# Introduction to Model Checking
# Exercises WS 2011/2012

René Thiemann

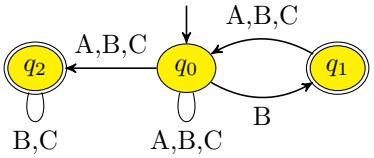
Institute of Computer Science

December 9, 2011

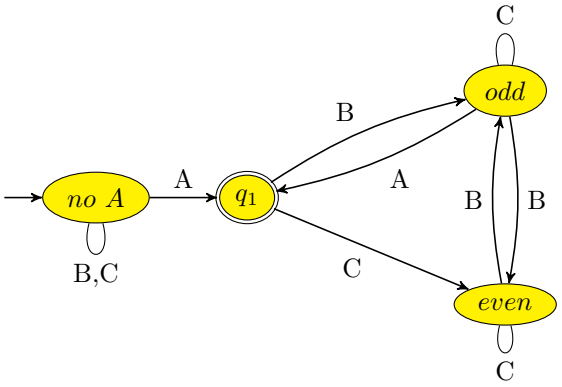
1. Consider the requirements 1, 2, and 4 on Slide 27, Chapter 2. For each of these requirements construct a GNBA which accepts the $\omega$-words that violate the requirement.

2. Let $\Sigma = \{A, B, C\}$. Construct GNBAs for the languages:

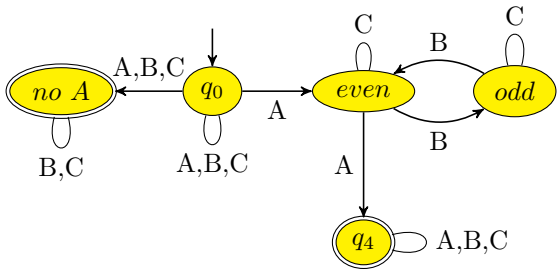
   - $L_1 = \{w \mid w$ contains only finitely many $A$'s or infinitely many $B$'s$\}$




   - $L_2 = \{w \mid w$ contains infinitely many $A$'s and between every two $A$'s there is an odd number of $B$'s$\}$




   - $L_3 = \Sigma^\omega \setminus L_2$




3. Show that GNBAs are closed under union, i.e., given two GNBAs $\mathcal{A}_1$ and $\mathcal{A}_2$ over the same alphabet $\Sigma$, construct a new GNBA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$. You only have to give a precise (formal) construction, a correctness proof is not required.

   Let $\mathcal{A}_i = (\mathcal{Q}_i, \Sigma, \delta_i, q_{0,i}, F_{1,i}, \ldots, F_{k_i,i})$ for $1 \leqslant i \leqslant 2$ and where w.l.o.g. $\mathcal{Q}_1 \cap \mathcal{Q}_2 = \varnothing$, $q_0 \notin \mathcal{Q}_1 \uplus \mathcal{Q}_2$, and $k_1 \leqslant k_2$.

   Then we define $\mathcal{A}$ as $(\mathcal{Q}_1 \uplus \mathcal{Q}_2 \uplus \{q_0\}, \Sigma, q_0, \delta, F_{1,1} \uplus F_{1,2}, \ldots, F_{k_1,1} \uplus F_{k_1,2}, \mathcal{Q}_1 \uplus F_{k_1+1,2}, \ldots, \mathcal{Q}_1 \uplus F_{k_2,2})$ where

   - $\delta(q, A) = \delta_1(q, A)$ for all $q \in \mathcal{Q}_1$
   - $\delta(q, A) = \delta_2(q, A)$ for all $q \in \mathcal{Q}_2$
   - $\delta(q_0, A) = \delta_1(q_{0,1}, A) \cup \delta_2(q_{0,2}, A)$

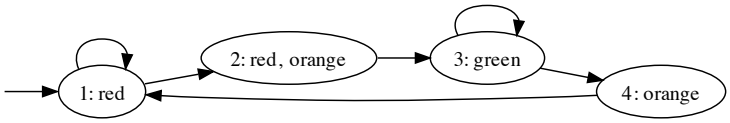
   Let $w \in \Sigma^\omega$.

