Model Checking
Exercises with (Some) Solutions

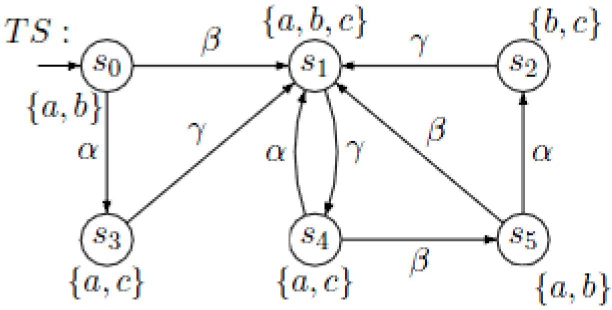
Teacher: Luca Tesei

Master of Science in Computer Science - University of Camerino

Contents

1 Regular Properties

Exercise 1.1. *Consider the following transition system TS:*



and the regular safety property

$$P_{safe} = \begin{array}{l}\text{“always if } a \text{ is valid and } b \wedge \neg c \text{ was valid somewhere before,} \\ \text{then } a \text{ and } b \text{ do not hold thereafter at least until } c \text{ holds”}\end{array}$$

As an example, it holds:

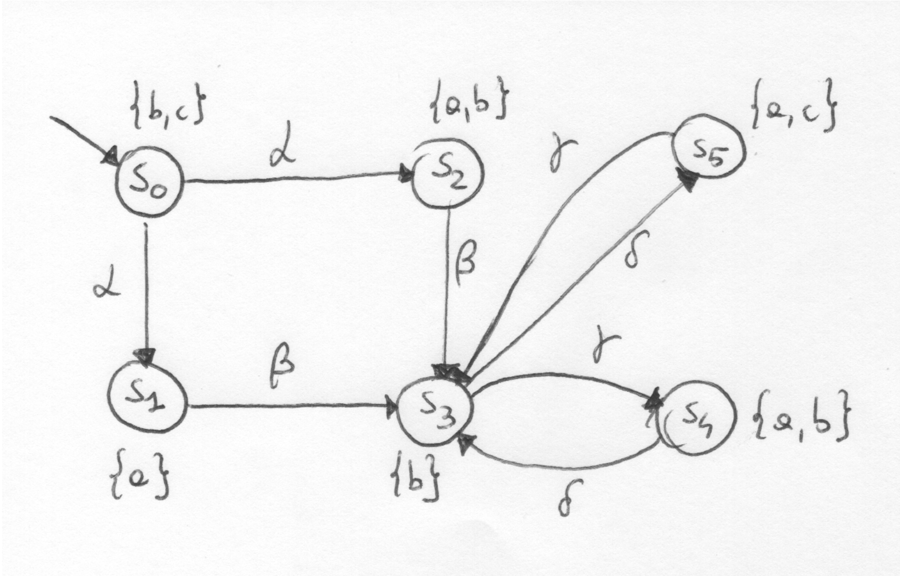
$$\begin{aligned} \{b\}\emptyset\{a,b\}\{a,b,c\} &\in pref(P_{safe}) \\ \{a,b\}\{a,b\}\emptyset\{b,c\} &\in pref(P_{safe}) \\ \{b\}\{a,c\}\{a\}\{a,b,c\} &\in BadPref(P_{safe}) \\ \{b\}\{a,c\}\{a,c\}\{a\} &\in BadPref(P_{safe}) \end{aligned}$$

Questions:

(a) Define an NFA A such that $L(A) = MinBadPref(P_{safe})$

(b) Decide whether $TS \models P_{safe}$ *using the* $TS \otimes A$ *construction.Provide a counterexample if* $TS \not\models P_{safe}$

Exercise 1.2. *Consider the following transition system* TS*:*



and the regular safety property

P_{safe} = *"always if b is holding and a was held somewhere before, then c must* **not** *hold in the position just after the current b"*

1. *Define an NFA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = \mathrm{MinBadPref}(P_{\text{safe}})$*

