

# **Sbírka řešených příkladů**

**B4B01JAG**

29. ledna 2025

# Obsah

	Strana
<b>1 První cvičení</b>	<b>1</b>
1.1 Charakterizace jazyka . . . . .	1
1.2 Práce na konečném automatu . . . . .	2
1.3 Posuvný registr . . . . .	2
1.4 Stavové diagramy pro DFA . . . . .	3
<b>2 Druhé cvičení</b>	<b>4</b>
2.1 Návrh DFA dle jazyka . . . . .	4
2.2 Návrh DFA dle jazyka . . . . .	4
2.3 Návrh DFA dle jazyka . . . . .	4
2.4 Nerodova věta a Pumping lemma . . . . .	6
<b>3 Třetí cvičení</b>	<b>7</b>
3.1 Nerodova věta a Pumping lemma . . . . .	7
3.2 Nalezení slova rozlišujícího stavy . . . . .	7
3.3 Návrh a redukce DFA podle jazyka . . . . .	8
3.4 Srovnání dvou DFA automatů . . . . .	9
3.5 Návrh a redukce DFA podle jazyka . . . . .	11
<b>4 Čtvrté cvičení</b>	<b>12</b>
4.1 Sestrojení DFA z NFA . . . . .	12
4.2 Sestrojení DFA z NFA . . . . .	12
4.3 Návrh NFA dle sjednocení jazyků a redukované DFA . . . . .	13
4.4 Srovnání dvou NFA automatů . . . . .	14
4.5 Konstrukce DFA z $\varepsilon$ -NFA . . . . .	15
<b>5 Páté cvičení</b>	<b>16</b>
5.1 Důkaz obecné vlastnosti dvou libovolných jazyků . . . . .	16
5.2 Tvorba regulárního výrazu reprezentujícího jazyk . . . . .	16
5.3 Hledání slov splňující různé regulární výrazy . . . . .	17
5.4 Konstrukce redukovaného DFA k regulárnímu výrazu . . . . .	17
5.5 Tvorba DFA a regulárního výrazu dle jazyka . . . . .	20

<b>6</b>	<b>Šesté cvičení</b>	<b>21</b>
6.1	Návrh NFA a redukovaný DFA . . . . .	21
6.2	Návrh NFA a redukovaný DFA . . . . .	22
6.3	Důkaz minimálního počtu stavů DFA pro jazyk . . . . .	22
6.4	Pumping lemma pro doplněk . . . . .	23
6.5	Návrh DFA dle jazyka . . . . .	23
6.6	Návrh DFA součinovou konstrukcí . . . . .	23
6.7	Návrh DFA . . . . .	24
<b>7</b>	<b>Sedmé cvičení</b>	<b>25</b>
7.1	Konstrukce redukovaného DFA k regulárnímu výrazu . . . . .	25
7.2	Konstrukce redukovaného DFA k regulárnímu výrazu . . . . .	25
7.3	Konstrukce redukovaného DFA k regulárnímu výrazu . . . . .	26
7.4	Tvorba regulárního výrazu z DFA . . . . .	27
<b>8</b>	<b>Osmé cvičení</b>	<b>29</b>
8.1	Konstrukce regulární gramatiky k automatu . . . . .	29
8.2	Tvorba DFA ke gramatice 3. typu . . . . .	29
8.3	Práce s derivačním stromem bezkontextové gramatiky . . . . .	30
8.4	Tvorba derivačního stromu k bezkontextové gramatice . . . . .	30
8.5	Návrh bezkontextové gramatiky pro jazyk . . . . .	31
8.6	Tvorba nevypouštěcí gramatiky . . . . .	31
8.7	Konstrukce gramatiky 3. typu k automatu . . . . .	31
8.8	Návrh bezkontextové gramatiky . . . . .	32
<b>9</b>	<b>Deváté cvičení</b>	<b>33</b>
9.1	Návrh bezkontextové gramatiky . . . . .	33
9.2	Konstrukce nevypouštěcí gramatiky . . . . .	33
9.3	Redukce gramatiky . . . . .	34
9.4	Důkaz, že gramatika generuje alespoň jedno slovo . . . . .	34
9.5	Převod gramatiky do Chomského normálního tvaru . . . . .	35
9.6	Práce s derivačním stromem bezkontextové gramatiky . . . . .	35
<b>10</b>	<b>Desáté cvičení</b>	<b>37</b>
10.1	Algoritmus CYK . . . . .	37
10.2	Algoritmus CYK . . . . .	38
10.3	Bezkontextové Pumping lemma . . . . .	38
10.4	Důkaz generování slova matematickou indukcí . . . . .	39

10.5 Algoritmus CYK . . . . .	40
<b>11 Jedenácté cvičení</b>	<b>41</b>
11.1 Stavový diagram zásobníkového automatu, práce na automatu . . . . .	41
<b>12 Dvanácté cvičení</b>	<b>43</b>
12.1 Tvorba zásobníkového automatu podle jazyka . . . . .	43
12.2 Tvorba zásobníkového automatu podle jazyka . . . . .	44
12.3 Tvorba zásobníkového automatu podle jazyka . . . . .	44
12.4 Důkaz bezkontextovosti jazyka . . . . .	45
<b>13 Třinácté cvičení</b>	<b>46</b>
13.1 Tvorba a práce zásobníkového automatu podle jazyka . . . . .	46
13.2 Tvorba nevypouštěcí gramatiky . . . . .	46
13.3 Převod gramatiky do Greibachové normální formy . . . . .	47
13.4 Převod gramatiky do Greibachové normální formy . . . . .	48
<b>14 Čtrnácté cvičení</b>	<b>49</b>
14.1 Převod gramatiky do Greibachové normální formy . . . . .	49
14.2 Zkoušková ukázka práce na bezkontextové gramatice . . . . .	49

# 1 První cvičení

## 1.1 Charakterizace jazyka

Jazyk  $L$  nad abecedou  $\Sigma = \{a, b\}$  je dán induktivně

$$\begin{aligned}\varepsilon &\in L \\ u \in L &\implies aub \in L \\ u \in L &\implies bua \in L \\ u, v \in L &\implies uv \in L\end{aligned}$$

Charakterizujte slova jazyka  $L$ , tj. najděte vlastnost  $\mathcal{V}$  takovou, že  $L = \{u \mid \text{slovo } u \text{ má vlastnost } \mathcal{V}\}$ . Své tvrzení dokažte.

$$L_1 = \{w \mid w \in \{a, b\}^*, |w|_a = |w|_b\}.$$

Důkaz:

a)  $L \subseteq L_1$

1.  $|\varepsilon|_a = 0 = |\varepsilon|_b$
2.  $|u|_a = |u|_b \Rightarrow |aub|_a = |u|_a + 1 = |aub|_b = |u|_b + 1$

b)  $L_1 \subseteq L$

1.  $|\varepsilon|_a = |\varepsilon|_b = 0, \varepsilon \in L_1, \varepsilon \in L$ .
2. Každé slovo  $w \in L_1$  lze rozdělit na následující případy, které umožňují jeho postupné rozdělení až na prázdné slovo  $\varepsilon$ :

**Možnost 1:**  $w$  začíná  $a$  a končí  $b$ ,

**Možnost 2:**  $w$  začíná  $b$  a končí  $a$ ,

**Možnost 3:**  $w$  začíná a končí tím stejným písmenem ( $a$  nebo  $b$ ).

**Možnost 1:**  $w = bua$ . Rozdělíme slovo  $w$  na  $w = bua$ , kde  $u$  je prostřední část slova splňující  $|u|_a = |u|_b$ . Podle definice pravidel  $L$ , pokud  $u \in L$ , pak  $aub \in L$ .

**Možnost 2:**  $w = aub$ . Rozdělíme slovo  $w$  na  $w = aub$ , kde  $u$  je prostřední část slova splňující  $|u|_a = |u|_b$ . Podle definice pravidel  $L$ , pokud  $u \in L$ , pak  $bua \in L$ .

**Možnost 3:**  $w = axa, w = bxb$ . Předpokládejme, že  $w$  začíná i končí znakem  $a$ . Procházíme  $a$  zleva doprava a hledáme první  $b$ , kde počet znaků  $a$  od začátku do tohoto  $b$  je stejný jako počet znaků  $b$  (včetně tohoto  $b$ ). Toto  $b$  rozděluje  $w$  na dvě části:  $w = uv$ , kde  $u$  obsahuje první část  $w$  (od prvního znaku do tohoto  $b$  včetně), kde  $|u|_a = |u|_b$ , a  $v$  obsahuje druhou část slova  $w$ , kde  $|v|_a = |v|_b$ . Podle pravidel  $L$ , pokud  $u, v \in L$ , pak  $uv \in L$ .

Analogicky postupujeme pro slovo začínající i končící znakem  $b$ .

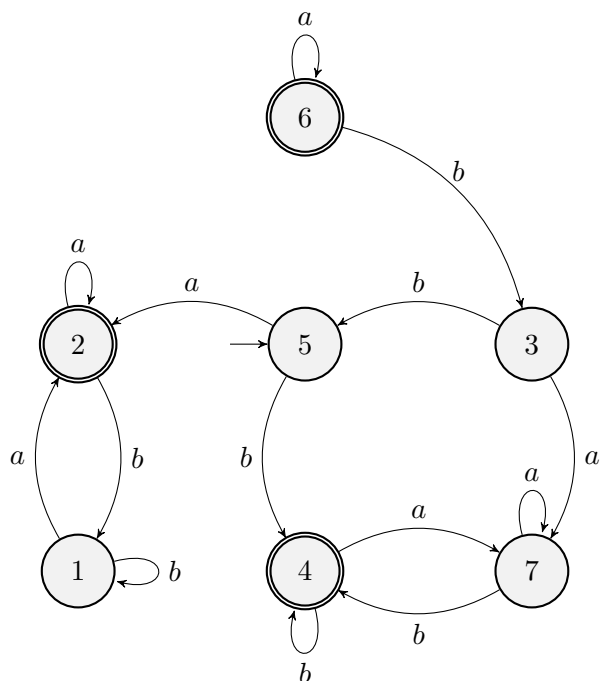
## 1.2 Práce na konečném automatu

Je dán konečný automat  $M$  tabulkou

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

1. Nakreslete stavový diagram automatu.
2. Simulujte krok po kroku výpočet automatu nad slovem  $bbaaab$ .
3. Z indukční definice odvoďte  $\delta^*(2, bab)$ .

1.