   By construction obviously each infinite run $q_0\, q_1\, q_2 \ldots$ of $w$ on $\mathcal{A}$ corresponds to either the infinite run $q_{0,1}\, q_1\, q_2 \ldots$ of $\mathcal{A}_1$ where all $q_j \in \mathcal{Q}_1$ for $j > 0$ or it corresponds to the infinite run $q_{0,2}\, q_1\, q_2 \ldots$ of $\mathcal{A}_2$ where all $q_j \in \mathcal{Q}_2$ for $j > 0$. Moreover, if the run is accepting then in the first case $F_{1,1} \uplus F_{1,2}, \ldots, F_{k_1,1} \uplus F_{k_1,2}$ are all visited infinitely often. But since no state of $\mathcal{Q}_2$ appears in the run and since $F_{j,2} \subseteq \mathcal{Q}$ we know that each of the sets $F_{1,1}, \ldots, F_{k_1,1}$ is visited infinitely often. Thus, then the run is also an accepting run of $\mathcal{A}_1$. Reasoning in the same way shows that also in the second case we get an accepting run of $\mathcal{A}_2$ since each $F_{1,i}$ and $\mathcal{Q}_1$ are subsets of $\mathcal{Q}_1$.

   Similarly, each infinite run of $w$ on $\mathcal{A}_1$ or $\mathcal{A}_2$ has a corresponding infinite run in $\mathcal{A}$ where just the first state $q_{0,i}$ is replaced by $q_0$. Moreover, if we have an accepting run of $\mathcal{A}_1$ then $F_{1,1}, \ldots, F_{1,k_1}$ is visited infinitely often, and obviously, $\mathcal{Q}_1$ is visited infinitely often. Hence, also each set $F_{1,1} \uplus F_{1,2}, \ldots, F_{k_1,1} \uplus F_{k_1,2}, \mathcal{Q}_1 \uplus F_{k_1+1,2}, \ldots, \mathcal{Q}_1 \uplus F_{k_2,2})$ is visited infinitely often, i.e., we obtain an accepting run of $\mathcal{A}$. Similarly, an accepting run of $\mathcal{A}_2$ yields an accepting run of $\mathcal{A}$.

   In total we have shown that for an arbitrary word $w$, $w$ has an accepting run of $\mathcal{A}$ iff it has an accepting run of $\mathcal{A}_1$ or $\mathcal{A}_2$. Thus, $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$.

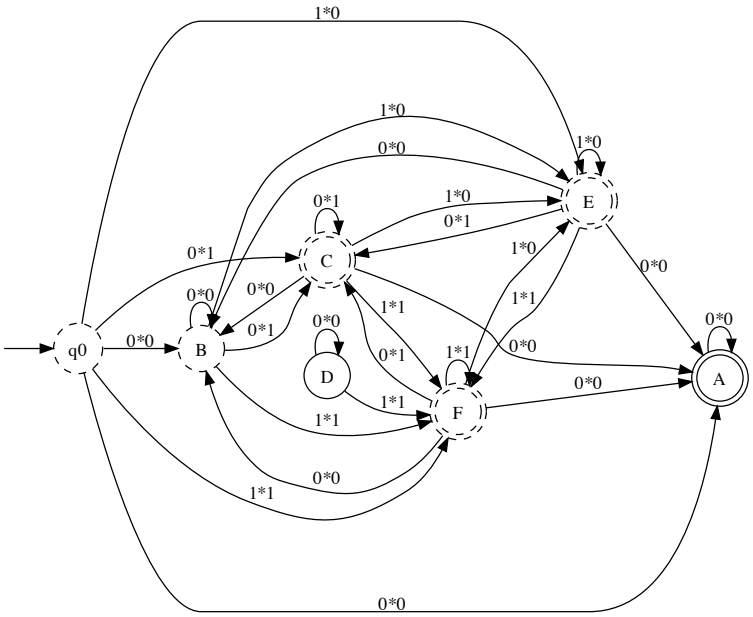
4. Consider the following transition system for a traffic light over $AP = \{\mathsf{red}, \mathsf{orange}, \mathsf{green}\}$:




(a) Consider the requirement "there is no direct switch from red to green". Does the traffic light system satisfy this requirement (informal answer suffices)? Do you see any problems with this requirement?

