Automata and Formal Languages — Exercise Sheet 2

Exercise 2.1
Consider the regular expression \( r = (b + bab)^* \).

(a) Convert \( r \) into an equivalent NFA-\( \varepsilon \) \( A \).
(b) Convert \( A \) into an equivalent NFA \( B \). (It is not necessary to use algorithm \( \text{NFA}_\varepsilon \text{toNFA} \))
(c) Convert \( B \) into an equivalent DFA \( C \).
(d) By inspecting \( B \), merge two states to get an equivalent DFA \( D \) with less states if possible (depending on how you answered previous questions, this may not be possible). No algorithm needed.
(e) Convert \( D \) into an equivalent regular expression \( r' \).
(f) Prove formally that \( L(r) = L(r') \).

Exercise 2.2
Prove that if \( L \) is a finite language, then the complement of \( L \) is a regular language.

Exercise 2.3
Let \( \Sigma = \{a, b, c\} \). Show that the language described by the regular expression \( (((b + c)^* a + c^*) + (be^*))^* \) is the set of all words over \( \Sigma \).

Exercise 2.4
Let \( n \geq 1 \) be some natural number and let \( \Sigma = \{a : 1 \leq a \leq n\} \). Consider the following language over \( \Sigma \):
\[
L = \{aaa : a \in \Sigma\}
\]
- Show that there is a NFA with \( 2n + 2 \) states which recognizes \( L \).
- Show that any NFA recognizing \( L \) must have at least \( 2n + 2 \) states.
Solution 2.1
There are different correct answers for the following exercises, the following is one possible set of answers.

(a)

<table>
<thead>
<tr>
<th>Iter.</th>
<th>Automaton obtained</th>
<th>Rule applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$p \rightarrow (b + bab)^* q$</td>
<td>Initial automaton from reg. expr.</td>
</tr>
</tbody>
</table>
| 2     | $p \rightarrow \varepsilon q \rightarrow \varepsilon r \leftarrow b + bab$ | $p \rightarrow r^* q$
|       | $p \rightarrow \varepsilon q \rightarrow \varepsilon r$ | $p \rightarrow \varepsilon q$ |
| 3     | $p \rightarrow \varepsilon q \rightarrow \varepsilon r \leftarrow bab$ | $p \rightarrow r_1 + r_2 q$
|       | $p \rightarrow \varepsilon q$ | $p \rightarrow \varepsilon q$
| 4     | $p \rightarrow \varepsilon q \rightarrow \varepsilon r \leftarrow b b$ | $p \rightarrow r_1 r_2 q$
|       | $p \rightarrow \varepsilon q \rightarrow \varepsilon r$ | $p \rightarrow \varepsilon q$
|       | $p \rightarrow \varepsilon q$ | $p \rightarrow \varepsilon q$
<p>|       | $p \rightarrow \varepsilon q$ | $p \rightarrow \varepsilon q$ |</p>
<table>
<thead>
<tr>
<th>Iter.</th>
<th>Automaton obtained</th>
<th>Rule applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><img src="image1.png" alt="Diagram 1" /></td>
<td><img src="image2.png" alt="Rule 1" /></td>
</tr>
<tr>
<td>2</td>
<td><img src="image3.png" alt="Diagram 2" /></td>
<td><img src="image4.png" alt="Rule 2" /></td>
</tr>
<tr>
<td>3</td>
<td><img src="image5.png" alt="Diagram 3" /></td>
<td><img src="image6.png" alt="Rule 3" /></td>
</tr>
</tbody>
</table>
(d) States \( \{p\} \) and \( \{q,r\} \) have the exact same behaviours, so we can merge them. Indeed, both states are final and \( \delta(\{p\}, \sigma) = \delta(\{q,r\}, \sigma) \) for every \( \sigma \in \{a,b\} \). We obtain:

\[
\begin{array}{c}
\text{Rule applied} \\
\text{Add single initial and final states.}
\end{array}
\]
<table>
<thead>
<tr>
<th>3</th>
<th><img src="image1.png" alt="Diagram 3" /></th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td><img src="image2.png" alt="Diagram 4" /></td>
</tr>
<tr>
<td>5</td>
<td><img src="image3.png" alt="Diagram 5" /></td>
</tr>
<tr>
<td>6</td>
<td><img src="image4.png" alt="Diagram 6" /></td>
</tr>
<tr>
<td>7</td>
<td><img src="image5.png" alt="Diagram 7" /></td>
</tr>
</tbody>
</table>

| Extract regular expression from the unique transition. |

(f) Let us first show that $b(b + abb)^i = (b + bab)^ib$ for every $i \in \mathbb{N}$. We proceed by induction on $i$. If $i = 0$, then the claim trivially holds. Let $i > 0$. Assume the claims holds at $i - 1$. We have

\[
b(b + abb)^i = b(b + abb)^{i-1}(b + abb) = (b + bab)^{i-1}b(b + abb) = (b + bab)^{i-1}(bb + bab) = (b + bab)^{i-1}(b + bab)b = (b + bab)^i.b.\]

This implies that

\[
b(b + abb)^* = (b + bab)^*b. \quad (1)\]
We may now prove the equivalence of the two regular expressions:

\[
\varepsilon + b(b + ab)^*(\varepsilon + ab) = \varepsilon + (b + bab)^*b(\varepsilon + ab) = \varepsilon + (b + bab)^*(b + bab) = \varepsilon + (b + bab)^+ = (b + bab)^*.
\]

(by (1))

(by distributivity)

Solution 2.2
Suppose \(L\) is a finite language. We shall first show that \(L\) is a regular language, by providing an NFA for \(L\).

