Prof. Javier Esparza
Philipp Czerner

Technical University of Munich
Chair for Foundations of Software Reliability

Automata and Formal Languages

Winter Term 2023/24 – Exercise Sheet 10

Exercise 10.1.

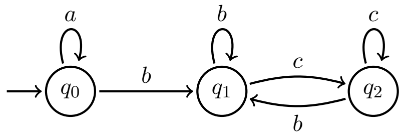
Consider automata with the set of states $Q = \{q_0, q_1, q_2\}$ and the acceptance conditions $\alpha_1, \alpha_2, \alpha_3, \alpha_4$ given by the following table:

	$\{q_0\}$	$\{q_1\}$	$\{q_2\}$	$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_1, q_2\}$	$\{q_0, q_1, q_2\}$
α_1	1	0	0	1	1	0	1
α_2	0	1	0	1	0	0	0
α_3	1	1	0	1	0	0	0
α_4	0	0	0	0	0	0	1

(a) For each of the conditions determine if they are Büchi, co-Büchi, Rabin, Muller.

(b) Can it happen that an accepting condition is neither Büchi nor co-Büchi nor Rabin nor Muller? If yes, give an example of such a condition.

(c) Consider the following semi-automaton and acceptance conditions $\alpha_1, \alpha_2, \alpha_3, \alpha_4$. What are the languages accepted by the obtained automata?



Solution.

(a) α_1 is a Büchi condition with $F = \{q_0\}$

α_2 is a Rabin condition with the set of Rabin pairs $\{\langle\{q_1\}, \{q_2\}\rangle\}$

α_3 is a co-Büchi condition with $F = \{q_2\}$

α_4 is a Muller condition with the Muller set $\{\{q_0, q_1, q_2\}\}$

(b) No. If a condition is neither Büchi nor co-Büchi nor Rabin, then it must be Muller. A Muller condition is an arbitrary condition.

(c) L_1 is defined by the expression a^ω

L_2 is defined by $a^*(bc^*)^*b^\omega$

L_3 is the union of L_1 and L_2

L_4 is the empty set, as we cannot have a run in which all 3 states are visited infinitely often.

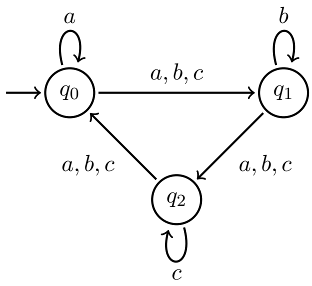
Exercise 10.2.

Let language $L = \{w \in \{a, b\}^\omega : w \text{ contains finitely many } a\}$

(a) Give a deterministic Rabin automaton for L.

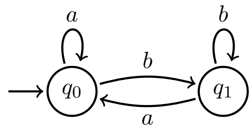
(b) Give an NBA for L and try to "determinize" it by using the NFA to DFA powerset construction. What is the language accepted by the resulting DBA?

(c) What ω-language is accepted by the following Muller automaton with acceptance condition $\{\{q_0\}, \{q_1\}, \{q_2\}\}$? And with acceptance condition $\{\{q_0, q_1\}, \{q_1, q_2\}, \{q_2, q_0\}\}$?

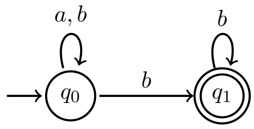


Solution.

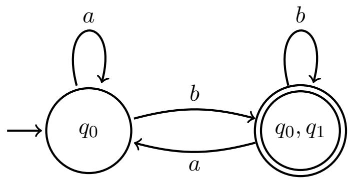
(a) The following DRA, with acceptance condition $\{\langle\{q_1\}, \{q_0\}\rangle\}$, i.e., a run is accepting iff it visits q_1 infinitely often and q_0 finitely often, recognizes L:



(b) This NBA accepts L:



The powerset construction yields the DBA below. It recognizes the language $\{w : w \text{ contains infinitely many b}\}$, which is different from $(a + b)^*b^\omega$:



(c) With the first acceptance condition the language is $\Sigma^*(a^\omega + b^\omega + c^\omega)$. With the second, the automaton does not accept any word. Indeed, every run that visits both q_0 and q_1 infinitely often must also visit q_2 infinitely often, and the same holds for q_1 and q_2, and for q_2 and q_0.

Exercise 10.3.

Let $L_1 = (ab)^\omega$ and let L_2 be the language of all words over $\{a, b\}$ containing infinitely many a and infinitely many b.