There is the problem of ambiguity what "red" in the requirement refers to: one can read it as "red and no other color" or "red and whatever other color", and similarly for "green". In the former case the property is satisfied, otherwise not.

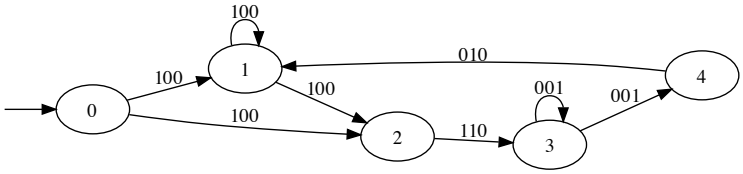
(b) For another requirement the following GNBA was automatically generated which specifies the set of forbidden traces.



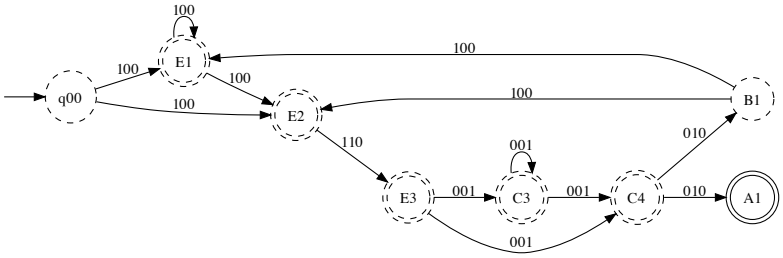

Here, the alphabet consists of vectors $(red, orange, green)$. Moreover, the double-circled states form one final state set ($\{A, C, E, F\}$)

and another final state set is given by the solid lined states ($\{A, D\}$).

- Decide whether the traffic light systems satisfies the requirements using the algorithms of Chapter 2. Here, for the intersection GNBA, only construct the reachable states (there are less than 10 reachable states)!
  Constructing the GNBA for the traffic light system yields




  Then the intersection with the given specification GNBA yields




  As the only reachable SCC has no solid state, the language of this GNBA is empty, and thus, the requirement is satisfied.
- Optional: Describe the requirement in words.
  It is easy to observe, that every accepting run must end in state $A$, and hence every word of the language must end in $(0 * 0)^\omega$. Moreover, one can show that indeed every word of this shape is accepted by the GNBA. Hence, the forbidden words are those that end in $(0 * 0)^\omega$. Hence, the allowed words are those where green or red occur infinitely often.

5. Formalize the following requirements in LTL over $AP = \{\mathsf{red}, \mathsf{orange}, \mathsf{green}\}$. (increasing difficulty)

   (a) The traffic light is never red and green at the same time.

   $$\mathsf{G}\,\neg(\mathsf{red} \wedge \mathsf{green})$$

   (b) Under the assumption that the traffic light is orange infinitely often, it is green infinitely often and red infinitely often.

   $$\mathsf{G}\,\mathsf{F}\,\mathsf{orange} \Rightarrow (\mathsf{G}\,\mathsf{F}\,\mathsf{green} \wedge \mathsf{G}\,\mathsf{F}\,\mathsf{red})$$

(c) The sequence of lights is exactly $(\mathsf{red}, \mathsf{red\,orange}, \mathsf{green}, \mathsf{orange})^\omega$.

$$\begin{aligned}\mathsf{red} \wedge \neg\mathsf{orange} \wedge \neg\mathsf{green} \wedge \mathsf{G}\,((\mathsf{red} \wedge \neg\mathsf{orange} \wedge \neg\mathsf{green}) \Rightarrow \mathsf{X}\,(\\ \mathsf{red} \wedge \mathsf{orange} \wedge \neg\mathsf{green} \wedge \mathsf{X}\,(\\ \neg\mathsf{red} \wedge \neg\mathsf{orange} \wedge \mathsf{green} \wedge \mathsf{X}\,(\\ \neg\mathsf{red} \wedge \mathsf{orange} \wedge \neg\mathsf{green} \wedge \mathsf{X}\,(\\ \mathsf{red} \wedge \neg\mathsf{orange} \wedge \neg\mathsf{green})))))\end{aligned}$$