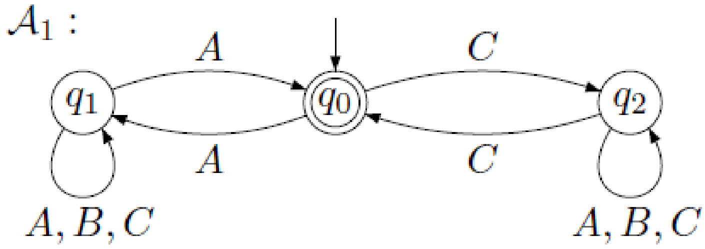
2. *Decide whether $\mathrm{TS} \models P_{\text{safe}}$ using the $\mathrm{TS} \otimes \mathcal{A}$ construction. Provide a counterexample if $\mathrm{TS} \not\models P_{\text{safe}}$*

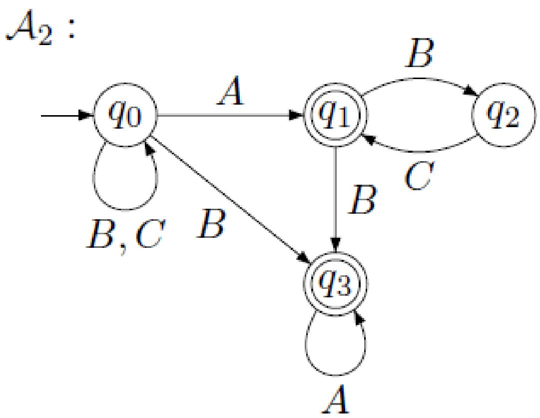
Exercise 1.3. *Find nondeterministic Büchi automata that accept the following ω−regular languages:*

a) $L_1 = \{\sigma \in \{A, B\}^\omega |$ *contains ABA infinitely often, but AA only finitely often* $\}$

b) $L_2 = L_\omega((AB + C) * ((AA + B)C)^\omega + (A * C)^\omega)$

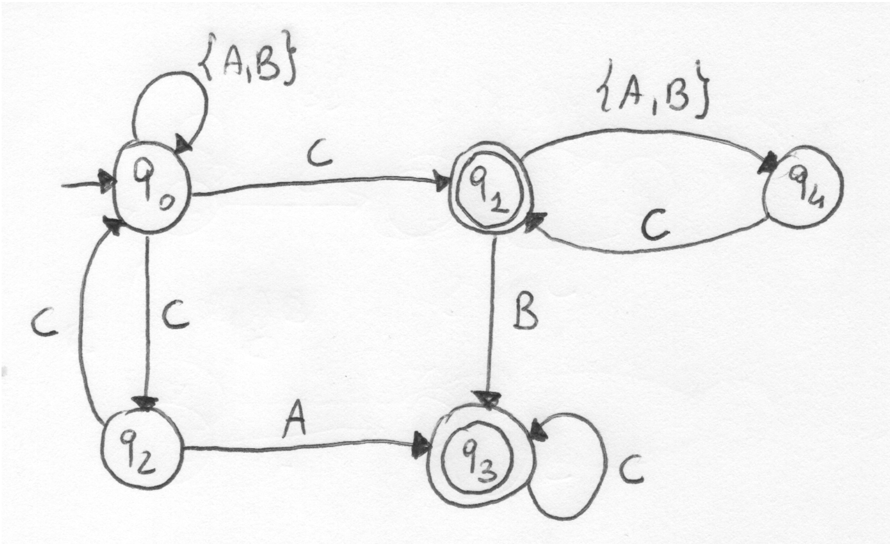
Exercise 1.4. *Consider the following NBA A_1 and A_2 over the alphabet $\sum = \{A, B, C\}$:*





Find ω−regular expressions for the languages accepted by A_1 and A_2, respectively.

Exercise 1.5. *Consider the following* NBA $\mathcal{A}_1$ *over the alphabet* $\Sigma = \{A, B, C\}$.



1. *Write an* ω*-regular expression for the language accepted by* $\mathcal{A}_1$.

