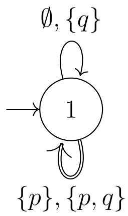
**Technische Universität München (I7)** **Summer Semester 2020**
**J. Křetínský / P. Ashok / K. Grover / T. Meggendorfer** **28.05.2020**

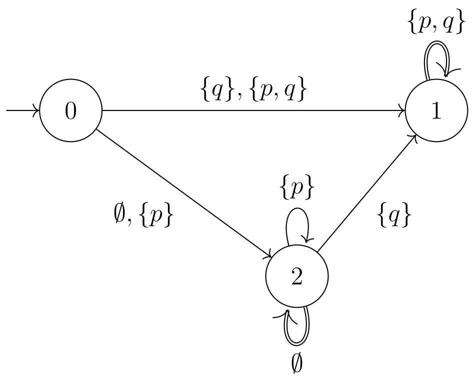
# Model Checking – Exercise sheet 5

**Exercise 5.1**
Convert the following Büchi automata with transition-based acceptance condition ("doubled"-transitions have to be seen infinitely often) to equivalent Büchi automata with state-based acceptance conditions. Moreover, give a general procedure to perform this conversion.

(a)




(b)




**Exercise 5.2**
Extend the set of rules of the LTL to Büchi automata translation to directly deal with the **F** and **G** operators. Use it to construct a Büchi automaton for $\phi = \mathbf{GF}p$. Is it necessary to construct states which do not contain $\phi$?

**Exercise 5.3**
Let $\phi = \mathbf{G}\left((\mathbf{X}(p\ \mathbf{U}\ q)) \rightarrow ((\neg p \wedge \mathbf{F}q) \vee (q\ \mathbf{U}\ \mathbf{X}q))\right)$ and $\mathcal{G}$ be a generalized Büchi automaton translated from $\phi$ using the construction presented in the lecture and the extended set of rules defined in the previous exercise.

(a) Write down the set of subformulae $Sub(\phi)$.

(b) What is the size of $CS(\phi)$?

(c) How many sets of accepting states does $\mathcal{G}$ have?

(d) Is $\{\phi\}$ an accepting state of $\mathcal{G}$?

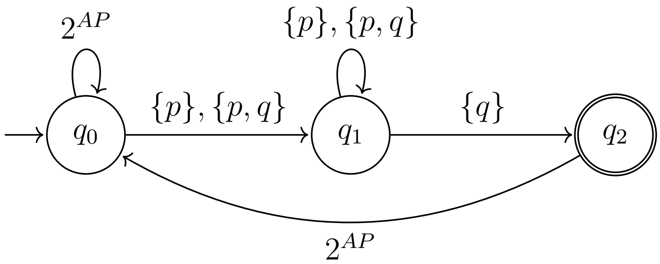
(e) Give a reachable state that has no successors.

(f) Give a successor state of the smallest consistent state containing $\{\phi, q, q\ \mathbf{U}\ \mathbf{X}q, \mathbf{F}q\}$.

(g) Give a predecessor state of the smallest consistent state containing $\{\phi, q, q\ \mathbf{U}\ \mathbf{X}q, \mathbf{F}q\}$.

**Exercise 5.4**

Consider the following Büchi automaton $\mathcal{B}$:




(a) Give an LTL formula $\phi$ such that $\mathcal{L}(\mathcal{B}) = [\![\phi]\!]$.

(b) By using the result from (a), propose a method to construct a Büchi automaton that accepts the complement of the language accepted by $\mathcal{B}$.

(c) Construct a Büchi automaton for the formula $\mathbf{G}(\neg p \vee (\neg p\ \mathbf{R}\ (p \vee \neg q)))$.

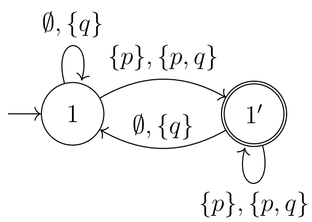
(d) By using the result from (c), construct a Büchi automaton that accepts the complement of the language accepted by $\mathcal{B}$.