(d) The sequence of lights is of the form $(\mathsf{red}^+\,\mathsf{green}^+\,\mathsf{orange}^+)^\omega$.

$$\begin{aligned}\mathsf{G}\,(\mathsf{red} \Rightarrow (\neg\mathsf{orange} \wedge \neg\mathsf{green})) \wedge\\ \mathsf{G}\,(\mathsf{orange} \Rightarrow (\neg\mathsf{red} \wedge \neg\mathsf{green})) \wedge\\ \mathsf{G}\,(\mathsf{green} \Rightarrow (\neg\mathsf{red} \wedge \neg\mathsf{orange})) \wedge\\ \mathsf{red} \wedge\\ \mathsf{G}\,(\mathsf{red} \Rightarrow \mathsf{X}\,(\mathsf{red} \vee \mathsf{green})) \wedge\\ \mathsf{G}\,(\mathsf{green} \Rightarrow \mathsf{X}\,(\mathsf{green} \vee \mathsf{orange})) \wedge\\ \mathsf{G}\,(\mathsf{orange} \Rightarrow \mathsf{X}\,(\mathsf{orange} \vee \mathsf{red})) \wedge\\ \mathsf{G\,F\,red} \wedge \mathsf{G\,F\,green} \wedge \mathsf{G\,F\,orange} \wedge\end{aligned}$$

(e) The sequence of lights is of the form $(\mathsf{red}^+\,\mathsf{orange}^+\,\mathsf{green}^+\,\mathsf{orange}^+)^\omega$.

$$\begin{aligned}&\mathsf{G}\,(\mathsf{red} \Rightarrow (\neg\mathsf{orange} \wedge \neg\mathsf{green})) \wedge\\ &\mathsf{G}\,(\mathsf{orange} \Rightarrow (\neg\mathsf{red} \wedge \neg\mathsf{green})) \wedge\\ &\mathsf{G}\,(\mathsf{green} \Rightarrow (\neg\mathsf{red} \wedge \neg\mathsf{orange})) \wedge\\ &\mathsf{G}\,(\mathsf{green} \vee \mathsf{red} \vee \mathsf{orange}) \wedge\\ &\mathsf{red} \wedge\\ &\mathsf{G}\,(\mathsf{red} \Rightarrow \mathsf{X}\,(\mathsf{red} \vee \mathsf{orange})) \wedge\\ &\mathsf{G}\,(\mathsf{green} \Rightarrow \mathsf{X}\,(\mathsf{green} \vee \mathsf{orange})) \wedge\\ &\mathsf{G\,F\,red} \wedge \mathsf{G\,F\,green} \wedge \mathsf{G\,F\,orange}\\ &\neg\mathsf{F}\,(\mathsf{red} \wedge \mathsf{X}\,(\mathsf{orange} \wedge \mathsf{orange\,U\,red})) \wedge \quad \text{no } ro^+r\\ &\neg\mathsf{F}\,(\mathsf{green} \wedge \mathsf{X}\,(\mathsf{orange} \wedge \mathsf{orange\,U\,green})) \quad \text{no } go^+g\end{aligned}$$

or alternatively,

$$\begin{aligned}\mathsf{G}\,(\mathsf{red} \Rightarrow (\neg\mathsf{orange} \wedge \neg\mathsf{green})) \wedge\\ \mathsf{G}\,(\mathsf{orange} \Rightarrow (\neg\mathsf{red} \wedge \neg\mathsf{green})) \wedge\\ \mathsf{G}\,(\mathsf{green} \Rightarrow (\neg\mathsf{red} \wedge \neg\mathsf{orange})) \wedge\\ \mathsf{red} \wedge\\ \mathsf{G}\,(\mathsf{red} \Rightarrow \mathsf{red\,U}\,(\mathsf{orange} \wedge \mathsf{orange\,U}\,(\mathsf{green} \wedge \mathsf{green\,U}\,(\mathsf{orange} \wedge \mathsf{orange\,U\,red}))))\end{aligned}$$