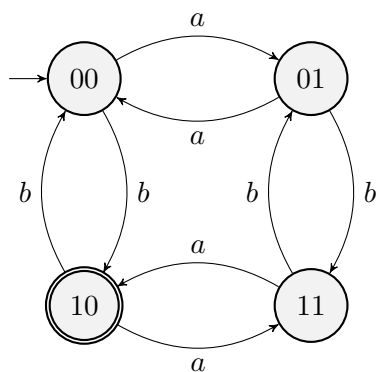


2.  $\rightarrow 5 - 4 - 4 - 7 - 7 - 7 - 4 \rightarrow$

3.  $\delta^*(2, bab) = \delta(\delta^*(2, ba), b) = \delta(\delta(\delta(2, b), a), b)$ .

## 1.3 Posuvný registr

Navrhněte automat modelující posuvný registr, který provádí celočíselné dělení 4 binárně zadaného čísla (číslo se čte od nejvyššího řádu). O jaký typ automatu se jedná?



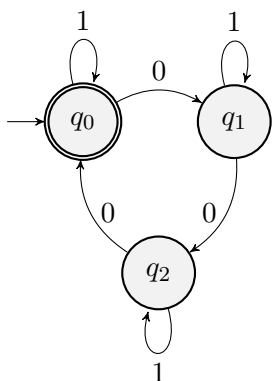
Jedná se o *Mealyho automat*

$$M = (Q, \Sigma, Y, \delta, q_o, \lambda).$$

## 1.4 Stavové diagramy pro DFA

Pro uvedené automaty nakreslete stavový diagram. Najděte vlastnost  $\mathcal{V}$ , která charakterizuje slova přijímaná daným automatem. Dokažte, že automat přijímá právě všechna slova s vlastností  $\mathcal{V}$ .

	0	1
$\leftrightarrow q_0$	$q_1$	$q_0$
$q_1$	$q_2$	$q_1$
$q_2$	$q_0$	$q_2$

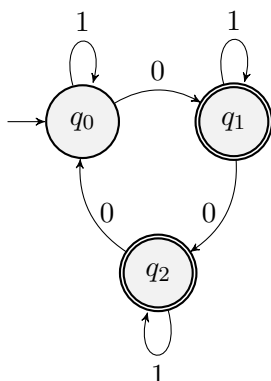


$w \in L$  iff  $|w|_0$  je dělitelný 3.

Invarianty:

- $q_0$ : in, out,  $|w|_0 = 3k$ ,
- $q_1$ :  $|w|_0 = 3k + 1$ ,
- $q_2$ :  $|w|_0 = 3k + 2$ ,  $k \in \mathbb{Z}$ .

	0	1
$\rightarrow q_0$	$q_1$	$q_0$
$\leftarrow q_1$	$q_2$	$q_1$
$\leftarrow q_2$	$q_0$	$q_2$

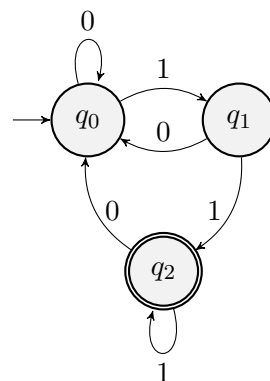


$w \in L$  iff  $|w|_0$  není dělitelný 3.

Invarianty:

- $q_0$ : in,  $|w|_0 = 3k$ ,
- $q_1$ : out,  $|w|_0 = 3k + 1$ ,
- $q_2$ : out,  $|w|_0 = 3k + 2$ ,  $k \in \mathbb{Z}$ .

	0	1
$\rightarrow q_0$	$q_0$	$q_1$
$q_1$	$q_0$	$q_2$
$\leftarrow q_2$	$q_0$	$q_2$



$w \in L$  iff  $w$  končí 11.

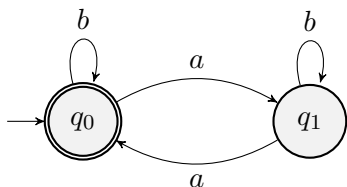
Invarianty:

- $q_0$ : in, končí 0,
- $q_1$ : nekončí 11, končí 1,
- $q_2$ : out, končí 11.

## 2 Druhé cvičení

### 2.1 Návrh DFA dle jazyka

Je dán jazyk  $L = \{w \mid \text{počet } |w|_a \text{ je sudý}\}$  nad abecedou  $\Sigma = \{a, b\}$ . Navrhněte konečný automat přijímající jazyk  $L$  a dokažte, že skutečně tento jazyk přijímá.

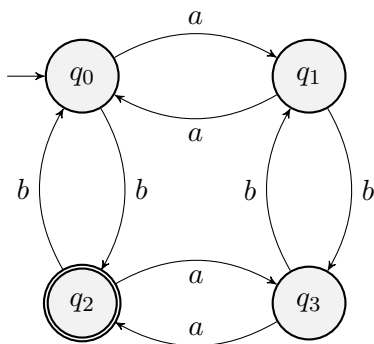


Důkaz pomocí invariantů:

- $q_0$ :  $|w|_a = 2k$ , **in**, **out**,
- $q_1$ :  $|w|_a = 2k + 1$ ,  $k \in \mathbb{Z}$ .

### 2.2 Návrh DFA dle jazyka

Je dán jazyk  $L = \{w \mid \text{počet } |w|_a \text{ je sudý a počet } |w|_b \text{ je lichý}\}$  nad abecedou  $\Sigma = \{a, b\}$ . Navrhněte konečný automat přijímající jazyk  $L$  a dokažte, že skutečně tento jazyk přijímá.



Důkaz pomocí invariantů:

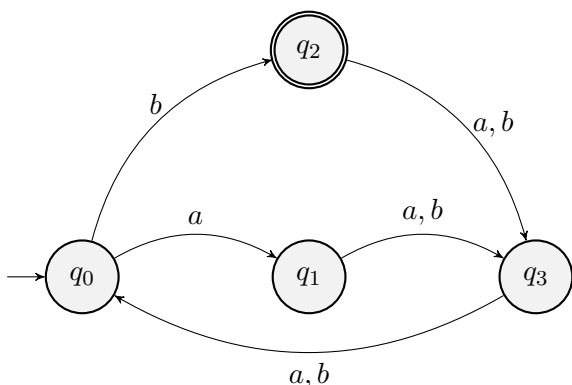
- $q_0$ :  $|w|_a = 2k$ ,  $|w|_b = 2k$ , **in**,
- $q_1$ :  $|w|_a = 2k + 1$ ,  $|w|_b = 2k$ ,
- $q_2$ :  $|w|_a = 2k$ ,  $|w|_b = 2k + 1$ , **out**,
- $q_3$ :  $|w|_a = 2k + 1$ ,  $|w|_b = 2k + 1$ ,  $k \in \mathbb{Z}$ .

### 2.3 Návrh DFA dle jazyka

Pro daný jazyk  $L$  navrhněte konečný automat, který tento jazyk přijímá. O automatu ukažte, že opravdu přijímá daný jazyk.

- $\Sigma = \{a, b\}$ , jazyk  $L$  obsahuje právě všechna slova, která končí  $b$  a mají délku  $3k + 1$ .
- $\Sigma = \{0, 1\}$ , jazyk  $L$  obsahuje právě všechna slova, která obsahují podslovo 0101.
- $\Sigma = \{0, 1\}$ , jazyk  $L$  obsahuje právě všechna slova, jejichž každé podslovo délky 3 obsahuje znak 0.

a)

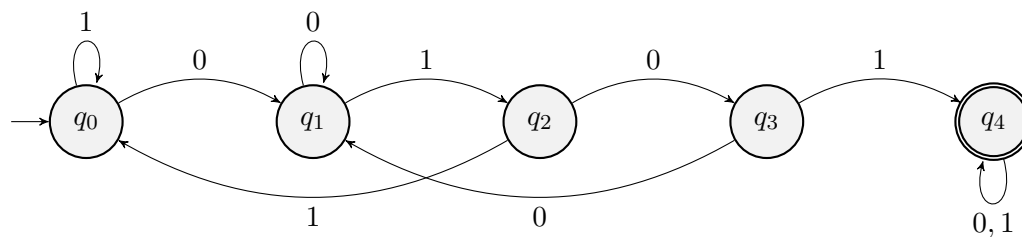


Důkaz pomocí invariantů:

- $q_0$ : končí  $a, b$ ,  $|w| = 3k$ , **in**,
- $q_1$ : končí  $a$ ,  $|w| = 3k + 1$ ,
- $q_2$ : končí  $b$ ,  $|w| = 3k + 1$ , **out**,
- $q_3$ : končí  $a, b$ ,  $|w| = 3k + 2$ ,  $k \in \mathbb{Z}$ .



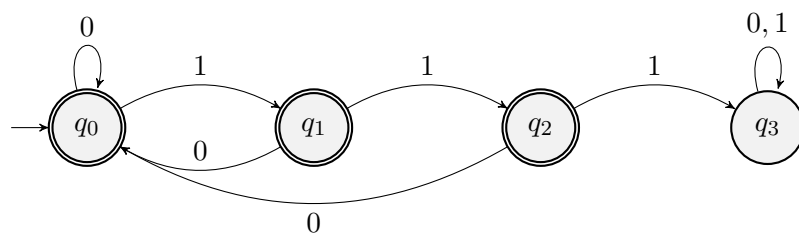
b)



Důkaz pomocí invariantů:

- $q_0$ :  $\varepsilon$ , končí 1, nekončí 01, neobsahuje 010, **in**,
- $q_1$ : končí 0, nekončí 010, neobsahuje 0101,
- $q_2$ : končí 01, neobsahuje 0101,
- $q_3$ : končí 010, neobsahuje 0101,
- $q_4$ : obsahuje 0101, **out**.

c) řešíme doplňkem, tedy jazyk  $\mathcal{L}$  obsahuje právě všechna slova, jejichž každé podslovo délky 3 **neobsahuje** znak 0. Následně všechny stavy, které nebyly výstupní, se stanou výstupními a stavy, které byly výstupní, se stanou nevýstupními.



Důkaz pomocí invariantů:

- $q_0$ : končí 0, neobsahuje 111, **in**, **out**,
- $q_1$ : končí 1, neobsahuje 111, **out**,
- $q_2$ : končí 11, neobsahuje 111, **out**,
- $q_3$ : obsahuje 111.

