Tutorial 2

Exercise 1: Write regular expressions for the following languages:

a) The language $\{ab, ba, abb, bab, abbb, babb\}$

Solution: $ab + ba + abb + bab + abbb + babb$ or $(ab + ba)(\varepsilon + b + bb)$

b) The language over alphabet $\{a, b, c\}$ containing exactly those words that contain subword abb.

Solution: $(a + b + c)^*abb(a + b + c)^*$

c) The language over alphabet $\{a, b, c\}$ containing exactly those words that start with prefix bca or end with suffix $ccab$.

Solution: $bca(a + b + c)^* + (a + b + c)^*ccab$

d) The language $\{w \in \{0, 1\}^* \mid |w|_0 \bmod 2 = 0\}$.

Solution: $1^*(01^*01^*)^*$

e) The language $\{w \in \{0, 1\}^* \mid |w|_0 \bmod 3 = 1\}$.

Solution: $1^*01^*(01^*01^*01^*)^*$

f) The language $\{w \in \{0, 1\}^* \mid w$ contains subwords 010 and $111\}$

Solution: $(0 + 1)^*010(0 + 1)^*111(0 + 1)^* + (0 + 1)^*111(0 + 1)^*010(0 + 1)^*$

g) The language $\{w \in \{a, b\}^* \mid w$ contains subword bab or $|w|_b \leq 3\}$

Solution: $(a + b)^*bab(a + b)^* + a^*(ba^* + \varepsilon)(ba^* + \varepsilon)(ba^* + \varepsilon)$

h) The language $\{w \in \{a, b\}^* \mid w$ contains subword bab and $|w|_b \leq 3\}$

Solution: $a^*ba^*baba^* + a^*baba^*ba^* + a^*baba^*$ or $(\varepsilon + a^*b)a^*baba^* + a^*baba^*ba^*$

i) The language of all words over $\{a, b, c\}$ that contain no two consecutive a's.

Solution: $\big((b + c + a(b + c)\big)^*(\varepsilon + a)$

Exercise 2: Let us have two languages L_1 and L_2 described by the regular expressions

$$L_1 = \mathcal{L}(\, 0^*1^*0^*1^*0^* \,), \qquad L_2 = \mathcal{L}(\, (01 + 10)^* \,).$$

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

Solution: The shortest words is ε and the longest 01100110, since the language L_2 does not contain any word where the same symbol would be repeated more than twice.

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

Solution: Because $1 \in L_1 - L_2$ and $010101 \in L_2 - L_1$.

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

Solution: 10101, it is unambiguous.

Exercise 3: 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.

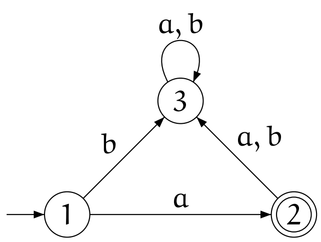
b) Propose how number constants will look like, and describe them using a regular expression.

   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 4: For each of the following languages, construct a DFA accepting the given language. Represent the constructed automata by graphs and tables.

a) $L_1 = \{w \in \{a, b\}^* \mid w = a\}$

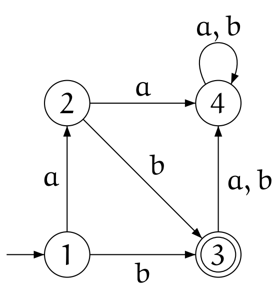
Solution:



	a	b
→ 1	2	3
← 2	3	3
3	3	3

b) $L_2 = \{b, ab\}$

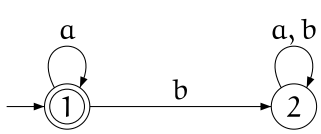
Solution:



	a	b
→ 1	2	3
2	4	3
← 3	4	4
4	4	4

c) $L_3 = \{w \in \{a, b\}^* \mid \exists n \in \mathbb{N} : w = a^n\}$

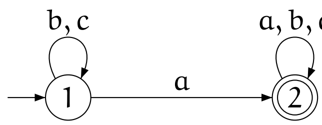
Solution:



	a	b
↔ 1	1	2
2	2	2

d) $L_4 = \{w \in \{a, b, c\}^* \mid |w|_a \geq 1\}$

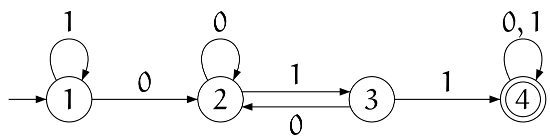
Solution:



	a	b	c
→ 1	2	1	1
← 2	2	2	2

e) $L_5 = \{w \in \{0, 1\}^* \mid w$ contains subword $011\}$

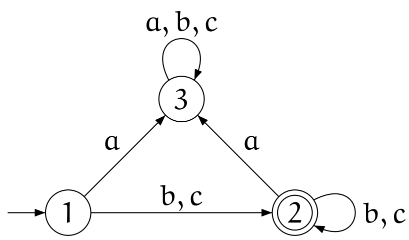
Solution:



	0	1
→ 1	2	1
2	2	3
3	2	4
← 4	4	4

f) $L_6 = \{w \in \{a, b, c\}^* \mid |w| > 0 \ \wedge \ |w|_a = 0\}$

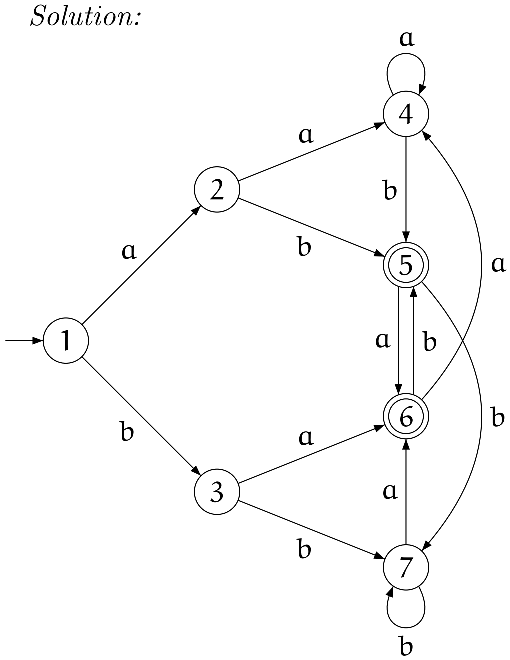
Solution:



	a	b	c
→ 1	3	2	2
← 2	3	2	2
3	3	3	3

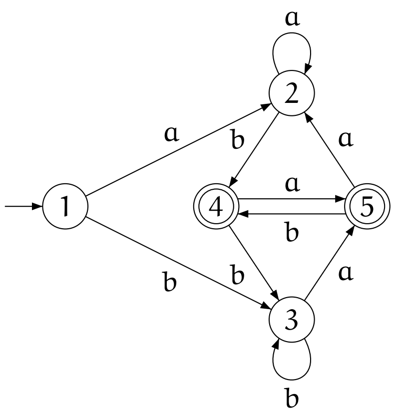
g) $L_7 = \{w \in \{a, b\}^* \mid |w| \geq 2$ and the last two symbols of w are not the same$\}$

Solution:



	a	b
→ 1	2	3
2	4	5
3	6	7
4	4	5
← 5	6	7
← 6	4	5
7	6	7

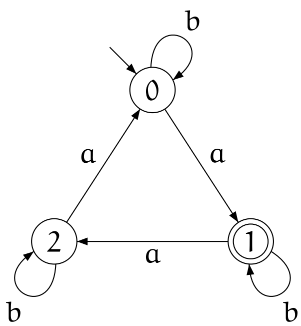
Alternative solution:



	a	b
$\to 1$	2	3
2	2	4
3	5	3
$\leftarrow 4$	5	3
$\leftarrow 5$	2	4

h) $L_8 = \{w \in \{a, b\}^* \mid |w|_a \bmod 3 = 1\}$

Solution:

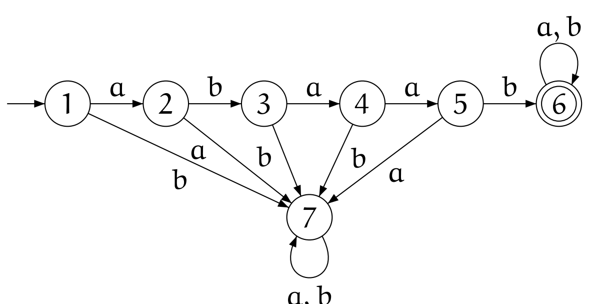


	a	b
$\to 0$	1	0
$\leftarrow 1$	2	1
2	0	2

Exercise 5: Construct DFA accepting words beginning with $abaab$, ending with $abaab$, and containing $abaab$, i.e., construct deterministic finite automata accepting the following three languages:

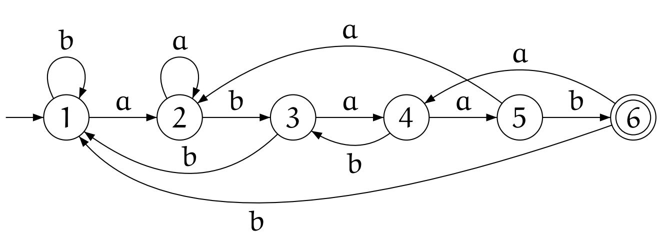
a) $L_1 = \{abaabw \mid w \in \{a, b\}^*\}$

Solution:



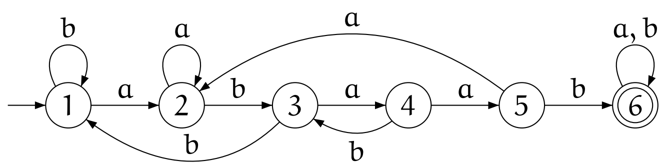
b) $L_2 = \{wabaab \mid w \in \{a, b\}^*\}$

Solution:



c) $L_3 = \{w_1 abaab w_2 \mid w_1, w_2 \in \{a, b\}^*\}$

Solution:



Exercise 6: Describe how to find out for a given DFA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ if:

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

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

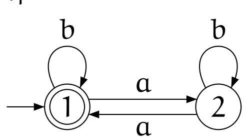
Solution: It is sufficient to compute the set of states that are reachable from q_0. We can use for example breadth-first search for this.

It holds that $\mathcal{L}(\mathcal{A}) = \emptyset$ iff none of reachable states is accepting, and $\mathcal{L}(\mathcal{A}) = \Sigma^*$ holds iff every reachable state is accepting.

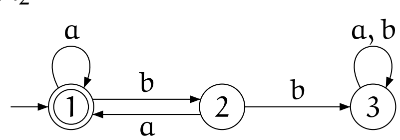
Exercise 7: Construct DFA $\mathcal{A}_1, \mathcal{A}_2$ such that:

$$\mathcal{L}(\mathcal{A}_1) = \{w \in \{a, b\}^* \mid |w|_a \bmod 2 = 0\}$$
$$\mathcal{L}(\mathcal{A}_2) = \{w \in \{a, b\}^* \mid \text{every occurence of symbol } b \text{ in } w \text{ is followed with symbol } a\}$$

Solution: $\mathcal{A}_1$:



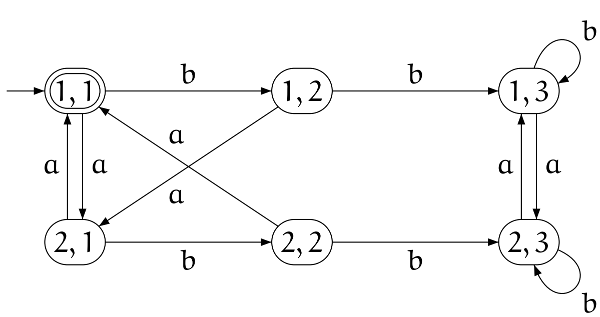
$\mathcal{A}_2$:



Using automata $\mathcal{A}_1, \mathcal{A}_2$, construct DFA accepting the following languages:

a) $L_1 = \{w \in \{a, b\}^* \mid |w|_a \bmod 2 = 0$ and every occurence of symbol b in w is followed with symbol $a\}$

Solution:



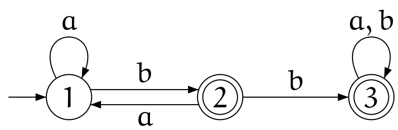
b) $L_2 = \{w \in \{a, b\}^* \mid |w|_a \bmod 2 = 0$ or every occurrence of symbol b in w is followed with symbol $a\}$

Solution: The same automaton as in (a) but with the set of accepting states

$$F = \{(1,1),\ (1,2),\ (1,3),\ (2,1)\}$$

c) $L_3 = \{w \in \{a, b\}^* \mid$ some occurrence of symbol b in w is not followed with symbol $a\}$

Solution:



d) $L_4 = \{w \in \{a, b\}^* \mid |w|_a \bmod 2 = 0$ and some occurrence of symbol b in w is not followed with symbol $a\}$

Solution: The same automaton as in (a) but with the set of accepting states

$$F = \{(1,2),\ (1,3)\}$$

e) $L_5 = \{w \in \{a, b\}^* \mid$ if $|w|_a \bmod 2 = 0$ then every occurrence of symbol b in w is followed with symbol $a\}$

Solution: The same automaton as in (a) but with the set of accepting states

$$F = \{(1,1),\ (2,1),\ (2,2),\ (2,3)\}$$

f) $L_6 = \{w \in \{a, b\}^* \mid |w|_a \bmod 2 = 0$ iff every occurrence of symbol b in w is followed with symbol $a\}$

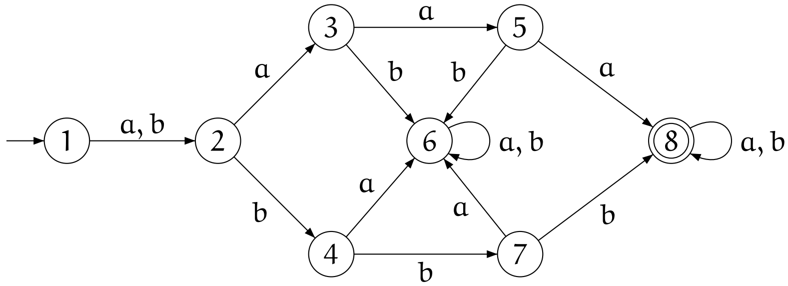
Solution: The same automaton as in (a) but with the set of accepting states

$$F = \{(1,1),\ (2,2),\ (2,3)\}$$

Exercise 8: For each of the following languages, construct a DFA accepting the given language. Represent the constructed automata by graphs and tables.

a) $L_1 = \{w \in \{a, b\}^* \mid |w| \geq 4$ and the second, third, and fourth symbol of w are the same$\}$

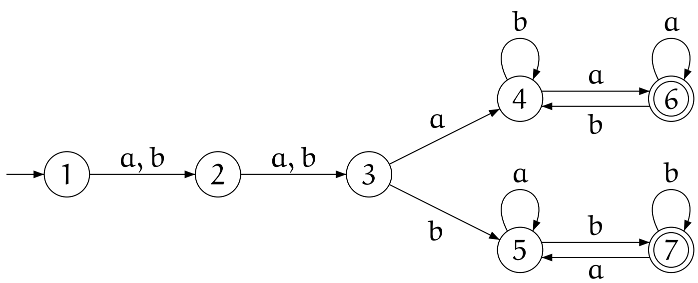
Solution:



	a	b
→ 1	2	2
2	3	4
3	5	6
4	6	7
5	8	6
6	6	6
7	6	8
(8)	8	8

b) $L_2 = \{w \in \{a, b\}^* \mid |w| \geq 4$ and the third symbol and the last symbol of w are the same$\}$

Solution:

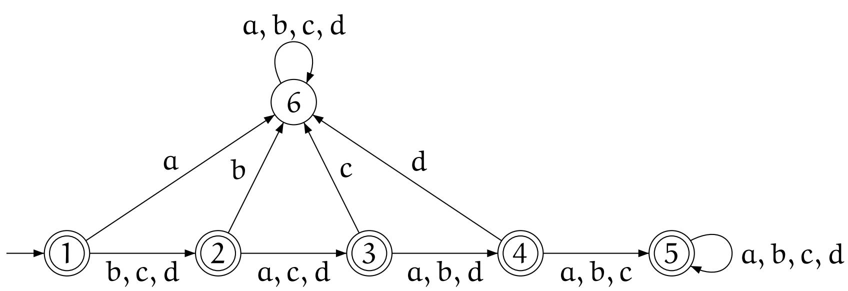


	a	b
→ 1	2	2
2	3	3
3	4	5
4	6	4
5	5	7
(6)	6	4
(7)	5	7

c) $L_3 = \{ w \in \{a, b, c, d\}^* \mid w$ does not start with a, the second symbol is not b, the third symbol is not c, and the fourth symbol is not d$\}$

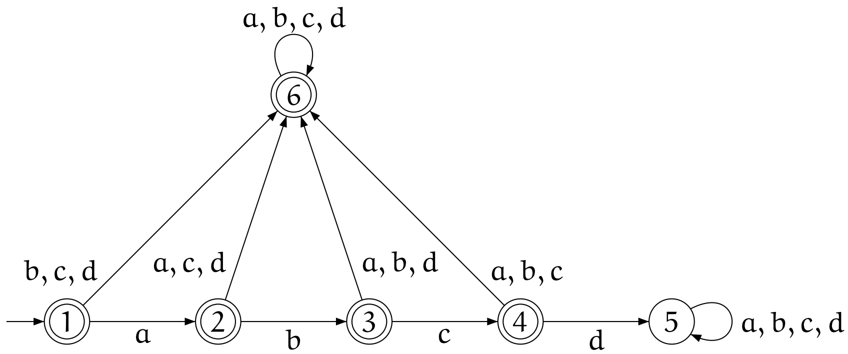
Remark: This language includes also those words w where $|w| < 4$.

Solution:



d) $L_4 = \{ w \in \{a, b, c, d\}^* \mid w$ does not start with a or the second symbol is not b or the third symbol is not c or the fourth symbol is not d$\}$

Solution:



Exercise 9: Desribe how to find out for given DFA $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ and $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ if $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$.

Solution: One of the possibilities is to use the fact that for arbitrary languages L_1, L_2 we have $L_1 = L_2$ iff

$$(L_1 \cap \overline{L_2}) \,\cup\, (\overline{L_1} \cap L_2) = \emptyset\,.$$

So it is sufficient to construct a DFA $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A}) = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$, where $L_1 = \mathcal{L}(\mathcal{A}_1)$ and $L_2 = \mathcal{L}(\mathcal{A}_2)$, and then to determine whether $\mathcal{L}(\mathcal{A}) = \emptyset$, for which we can use the approach from Exercise 6.

Another possible approach (which is basically a variant of the previous one) can be based on a construction similar as in the case of constructions for the intersection of the union (i.e., to construct an automaton with set of states $Q_1 \times Q_2$that simulates computations of automata $\mathcal{A}_1$ and $\mathcal{A}_2$ in parallel). For this automaton, it is sufficient to find out wheher there is some reachable state from the set

$$(F_1 \times (Q_2 - F_2)) \,\cup\, ((Q_1 - F_1) \times F_2),$$

i.e., a state corresponding to a situation where one of automata $\mathcal{A}_1, \mathcal{A}_2$ accepts the given word, and the other does not. If there is such reachable state, then $\mathcal{L}(\mathcal{A}_1) \neq \mathcal{L}(\mathcal{A}_2)$, otherwise $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$.

Remark: There are other possible approaches how this problem can be solved. The most efficient algorithms are based on a construction of a decomposition of the set of states into classes of equivalent states. We will not discuss these approaches in this introductory course.