Cvičení 2

Příklad 1: Napište regulární výrazy pro následující jazyky:

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

Řešení: $ab + ba + abb + bab + abbb + babb$ nebo $(ab + ba)(\varepsilon + b + bb)$

b) Jazyk nad abecedou $\{a, b, c\}$ obsahující právě ta slova, která obsahují podslovo abb.

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

c) Jazyk nad abecedou $\{a, b, c\}$ obsahující právě ta slova, která začínají prefixem bca nebo končí sufixem $ccab$.

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

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

Řešení: $1^*(01^*01^*)^*$

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

Řešení: $1^*01^*(01^*01^*01^*)^*$

f) Jazyk $\{w \in \{0, 1\}^* \mid w$ obsahuje podslova 010 a $111\}$

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

g) Jazyk $\{w \in \{a, b\}^* \mid w$ obsahuje podslovo bab nebo $|w|_b \le 3\}$

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

h) Jazyk $\{w \in \{a, b\}^* \mid w$ obsahuje podslovo bab a $|w|_b \le 3\}$

Řešení: $a^*ba^*baba^* + a^*baba^*ba^* + a^*baba^*$ nebo $(\varepsilon + a^*b)a^*baba^* + a^*baba^*ba^*$

i) Jazyk všech slov nad abecedou $\{a, b, c\}$, ve kterých se nikde nevyskytují dva znaky a hned za sebou.

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

Příklad 2: Mějme dva jazyky L_1 a L_2 popsané regulárními výrazy

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

a) Jaké je nejkratší a nejdelší slovo v průniku $L_1 \cap L_2$?

Řešení: Nejkratší je ε a nejdelší 01100110, neboť jazyk L_2 nedovoluje opakovat stejný znak za sebou více než dvakrát.

b) Proč žádný z těchto jazyků L_1 a L_2 není podmnožinou toho druhého?

Řešení: Protože $1 \in L_1 - L_2$ a $010101 \in L_2 - L_1$.

c) Jaké je nejkratší slovo, které nepatří do sjednocení $L_1 \cup L_2$? Je to jednoznačné?

Řešení: `10101`, jednoznačně.

Příklad 3: Řekněme, že bychom chtěli navrhnout syntaxi pro zápis jednoduchých aritmetických výrazů pomocí slov nad abecedou

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

a) Navrhněte, jak budou vypadat identifikátory, a popište to pomocí regulárního výrazu.

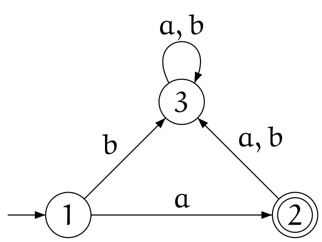
b) Navrhněte, jak budou vypadat číselné konstanty, a popište to pomocí regulárního výrazu.

Poznámka: Při popisu číselných konstant umožněte jak celočíselné konstanty, např. `129` nebo `0`, tak neceločíselné konstanty, např. `3.14`, `-1e10` nebo `4.2E-23`. Zvažte i možnost zápisu číselných konstant v dalších číselných soustavách kromě desítkové (např. hexadecimální, oktalové, binární).

Příklad 4: Pro každý z následujících jazyků sestrojte determinitický konečný automat (DKA), který ho rozpoznává. Vytvořené automaty znázorněte grafem a zapište tabulkou.

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

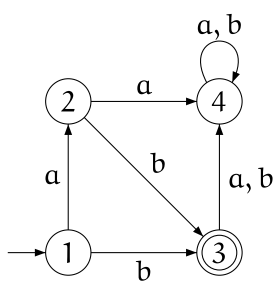
Řešení:



	a	b
$\rightarrow$ 1	2	3
$\leftarrow$ 2	3	3
3	3	3

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

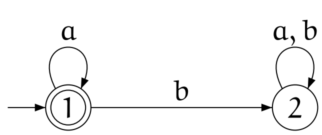
Řešení:



	a	b
$\rightarrow$ 1	2	3
2	4	3
$\leftarrow$ 3	4	4
4	4	4

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

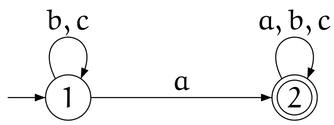
Řešení:



	a	b
$\leftrightarrow$ 1	1	2
2	2	2

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

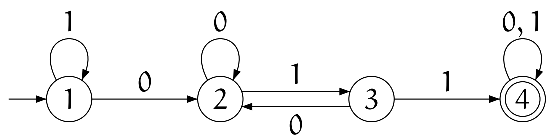
Řešení:



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

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

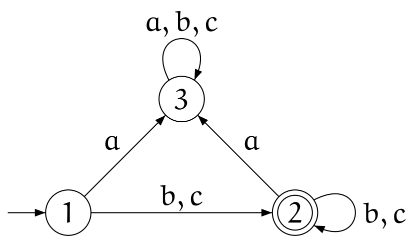
Řešení:



	0	1
$\to 1$	2	1
2	2	3
3	2	4
$\leftarrow 4$	4	4

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

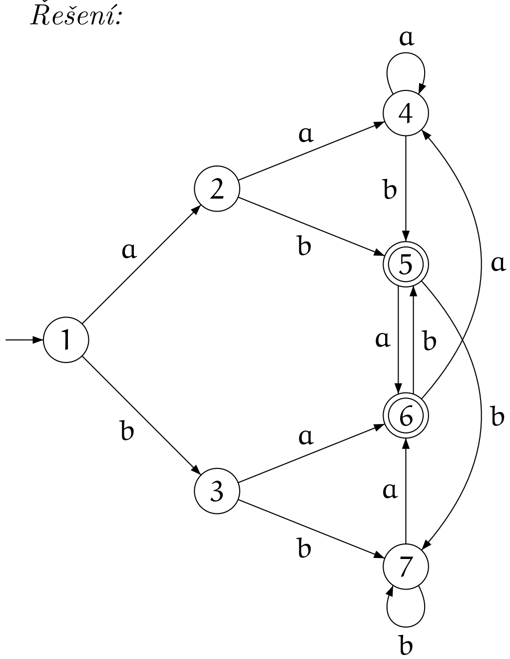
Řešení:



	a	b	c
$\to 1$	3	2	2
$\leftarrow 2$	3	2	2
3	3	3	3

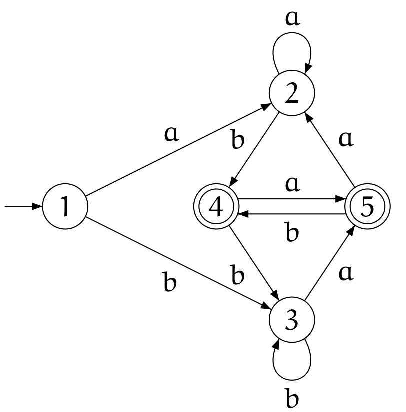
g) $L_7 = \{w \in \{a, b\}^* \mid |w| \geq 2$ a poslední dva symboly slova w nejsou stejné$\}$

Řešení:



	a	b
$\to 1$	2	3
2	4	5
3	6	7
4	4	5
$\leftarrow 5$	6	7
$\leftarrow 6$	4	5
7	6	7

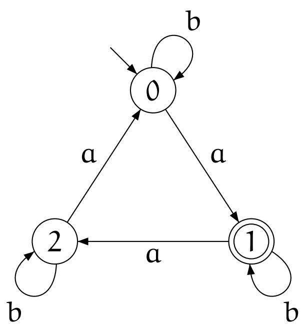
Alternativní řešení:



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

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

Řešení:

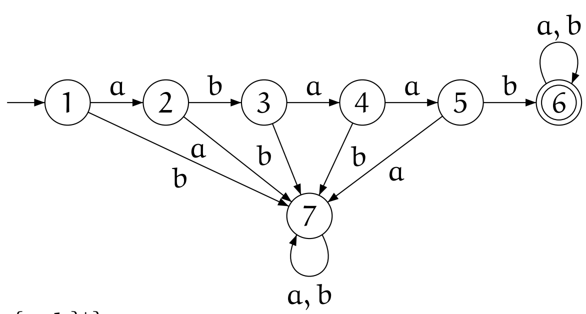


	a	b
$\rightarrow 0$	1	0
$\leftarrow 1$	2	1
2	0	2

Příklad 5: Sestrojte DKA přijímající slova začínající $abaab$, končící $abaab$ a obsahující $abaab$, tj. sestrojte DKA rozpoznávající následující tři jazyky:

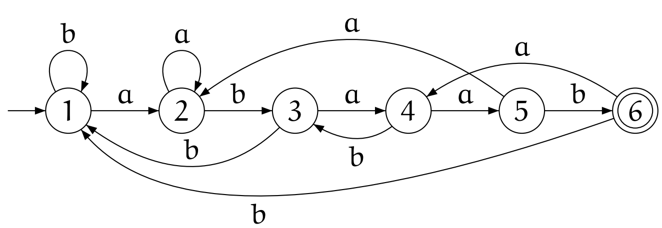
a) $L_1 = \{abaabw \mid w \in \{a, b\}^*\}$

Řešení:



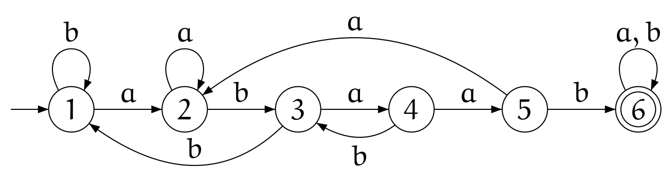
b) $L_2 = \{wabaab \mid w \in \{a, b\}^*\}$

Řešení:



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

Řešení:



Příklad 6: Navrhněte obecný postup, jak pro daný DKA $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ zjistit, zda:

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

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

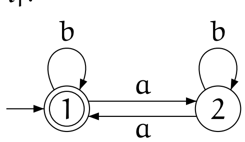
Řešení: Stačí zjistit množinu všech stavů dosažitelných z q_0. Pro toto zjištění můžeme použít například algoritmus prohledávání do šířky (breadth-first search).

Platí, že $\mathcal{L}(\mathcal{A}) = \emptyset$ právě tehdy, když žádný z dosažitelných stavů není přijímající, a $\mathcal{L}(\mathcal{A}) = \Sigma^*$ právě tehdy, když všechny dosažitelné stavy jsou přijímající.

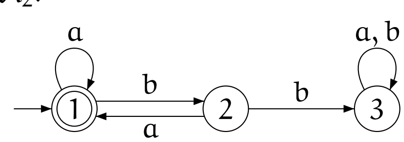
Příklad 7: Navrněte DKA $\mathcal{A}_1, \mathcal{A}_2$ takové, že:

$$\mathcal{L}(\mathcal{A}_1) = \{w \in \{a, b\}^* \mid |w|_a \bmod 2 = 0\}$$
$$\mathcal{L}(\mathcal{A}_2) = \{w \in \{a, b\}^* \mid \text{ve } w \text{ je každý výskyt symbolu } b \text{ následován symbolem } a\}$$

Řešení: $\mathcal{A}_1$:



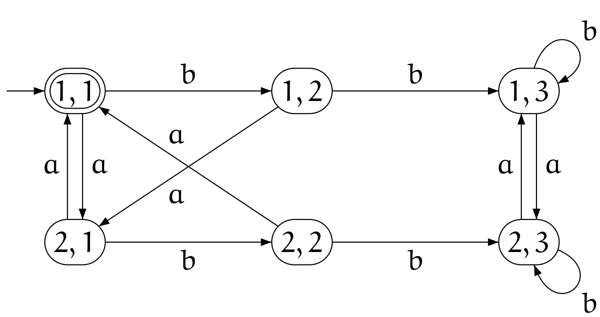
$\mathcal{A}_2$:



S využitím automatů $\mathcal{A}_1, \mathcal{A}_2$ sestrojte DKA rozpoznávající následující jazyky:

a) $L_1 = \{w \in \{a, b\}^* \mid |w|_a \bmod 2 = 0$ a ve w je každý výskyt symbolu b následován symbolem $a\}$

Řešení:



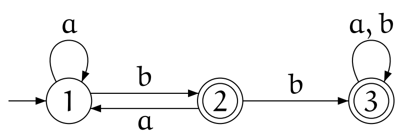
b) $L_2 = \{w \in \{a,b\}^* \mid |w|_a \bmod 2 = 0$ nebo je ve w každý výskyt symbolu b následován symbolem $a\}$

Řešení: Stejný automat jako v (a), akorát, že množina přijímajících stavů je

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

c) $L_3 = \{w \in \{a,b\}^* \mid$ ve w není nějaký výskyt symbolu b následován symbolem $a\}$

Řešení:



d) $L_4 = \{w \in \{a,b\}^* \mid |w|_a \bmod 2 = 0$ a ve w není nějaký výskyt symbolu b následován symbolem $a\}$

Řešení: Stejný automat jako v (a), akorát, že množina přijímajících stavů je

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

e) $L_5 = \{w \in \{a,b\}^* \mid$ jestliže $|w|_a \bmod 2 = 0$, pak je ve w každý výskyt symbolu b následován symbolem $a\}$

Řešení: Stejný automat jako v (a), akorát, že množina přijímajících stavů je

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

f) $L_6 = \{w \in \{a,b\}^* \mid |w|_a \bmod 2 = 0$ právě, když je ve w každý výskyt symbolu b následován symbolem $a\}$

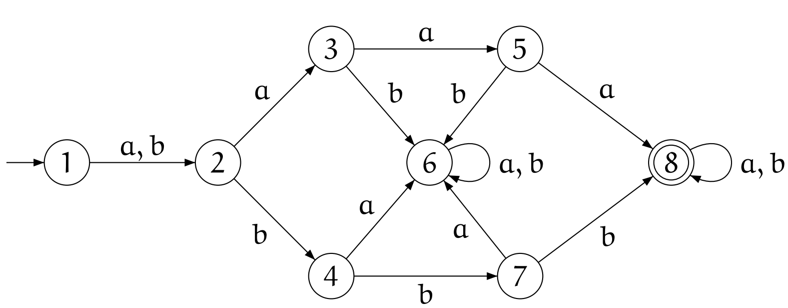
Řešení: Stejný automat jako v (a), akorát, že množina přijímajících stavů je

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

Příklad 8: Pro každý z následujících jazyků sestrojte DKA, který ho rozpoznává. Vytvořené automaty znázorněte grafem a zapište tabulkou.

a) $L_1 = \{w \in \{a,b\}^* \mid |w| \geq 4$ a druhý, třetí a čtvrtý symbol slova w jsou stejné$\}$

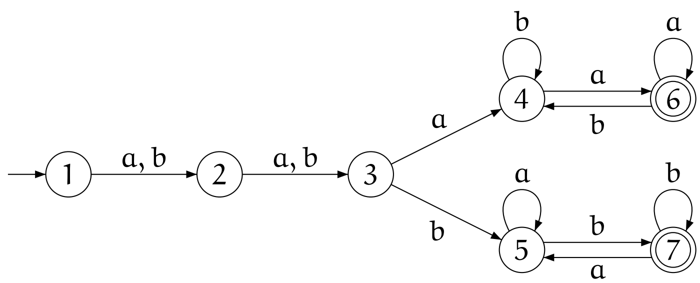
Řešení:



	a	b
$\to 1$	2	2
2	3	4
3	5	6
4	6	7
5	8	6
6	6	6
7	6	8
8	8	8

b) $L_2 = \{w \in \{a, b\}^* \mid |w| \geq 4$ a třetí a poslední symbol slova w jsou stejné$\}$

Řešení:

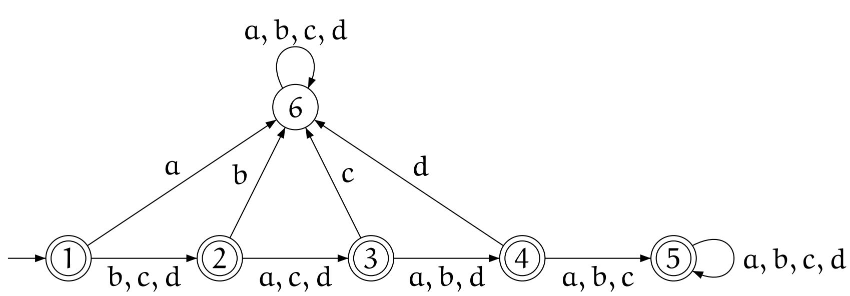


	a	b
$\to 1$	2	2
2	3	3
3	4	5
4	6	4
5	5	7
6	6	4
7	5	7

c) $L_3 = \{\, w \in \{a, b, c, d\}^* \mid w$ nezačíná a, druhý znak není b, třetí znak není c a čtvrtý znak není $d \,\}$

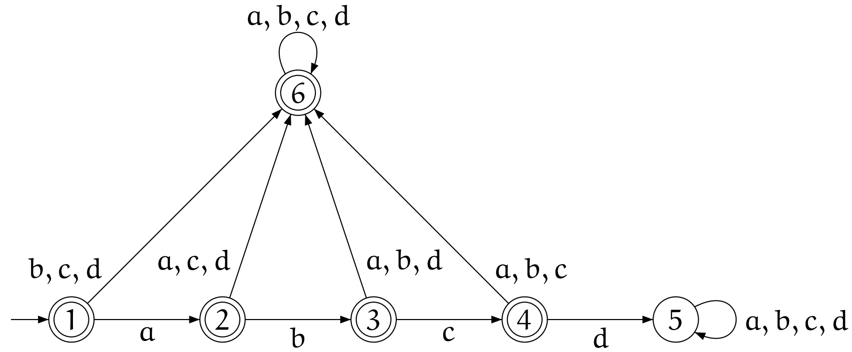
Poznámka: Tento jazyk zahrnuje i ta slova w, kde $|w| < 4$.

Řešení:



d) $L_4 = \{\, w \in \{a, b, c, d\}^* \mid w$ nezačíná a nebo druhý znak není b nebo třetí znak není c nebo čtvrtý znak není $d \,\}$

Řešení:



Příklad 9: Navrhněte obecný postup, jak pro dané DKA $\mathcal{A}_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ a $\mathcal{A}_2 = (Q_2, \Sigma, \delta_2, q_2, F_2)$ zjistit, zda $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$.

Řešení: Jednou z možností je využít toho, že pro libovolné jazyky L_1, L_2 platí $L_1 = L_2$ právě tehdy, když

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

Stačí tedy sestrojit DKA A takový, že $L(A) = (L_1 \cap \overline{L_2}) \cup (\overline{L_1} \cap L_2)$, kde $L_1 = L(A_1)$ a $L_2 = L(A_2)$, a zjistit zda $L(A) = \emptyset$, k čemuž můžeme použít postup z příkladu 6.

Jiná varianta je založená na konstrukci podobné jako u průniku nebo sjednocení (tj. automat s množinou stavů $Q_1 \times Q_2$, který simuluje činnost automatů A_1 a A_2 současně). U tohoto automatu stačí zjistit, zda je dosažitelný nějaký stav z množiny

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

tj. stav odpovídající situaci, kdy by jeden z automatů A_1, A_2 dané slovo přijímal a druhý ne. Pokud nějaký takový dosažitelný stav existuje, tak $L(A_1) \neq L(A_2)$, v opačném případě $L(A_1) = L(A_2)$.

Poznámka: Dá se vymyslet celá řada dalších různých postupů. Nejefektivnější algoritmy jsou založené na rozkladu na třídy ekvivalentních stavů. Tímto se však v UTI nezabýváme.