## 2.4 Nerodova věta a Pumping lemma

Pomocí Nerodovy věty a Pumping lemmatu dokažte, že jazyk  $L = \{0^i 1^j 0^k \mid 0 \leq i < j < k\}$  není regulární.

**Definice Pumping lemmatu.** Když  $L$  je regulární, tak existuje  $n \geq 0$  takové, že každé  $u \in L$ ,  $|u| > n$ , je možné rozložit na 3 slova  $u = xwy$  splňující:

- 1)  $|xw| \leq n$
- 2)  $w \neq \varepsilon$
- 3)  $xw^i y \in L, \forall i = 0, 1, 2, \dots$

Důkaz sporem: Kdyby  $L$  byl regulární, tak existuje  $n$  s vlastnostmi z Pumping lemma.

Zvolíme  $u = 0^n 1^{n+1} 0^{n+2}$ .

Pak 1) vlastnost říká, že  $xw = 0^l, l \leq n$ . Zároveň musí platit 2), tedy  $w = 0^k, 1 \leq k \leq l$ .

Když teď napumpujeme  $xw^i y$ , například  $i = 2$ , dostaneme  $xw^2 y = 0^{n+k} 1^{n+1} 0^{n+2} \notin L$ .

Tedy  $L$  není regulární. ■

**Definice Nerodovy věty.**  $L$  je regulární iff existuje ekvivalence  $T$  na  $\Sigma^*$  taková, že:

- 1)  $L$  je sjednocení některých tříd  $T$
- 2) pokud  $uTv$ , tak  $uwTvw$  pro každé  $w \in \Sigma^*$
- 3)  $T$  má konečný počet tříd

Důkaz sporem: Kdyby existovala  $T$ .

$1, 1^2, 1^3, \dots, 1^i, \dots, 1^n, \dots = \{1^j \mid j \geq 1\}$  je nekonečná posloupnost 0 a 1.

$T$  musí mít konečně mnoho tříd, proto musí existovat  $i > j, i \neq j \wedge 1^i T 1^j$ .

Zvolíme  $w = 0^{j+1}$ .

Pak podle vlastnosti 2)  $\underbrace{1^i 0^{j+1}}_{\substack{i \geq j+1 \\ \notin L}} T \underbrace{1^j 0^{j+1}}_{\substack{j < j+1 \\ \in L}}$ .

Tedy  $L$  není regulární. ■

### 3 Třetí cvičení

#### 3.1 Nerodova věta a Pumping lemma

Pomocí Nerodovy věty a pomocí Pumping lemmatu dokažte, že jazyk  $L = \{0^n 1^m \mid n > m \geq 0\}$  není regulární.

Definice **Nerodovy věty**.  $L$  je regulární iff existují ekvivalence  $T$  na  $\Sigma^*$  taková, že:

- 1)  $L$  je sjednocení některých tříd  $T$
- 2) pokud  $uTv$ , tak  $uwTvw$  pro každé  $w \in \Sigma^*$
- 3)  $T$  má konečný počet tříd

Důkaz sporem: Kdyby existovala  $T$  na  $\{0, 1\}^*$ .

$1, 1^2, 1^3, \dots, 1^i, \dots, 1^n, \dots = \{1^j \mid j \geq 1\}$  je nekonečná posloupnost z  $\{0, 1\}$ .

$T$  má konečně mnoho tříd, tudíž  $0^i T 0^j$  pro nějaké  $i \neq j, i > j$ .

Protože platí 2), tak  $0^i w T 0^j w$  pro  $w \in \{0, 1\}^*$ .

Zvolme  $w = 1^{i-1}$ . Pak  $\underbrace{0^i 1^{i-1}}_{\in L} T \underbrace{0^j 1^{i-1}}_{\notin L}$ . Tedy  $L$  není regulární. ■

Definice **Pumping lemmatu**. Když  $L$  je regulární, tak existuje  $n \in L, n \geq 1$ , takové, že každé  $u \in L, |u| > n$  je možné rozložit na 3 slova  $u = xwy$  splňující:

- 1)  $|xw| \leq n$
- 2)  $w \neq \varepsilon$
- 3)  $xw^i y \in L, \forall i = 0, 1, 2, \dots$

Důkaz sporem: Kdyby  $L$  byl regulární, tak existuje  $n$  s vlastnostmi z Pumping lemma.

Zvolíme  $u = 0^{n+1} 1^n$ .

Kdyby  $u = xwy$ , tak 1) vlastnost říká, že  $xw = 0^l, l \leq n$ . Zároveň musí platit 2), tedy  $w = 0^k, 1 \leq k \leq l$ . Když teď napumpujeme  $xw^i y$ , například  $i = 0$ , dostaneme  $xw^0 y = 0^{n+1-k} 1^n \notin L$ .

Tedy  $L$  není regulární. ■

#### 3.2 Nalezení slova rozlišujícího stavy

Je dán DFA tabulkou:

	$a$	$b$
$\leftrightarrow 0$	1	2
1	3	0
2	4	5
3	0	2
4	2	5
5	0	3

Najděte slovo nejkratší délky, jestliže existuje, které rozliší

- a) stavy 3 a 5.
- b) stavy 2 a 4.

To, že slovo  $u$  rozliší dva stavy znamená, že přechodová funkce při práci nad slovem  $u$  převede jeden ze stavů do koncového stavu a druhý do stavu, který není koncový.

a)  $\delta(3, a) = 0, \delta(5, a) = 0 \implies \delta^*(3, au) = \delta^*(5, au)$ . Slovo nezačíná  $a$ .

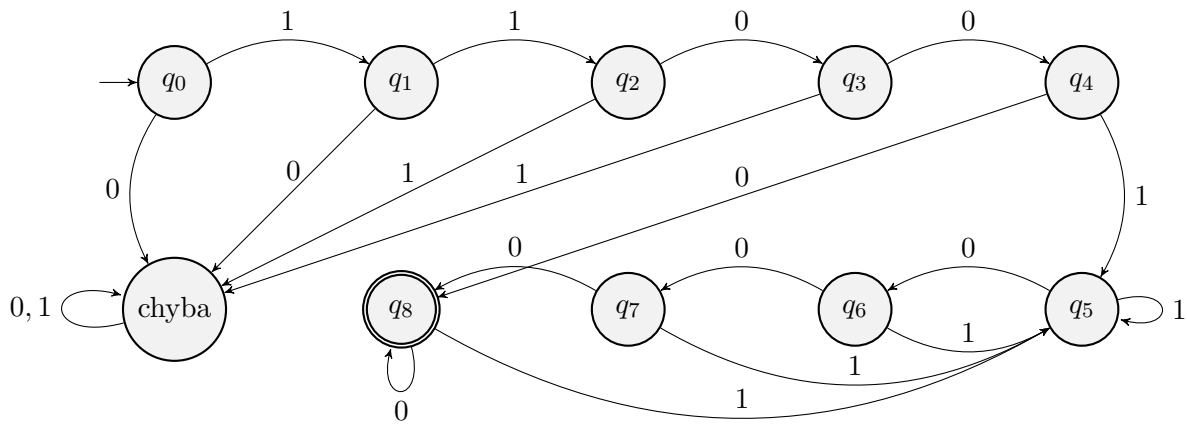
$\delta(3, b) = 2, \delta(5, b) = 3$ .

$$\left. \begin{array}{l} \delta^*(3, ba) = \delta(2, a) = 4 \notin F \\ \delta^*(5, ba) = \delta(3, a) = 0 \in F \end{array} \right\} \implies u = ba$$

b)  $\delta(2, a) = 4, \delta(4, a) = 2$ .  $\delta(2, b) = 5, \delta(4, b) = 5$ . Tyto stavy nelze rozlišit.

### 3.3 Návrh a redukce DFA podle jazyka

Navrhněte DFA, který přijímá jazyk  $L$  skládající se ze všech slov nad abecedou  $\Sigma = \{0, 1\}$ , která začínají 1100 a končí 000. Navržený automat redukuje.



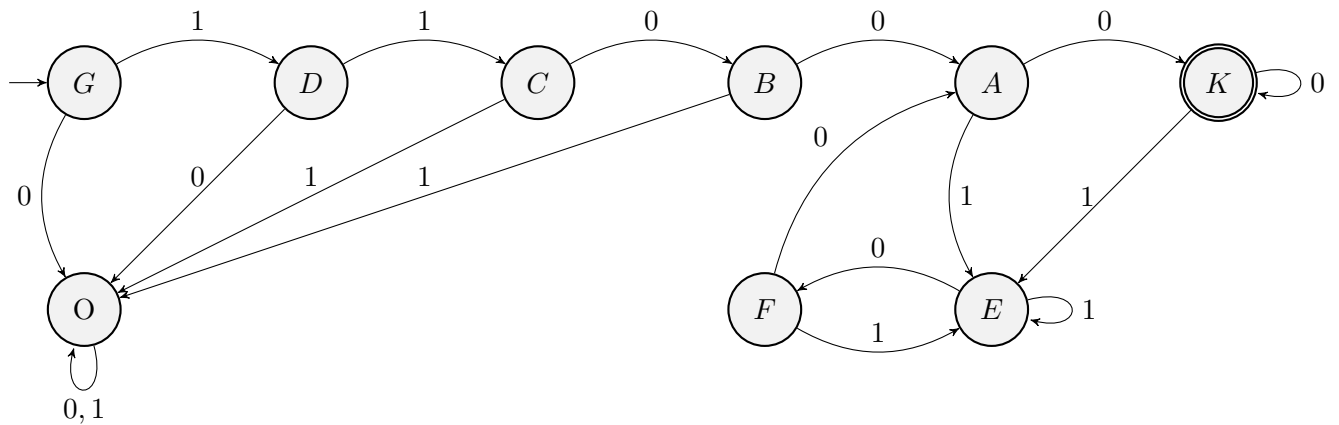
Invarianty:

- $q_0$ : in,
- $q_1$ : začíná 1, končí 1,
- $q_2$ : začíná 11, končí 11,
- $q_3$ : začíná 110, končí 110,
- $q_4$ : začíná 1100, končí 1100,
- $q_5$ : začíná 1100, končí 1,
- $q_6$ : začíná 1100, končí 10,
- $q_7$ : začíná 1100, končí 100,
- $q_8$ : začíná 1100, končí 000, out,
- chyba: nezačíná 1100.