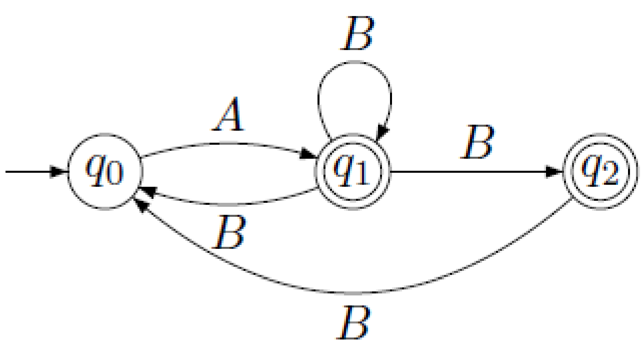
Exercise 1.6. *Prove or disprove the following equivalences for* ω*-regular expressions:*

$a)(E_1 + E_2).F^\omega \equiv E_1.F^\omega + E_2.F^\omega$
$b)E.(F_1 + F_2)^\omega \equiv E.F_1^\omega + E.F_2^\omega$
$c)E.(F.F^*)^\omega \equiv E.F^\omega$
$d)(E^*.F)^\omega \equiv E^*.F^\omega$

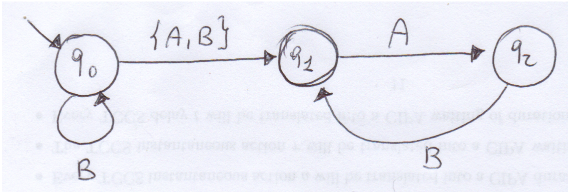
Here, $E, E1, E2, F, F1, F2$ *denote regular expressions with* $\epsilon \notin L(F) \cup L(F1) \cup L(F2)$.

Exercise 1.7. *Show that the class of languages accepted by DBA is not closed under complementation.*

Exercise 1.8. *Consider the GNBA outlined on the right with acceptance sets* $F1 = q1$ *and* $F2 = q2$. *Construct an equivalent NBA using the transformation introduced in the lecture.*



Exercise 1.9. *Consider the following GNBA:*



where the alphabet $\Sigma = \{A, B\}$ *and the acceptance sets are* $\mathcal{F} = \{F_1, F_2\}$ *with* $F_1 = \{q_1\}$ *and* $F_2 = \{q_2\}$.

1. *Construct an equivalent NBA* $\mathcal{A}$ *using the transformation introduced in the lectures.*
2. *Write an* ω*-regular expression denoting exactly* $\mathcal{L}_\omega(\mathcal{A})$.

Exercise 1.10. *Provide NBA A1 and A2 for the languages given by the expressions* $(AC + B)^*B^\omega$ *and* $(B^*AC)^\omega$ *and apply the product construction (using GNBA) to obtain an NBA A with* $L_\omega(A) = L_\omega(A_1) \cap L_\omega(A_2)$. *Justify, why* $L_\omega(G) = \emptyset$ *where G denotes the GNBA accepting the intersection.*

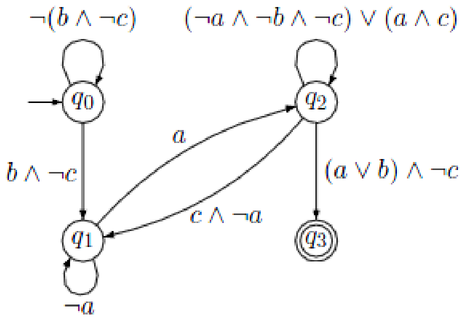
Exercise 1.11. *Draw nondeterministic Büchi automata that accept the following* ω*-regular languages:*

1. $\mathcal{L}_1 = \{\sigma \in \{A, B, C\}^\omega \mid \sigma$ *contains* C *only finitely many times and contains* AB *infinitely many times*$\}$
2. $\mathcal{L}_2 = (AB + AC)^*ABC(BCA + ACB)^\omega + (A + B)^*(CB)^\omega$

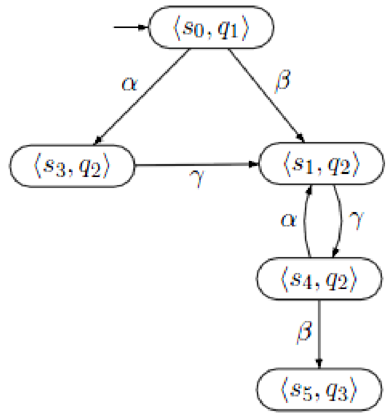
Solutions

Solution of Exercise 1.1

- The NFA that accepts the set of minimal bad prefixes:



- First we apply the $TS \otimes \mathcal{A}$ construction which yields:



A counterexample to $TS \models P_{safe}$ is given by the following initial path fragment in $TS \otimes \mathcal{A}$:

$$\pi_{\otimes} = \langle s_0, q_1 \rangle \langle s_3, q_2 \rangle \langle s_1, q_2 \rangle \langle s_4, q_2 \rangle \langle s_5, q_3 \rangle$$

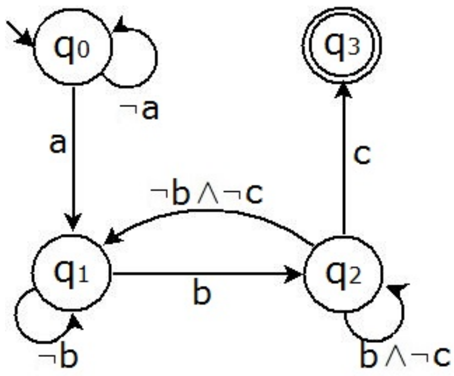
By projection on the state component, we get a path in the underlying transition system:

$$\pi = s_0 s_3 s_1 s_4 s_5 \text{ with } trace(\pi) = \{a, b\}\{a, c\}\{a, b, c\}\{a, c\}\{a, b\}$$

Obviously, $trace(\pi) \in BadPref(P_{safe})$, so we have $Traces_{fin}(TS) \cap BadPref(P_{safe}) \neq \emptyset$. By lemma 3.25, this is equivalent to $TS \not\models P_{safe}$.

Solution of Exercise 1.2

1. An NFA accepting the minimal bad prefixes for the property is $\mathcal{A}$:



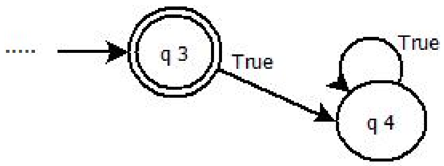
where:
$\neg a \equiv \{\{\}, \{b\}, \{c\}, \{b, c\}\}$
$a \equiv \{\{a\}, \{a, b\}, \{a, c\}, \{a, b, c\}\}$
The union of $\neg a$ and a is 2^{AP}

$\neg b \equiv \{\{\}, \{a\}, \{c\}, \{a, c\}\}$
$b \equiv \{\{b\}, \{a, b\}, \{b, c\}, \{a, b, c\}\}$
The union of $\neg b$ and b is 2^{AP}

$c \equiv \{\{c\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$
$b \wedge \neg c \equiv \{\{b\}, \{a, b\}\}$
$\neg b \wedge \neg c \equiv \{\{\}, \{a\}\}$
The union of c, $b \wedge \neg c$ and $\neg b \wedge \neg c$ is 2^{AP}

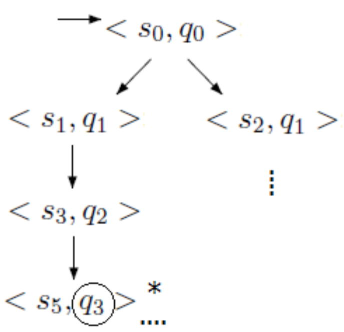
So the NFA is non-blocking apart from state q_3.

2. To apply the product $TS \otimes \mathcal{A}$, $\mathcal{A}$ should be non-blocking. Our $\mathcal{A}$ is deterministic and becomes non-blocking if we add a state q_4 and let



or alternatively we can add a self-loop on q_3. In this case the automaton would recognize all bad prefixes, not just the minimal ones. Let us consider $\mathcal{A}'$ made on one of these two ways.

Let's construct the product:
$L(s_0) = \{b, c\}$ $\delta(q_0, \{b, c\}) = \{q_0\}$
So the unique initial state of $TS \otimes \mathcal{A}'$ is $< s_0, q_0 >$



From $< s_0, q_0 >$:

- $s_0 \longrightarrow s_1 \; L(s_1) = \{a\}$
  $\delta(q_0, \{a\}) = \{q_1\}$.
- $s_0 \longrightarrow s_2 \; L(s_2) = \{a, b\}$
  $\delta(q_0, \{a, b\}) = \{q_1\}$.

From $< s_1, q_1 >$:

- $s_1 \longrightarrow s_3 \; L(s_3) = \{b\}$
  $\delta(q_1, \{b\}) = \{q_2\}$.

