : s3;
      state = s2 : {s0, s1, s2};
      state = s3 : s2;
    esac;
DEFINE
  p := state = s0 | state = s1 | state = s2;
  q := state = s1;

SPEC
  EG p
SPEC
  AX AF EG p
SPEC
  A [p U q]
SPEC
  AG (p -> AX p)
SPEC
  EX (!q & E [!p U q])
```

Solution 10.2

```
MODULE main
VAR
  op : 0..2;
  in_val  : 0..9;
  out_val : 0..9;
  ptr : 0..5;
  arr : array 0..4 of 0..9;

FROZENVAR
  i : 0..4;
  x : 0..9;

DEFINE
  empty := (ptr = 0);
  full  := (ptr = 5);
  push  := (op = 0);
```

```
  pop   := (op = 1);

ASSIGN
  init(ptr) := 0;
  next(ptr) := case
                 push & !full  : ptr + 1;
                 pop  & !empty : ptr - 1;
                 TRUE : ptr;
               esac;

  next(arr[0]) := push & ptr = 0 ? in_val : arr[0];
  next(arr[1]) := push & ptr = 1 ? in_val : arr[1];
  next(arr[2]) := push & ptr = 2 ? in_val : arr[2];
  next(arr[3]) := push & ptr = 3 ? in_val : arr[3];
  next(arr[4]) := push & ptr = 4 ? in_val : arr[4];

  next(out_val) := case
                     pop & !empty : arr[ptr - 1];
                     TRUE : out_val;
                   esac;

--  (a) The stack cannot be empty and full at the same time.
SPEC
  AG !(empty & full)

-- (b) There exists a path along which the stack is eventually always full.
SPEC
  EF EG full

-- (c) From any given point of time, there always exists a path in
-- which the stack will be full.
SPEC
  AG EF full

-- (d) The stack cannot be empty after a push.
SPEC
  AG (push -> AX !empty)

-- (e) The internal stack is correctly updated after a push or a pop.
SPEC
  AG ((push & !full & in_val = x & ptr = i) -> (AX (arr[i] = x)))

SPEC
  AG ((push & !full & ptr = i) -> (AX (ptr = i + 1)))
```

```
SPEC
  AG ((pop & !empty & ptr = i) -> (AX (ptr = i - 1)))

SPEC
  AG ((push & ptr >= 4) -> (AX full))

SPEC
  AG ((pop & ptr <= 1) -> (AX empty))

-- (f) Whenever the stack is full, there exists a path in which the
-- stack stays full forever or it remains full until a pop.
SPEC
  AG (full -> ((EG full) | E[full U pop]))

-- (g) For every push, there exists a path that pops the value without
-- pushing another value.
SPEC
  AG (push -> EX E[!push U pop])

-- (h) After every pop, out_val holds the correct value
SPEC
  AG ((pop & !empty & arr[ptr - 1] = x) -> (AX (out_val = x)))
```

Solution 10.3

(a) We prove the claim by induction on i. The validity of the base case follows from $\xi^0_{\varphi,\psi}(X) = X \subseteq X' = \xi^0_{\varphi',\psi'}(X')$. Assume the claim holds for $i > 0$. Let $x \in \xi^{i+1}_{\varphi,\psi}(X)$. By definition of ξ, we have

$$x \in [\![\psi]\!] \cup \left([\![\varphi]\!] \cap \mathrm{pre}(\xi^i_{\varphi,\psi}(X))\right).$$

If $x \in [\![\psi]\!]$, then $x \in [\![\psi']\!]$ and hence $x \in \xi^{i+1}_{\varphi',\psi'}(X')$ in which case we are done. Thus, assume $x \in [\![\varphi]\!] \cap \mathrm{pre}(\xi^i_{\varphi,\psi}(X))$. There exists $y \in \xi^i_{\varphi,\psi}(X)$ such that $x \to y$. By induction hypothesis, $\xi^i_{\varphi,\psi}(X) \subseteq \xi^i_{\varphi',\psi'}(X')$. Thus, $y \in \xi^i_{\varphi',\psi'}(X')$ and hence $x \in \mathrm{pre}(\xi^i_{\varphi',\psi'}(X'))$. Moreover, $x \in [\![\varphi]\!] \subseteq [\![\varphi']\!]$. Therefore, $x \in \xi^{i+1}_{\varphi',\psi'}(X')$. □

(b) If $(\varphi \Rightarrow \varphi') \land (\psi \Rightarrow \psi')$, then $[\![\varphi]\!] \subseteq [\![\varphi']\!]$ and $[\![\psi]\!] \subseteq [\![\psi]\!]$. As seen in class, there exist $i, j \in \mathbb{N}$ such that

$$[\![\varphi \ \mathbf{EU}\ \psi]\!] = \xi^\ell_{\varphi,\psi}(\emptyset) \qquad \text{for every } \ell \geq i,$$

$$[\![\varphi' \ \mathbf{EU}\ \psi']\!] = \xi^\ell_{\varphi',\psi'}(\emptyset) \qquad \text{for every } \ell \geq j.$$

Let $k = \max(i, j)$. We have:

$$\begin{aligned}
[\![\varphi \ \mathbf{EU} \ \psi]\!] &= \xi^k_{\varphi,\psi}(\emptyset) \\
&\subseteq \xi^k_{\varphi',\psi'}(\emptyset) && \text{(by (a))} \\
&= [\![\varphi' \ \mathbf{EU} \ \psi']\!].
\end{aligned}$$

This means that $(\varphi \ \mathbf{EU} \ \psi) \Rightarrow (\varphi' \ \mathbf{EU} \ \psi')$. By taking $\varphi = \varphi' = \texttt{true}$, it also follows that $\mathbf{EF}\varphi \Rightarrow \mathbf{EF}\varphi'$.

It remains to show that $\mathbf{AG}\varphi \Rightarrow \mathbf{AG}\varphi'$ holds. We have

$$\begin{aligned}
(\mathbf{AG}\varphi \Rightarrow \mathbf{AG}\varphi') &\equiv (\neg\mathbf{AG}\varphi \vee \mathbf{AG}\varphi') \\
&\equiv (\neg\neg\mathbf{EF}\neg\varphi \vee \neg\mathbf{EF}\neg\varphi') \\
&\equiv (\mathbf{EF}\neg\varphi \vee \neg\mathbf{EF}\neg\varphi') \\
&\equiv (\mathbf{EF}\neg\varphi' \Rightarrow \mathbf{EF}\neg\varphi).
\end{aligned}$$

Now, observe that $[\![\neg\varphi']\!] = \overline{[\![\varphi']\!]} \subseteq \overline{[\![\varphi]\!]} = [\![\neg\varphi]\!]$ since $[\![\varphi]\!] \subseteq [\![\varphi']\!]$. Thus, $\mathbf{EF}\neg\varphi' \Rightarrow \mathbf{EF}\neg\varphi$ holds which completes the proof. $\square$