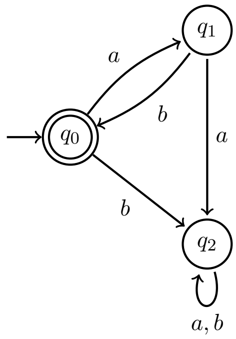
(a) Exhibit three different DBAs with three states recognizing L_1.

(b) Exhibit six different DBAs with three states recognizing L_2.

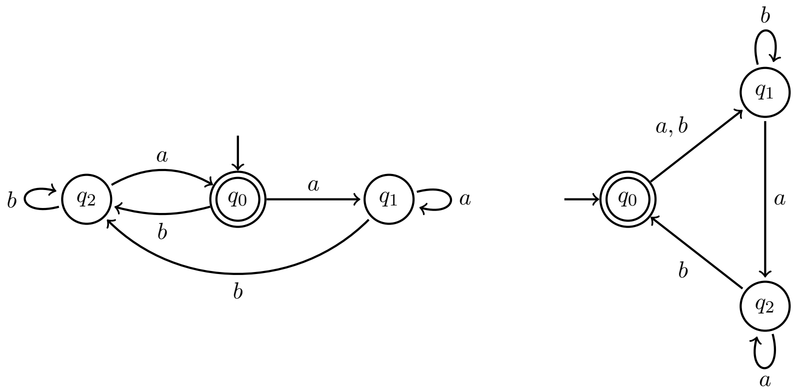
(c) Show that no DBA with at most two states recognizes L_1 or L_2.

Solution.

(a) We obtain three DBAs for L_1 from the one below by making either q_0, q_1 or both accepting:



(b) Here are two different DBAs for L_2. We obtain two further DBAs from each of these automata by making either q_1 or q_2 the initial state.



(c) A DBA with a single state either accepts the empty language or $(a+b)^\omega$ and so no single-state DBA can accept L_1 or L_2. Suppose B is a two-state DBA with states p and q which accepts the language L_1. Let p be the initial state of B.

If q is not reachable from p by means of any transition, then the language accepted by B is either the empty language or $(a+b)^\omega$. Hence, we can assume that either $p \xrightarrow{a} q$ or $p \xrightarrow{b} q$. Without loss of generality, we can assume that $p \xrightarrow{a} q$. Notice that either $q \xrightarrow{a} q$ or $q \xrightarrow{a} p$. In either case, it is clear that if q is a final state then a^ω will be accepted by B, leading to a contradiction as $a^\omega \notin L_1$. Hence, q is not a final state and so p must be a final state.

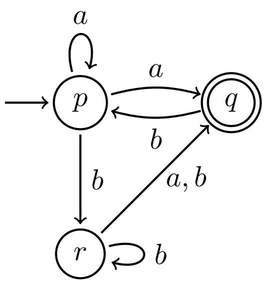
Notice that if $p \xrightarrow{b} p$ then b^ω will be accepted by B, once again leading to a contradiction. Hence we have $p \xrightarrow{a} q$ and $p \xrightarrow{b} q$. Because of this and because of the fact that p is the only final state, it must be the case that either $q \xrightarrow{a} p$ or $q \xrightarrow{b} p$. In the former case, a^ω is accepted by B and in the latter case b^ω is accepted by B, both leading to a contradiction.

It follows that no two-state DBA can accept L_1. If we replace L_1 with L_2 in the above argument, then we can also show that no two-state DBA can accept L_2 as well.

Exercise 10.4.

(a) Show that for every NCA there is an equivalent NBA.

(b) For the following NCA give an equivalent NBA, using the construction from (a):



Solution.

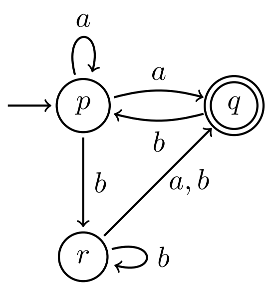
(a) Let $A = (Q, \Sigma, \delta, Q_0, F)$ be an NCA. We construct an NBA B which is equivalent to A. Observe that the co-Büchi accepting condition $\inf(\rho) \cap F = \emptyset$ is equivalent to $\inf(\rho) \subseteq Q \setminus F$. This condition holds iff ρ has an infinite suffix that only visits states of $Q \setminus F$. We design B in two stages. In the first one, we take two copies of A, that we call A_0 and A_1, and put them side by side; A_0 is a full copy, containing all states and transitions of A, and A_1 is a partial copy, containing only the states of $Q \setminus F$ and the transitions between these states. We write $[q, 0]$ to denote the copy of a state $q \in Q$ in A_0, and $[q, 1]$ for the copy of a state $q \in Q \setminus F$ in A_1. In the second stage, we add some transitions that "jump" from A_0 to A_1: for every transition $[q, 0] \xrightarrow{a} [q', 0]$ of A_0 such that $q' \in Q \setminus F$, we add a transition $[q, 0] \xrightarrow{a} [q', 1]$ that "jumps" to $[q', 1]$, the "twin state" of $[q', 0]$ in A_1. Note that $[q, 0] \xrightarrow{a} [q', 1]$ does not replace $[q, 0] \xrightarrow{a} [q', 0]$, it is an *additional* transition. As initial states of B, we choose the copy of Q_0 in A_0, i.e., $\{[q, 0] : q \in Q_0\}$, and as accepting states all the states of A_1, i.e., $\{[q, 1] : q \in Q \setminus F\}$.

