Technische Universität München **Summer Semester 2015**
I7
J. Esparza / A. Durand-Gasselin / D. Suwimonteerabuth **June 2015**

Model Checking – Solution sheet 5

Exercise 5.1: Solution

1. $\mathbf{F}\neg p$ is $\top \mathcal{U} p$ which is in NNF. $\mathbf{G}\varphi$ is $\bot \mathcal{R} \varphi$ which is also in NNF.
2. It matters little that φ is in NNF, since $\mathbf{G}\neg\varphi$ is to be rewritten as $\bot \mathcal{R} \neg\varphi$, one needs to compute a formula ψ in NNF, equivalent to $\neg\varphi$. Then $\bot \mathcal{R} \psi$ is equivalent to $\mathbf{G}\neg\varphi$ and is in NNF.
3. The extended construction has the same set of states (i.e. sets of subformulas of the formula in NNF) except this formula may also contain $\mathbf{F}$ and $\mathbf{G}$.

   We recall the transition relation was defined as follows: $(M, \sigma, M') \in \Delta$ iff $\sigma = M \cap AP$ and:

   - if $\mathbf{X}\varphi_1 \in Sub(\varphi)$ then $\mathbf{X}\varphi_1 \in M$ iff $\varphi_1 \in M'$;
   - if $\varphi_1 \mathcal{U} \varphi_2 \in Sub(\varphi)$ then
     $\varphi_1 \mathcal{U} \varphi 2 \in M$ iff $\varphi_2 \in M$ or ($\varphi_1 \in M$ and $\varphi_1 \mathcal{U} \varphi_2 \in M'$);
   - if $\varphi_1 \mathcal{R} \varphi_2 \in Sub(\varphi)$ then
     $\varphi_1 \mathcal{R} \varphi_2 \in M$ iff ($\varphi_1 \in M$ and $\varphi_2 \in M$) or ($\varphi_2 \in M$ and $\varphi_1 \mathcal{R} \varphi_2 \in M'$).

   The acceptance condition was defined as follows: F contains a set F_ψ for every subformula ψ of the form $\varphi_1 \mathcal{U} \varphi_2$, where $F_\psi = \{M \in CS(\varphi) \mid \varphi_2 \in M \text{ or } \neg(\varphi_1 \mathcal{U} \varphi_2) \in M\}$.

   We further **restrict** the transition function: $(M, \sigma, M') \in \Delta$ if furthermore:

   - if $\mathbf{F}\varphi_1 \in Sub(\varphi)$ then $\mathbf{F}\varphi_1 \in M$ iff $\varphi_1 \in M$ or $\mathbf{F}\varphi_1 \in M'$;
   - if $\mathbf{G}\varphi_1 \in Sub(\varphi)$ then $\mathbf{G}\varphi_1 \in M$ iff $\varphi_1 \in M$ and $\mathbf{G}\varphi_1 \in M'$.

   F also contains a set F_ψ for every subformula ψ of the form $\mathbf{F}\varphi_1$, where $F_\psi = \{M \in CS(\varphi) \mid \varphi_1 \in M \text{ or } \neg(\mathbf{F}\varphi_1) \in M\}$.
4. We prove this claim by showing the two automata accept the same language. Not building states that do not contain ψ provides an automaton whose language is included in the GBA of φ, indeed any accepting run in the former automaton is also an accepting run in the latter.

   Let us show that any accepting run in the automaton for φ does not visit any state not containing ψ. We prove this claim in two steps: first we show that every reachable state contains $\mathbf{G}\psi$, then we show that any state containing $\mathbf{G}\psi$ has a successor only if it also contains ψ. Finally we conclude that no accepting run contains a state not containing $\mathbf{G}\psi$.

The first claim derives from the construction of the GBA. The second claim derives from the first claim and the construction of the GBA.

Not building any state that do not contain ψ yields a GBA accepting the same language: it is therefore not necessary to build them.

Exercise 5.2: LTL to Büchi translation

We consider the following LTL formula:
$\varphi = \mathbf{G}((\mathbf{X}(p \,\mathcal{U}\, q)) \rightarrow ((\neg p \wedge \mathbf{F}q) \vee (q \,\mathcal{U}\, \mathbf{X}q)))$