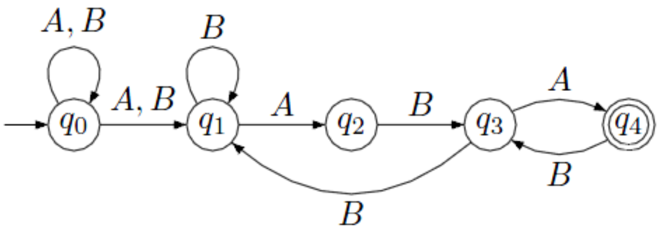
From $< s_3, q_2 >$:

- $s_3 \longrightarrow s_5 \; L(s_5) = \{a, c\}$
  $\delta(q_2, \{a, c\}) = \{q_3\}$.

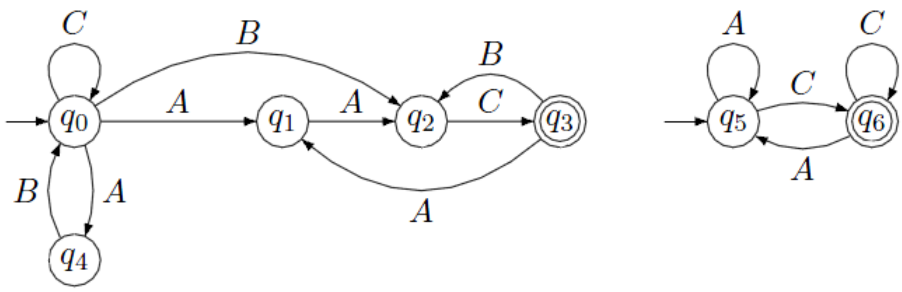
we can stop constructing $TS \otimes \mathcal{A}'$ because we can already decide that $TS \not\models P_{safe}$.
Indeed in $TS \otimes \mathcal{A}'$ a state in which q_3 is present is reachable *. The path gives us a counter-example for the property:
$s_0 s_1 s_3 s_5 ...$ whose trace is $\{b, c\}\{a\}\{b\}\{a, c\} ... \not\models P_{safe}$

Solution of Exercise 1.3

a) $L_1 = \{\sigma \in \{A, B\}^\omega \mid \sigma$ contains ABA infinitely often, but AA only finitely often$\}$



b) $L_2 = \mathcal{L}\left((AB + C)^*((AA + B)C)^\omega + (A^*C)^\omega\right)$



Solution of Exercise 1.4

$a) L_\omega(A_1) = L_{q_0q_0}.L^\omega_{q_0q_0} = L(C(A+B+C)^+C + A(A+B+C)^+A)^\omega$

$b)$ Here, we have $F = \{q_1, q_3\}$:

$$L_{q_0q_1} = (B+C)^*A(BC)^*$$

$$L_{q_0q_3} = (B+C)^*(B + A(BC)^*B)A^*$$

$$L_{q_1q_1} = (BC)*$$

$$L_{q_3q_3} = A*$$

The language accepted by A_2 then is:

$L_\omega(A_2) = \cup_{q\in F, q_0 \in Q_0} L_{q_0q}.(L_{q,q} \setminus \{\epsilon\})^\omega$

$$= L_{q_0q_1}.(L_{q_1,q_1} \setminus \{\epsilon\})^\omega \cup L_{q_0q_3}.(L_{q_3,q_3} \setminus \{\epsilon\})^\omega$$

$$= L_\omega([(B+C)^*A(BC)^*].[(BC)^+)]^\omega + [(B+C)^*(B + A(BC)^*B)A^*].[A^+)]^\omega$$

Solution of Exercise 1.5

Let's use the procedure given in the lecture slides.