**Solution 5.1**

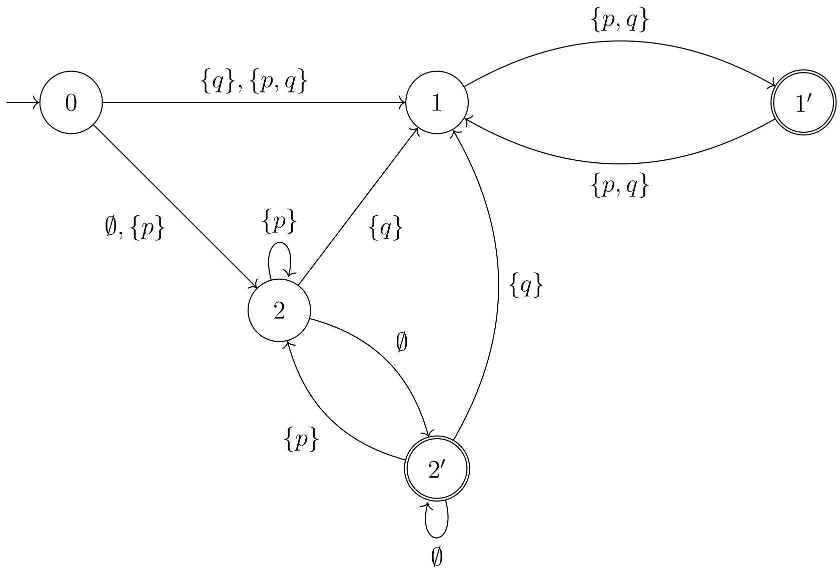
The general procedure is as follows.

Let the states of the original automaton be relabeled to $S \times \{1\}$ and create a copy of the states labeled by $S \times \{2\}$. For every accepting transition from $(s_1, 1) \to (s_2, 1)$, change the destination to $(s_2, 2)$. For every non-accepting transition from $(s_1, 2) \to (s_2, 2)$, change the destination to $(s_2, 1)$.

(a)




(b)




**Solution 5.2**

Extend the definition of NNF to include **F** and **G**, extend the corresponding $Sub(\phi)$:

- if $\mathbf{F}\phi_1 \in Sub(\phi)$ then $\phi_1 \in Sub(\phi)$
- if $\mathbf{G}\phi_1 \in Sub(\phi)$ then $\phi_1 \in Sub(\phi)$,

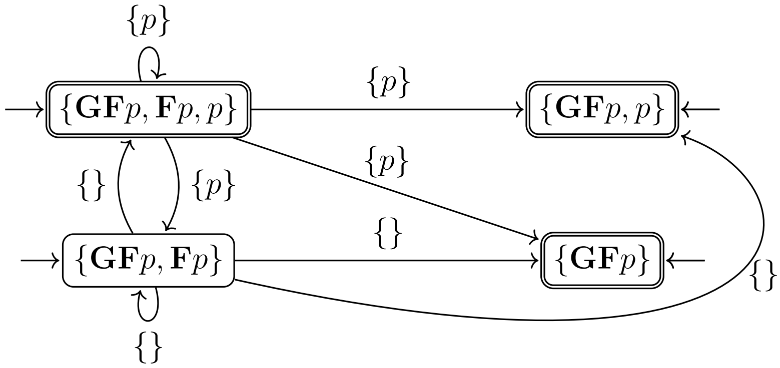
and extend rules for transitions as follows: $(M, \sigma, M') \in \Delta$ iff $\sigma = M \cap AP$ and

- if $\mathbf{F}\phi_1 \in Sub(\phi)$, then $\mathbf{F}\phi_1 \in M$ iff $\phi_1 \in M$ or $\mathbf{F}\phi_1 \in M'$
- if $\mathbf{G}\phi_1 \in Sub(\phi)$, then $\mathbf{G}\phi_1 \in M$ iff $\phi_1 \in M$ and $\mathbf{G}\phi_1 \in M'$

Also, the acceptance condition must be extended for $\mathbf{F}$: $\mathcal{F}$ contains a set $F_\psi$, for every subformula $\psi$ of the form $\mathbf{F}\phi_1$, where

$$F_\psi = \{M \in CS(\phi) \mid \phi_1 \in M \text{ or } \neg(\mathbf{F}\phi_1) \in M\}$$

The translated Büchi automaton for $\phi = \mathbf{GF}p$ is below. Notice that the initial states must contain $\mathbf{GF}p$, and from the translation rule successors of states with $\mathbf{GF}p$ must also contain $\mathbf{GF}p$. So, it is not necessary to construct states without $\mathbf{GF}p$.




**Solution 5.3**

Translate $\phi$ into an NNF formula:

$$\begin{aligned}\phi &= \mathbf{G}\left((\mathbf{X}(p \ \mathbf{U}\ q)) \rightarrow ((\neg p \wedge \mathbf{F}q) \vee (q \ \mathbf{U}\ \mathbf{X}q))\right) \\ &\equiv \mathbf{G}\left((\mathbf{X}(\neg p \ \mathbf{R}\ \neg q)) \vee ((\neg p \wedge \mathbf{F}q) \vee (q \ \mathbf{U}\ \mathbf{X}q))\right)\end{aligned}$$