Redukce podmnožinovou konstrukcí:

		0	1	$\sim_0$	0	1	$\sim_1$	0	1	$\sim_2$	0	1	$\sim_3$	0	1	$\sim_4$	0	1	$\sim_5$	0	1	$\sim_6$
$\rightarrow$	$q_0$	ch	1	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$D$	$G$	$O$	$D$	$G$
	$q_1$	ch	2	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$C$	$D$	$O$	$C$	$D$	$O$	$C$	$D$
	$q_2$	3	ch	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$B$	$O$	$C$	$B$	$O$	$C$	$B$	$O$	$C$	$B$	$O$	$C$
	$q_3$	4	ch	$O$	$O$	$O$	$O$	$A$	$O$	$B$	$A$	$O$	$B$	$A$	$O$	$B$	$A$	$O$	$B$	$A$	$O$	$B$
	$q_4$	8	5	$O$	$K$	$O$	$A$	$K$	$O$	$A$	$K$	$O$	$A$	$K$	$C$	$A$	$K$	$E$	$A$	$K$	$E$	$A$
	$q_5$	6	5	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$B$	$O$	$C$	$B$	$C$	$E$	$F$	$E$	$E$	$F$	$E$	$E$
	$q_6$	7	5	$O$	$O$	$O$	$O$	$A$	$O$	$B$	$A$	$O$	$B$	$A$	$C$	$F$	$A$	$E$	$F$	$A$	$E$	$F$
	$q_7$	8	5	$O$	$K$	$O$	$A$	$K$	$O$	$A$	$K$	$O$	$A$	$K$	$C$	$A$	$K$	$E$	$A$	$K$	$E$	$A$
$\leftarrow$	$q_8$	8	5	$K$	$K$	$O$	$K$	$K$	$O$	$K$	$K$	$O$	$K$	$K$	$C$	$K$	$K$	$E$	$K$	$K$	$E$	$K$
	ch	ch	ch	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$

Redukovaný automat:



### 3.4 Srovnání dvou DFA automatů

Jsou dány dva automaty. Rozhodněte, zda jsou ekvivalentní, tj. zda přijímají stejný jazyk.

$M_1 :$		$a$	$b$	$M_2 :$		$a$	$b$
	$\leftrightarrow 0$	0	5		$A$	$H$	$G$
	1	1	3		$B$	$B$	$A$
	2	2	7		$C$	$E$	$D$
	3	3	2		$D$	$D$	$B$
	$\leftarrow 4$	6	1		$E$	$C$	$D$
	5	5	1		$F$	$F$	$E$
	$\leftarrow 6$	4	2		$\leftrightarrow G$	$G$	$F$
	7	7	0		$H$	$A$	$G$

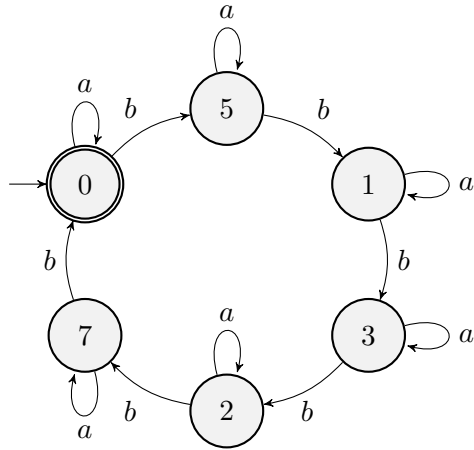
Nejdříve odstraníme nedosažitelné vztahy a pak provedeme redukci.

Odstranění nedosažitelných vztahů:

$M_1 :$		$a$	$b$	$M_2 :$		$a$	$b$
	$\leftrightarrow 0$	0	5		$A$	$H$	$G$
	1	1	3		$B$	$B$	$A$
	2	2	7		$C$	$E$	$D$
	3	3	2		$D$	$D$	$B$
	5	5	1		$E$	$C$	$D$
	7	7	0		$F$	$F$	$E$
					$\leftrightarrow G$	$G$	$F$
					$H$	$A$	$G$

A teď zredukovat oba automaty.

$M_1$ :



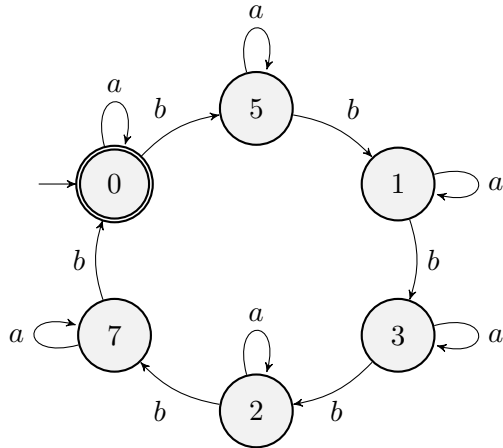
$M_1$  je již redukovaný.

$M_2$ :

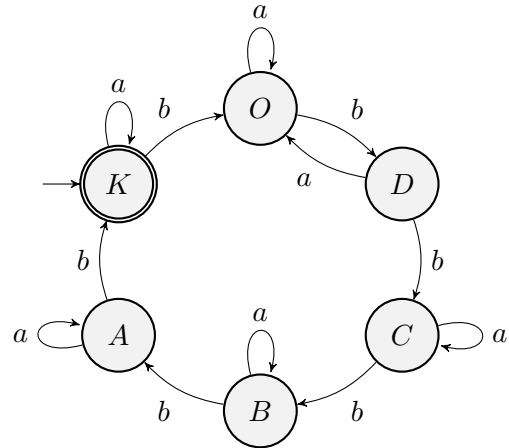
		$a$	$b$	$\sim_0$	$a$	$b$	$\sim_1$	$a$	$b$	$\sim_2$	$a$	$b$	$\sim_3$	$a$	$b$	$\sim_4$	$a$	$b$	$\sim_5$
$\leftrightarrow$	$A$	$H$	$G$	$O$	$O$	$K$	$A$	$A$	$K$	$A$	$A$	$K$	$A$	$A$	$K$	$A$	$A$	$K$	$A$
	$B$	$B$	$A$	$O$	$O$	$O$	$O$	$O$	$A$	$B$	$B$	$A$	$B$	$B$	$A$	$B$	$B$	$A$	$B$
	$C$	$E$	$D$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$C$	$O$	$D$	$C$	$E$
	$D$	$D$	$B$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$B$	$C$	$C$	$B$	$C$	$C$	$B$	$C$
	$E$	$C$	$D$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$C$	$D$	$O$	$C$	$D$
	$F$	$F$	$E$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$D$	$O$
	$G$	$G$	$F$	$K$	$K$	$O$	$K$	$K$	$O$	$K$	$K$	$O$	$K$	$K$	$O$	$K$	$K$	$O$	$K$
	$H$	$A$	$G$	$O$	$O$	$K$	$A$	$A$	$K$	$A$	$A$	$K$	$A$	$A$	$K$	$A$	$A$	$K$	$A$

Protože má každý řádek svou vlastní třídu,  $M_2$  je již redukovaný.

$M_1$ :



$M_2$ :



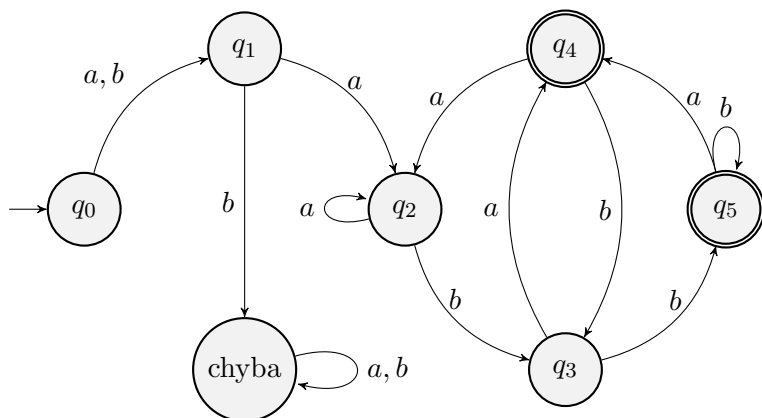
Nejsou ekvivalnetní -  $M_1$  přijme např. slovo  $bbabbbb$ , které  $M_2$  nepřijme.

### 3.5 Návrh a redukce DFA podle jazyka

Navrhnete DFA, který přijímá  $L$  nad abecedou  $\{a, b\}$ , kde  $L$  obsahuje právě všechna slova  $w$  taková, že

- druhý znak slova  $w$  je  $a$ ,
- předposlední znak slova  $w$  je  $b$ ,
- $|w| \geq 3$ .

Výsledný DFA redukuje.



Invarianty:

- $q_0$ :  $|w| < 3$ , in,
- chyba: druhý znak je  $b$ ,
- $q_1$ :  $|w| < 3$ , druhý znak je  $\varepsilon$ , předposlední znak je  $\varepsilon$ ,
- $q_2$ :  $|w| < 3$ , druhý znak je  $a$ , předposlední znak je  $a, b$ ,
- $q_3$ :  $|w| \geq 3$ , druhý znak je  $a$ , předposlední znak je  $a$ ,
- $q_4$ :  $|w| \geq 3$ , druhý znak je  $a$ , předposlední znak je  $b$ , končí  $ba$ , out,
- $q_5$ :  $|w| \geq 3$ , druhý znak je  $a$ , předposlední znak je  $b$ , končí  $bb$ , out.

Redukce podmnožinovou konstrukcí:

		$a$	$b$	$\sim_0$	$a$	$b$	$\sim_1$	$a$	$b$	$\sim_2$	$a$	$b$	$\sim_3$	$a$	$b$	$\sim_4$
→	$q_0$	$q_1$	$q_1$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$D$	$D$	$E$
	$q_1$	$q_2$	Chyba	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$C$	$O$	$D$	$C$	$O$	$D$
	$q_2$	$q_2$	$q_3$	$O$	$O$	$O$	$O$	$O$	$A$	$C$	$C$	$A$	$C$	$C$	$A$	$C$
	$q_3$	$q_4$	$q_5$	$O$	$K$	$K$	$A$	$B$	$K$	$A$	$B$	$K$	$A$	$B$	$K$	$A$
←	$q_4$	$q_2$	$q_3$	$K$	$O$	$O$	$B$	$O$	$A$	$B$	$C$	$A$	$B$	$C$	$A$	$B$
←	$q_5$	$q_4$	$q_5$	$K$	$K$	$K$	$K$	$B$	$K$	$K$	$B$	$K$	$K$	$B$	$K$	$K$
	Chyba	Chyba	Chyba	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$

Protože každý řádek má svou vlastní třídu, původní DFA je již redukovaný.

