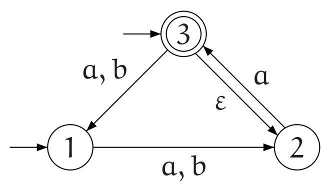
Tutorial 3

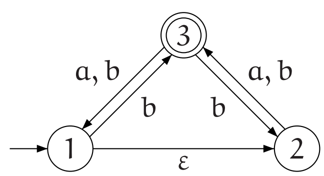
Exercise 1: Construct GNFA accepting languages L_1 and L_4:

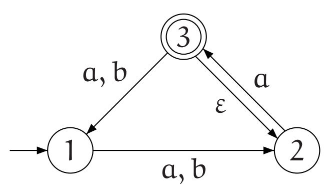
a) $L_1 = L_2 \cdot L_3$, where
$L_2 = \{w \in \{0,1\}^* \mid \text{every occurrence of } 00 \text{ in } w \text{ is immediately followed by } 1\}$
$L_3 = \{w \in \{0,1\}^* \mid |w|_1 \bmod 3 = 2\}$

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.

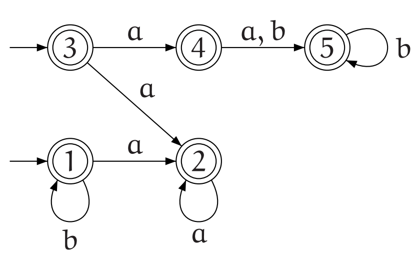
Exercise 2: Construct equivalent DFA for the given GNFA:

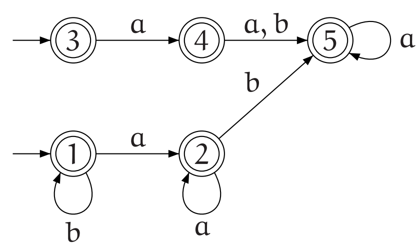
a)


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: Write regular expressions for the following languages:

a) The language $\{ab, ba, abb, bab, abbb, babb\}$
b) The language over alphabet $\{a, b, c\}$ containing exactly those words that contain subword abb.
c) The language over alphabet $\{a, b, c\}$ containing exactly those words that start with prefix bca or end with suffix $ccab$.
d) The language $\{w \in \{0, 1\}^* \mid |w|_0 \bmod 2 = 0\}$.
e) The language $\{w \in \{0, 1\}^* \mid |w|_0 \bmod 3 = 1\}$.
f) The language $\{w \in \{0, 1\}^* \mid w$ contains subwords 010 and $111\}$
g) The language $\{w \in \{a, b\}^* \mid w$ contains subword bab or $|w|_b \leq 3\}$
h) The language $\{w \in \{a, b\}^* \mid w$ contains subword bab and $|w|_b \leq 3\}$
i) The language of all words over $\{a, b, c\}$ that contain no two consecutive a's.

Exercise 5: Let us have two languages K and L described by the regular expressions

$$L_1 = [\,0^*1^*0^*1^*0^*\,], \qquad L_2 = [\,(01 + 10)^*\,].$$

a) What are the shortest and the longest words in the intersection $L_1 \cap L_2$?

b) Why none of the languages L_1 and L_2 is a subset of the other?

c) What is the shortest word that does not belong to the union $L_1 \cup L_2$? Is it unambiguous?

Exercise 6: Let us say that we would like to devise a syntax for representation of simple arithmetic expressions by words over alphabet

$$\Sigma = \{\texttt{A}, \texttt{B}, \ldots, \texttt{Z}, \texttt{a}, \texttt{b}, \ldots, \texttt{z}, \texttt{0}, \texttt{1}, \ldots, \texttt{9}, \texttt{.}, \texttt{+}, \texttt{-}, \texttt{*}, \texttt{/}, \texttt{(}, \texttt{)}\}.$$

a) Propose how identifiers will look like, and deribe them using a regular expression.

b) Propose how number constants will look like, and describe them using a regular expression.

Remark: Allow the number constants that would represent integers, e.g., `129` or `0`, and also floating-point number constants, e.g., `3.14`, `-1e10`, or `4.2E-23`. Consider also the possibility of representing number constants in other number systems except the decimal number system (e.g., hexadecimal, octal, binary).

Exercise 7: 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 8: Describe an algorithm that for a given NFA $A = (Q, \Sigma, \delta, I, F)$ decides if:

a) $L(A) = \emptyset$

b) $L(A) = \Sigma^*$

Exercise 9: Describe an algorithm that for given NFA $A_1 = (Q_1, \Sigma, \delta_1, I_1, F_1)$ and $A_2 = (Q_2, \Sigma, \delta_2, I_2, F_2)$ decides if $L(A_1) = L(A_2)$.

Exercise 10: Describe an algorithm that for given GNFA A constructs an equivalent NFA A' such that the sets of states of automata A and A' are the same.

***Exercise 11:** 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

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

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