Let the alphabet of \(L\) be \(\Sigma\) and let \(L = \{w_1, \ldots, w_n\}\). For each \(w_i\), we will construct an NFA \(A_i\) that accepts only the word \(w_i\). If \(w_i = \varepsilon\) then the following NFA satisfies the required property:

\[
\text{Suppose } w_i \text{ is not the empty word. Let } w_i = a_1, a_2, \ldots, a_m. \text{ Then the following NFA satisfies the required property:}
\]

\[
\begin{array}{ccc}
& a_1 & \\
& & \\
\circ & \cdots & \circ \\
& & \\
\end{array}
\]

Hence we have an NFA \(A_i := (Q^i, \Sigma, \delta^i, Q_0^i, F^i)\) for each word \(w_i\). Now, we will construct an NFA \(A\) which recognizes the “union” \(L = \cup_{1 \leq i \leq n} w_i\). Let \(Q := \cup_{1 \leq i \leq n} Q^i, Q_0 := \cup_{1 \leq i \leq n} Q_0^i, F := \cup_{1 \leq i \leq n} F^i\). Further, let \(\delta : Q \times \Sigma \rightarrow 2^Q\) be the function given by \(\delta(q, a) = \delta^i(q, a)\) for every \(q \in Q^i\) and let \(A := (Q, \Sigma, \delta, Q_0, F)\). Then, \(A\) is an NFA which recognizes the language \(L\).

Hence, we have shown that if \(L\) is a finite language, then it is regular. Hence, there must be a DFA \(B = (Q, \Sigma, \delta, Q_0, F)\) such that \(B\) recognizes the language \(L\). Consider the DFA \(B^- = (Q, \Sigma, \delta, Q_0, Q \setminus F)\) obtained from \(B\) by “swapping” the final and non-final states of \(B\). By construction, \(B^-\) accepts a word if and only if it is rejected by \(B\) and hence \(B^-\) recognizes the complement of the language \(L\).

Solution 2.3
Let \(r := (((b + c)^*a + c^*) + (bc^*)^*)^*\). Let \(w = a_1, a_2, \ldots, a_n\) be any word over \(\Sigma\). We have to show that \(w \in L(r)\).

Let \(r' := ((b + c)^*a + c^*) + (bc^*)^*\). We will first show that each \(a_i \in L(r')\). Indeed, if \(a_i = a\), then \(a \in L((b + c)^*a) \subseteq L(r')\). If \(a_i = b\), then \(b \in L((bc^*)^*) \subseteq L(r')\). Finally, if \(a_i = c\) then \(c \in L(c^*) \subseteq L(r')\). Hence, for each \(i\), the letter \(a_i \in L(r')\).

Notice that \(L(r) = (L(r'))^*\). Since each \(a_i \in L(r')\), it follows that \(w \in L(r)\). Hence, we have shown that any word over \(\Sigma\) is included in \(L(r)\), which is what we wanted to prove.

Solution 2.4
- The following is an NFA with \(2n + 2\) states which recognizes \(L\).
We shall now show that any NFA recognizing \( L \) must have at least \( 2n + 2 \) states.

Let \( A \) be any NFA recognizing \( L \).

For every \( a \in \Sigma \), let \( q^a_0, q^a_1, q^a_2, q^a_3 \) be an accepting run of the word \( aaa \) over the NFA \( A \). We claim that if \( a \neq b \), then \( q^a_i \neq q^b_j \) for any \( i, j \in \{1, 2\} \). Indeed if \( q^a_1 = q^b_1 \) (resp. \( q^a_2 = q^b_2 \)) then the word \( abb \) (resp. \( aab \)) has an accepting run given by \( q^a_0, q^a_1, q^b_2, q^b_3 \) (resp. \( q^a_0, q^a_1, q^b_3, q^b_4 \)). On the other hand, if \( q^a_1 = q^b_2 \) (resp. \( q^a_2 = q^b_1 \)) then the word \( ab \) (resp. \( aabb \)) has an accepting run given by \( q^a_0, q^a_1, q^a_3 \) (resp. \( q^a_0, q^a_1, q^a_2, q^b_3 \)). It then follows that the NFA \( A \) must have at least \( 2n \) states.

We now claim that for any \( a \in \Sigma \) and any \( i \in \{1, 2\} \), the state \( q^a_i \) cannot be an initial or a final state. Indeed, if \( q^a_1 \) (resp. \( q^a_2 \)) is a final state, then the word \( a \) (resp. \( aa \)) is accepted by \( A \). On the other hand, if \( q^a_1 \) (resp. \( q^a_2 \)) is an initial state, then the word \( aa \) (resp. \( a \)) is accepted by \( A \). Hence, there is at least one initial state and one final state of \( A \) which is not in the set \( \{q^a_i : i \in \{1, 2\}, a \in \Sigma\} \).

Notice that no initial state of \( A \) can be a final state, as otherwise \( A \) would accept \( \epsilon \). It follows that there are at least two states which are not in the set \( \{q^a_i : i \in \{1, 2\}, a \in \Sigma\} \). Hence, \( A \) has at least \( 2n + 2 \) states.