## 4 Čtvrté cvičení

### 4.1 Sestrojení DFA z NFA

Pro dané NFA sestrojte podmnožinovou konstrukcí DFA a výsledek redukujte.

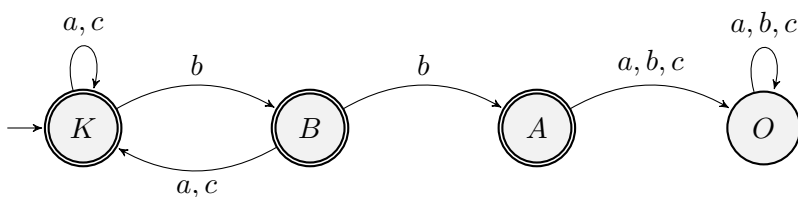
$$M_1 :$$

	$a$	$b$	$c$
$\leftrightarrow 1$	$\{1\}$	$\{2\}$	$\{1\}$
$\leftarrow 2$	$\{1\}$	$\{3\}$	$\{1\}$
$\leftarrow 3$	$\emptyset$	$\emptyset$	$\emptyset$

Všimněte si, že v automatu  $M_1$  jsou všechny stavy koncové. Co z toho lze usoudit o jazyku, které je automatem přijímán?

	$a$	$b$	$c$	$\sim_0$	$a$	$b$	$c$	$\sim_1$	$a$	$b$	$c$	$\sim_2$	$a$	$b$	$c$	$\sim_3$
$\leftrightarrow \{1\}$	$\{1\}$	$\{2\}$	$\{1\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$B$	$K$
$\leftarrow \{2\}$	$\{1\}$	$\{3\}$	$\{1\}$	$K$	$K$	$K$	$K$	$K$	$K$	$A$	$K$	$B$	$K$	$A$	$K$	$B$
$\leftarrow \{3\}$	$\emptyset$	$\emptyset$	$\emptyset$	$K$	$O$	$O$	$O$	$A$	$O$	$O$	$O$	$A$	$O$	$O$	$O$	$A$
$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$

Automat se redukcí nezmění.



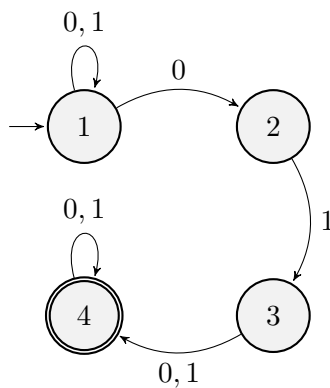
Automat nepřijímá slova obsahující sekvenci  $bb$  následovanou libovolným dalším znakem.

### 4.2 Sestrojení DFA z NFA

NFA  $M$  je dán tabulkou níže. Nakreslete jeho stavový diagram a podmnožinovou konstrukcí sestrojte DFA, který přijímá stejný jazyk.  $DFA$  zredukujte.

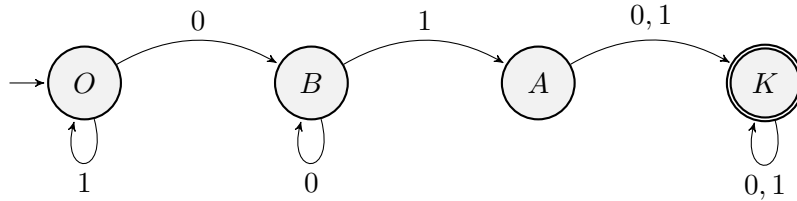
$$M :$$

	0	1
$\rightarrow 1$	$\{1, 2\}$	$\{1\}$
2	$\emptyset$	$\{3\}$
3	$\{4\}$	$\{4\}$
$\leftarrow 4$	$\{4\}$	$\{4\}$



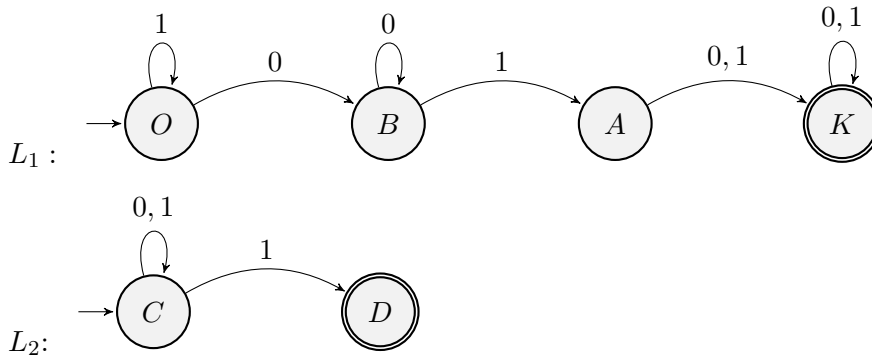


		0	1	$\sim_0$	0	1	$\sim_1$	0	1	$\sim_2$	0	1	$\sim_3$
$\rightarrow$	$\{1\}$	$\{1, 2\}$	$\{1\}$	$O$	$O$	$O$	$O$	$K$	$K$	$K$	$K$	$K$	$O$
	$\{1, 2\}$	$\{1, 2\}$	$\{1, 3\}$	$O$	$O$	$O$	$O$	$K$	$K$	$K$	$K$	$K$	$B$
	$\{1, 3\}$	$\{1, 2, 4\}$	$\{1, 4\}$	$O$	$K$	$K$	$A$	$K$	$K$	$A$	$K$	$K$	$A$
$\leftarrow$	$\{1, 2, 4\}$	$\{1, 2, 4\}$	$\{1, 3, 4\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$
$\leftarrow$	$\{1, 4\}$	$\{1, 2, 4\}$	$\{1, 4\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$
$\leftarrow$	$\{1, 3, 4\}$	$\{1, 2, 4\}$	$\{1, 4\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$



### 4.3 Návrh NFA dle sjednocení jazyků a redukované DFA

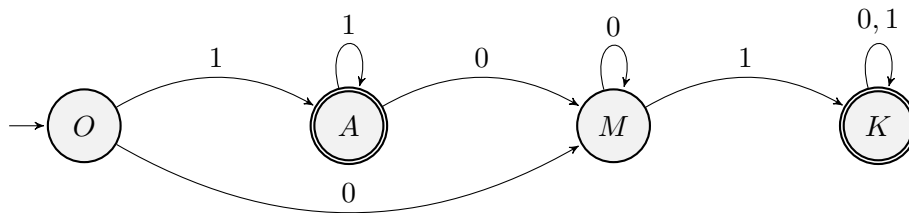
Navrhněte NFA přijímající jazyk  $L = L_1 \cup L_2$ , kde  $L_1 = L(M)$ , kde  $M$  je automat z 4.2, a  $L_2 = \{u \mid u \text{ končí } 1\}$ . K tomuto NFA zkonstruuje DFA přijímající stejný jazyk. DFA redukuje.



Podmnožinová konstrukce:

		0	1	$\sim_0$	0	1	$\sim_1$	0	1	$\sim_2$	0	1	$\sim_3$
$\rightarrow$	$\{O, C\}$	$\{B, C\}$	$\{O, C, D\}$	$O$	$O$	$K$	$O$	$O$	$A$	$O$	$M$	$A$	$O$
	$\{B, C\}$	$\{B, C\}$	$\{A, C, D\}$	$O$	$O$	$K$	$O$	$O$	$K$	$M$	$M$	$K$	$M$
$\leftarrow$	$\{O, C, D\}$	$\{B, C\}$	$\{O, C, D\}$	$K$	$O$	$K$	$A$	$O$	$A$	$A$	$M$	$A$	$A$
$\leftarrow$	$\{A, C, D\}$	$\{K, C\}$	$\{K, C, D\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$
$\leftarrow$	$\{K, C\}$	$\{K, C\}$	$\{K, C, D\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$
$\leftarrow$	$\{K, C, D\}$	$\{K, C\}$	$\{K, C, D\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$

Výsledný redukováný DFA:



#### 4.4 Srovnání dvou NFA automatů

Jsou dány dva  $\varepsilon$ -NFA. Rozhodněte, zda přijímají stejný jazyk. Pro oba  $\varepsilon$ -NFA sestrojte redukované DFA.

$$M_1 :$$

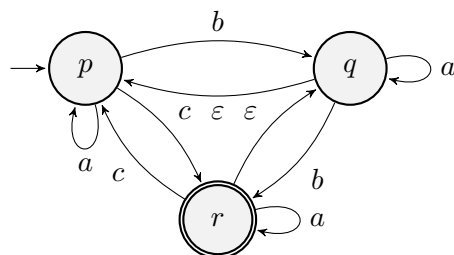
	$\varepsilon$	$a$	$b$	$c$
$\rightarrow p$	$\emptyset$	$\{p\}$	$\{q\}$	$\{r\}$
$q$	$\{p\}$	$\{q\}$	$\{r\}$	$\emptyset$
$\leftarrow r$	$\{q\}$	$\{r\}$	$\emptyset$	$\{p\}$

$$M_2 :$$

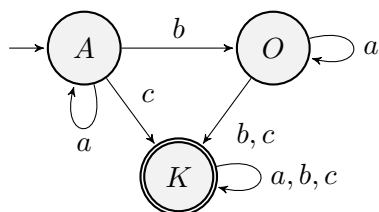
	$\varepsilon$	$a$	$b$	$c$
$\rightarrow p$	$\{q, r\}$	$\emptyset$	$\{q\}$	$\{r\}$
$q$	$\emptyset$	$\{p\}$	$\{q\}$	$\{p, q\}$
$\leftarrow r$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$

$M_1$ :

	$\varepsilon$	$a$	$b$	$c$	$\varepsilon$ -uzávěry
$\rightarrow p$	$\emptyset$	$\{p\}$	$\{q\}$	$\{r\}$	$\varepsilon\text{-uz}(p) = \{p\}$
$q$	$\{p\}$	$\{q\}$	$\{r\}$	$\emptyset$	$\varepsilon\text{-uz}(q) = \{p, q\}$
$\leftarrow r$	$\{q\}$	$\{r\}$	$\emptyset$	$\{p\}$	$\varepsilon\text{-uz}(r) = \{p, q, r\}$



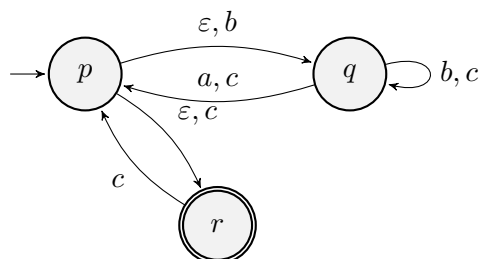
Redukovaný DFA a podmnožinová konstrukce pro  $M_1$ :