(a) Let $\phi_1 = \neg p \ \mathbf{R}\ \neg q$, $\phi_2 = \neg p \wedge \mathbf{F}q$, and $\phi_3 = q \ \mathbf{U}\ \mathbf{X}q$. We have $\phi = \mathbf{G}(\mathbf{X}\phi_1 \vee (\phi_2 \vee \phi_3))$ and $Sub(\phi) = \{\mathbf{true}, \phi, \mathbf{X}\phi_1 \vee (\phi_2 \vee \phi_3), \mathbf{X}\phi_1, \phi_2 \vee \phi_3, \phi_1, \phi_2, \phi_3, \mathbf{F}q, \mathbf{X}q, p, q\} \cup \{\mathbf{false}, \neg\phi, \neg(\mathbf{X}\phi_1 \vee (\phi_2 \vee \phi_3)), \neg\mathbf{X}\phi_1, \neg(\phi_2 \vee \phi_3), \neg\phi_1, \neg\phi_2, \neg\phi_3, \neg\mathbf{F}q, \neg\mathbf{X}q, \neg p, \neg q\}$.

(b) Only $\phi, \mathbf{X}\phi_1, \phi_1, \phi_3, \mathbf{F}q, \mathbf{X}q, p, q$ can independently form consistent states. So, $|CS(\phi)| = 2^8 = 256$ states

(c) $\mathcal{F} = \{F_{q\mathbf{U}\mathbf{X}q}, F_{\mathbf{F}q}\}$

(d) $\{\phi\} \in F_{q\mathbf{U}\mathbf{X}q}$ and $\{\phi\} \in F_{\mathbf{F}q}$

(e) $\{\phi\}$ is reachable because it is an initial state, and it has no successors because $\mathbf{X}\phi_1 \vee (\phi_2 \vee \phi_3) \notin \{\phi\}$.

(f) $\{\phi, q \ \mathbf{U}\ \mathbf{X}q\}$

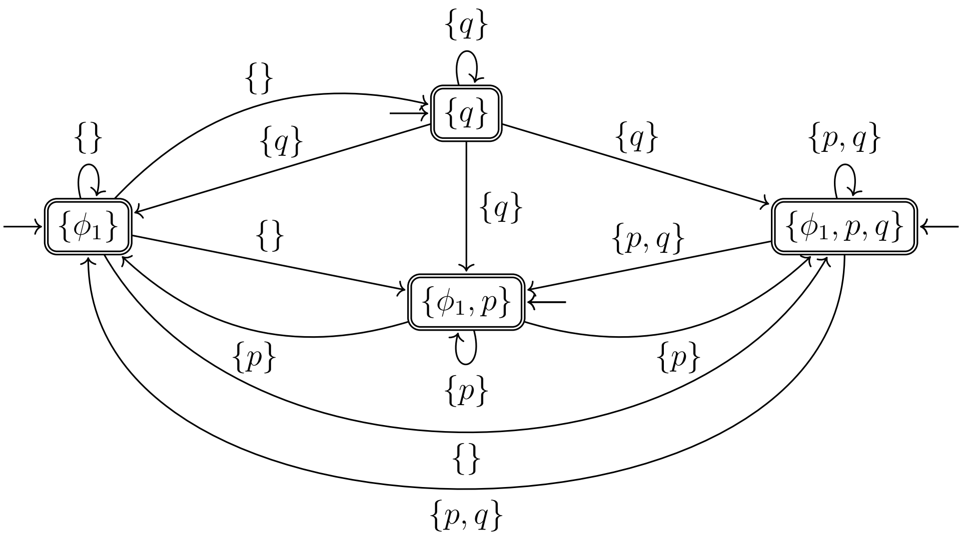
(g) $\{\phi, q, q\ \mathbf{U}\ \mathbf{X}q, \mathbf{F}q, \mathbf{X}q\}$

**Solution 5.4**

(a) $\phi = \mathbf{GF}(p \wedge (p\ \mathbf{U}\ (\neg p \wedge q)))$

(b) Construct a Büchi automaton for $\neg\phi$ by using e.g. the translation in the lecture.

(c) Let $\phi_1 = \neg p\ \mathbf{R}\ (p \vee \neg q)$. Note that $\phi_1, p, q$ are enough to form consistent sets, i.e. we assume that $\phi$ and $\neg p \vee (\neg p\ \mathbf{R}\ (p \vee \neg q))$ are implicitly in every state. So, $CS(\phi) = 2^{\{\phi_1, p, q\}}$. However, we know that $\{p\}$ and $\{p, q\}$ have no successors because of $\mathbf{G}$, and $\{\}$ and $\{\phi_1, q\}$ have no successors because of $\mathbf{R}$.




(d) Notice that $\neg\phi \equiv \mathbf{FG}(\neg p \vee (\neg p\ \mathbf{R}\ (p \vee \neg q)))$. It suffices to add a self-looping initial state and transitions from it to all states in (c).