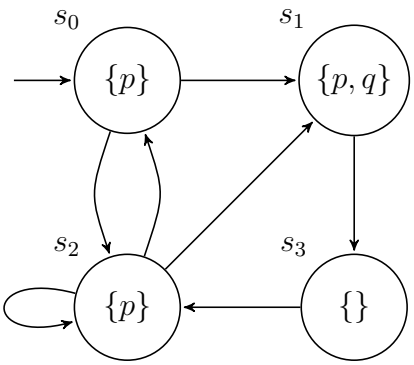
Technische Universität München (I7) **Summer Semester 2017**
J. Křetínský / T. Meggendorfer / P. Ashok **30.03.2017**

Model Checking – Exercise sheet 9

Exercise 9.1

Create a NuSMV model for the following Kripke structure:



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

1. **EG** p
2. **AX AF EG** p
3. p **AU** q
4. **AG** $(p \rightarrow \textbf{AX}\ p)$
5. **EX** $(\neg q \wedge (\neg p\ \textbf{EU}\ q))$

Exercise 9.2

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 top of the stack and outputs it via `out_val`. 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 formulae:

1. The stack cannot be empty and full at the same time.
2. There exists a path in which the stack is always full.
3. From any given point of time, there always exists a path in which the stack will be full.
4. The stack cannot be empty after a push.
5. The internal stack is correctly updated after a push.
6. Whenever the stack is full, there exists a path in which the stack stays full forever or it remains full until a pop.
7. For every push, there exists a path that pops the value without pushing another value.