		$a$	$b$	$c$	$\sim_0$	$a$	$b$	$c$	$\sim_1$
$\rightarrow$	$\{p\}$	$\{p\}$	$\{p, q\}$	$\{p, q, r\}$	$O$	$O$	$O$	$K$	$A$
	$\{p, q\}$	$\{p, q\}$	$\{p, q, r\}$	$\{p, q, r\}$	$O$	$O$	$K$	$K$	$O$
$\leftarrow$	$\{p, q, r\}$	$\{p, q, r\}$	$\{p, q, r\}$	$\{p, q, r\}$	$K$	$K$	$K$	$K$	$K$

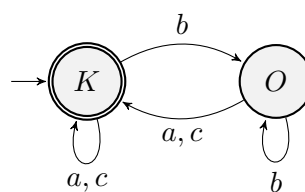
$M_2$ :

	$\varepsilon$	$a$	$b$	$c$	$\varepsilon$ -uzávěry
$\rightarrow p$	$\{q, r\}$	$\emptyset$	$\{q\}$	$\{r\}$	$\varepsilon\text{-uz}(p) = \{p, q, r\}$
$q$	$\emptyset$	$\{p\}$	$\{q\}$	$\{p, q\}$	$\varepsilon\text{-uz}(q) = \{q\}$
$\leftarrow r$	$\emptyset$	$\emptyset$	$\emptyset$	$\{p\}$	$\varepsilon\text{-uz}(r) = \{r\}$



Redukovaný DFA a podmnožinová konstrukce pro  $M_2$ :

		$a$	$b$	$c$	$\sim_0$
$\leftrightarrow$	$\{p, q, r\}$	$\{p, q, r\}$	$\{q\}$	$\{p, q, r\}$	$K$
	$\{q\}$	$\{p, q, r\}$	$\{q\}$	$\{p, q, r\}$	$O$



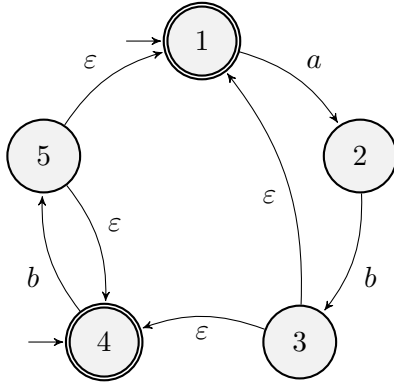
Z redukovaných DFA tabulek je očividné, že  $M_1$  a  $M_2$  nepřijímají stejný jazyk.  $M_2$  například přijme slovo  $w = a$ , zatímco  $M_1$  takové nepřijme.

#### 4.5 Konstrukce DFA z $\varepsilon$ -NFA

Je dán  $\varepsilon$ -NFA. Zkonstruuje redukovaný DFA přijímající stejný jazyk jako M.

	$\varepsilon$	$a$	$b$
$\leftrightarrow 1$	$\emptyset$	$\{2\}$	$\emptyset$
2	$\emptyset$	$\emptyset$	$\{3\}$
3	$\{1, 4\}$	$\emptyset$	$\emptyset$
$\leftrightarrow 4$	$\emptyset$	$\emptyset$	$\{5\}$
5	$\{1, 4\}$	$\emptyset$	$\emptyset$

$\varepsilon$ -NFA:

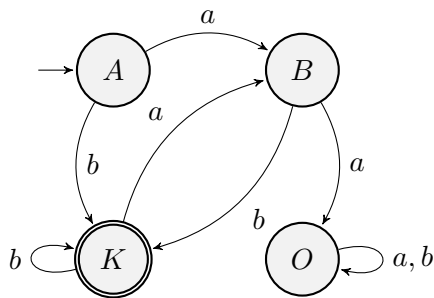


	$\varepsilon$	$a$	$b$	$\varepsilon$ -uzávěry
$\leftrightarrow 1$	$\emptyset$	$\{2\}$	$\emptyset$	$\varepsilon\text{-uz}(1) = \{1\}$
2	$\emptyset$	$\emptyset$	$\{3\}$	$\varepsilon\text{-uz}(2) = \{2\}$
3	$\{1, 4\}$	$\emptyset$	$\emptyset$	$\varepsilon\text{-uz}(3) = \{1, 3, 4\}$
$\leftrightarrow 4$	$\emptyset$	$\emptyset$	$\{5\}$	$\varepsilon\text{-uz}(4) = \{4\}$
5	$\{1, 4\}$	$\emptyset$	$\emptyset$	$\varepsilon\text{-uz}(5) = \{1, 4, 5\}$

Podmnožinová konstrukce:

	$a$	$b$	$\sim_0$	$a$	$b$	$\sim_1$	$a$	$b$	$\sim_2$	$a$	$b$	$\sim_3$	
$\rightarrow$	$\{1, 4\}$	$\{2\}$	$\{1, 4, 5\}$	$O$	$O$	$K$	$A$	$A$	$K$	$A$	$B$	$K$	$A$
	$\{2\}$	$\emptyset$	$\{1, 3, 4\}$	$O$	$O$	$K$	$A$	$O$	$K$	$B$	$O$	$K$	$B$
$\leftarrow$	$\{1, 4, 5\}$	$\{2\}$	$\{1, 4, 5\}$	$K$	$O$	$K$	$K$	$A$	$K$	$K$	$B$	$K$	$K$
	$\emptyset$	$\emptyset$	$\emptyset$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$
$\leftarrow$	$\{1, 3, 4\}$	$\{2\}$	$\{1, 4, 5\}$	$K$	$O$	$K$	$K$	$A$	$K$	$K$	$B$	$K$	$K$

Výsledný DFA:



## 5 Páté cvičení

### 5.1 Důkaz obecné vlastnosti dvou libovolných jazyků

Dokažte, že pro libovolné jazyky  $L_1$  a  $L_2$  nad stejnou abecedou platí  $(L_1 \cup L_2)^* = (L_1^* L_2^*)^*$ .

a)  $(L_1 \cup L_2)^* \subseteq (L_1^* L_2^*)^*$

Nechť  $w \in (L_1 \cup L_2)^*$ . Tedy  $w = x_1 x_2 \dots x_k, x_i \in (L_1 \cup L_2)$ .

Protože každý  $x_i$  patří buď do  $L_1$  nebo do  $L_2$ , můžeme každé takové  $w$  chápat jako zřetězení slov, která střídavě patří do  $L_1^*$  nebo  $L_2^*$ . Tedy slova z  $(L_1 \cup L_2)^*$  lze seskupit do bloků tak, že tyto bloky patří buď do  $L_1^*$  nebo  $L_2^*$ .

Například:  $w = u_1 v_1 u_2 v_2 \dots u_m v_m$ , kde  $u_i \in L_1^*, v_i \in L_2^*$ .

To znamená, že  $w \in (L_1^* \cup L_2^*)^*$ . Jelikož  $w$  je libovolné slovo z  $(L_1^* L_2^*)^*$  platí  $(L_1 \cup L_2)^* \subseteq (L_1^* L_2^*)^*$ .

b)  $(L_1^* L_2^*)^* \subseteq (L_1 \cup L_2)^*$

Nechť  $w \in (L_1^* L_2^*)^*$ . Tedy  $w = y_1 y_2 \dots y_k, y_i \in L_1^* L_2^*$ .

Každé  $y_i \in L_1^* L_2^*$  lze dále rozepsat jako zřetězení slov  $u_1 v_1 \dots u_m v_m$ , kde  $u_j \in L_1^*, v_j \in L_2^*$ .

Tedy každé slovo  $y_i$  je posloupnost slov z  $L_1$  a  $L_2$ , což znamená, že všechny  $y_i$  jsou tvořeny prvky z  $L_1 \cup L_2$ . Proto celé  $w$  lze vyjádřit jako zřetězení slov z  $L_1 \cup L_2$ , což znamená, že  $w \in (L_1 \cup L_2)^*$ . A tedy platí  $(L_1^* L_2^*)^* \subseteq (L_1 \cup L_2)^*$ .

### 5.2 Tvorba regulárního výrazu reprezentujícího jazyk

Napište regulární výraz, který reprezentuje jazyk  $L$  nad abecedou  $\Sigma = \{0, 1\}$ , jestliže výraz existuje.

- a)  $L$  se skládá ze všech slov, které obsahují pouze 0.
- b)  $L$  se skládá ze všech slov, které obsahují přesně jednu 1.
- c)  $L$  se skládá ze všech slov, které obsahují alespoň jednu 1.
- d)  $L$  se skládá ze všech slov, které obsahují alespoň dvě 1.
- e)  $L$  se skládá ze všech slov, které obsahují sudý počet 1.
- f)  $L$  se skládá ze všech slov, které obsahují lichý počet 1.

Odpovědi zdůvodněte.

- |                    |                            |
|--------------------|----------------------------|
| a) $0^*00^*$ .     | d) $0^*10^*1(0+1)^*$ .     |
| b) $0^*10^*$ .     | e) $(0^*10^*1)^*0^*$ .     |
| c) $0^*1(0+1)^*$ . | f) $0^*1(0^*10^*1)^*0^*$ . |

Zdůvodnění:

- a) Vynutíme si alespoň jednu nulu, protože prázdné slovo neobsahuje 0. Zároveň umožníme pomocí Kleeného operátoru libovolný počet 0 na obou stranách slova. Žádnou 1 nepovolíme.
- b) Před první 1 povolíme libovolný počet 0. Po první 1 následně povolíme pouze libovolný počet 0.
- c) Před první 1 povolíme libovolný počet 0. Po první 1 následně povolíme všechny znaky abecedy  $\Sigma$  v libovolném počtu.
- d) Dovolíme libovolný počet 0, po kterých následuje právě jedna 1. Následovat může následovat libovolný počet 0. Po té zas vynutíme právě jednu 1. Po této sekvenci může následovat libovolný počet znaků abecedy  $\Sigma$ .