6. Prove the following equivalences

- $\mathsf{F}\,\mathsf{X}\,\varphi \quad \equiv \quad \mathsf{X}\,\mathsf{F}\,\varphi$

$$
\begin{array}{ll}
 & w \models \mathsf{F}\,\mathsf{X}\,\varphi \\
\text{iff} & \exists i \geqslant 0 : w[i..] \models \mathsf{X}\,\varphi \\
\text{iff} & \exists i \geqslant 0 : w[i..][1..] \models \varphi \\
\text{iff} & \exists i \geqslant 0 : w[1..][i..] \models \varphi \\
\text{iff} & w[1..] \models \mathsf{F}\,\varphi \\
\text{iff} & w \models \mathsf{X}\,\mathsf{F}\,\varphi
\end{array}
$$

- $\mathsf{F}\,\mathsf{G}\,\mathsf{F}\,\varphi \quad \equiv \quad \mathsf{G}\,\mathsf{F}\,\varphi$

$$
\begin{array}{ll}
 & w \models \mathsf{F}\,\mathsf{G}\,\mathsf{F}\,\varphi \\
\text{iff} & \exists i \geqslant 0 : w[i..] \models \mathsf{G}\,\mathsf{F}\,\varphi \\
\text{iff} & \exists i \geqslant 0 : \forall j \geqslant 0 : w[i..][j..] \models \mathsf{F}\,\varphi \\
\text{iff} & \exists i \geqslant 0 : \forall j \geqslant 0 : \exists k \geqslant 0 : w[i..][j..][k..] \models \varphi \\
\text{iff} & \exists i \geqslant 0 : \forall j \geqslant 0 : \exists k \geqslant 0 : w[i+j+k..] \models \varphi \\
\text{iff} & \forall j \geqslant 0 : \exists k' \geqslant 0 : w[j+k'..] \models \varphi \qquad \text{use } (k' := k+i) \text{ and } (k := k', i := 0) \\
\text{iff} & \forall j \geqslant 0 : \exists k' \geqslant 0 : w[j..][k'..] \models \varphi \\
\text{iff} & \forall j \geqslant 0 : w[j..] \models \mathsf{F}\,\varphi \\
\text{iff} & w \models \mathsf{G}\,\mathsf{F}\,\varphi
\end{array}
$$

7. Consider the LTL-formula

$$
\begin{array}{l}
\mathsf{G}\,(\mathsf{green} \Rightarrow (\neg\mathsf{orange} \wedge \mathsf{X}\,(\mathsf{green}\,\mathsf{U}\,\mathsf{orange}))) \\
\equiv \neg(\text{true}\,\mathsf{U}\,(\mathsf{green} \wedge \neg(\neg\mathsf{orange} \wedge \mathsf{X}\,(\mathsf{green}\,\mathsf{U}\,\mathsf{orange})))) = \psi
\end{array}
$$

Compute $\mathcal{A}_\psi$ by using the improved translation where $cl'(\varphi) = \mathsf{green}, \mathsf{orange}, \ldots$:

(a) compute the transition function symbolically in the form of constraints

(b) compute the sets of final states symbolically

(c) compute all outgoing transitions of state $(0,0,0,0,1)^T$ explicitly and for each of the states you encounter determine to which final state sets it belongs

| $cl'(\psi)$ | expansion |
|---|---|
| $1 : \mathsf{green}$ | $d_1 \Leftrightarrow c_1$ |
| $2 : \mathsf{orange}$ | $d_2 \Leftrightarrow c_2$ |
| $3 : \mathsf{green}\,\mathsf{U}\,\mathsf{orange}$ | |
| $4 : \mathsf{X}\,(\mathsf{green}\,\mathsf{U}\,\mathsf{orange})$ | |
| $5 : \text{true}\,\mathsf{U}\,(\mathsf{green} \wedge \neg(\neg\mathsf{orange} \wedge \mathsf{X}\,(\mathsf{green}\,\mathsf{U}\,\mathsf{orange})))$ | |