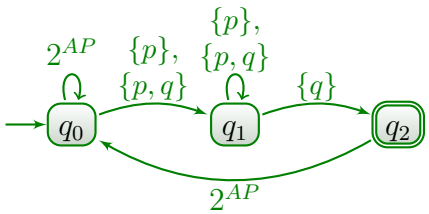
1. First we put φ in NNF: $\varphi = \mathbf{G}((\mathbf{X}(\neg p \,\mathcal{R}\, \neg q)) \vee (\neg p \wedge \mathbf{F}q) \vee (q \,\mathcal{U}\, \mathbf{X}q))$ thus subformulas are: $p, q, \neg p, \neg q, \mathbf{G}((\mathbf{X}(\neg p \,\mathcal{R}\, \neg q)) \vee (\neg p \wedge \mathbf{F}q) \vee (q \,\mathcal{U}\, \mathbf{X}q))$ $(\mathbf{X}(\neg p \,\mathcal{R}\, \neg q)) \vee (\neg p \wedge \mathbf{F}q) \vee (q \,\mathcal{U}\, \mathbf{X}q), (\neg p \wedge \mathbf{F}q) \vee (q \,\mathcal{U}\, \mathbf{X}q)$, $\mathbf{X}(\neg p \,\mathcal{R}\, \neg q)$, $\neg p \wedge \mathbf{F}q$, $(q \,\mathcal{U}\, \mathbf{X}q)$, $\neg p \,\mathcal{R}\, \neg q$, $\mathbf{X}q$.
2. Only 8 subformulas are booleanly independent: φ, $\mathbf{X}(\neg p \,\mathcal{R}\, \neg q)$, $\neg p \wedge \mathbf{F}q$, $(q \,\mathcal{U}\, \mathbf{X}q)$, $\neg p \,\mathcal{R}\, \neg q$, Xq, p, q, therefore there are $2^8 = 256$ states
3. $F = \{F_{\mathbf{F}q}, F_{q \,\mathcal{U}\, \mathbf{X}q}\}$
4. Is $\{\varphi\} \in F_{\mathbf{F}q}$ and $\{\varphi\} \in \{F_{q \,\mathcal{U}\, \mathbf{X}q}\}$. Any run visiting $\{\varphi\}$ infinitely often is thus accepting. $\{\varphi\}$ may therefore (abusively) be called accepting.
5. $\{\varphi\}$ is reachable since it is an initial state, and has no successor.
6. First of all the questions should have read give a successor (resp. predecessor) state of the smallest consistent state containing $\{\varphi, p, q, q \,\mathcal{U}\, \mathbf{X}q, \mathbf{F}q\}$

   Typically the smallest consistent state containing $\{\varphi, p, q \,\mathcal{U}\, \mathbf{X}q, \mathbf{F}q\}$ is a successor.

   The smallest consistent state containing $\{\varphi, p, q, \mathbf{F}q\}$ is a predecessor.

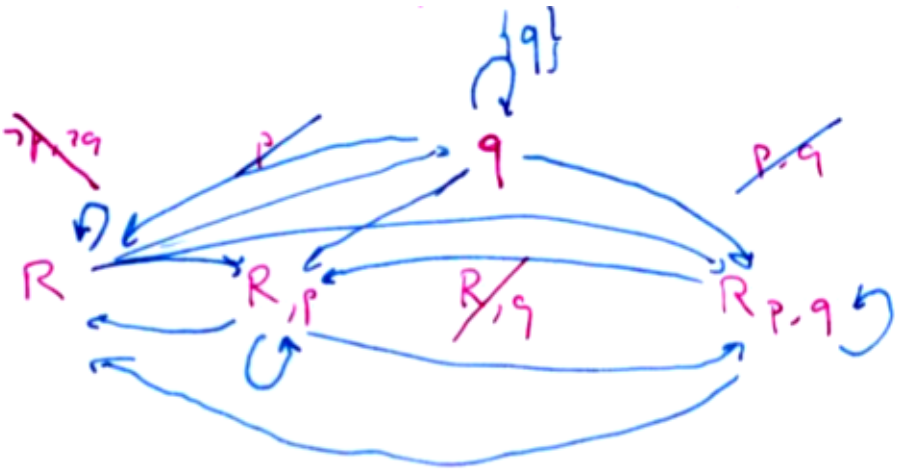
Exercise 5.3: From Büchi to LTL

We give the following automaton $\mathcal{B}$:



1. It should not be too hard to find a formula such as $\mathbf{GF}(p \wedge \mathbf{X}(p \,\mathcal{U}\, (q \wedge \neg p)))$. This formula is actually equivalent to $\psi = \mathbf{GF}(p \wedge (p \,\mathcal{U}\, (q \wedge \neg p)))$.
2. To build a Büchi automaton that accepting the complement of $L_{\mathcal{B}}$, one can simply build the LTL-to-Büchi translation of the LTL formula $\neg\psi$.
3. To build the Büchi automaton for the formula $\mathbf{G}(\neg p \vee (\neg p \,\mathcal{R}(p \vee \neg q)))$. we make 3 remarks: first that only formulas $p, q, \neg p \,\mathcal{R}(p \vee \neg q)$ suffice to form a boolean basis. $\mathbf{G}(\neg p \vee (\neg p \,\mathcal{R}(p \vee \neg q)))$ should be in each state, and the

valuation of the other subformulas can be deduced from these 3 formulas. The 5 states are initial.



4. Remark that $\neg\psi = \mathbf{FG}(\neg p \vee (\neg p \,\mathcal{R}(p \vee \neg q)))$. It suffices to add a self looping initial state that may also to any initial state of the previous automaton.