- e) Zde mohou nastat dva případy: slovo neobsahuje žádnou 1, nebo jich obsahuje sudý počet. Pokud neobsahuje žádnou 1, můžeme celou závorku přeskočit díky Kleenyho operátoru, pak přijmeme pouze libovolný počet 0. Pokud obsahuje jedničku, závorka nám zaručí, že jich bude nutně sudý počet. Mezi nimi na všech stranách může být libovolný počet 0.
- f) Zde vynutíme alespoň jednu 1 a pokud by se mělo ve slově vyskytovat více 1, tak závorka zaručí, že vždy přijmeme dvojici 1. Tím bude v celém slově stále lichý počet již zpracovaných 1. Znovu z obou stran obklopeny libovolným počtem 0.

### 5.3 Hledání slov splňujících různé regulární výrazy

Jazyk  $L_1$  je reprezentován regulárním výrazem  $r_1 = 0^*1^*0^*1^*0^*$  a jazyk  $L_2$  je reprezentován regulárním výrazem  $r_2 = (01 + 10)^*$ .

- a) Najděte nejkratší neprázdné slovo, které patří do průniku  $L_1 \cap L_2$ .
- b) Najděte nejdelší neprázdné slovo, které patří do průniku  $L_1 \cap L_2$ .
- c) Najděte nejkratší slovo, které leží v  $L_1$ , ale neleží v  $L_2$ .
- d) Najděte nejkratší slovo, které neleží ve sjednocení  $L_1 \cup L_2$ .

Odpovědi zdůvodněte.

- |                |   |
|----------------|---|
| a) 01 nebo 10. | c) 0 nebo 1, protože délka $\notin L_2$ . |
| b) 01100110.   | d) 10101.                                 |

Zdůvodnění:

- a) Silnější restrikcí do průniku vnáší  $L_2$ .  $L_2$  vynucuje vybrání si jedné z dvojic 10 nebo 01.  $L_1$  nám obě varianty povoluje.
- b) Teď naopak využijeme flexibility  $L_1$ . Jedinou povinností je používat vždy dvojice, kvůli  $L_2$ . Pro vizualizaci:  $01^20^21^20$ . To má přesně tvar jako  $r_1$ .
- c) Protože  $L_2$  vyžaduje dvojice, zaměříme se na slova s menší délkou: 0, 1. Obě varianty  $L_1$  přijme.
- d) Musíme porušit dvojice z  $L_2$  a zároveň pravidelné střídání 0 a 1 z  $L_1$ . Stačí tedy prohodit pořadí střídání a aby slovo mělo lichou délku. Také je potřeba, aby délka byla alespoň  $|r_1|$ , kvůli  $L_1$ .

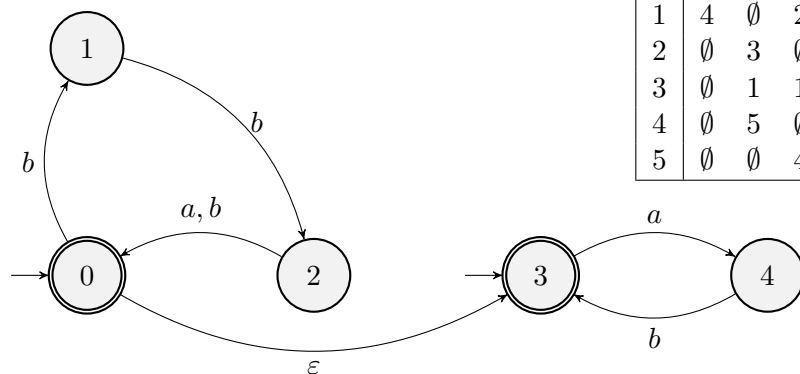
### 5.4 Konstrukce redukovaného DFA k regulárnímu výrazu

Je dán regulární výraz  $r = (baa + bab)^*(ab)^*$ . K  $r$  zkonstruujte redukovaný DFA, který přijímá jazyk reprezentovaný tímto regulárním výrazem.

(Návod: Postupujte dvěma způsoby; jednak obecným postupem z přednášky, jednak rozdělením na podvýrazy, pro které je možné najít NFA přímo a pak použitím konstrukcí z důkazů faktu, že třída regulárních jazyků je uzavřena na sjednocení, zřetězení a Kleeného operátor.)

## 1. Obecný postup

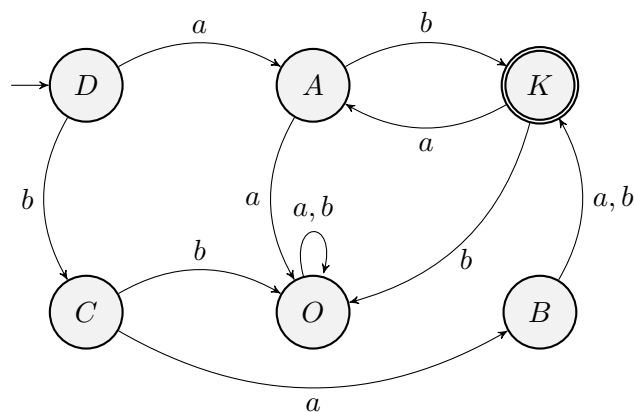
$$(baa + bab)^* \equiv (ba(a + b))^*$$



	$\varepsilon$	$a$	$b$	$\varepsilon$ uzávěry
1	4	$\emptyset$	2	$\varepsilon\text{-uz}(1) = \{1, 4\}$
2	$\emptyset$	3	$\emptyset$	$\varepsilon\text{-uz}(2) = \{2\}$
3	$\emptyset$	1	1	$\varepsilon\text{-uz}(3) = \{3\}$
4	$\emptyset$	5	$\emptyset$	$\varepsilon\text{-uz}(4) = \{4\}$
5	$\emptyset$	$\emptyset$	4	$\varepsilon\text{-uz}(5) = \{5\}$

Podmnožinová konstrukce:

		$a$	$b$	$\sim_0$	$a$	$b$	$\sim_1$	$a$	$b$	$\sim_2$	$a$	$b$	$\sim_3$
$\leftrightarrow$	$\{1, 4\}$	5	2	$K$	$O$	$O$	$K$	$A$	$O$	$K$	$A$	$C$	$D$
	5	$\emptyset$	4	$O$	$O$	$K$	$A$	$O$	$K$	$A$	$O$	$K$	$A$
	2	3	$\emptyset$	$O$	$O$	$O$	$O$	$B$	$O$	$C$	$B$	$O$	$C$
$\leftarrow$	4	5	$\emptyset$	$K$	$O$	$O$	$K$	$A$	$O$	$K$	$A$	$O$	$K$
	3	$\{1, 4\}$	$\{1, 4\}$	$O$	$K$	$K$	$B$	$K$	$K$	$B$	$K$	$K$	$B$
	$\emptyset$	$\emptyset$	$\emptyset$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$



## 2. postup Rozdělení na podvýrazy

I. krok očíslování

$$(b_1a_2a_3 + b_4a_5b_6)^*(a_7b_8)^* \equiv (b_1a_2(a_3 + b_4))^*(a_5b_6)^*$$

II. krok

Pro jazyk, který je přijímán regulárním výrazem  $r$  platí:

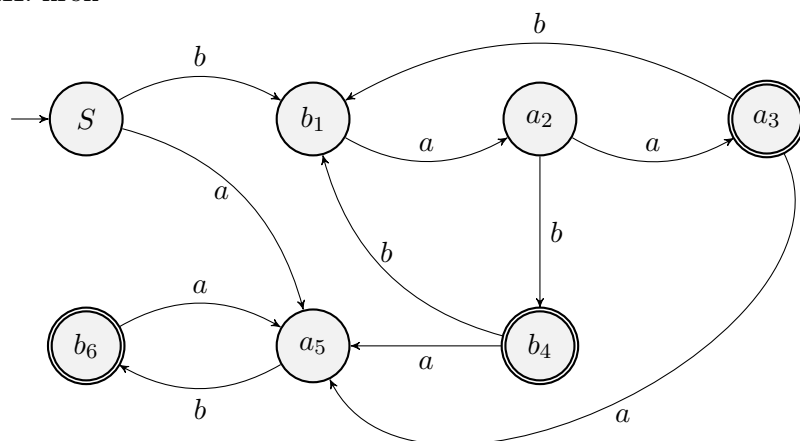
výraz může začínat:  $b_1, a_5$

mohou po sobě následovat:  $\mathbf{b_1}$ :  $a_2$ ;  $\mathbf{a_2}$ :  $a_3, b_4$ ;  $\mathbf{a_3}$ :  $b_1, a_5$ ;  $\mathbf{b_4}$ :  $b_1, a_5$ ;  $\mathbf{a_5}$ :  $b_6$ ;  $\mathbf{b_6}$ :  $a_5$

výraz může končit:  $a_3, b_4, b_6$

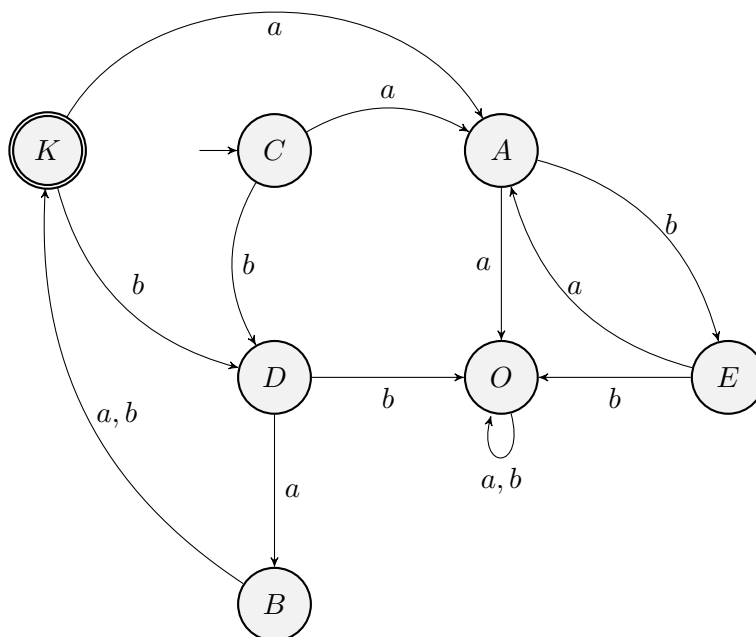
je  $\varepsilon$  v  $L$ ? Ano.