$L_{q_0q_1} = ((A+B)^*(CC)^*)C((A+B)C)^*$
$L_{q_0q_3} = ((A+B)^*(CC)^*)CAC^* + ((A+B)^*(CC)^*)C((A+B)C)^*BC^*$
$L_{q_1q_1} = ((A+B)C)^* \implies L_{q_1q_1} \setminus \{\varepsilon\} = ((A+B)C)^+$
$L_{q_3q_3} = C^* \implies L_{q_3q_3} \setminus \{\varepsilon\} = C^+$

Then $L_\omega(\mathcal{A}_\infty) = ((A+B)^*(CC)^*)C((A+B)C)^\omega + [((A+B)^*(CC)^*)CAC^* + ((A+B)^*(CC)^*)C((A+B)C)^*B]C^\omega$
(already simplified)

Solution of Exercise 1.6

a) $(E_1 + E_2).F^\omega \equiv E_1.F^\omega + E_2.F^\omega$
True, since:

$$\begin{aligned}
\mathcal{L}_\omega((E_1 + E_2).F^\omega) &= \mathcal{L}(E_1 + E_2).\mathcal{L}(F)^\omega \\
&= \big(\mathcal{L}(E_1) \cup \mathcal{L}(E_2)\big).\mathcal{L}(F)^\omega \\
&= \mathcal{L}(E_1).\mathcal{L}(F)^\omega \cup \mathcal{L}(E_2).\mathcal{L}(F)^\omega \\
&= \mathcal{L}_\omega(E_1.F^\omega) \cup \mathcal{L}_\omega(E_2.F^\omega) \\
&= \mathcal{L}_\omega(E_1.F^\omega + E_2.F^\omega)
\end{aligned}$$

b) $E.(F_1 + F_2)^\omega \equiv E.F_1^\omega + E.F_2^\omega$
False: Consider $E = \underline{\varepsilon}$ and $F_1 = A$, $F_2 = B$ where $\underline{\varepsilon}$ denotes the language consisting of the empty word only, i.e. $\{\varepsilon\}$.
Then $\mathcal{L}_\omega(E.(F_1 + F_2)^\omega) = \{A, B\}^\omega$, but $(AB)^\omega \notin \mathcal{L}_\omega(E.F_1^\omega + E.F_2^\omega) = \{A^\omega, B^\omega\}$.

c) $E.(F.F^*)^\omega \equiv E.F^\omega$
True, since:

$$\begin{aligned}
\mathcal{L}_\omega(E.(F.F^*)^\omega) &= \mathcal{L}(E).\mathcal{L}(F.F^*)^\omega \\
&= \mathcal{L}(E).\mathcal{L}(F^+)^\omega \\
&= \mathcal{L}(E).\left(\{w_0 w_1 w_2 \dots w_k \mid k > 0 \wedge w_i \in \mathcal{L}(F) \text{ for all } i \in \{0, \dots, k\}\}\right)^\omega \\
&= \mathcal{L}(E).\{v_1 v_2 \dots \mid v_i \in \mathcal{L}(F^+)\} \\
&= \mathcal{L}(E).\left\{w_{1,1}w_{1,2}\dots w_{1,k_1}w_{2,1}\dots w_{2,k_2}w_{3,1}\dots \mid w_{i,j_i} \in \mathcal{L}(F)\ \forall i \geq 1 \wedge \forall j_i \in \{1, \dots, k_i\}\right\} \\
&= \mathcal{L}(E).\mathcal{L}(F)^\omega \\
&= \mathcal{L}_\omega(E.F^\omega)
\end{aligned}$$

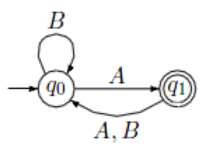
d) $(E^*.F)^\omega \equiv E^*.F^\omega$
False: Consider $E = A$, $F = B$. Then, $(AB)^\omega \in \mathcal{L}_\omega((E^*.F)^\omega)$ but $(AB)^\omega \notin \mathcal{L}_\omega(E^*.F^\omega)$

Solution of Exercise 1.7

To show that the class of DBA-acceptable languages is not closed under complementation, consider the following ω-regular language over $\Sigma = \{A, B\}$:

$$L = \mathcal{L}_\omega\big(((A + B)^* A)^\omega\big)$$

It is recognizable by the following deterministic Büchi automaton:



It remains to show that its complement language $\bar{L} = \{A, B\}^\omega \setminus L = \mathcal{L}_\omega\big((A{+}B)^* B^\omega\big)$ cannot be recognized by a deterministic Büchi automaton.
This is proven in Theorem 4.46 in the lecture notes.

