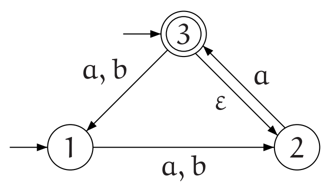
Tutorial 9

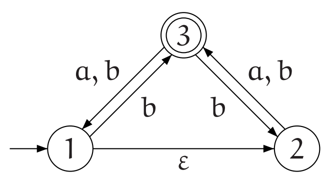
Exercise 1: Construct GNFA accepting languages L_1 and L_4:

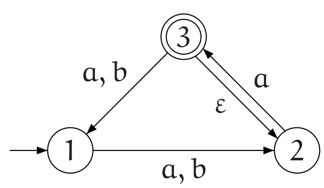
a) $L_1 = L_2 \cdot L_3$, where
$L_2 = \{w \in \{0,1\}^* \mid \text{every occurrence of } 00 \text{ in } w \text{ is immediately followed by } 1\}$
$L_3 = \{w \in \{0,1\}^* \mid |w|_1 \bmod 3 = 2\}$

b) $L_4 = \{w \in \{a,b\}^* \mid w \text{ is obtained from some word } w' \in L_5 \text{ by ommiting of one symbol}\}$, where L_5 is the language consisting of those words over alphabet $\{a,b\}$ that contain subword $abba$ and end with suffix abb.

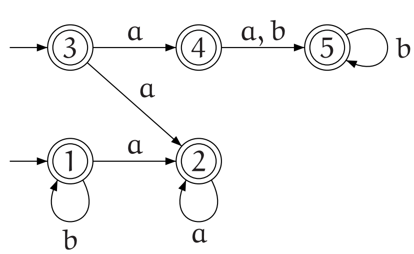
Exercise 2: Construct equivalent DFA for the given GNFA:

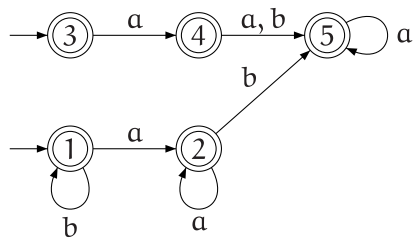
a)


b)


c)


Exercise 3: For each of the following automata find at least one word over alphabet $\{a,b\}$, which is not accepted by the given automaton.

a)


b) 

Exercise 4: For each of the following regular expressions, construct an equivalent finite automaton (it can be a GNFA):

a) $(0 + 11)^*01$
b) $(0 + 11)^*00^*1$
c) $(a + bab)^* + a^*(ba + \varepsilon)$

Exercise 5: Describe an algorithm that for a given NFA $\mathcal{A} = (Q, \Sigma, \delta, I, F)$ decides if:

a) $\mathcal{L}(\mathcal{A}) = \emptyset$

b) $\mathcal{L}(\mathcal{A}) = \Sigma^*$

Exercise 6: Describe an algorithm that for given NFA $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, I_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, I_2, F_2)$ decides if $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$.

Exercise 7: Describe an algorithm that for given GNFA $\mathcal{A}$ constructs an equivalent NFA $\mathcal{A}'$ such that the sets of states of automata $\mathcal{A}$ and $\mathcal{A}'$ are the same.

***Exercise 8:** Consider an arbitrary alphabet Σ.
The ***Hamming distance*** $h(u, v)$ of a pair of words $u, v \in \Sigma^*$, such that $|u| = |v|$, is the number of positions in the words u, v where these two words differ. Formally, $h(u, v)$ can be defined as follows: $h(\varepsilon, \varepsilon) = 0$, and for all symbols $a, b \in \Sigma$ and words $u, v \in \Sigma^*$, such that $|u| = |v|$, we have

$$h(au, bv) = \begin{cases} h(u, v) & \text{if } a = b \\ 1 + h(u, v) & \text{if } a \neq b \end{cases}$$

For a language $L \subseteq \Sigma^*$ and each $k \geq 0$ we define the language $H_k(L)$ as

$$H_k(L) = \{w \in \Sigma^* \mid \exists w' \in L : |w| = |w'| \wedge h(w, w') \leq k\}.$$

Show that for each $k \geq 0$ holds that if a language L is regular then also language $H_k(L)$ is regular.