Automata and Formal Languages — Exercise Sheet 11

Exercise 11.1
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 $\omega$-language is accepted by the following Muller automaton with acceptance condition $\{ \{q_0\}, \{q_1\}, \{q_2\}\}$? And with acceptance condition $\{ \{q_0, q_1\}, \{q_1, q_2\}, \{q_2, q_0\}\}$?

Exercise 11.2
Give a procedure that translates non-deterministic Rabin automata to non-deterministic Büchi automata.

Exercise 11.3
Let $L_\sigma = \{ w \in \{a, b, c\}^\omega : w \text{ contains infinitely many } \sigma \text{'s} \}$. Give deterministic Büchi automata for languages $L_a, L_b$ and $L_c$, and construct the intersection of these automata.

Exercise 11.4

(a) Consider the following Büchi automaton $A$ over $\Sigma = \{a, b\}$:

Draw $\text{dag}(abab^\omega)$ and $\text{dag}((ab)^\omega)$.

(b) Let $r_w$ be the ranking of $\text{dag}(w)$ defined by

$$r_w(q, i) = \begin{cases} 1 & \text{if } q = q_0 \text{ and } (q_0, i) \text{ appears in } \text{dag}(w), \\ 0 & \text{if } q = q_1 \text{ and } (q_1, i) \text{ appears in } \text{dag}(w), \\ \bot & \text{otherwise}. \end{cases}$$

Are $r_{abab^\omega}$ and $r_{(ab)^\omega}$ (over $A$) odd rankings?
(c) Consider the following Büchi automaton $B$ over $\Sigma = \{a, b\}$:

\[
\begin{array}{c}
q_0 \xrightarrow{a} \quad q_1 \xrightarrow{a} \quad q_2 \\
\end{array}
\]

Draw $\text{dag}(a^\omega)$. Show that any odd ranking for this dag must contain a rank of 3 or more.
Solution 11.1

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

\[
\begin{array}{ccc}
 q_0 & \xrightarrow{a} & q_1 \\
 & \xrightarrow{b} & \\
 & \xrightarrow{a} & q_0
\end{array}
\]

(b) This NBA accepts \(L\):

\[
\begin{array}{ccc}
 q_0 & \xrightarrow{a,b} & q_1 \\
 & \xrightarrow{b} & \\
 & \xrightarrow{a,c} & q_0
\end{array}
\]

The powerset construction yields the DBA below (with the trap state omitted). It recognizes the language \(a^*b^*\), which is different from \((a + b)^*b^*\):

\[
\begin{array}{ccc}
 q_0 & \xrightarrow{a,b} & q_0, q_1 \\
 & \xrightarrow{b} & q_0
\end{array}
\]

(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 \(q_0\).

Solution 11.2

Given a Rabin automaton \(A = (Q, \Sigma, Q_0, \delta, \{\langle F_0, G_0 \rangle, \ldots, \langle F_{m-1}, G_{m-1} \rangle\})\), 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 sheet). To construct \(B_i\), we take two copies of \(A_i\), say \(A_0^i\) and \(A_1^i\), where \(A_0^i\) is a full copy of \(A_i\) and \(A_1^i\) 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 \([\{0, 1\} : q \in Q_0\}\) 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 \(L_\omega(A_i)\).

Solution 11.3

The following deterministic Büchi automata respectively accept \(L_a\), \(L_b\) and \(L_c\):

\[
\begin{array}{ccc}
 p_0 & \xrightarrow{b,c} & p_0 \\
 & \xrightarrow{a} & p_1 \\
 & \xrightarrow{a,c} & p_1 \\
 q_0 & \xrightarrow{a,c} & q_1 \\
 & \xrightarrow{b} & q_0 \\
 r_0 & \xrightarrow{a,b} & r_1 \\
 & \xrightarrow{c} & r_1
\end{array}
\]

Taking their intersection leads to the following deterministic Büchi automaton:
Note that $L_a \cap L_b \cap L_c$ is accepted by a smaller DBA:

Solution 11.4

(a) $\text{dag}(abab^\omega)$:

(b) $r$ is not an odd rank for $\text{dag}(abab^\omega)$ since

$$
\langle q_0, 0 \rangle \xrightarrow{a} \langle q_0, 1 \rangle \xrightarrow{b} \langle q_0, 2 \rangle \xrightarrow{a} \langle q_0, 3 \rangle \xrightarrow{b} \langle q_1, 4 \rangle \xrightarrow{b} \langle q_1, 5 \rangle \xrightarrow{b} \cdots
$$

is an infinite path of $\text{dag}(abab^\omega)$ not visiting odd nodes infinitely often.
• $r$ is an odd rank for $\text{dag}((ab)^\omega)$ since it has a single infinite path:

$$
\langle q_0, 0 \rangle \xrightarrow{a} \langle q_0, 1 \rangle \xrightarrow{b} \langle q_0, 2 \rangle \xrightarrow{a} \langle q_0, 3 \rangle \xrightarrow{b} \langle q_0, 4 \rangle \xrightarrow{a} \langle q_0, 5 \rangle \xrightarrow{b} \ldots
$$

which only visits odd nodes.

(c) $\text{dag}(a^\omega)$:

Let $r$ be an odd rank for $\text{dag}(a^\omega)$. It exists since $a^\omega$ is not accepted by $B$. Since $r$ is odd, all infinite paths must visit odd nodes infinitely often (i.o.). In particular the bottom infinite path of $q_0$ nodes must stabilize to nodes with odd rank.

Let us assume the nodes $\langle q_0, j \rangle$ have rank 1 for all $j \geq i$ for some $i \geq 0$. Consider the infinite path $\rho = \langle q_0, i \rangle \xrightarrow{a} \langle q_1, i + 1 \rangle \xrightarrow{a} \langle q_2, i + 2 \rangle \xrightarrow{a} \langle q_2, i + 3 \rangle \ldots$. Node $\langle q_1, i + 1 \rangle$ must have an even rank (since $q_1$ is accepting) smaller or equal to 1, so it has rank 0. This entails that $\langle q_2, k \rangle$ has rank 0 for all $k \geq i + 2$. This contradicts $r$ being an odd ranking because the path $\rho$ is infinite yet does not visit odd nodes infinitely often.

Thus the bottom infinite path of $q_0$ nodes must stabilize to nodes with odd rank strictly bigger than 1, i.e., bigger or equal to 3.