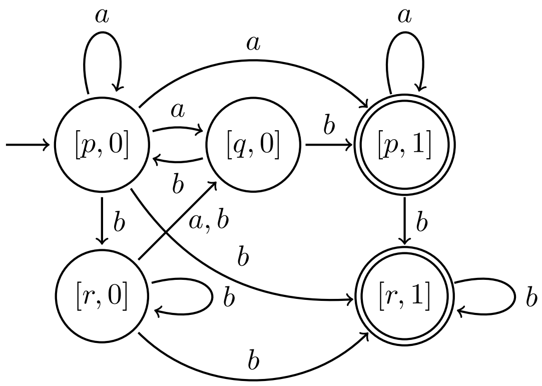
It remains to show that $L_\omega(A) = L_\omega(B)$.

$\subseteq$) Let $w \in L_\omega(A)$. There is a run ρ of A on word w such that $\inf \rho \cap F = \emptyset$. It follows that $\rho = \rho_0\, \rho_1$, where ρ_0 is a finite prefix of ρ, and ρ_1 is an infinite suffix that only contains states of $Q \setminus F$. Let ρ' be the run of B on w that simulates ρ_0 on A_0, and then "jumps" to A_1 and simulates ρ_1 in A_1. Notice that ρ' exists because ρ_1 only visits states of $Q \setminus F$. Since all states of A_1 are accepting, ρ' is an accepting run of the NBA B, and so $w \in L_\omega(B)$.

$\supseteq$) Let $w \in L_\omega(B)$. There is an accepting run ρ of B on word w. Thus, ρ visits states of A_1 infinitely often. Since a run of B that enters A_1 can never return to A_0 (there are no "back-jumps" from A_1 to A_0,) ρ has an infinite suffix ρ_1 that only visits states of A_1, i.e., states $[q, 1]$ such that $q \in Q \setminus F$. Let ρ' be the result of replacing $[q, 0]$ and $[q, 1]$ by q everywhere in ρ. Clearly, ρ' is a run of A on w that only visits F finitely often. Thus, ρ' is an accepting run of A, and $w \in L_\omega(A)$. □

(b) The NCA below on the left is transformed into the NBA on the right:





Exercise 10.5.

Give a procedure that translates non-deterministic Rabin automata to non-deterministic Büchi automata.

Solution.

Given a Rabin automaton $A = (Q, \Sigma, Q_0, \delta, \{\langle F_0, G_0\rangle, \dots, \langle F_{m-1}, G_{m-1}\rangle\})$, it follows easily that $L_\omega(A) = \bigcup_{i=0}^{m-1} L_\omega(A_i)$ where each $A_i = (Q, \Sigma, Q_0, \delta, \{\langle F_i, G_i\rangle\})$. So it suffices to translate each A_i into an NBA B_i and take the union of the B_i's. For this, we use the same idea that we used for converting an NCA into an NBA (as shown in the previous exercise). To construct B_i, we take two copies of A_i, say A_i^0 and A_i^1, where A_i^0 is a full copy of A_i and A_i^1 is a partial copy containing only the states of $Q \setminus G_i$ and the transitions between these states. We let $[q, i]$ denote the i^{th} copy of the state q and for every transition $q \xrightarrow{a} q'$ in A_i with $q' \in Q \setminus G_i$, we add a transition $[q, 0] \xrightarrow{a} [q', 1]$ to B_i. We set the initial states to be $\{[q, 0], q \in Q_0\}$ and we set the final states to be $\{[q, 1] : q \in F_i\}$. Similar to the last exercise of the previous sheet, we can show that B_i accepts $L_\omega(A_i)$.