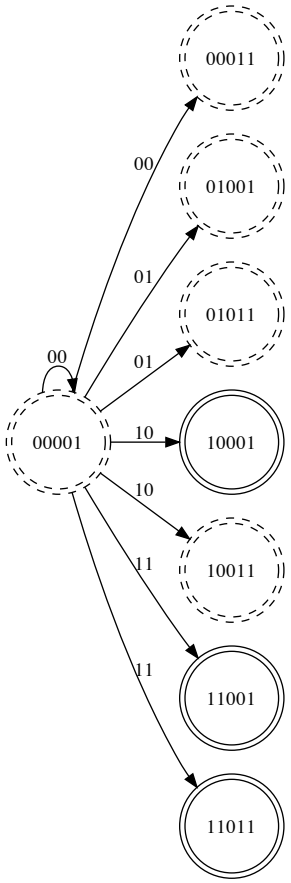
consistency

$$
\begin{aligned}
b_3 &\Leftrightarrow (b_2 \vee (b_1 \wedge c_3)) \\
b_4 &\Leftrightarrow c_3 \\
b_5 &\Leftrightarrow ((b_1 \wedge \neg(\neg b_2 \wedge b_4)) \vee (\text{true} \wedge c_5))
\end{aligned}
$$

- $(c_1, \ldots, c_5)^T \in \delta((b_1, \ldots, b_5)^T, (d_1, d_2)^T)$ iff expansion and consistent
- $(c_1, \ldots, c_5)^T \in \delta(q_0, (d_1, d_2)^T)$ iff expansion and $\neg c_5$
- $F_1 = \{(b_1, \ldots, b_5) \mid \neg b_3 \vee b_2\}$
- $F_2 = \{(b_1, \ldots, b_5) \mid \neg b_5 \vee (b_1 \wedge \neg(\neg b_2 \wedge b_4))\}$
- simplifiying consistency conditions for $\vec{b} = (0, 0, 0, 0, 1)^T$ yields

$$
\begin{aligned}
0 \Leftrightarrow (0 \vee (0 \wedge c_3)) &\equiv \text{true} \\
0 \Leftrightarrow c_3 &\equiv \neg c_3 \\
1 \Leftrightarrow (0 \wedge \neg(\neg 0 \wedge 0)) \vee (\text{true} \wedge c_5)) &\equiv c_5
\end{aligned}
$$

hence, $(0, 0, 0, 0, 1)^T \xrightarrow{(d_1, d_2)} (d_1, d_2, 0, *, 1)$




(double circle belong to $F_1$, solid lines belong to $F_2$)

8. Consider 3 processes (where each process has its own register $r$) which each increases a shared variable $x$ by 1 for 6 times.

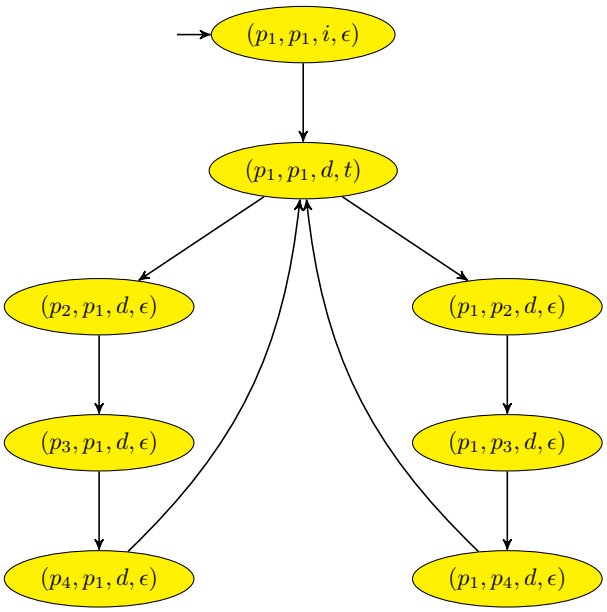
```
LOOP 6 TIMES
  LOAD x INTO r
  INCR r
  STORE r INTO x
```

   - Model these processes in Spin.
   - Determine whether it is possible that the variable $x$ has value 2 when all processes have been executed, assuming that the value of $x$ is initially 0. To this end, formalize a corresponding property in LTL and let Spin determine the answer.

9. Confirm yourself that it is not a good idea to use asynchronous communication for the traffic lights. To this end, create the transition system for the channel system

$$[\mathit{TrafficLight} \mid \mathit{TrafficLight} \mid \mathit{Init}]$$

where $c$ has a buffer of size 1.




The problem with asynchronous communication is that now the crossing is not fair any more, i.e., one of the traffic lights might stay red forever.