Solution of Exercise 1.8

The acceptance condition for GNBA $A = (Q, \sum, \delta, Q_0, F)$ with $F = \{F1, ..., Fn\}$ and $Fi \subseteq Q$for $(1 \leq i \leq n)$:

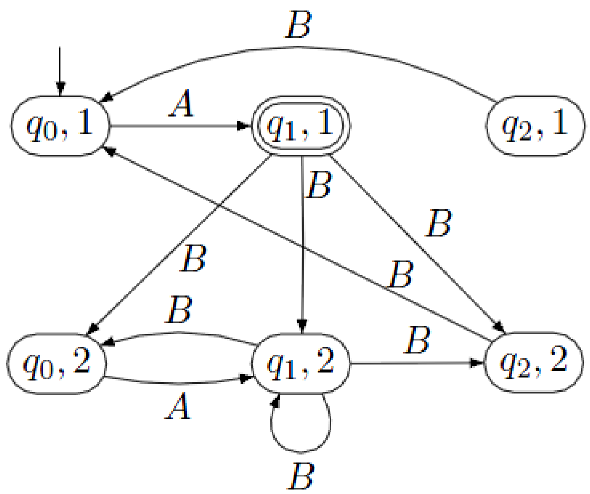
$$\mathcal{A} \text{ accepts } \alpha \in \Sigma^\omega \iff \text{ ex. infinite run } \rho \text{ of } \mathcal{A} \text{ on } \alpha \text{ s.t. } \forall F \in \mathcal{F}. \left(\overset{\infty}{\exists}\, j \geq 0.\ \rho[j] \in F\right)$$

Using the construction from the lecture, we infer the following NBA

$$\mathcal{A}' = (Q', \Sigma, \delta', Q'_0, F) \text{ where}$$

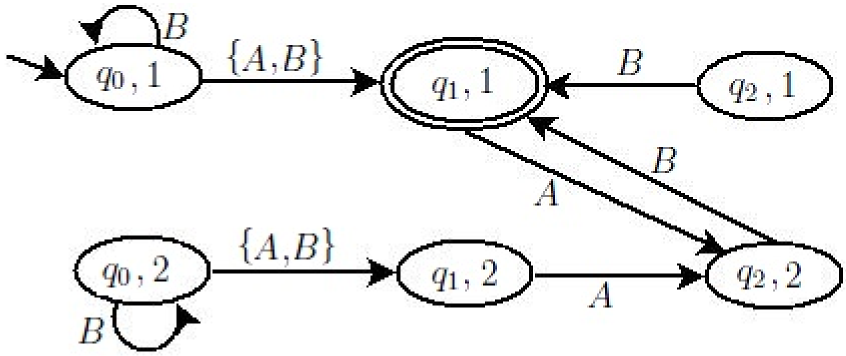
- $Q' = Q \times \{1, 2\}$
- $\delta'((q, i), A) = \begin{cases} \{(q', i) \mid q' \in \delta(q, A)\} & \text{if } q \notin F_i \\ \{(q', (i \bmod 2) + 1) \mid q' \in \delta(q, A)\} & \text{otherwise} \end{cases}$
- $Q'_0 = \{(q_0, 1)\}$
- $F = \{(q_1, 1)\}$

The automaton can be outlined as follows: Using the construction from the lecture, we infer the following NBA

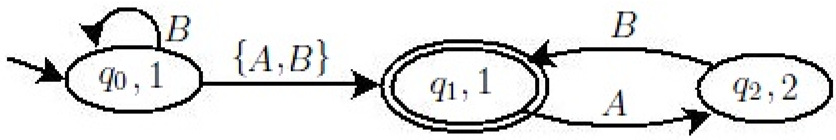


Solution of Exercise 1.9

1. The state space of the NBA is $\{q_0, q_1, q_2\} \times \{1, 2\}$



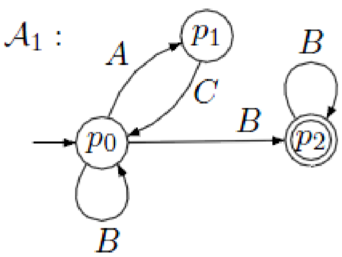
where $F = \{q_1, 1\}$.
Eliminating the unreachable states is: $\mathcal{A}$:

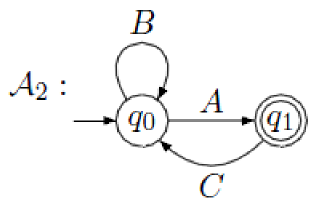


2. An ω-regular expression for the language of $\mathcal{A}$ is $\alpha = B^*(A + B)(AB)^\omega$

Solution of Exercise 1.10

NBA $A_1 = (Q_1, \sum, \delta_1, Q_{0,1}, F_1)$ and $A_2 = (Q_2, \sum, \delta_2, Q_{0,2}, F_2)$ for the languages:



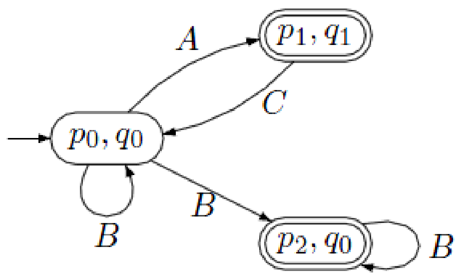


The corresponding GNBA are given by:
$G_1 = (Q_1, \sum, \delta_1, Q_{0,1}, \{F_1\})$
$G_2 = (Q_2, \sum, \delta_2, Q_{0,2}, \{F_1\})$

Applying the product construction (cf. Lemma 4.60) yields the following GNBA:
$G = (Q_1 \times Q_2, \sum, \delta, Q_{0,1} \times Q_{0,2}, \mathcal{F})$ where

- $\delta((p,q),A) = \delta_1(p,A) \times \delta_2(q,A)$
- $\mathcal{F} = \{F_1 \times Q_2\} \cup \{Q_1 \times F_2\} = \{\{(p_2,q_0),(p_2,q_1)\},\{(p_0,q_1),(p_1,q_1),(p_2,q_1)\}\}$

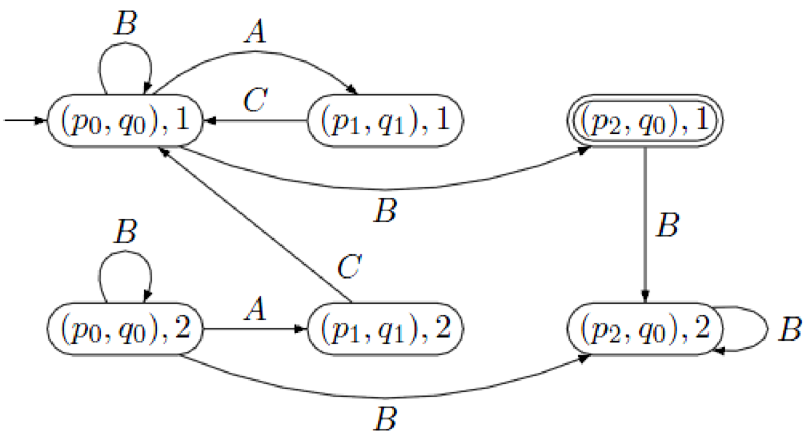
The automaton G can be outlined as follows (only reachable states are outlined below):



Its acceptance component is $\mathcal{F} = \{\{(p_2,q_0)\},\{(p_1,q_1)\}\}$.

According to the acceptance condition of GBNA, G accepts an input word if and only if for each $F \in \mathcal{F}$ some states are visited infinitely often. But as soon as (p_2,q_0) is visited, F_1 is not reachable any longer.
Therefore G only accepts the empty language.

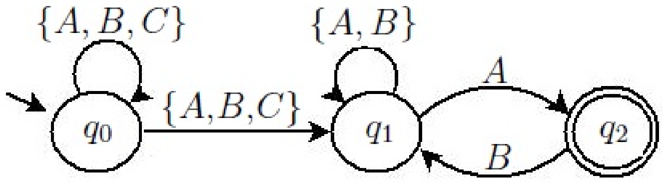
Given G, construct an equivalent NBA A:



Again, on each possible run, the state $((p_2,q_0),2)$ of A can be visited only once. Therefore also $L_\omega(A) = \emptyset$ holds.

Solution of Exercise 1.11

1. In the prefix there could be As, Bs and Cs any order, the tail should be of the form $(A^+B^+)^\omega = AAABBABAABA...$



   Switches from A to B infinitely many times.

2. .