III. krok



IV. podmnožinová konstrukce DFA + redukce

		<i>a</i>	<i>b</i>	$\sim_0$	<i>a</i>	<i>b</i>	$\sim_1$	<i>a</i>	<i>b</i>	$\sim_2$	<i>a</i>	<i>b</i>	$\sim_3$
$\rightarrow$	<i>S</i>	5	1	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>A</i>	<i>O</i>	<i>C</i>	<i>A</i>	<i>D</i>	<i>C</i>
	5	$\emptyset$	6	<i>O</i>	<i>O</i>	<i>K</i>	<i>A</i>	<i>O</i>	<i>K</i>	<i>A</i>	<i>O</i>	<i>E</i>	<i>A</i>
	1	2	$\emptyset$	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>B</i>	<i>O</i>	<i>D</i>	<i>B</i>	<i>O</i>	<i>D</i>
	$\emptyset$	$\emptyset$	$\emptyset$	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>	<i>O</i>
$\leftarrow$	6	5	$\emptyset$	<i>K</i>	<i>O</i>	<i>O</i>	<i>K</i>	<i>A</i>	<i>O</i>	<i>E</i>	<i>A</i>	<i>O</i>	<i>E</i>
	2	3	4	<i>O</i>	<i>K</i>	<i>K</i>	<i>B</i>	<i>K</i>	<i>K</i>	<i>B</i>	<i>K</i>	<i>K</i>	<i>B</i>
$\leftarrow$	3	5	1	<i>K</i>	<i>O</i>	<i>O</i>	<i>K</i>	<i>A</i>	<i>O</i>	<i>K</i>	<i>A</i>	<i>D</i>	<i>K</i>
$\leftarrow$	4	5	1	<i>K</i>	<i>O</i>	<i>O</i>	<i>K</i>	<i>A</i>	<i>O</i>	<i>K</i>	<i>A</i>	<i>D</i>	<i>K</i>

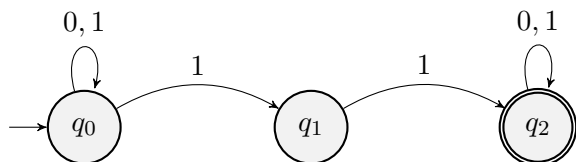


## 5.5 Tvorba DFA a regulárního výrazu dle jazyka

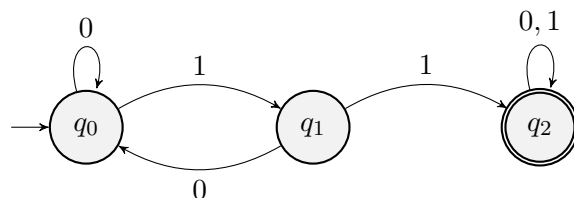
Je dán jazyk  $L$  nad  $\Sigma = \{0, 1\}$ , kde  $L = \{w \mid w \text{ neobsahuje } 11 \text{ jako podslovo}\}$ . Navrhnete redukovaný DFA  $M$ , který přijímá  $L$ . Pro jazyk  $L$  najděte regulární výraz, který ho reprezentuje (použijte úpravy grafu z přednášky).

$\mathcal{L} = \{w \mid w \text{ obsahuje } 11 \text{ jako podslovo}\}$ .

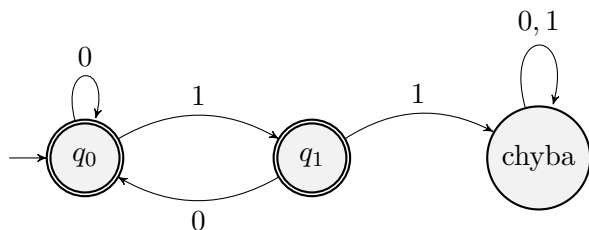
$\mathcal{L}$  NFA:



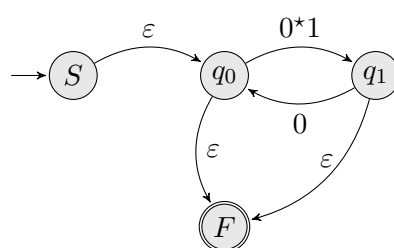
$\mathcal{L}$  DFA:



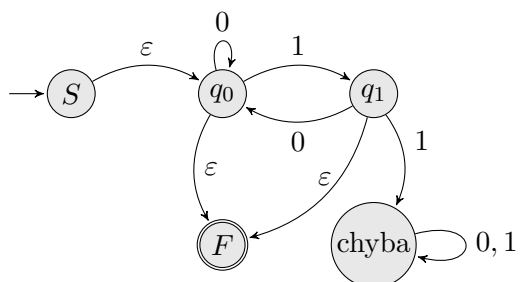
$L$  DFA:



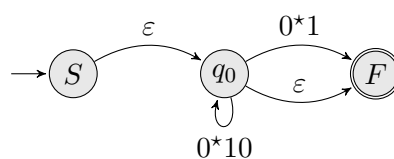
3. odstraňuji vrchol *chyba*:



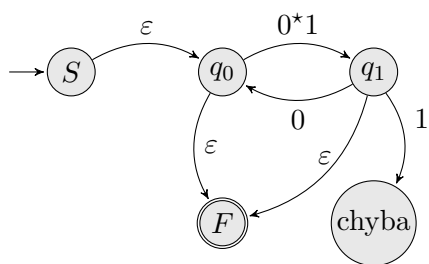
1. zavedu stavy  $S, F$ :



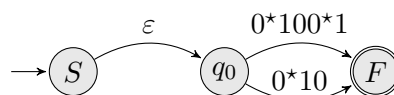
4. odstraňuji vrchol  $q_1$ :



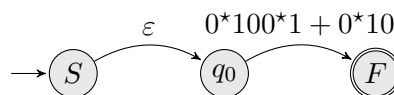
2. odstraňuji smyčky:



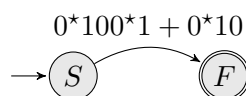
5. odstraňuji smyčky:



6. odstraňuji paralelní hrany:



7. odstraňuji vrchol  $A$ :



Finální regulární výraz pro  $L$ :  $r = 0^*100^*1 + 0^*10$ .



## 6 Šesté cvičení

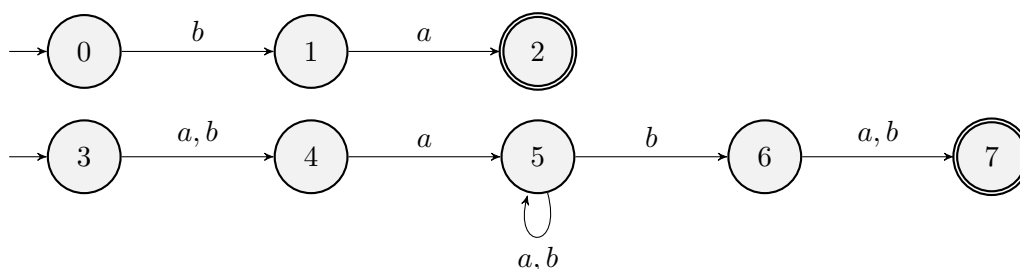
### 6.1 Návrh NFA a redukovaný DFA

Navrhněte NFA, který přijímá jazyk  $L$  nad abecedou  $\{a, b\}$ , kde  $L$  obsahuje právě všechna slova  $w$  taková, že

- druhý znak slova  $w$  je  $a$ ,
- předposlední znak slova  $w$  je  $b$ .

K danému NFA (není-li již DFA) sestrojte podmnožinovou konstrukcí DFA přijímající stejný jazyk. Výsledný DFA redukuje.

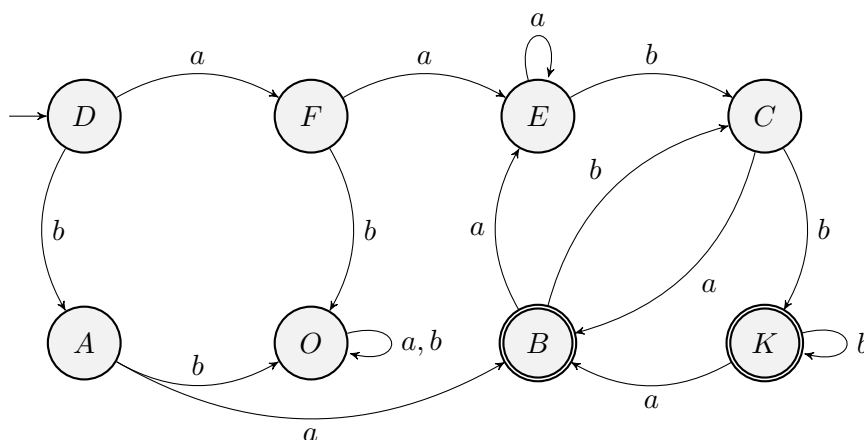
NFA:



Podmnožinová konstrukce:

		$a$	$b$	$\sim_0$	$a$	$b$	$\sim_1$	$a$	$b$	$\sim_2$	$a$	$b$	$\sim_3$	$a$	$b$	$\sim_4$
$\rightarrow$	$\{0, 3\}$	$\{4\}$	$\{1, 4\}$	$O$	$O$	$O$	$O$	$O$	$A$	$D$	$O$	$A$	$D$	$F$	$A$	$D$
	$\{4\}$	$\{5\}$	$\emptyset$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$E$	$O$	$F$	$E$	$O$	$F$
	$\{1, 4\}$	$\{2, 5\}$	$\emptyset$	$O$	$K$	$O$	$A$	$B$	$O$	$A$	$B$	$O$	$A$	$B$	$O$	$A$
	$\{5\}$	$\{5\}$	$\{5, 6\}$	$O$	$O$	$O$	$O$	$O$	$C$	$E$	$E$	$C$	$E$	$E$	$C$	$E$
	$\emptyset$	$\emptyset$	$\emptyset$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$
$\leftarrow$	$\{2, 5\}$	$\{5\}$	$\{5, 6\}$	$K$	$O$	$O$	$B$	$O$	$C$	$B$	$E$	$C$	$B$	$E$	$C$	$B$
	$\{5, 6\}$	$\{5, 7\}$	$\{5, 6, 7\}$	$O$	$K$	$K$	$C$	$B$	$K$	$C$	$B$	$K$	$C$	$B$	$K$	$C$
$\leftarrow$	$\{5, 7\}$	$\{5\}$	$\{5, 6\}$	$K$	$O$	$O$	$B$	$O$	$C$	$B$	$E$	$C$	$B$	$E$	$C$	$B$
$\leftarrow$	$\{5, 6, 7\}$	$\{5, 7\}$	$\{5, 6, 7\}$	$K$	$K$	$K$	$K$	$B$	$K$	$K$	$B$	$K$	$K$	$B$	$K$	$K$

DFA:



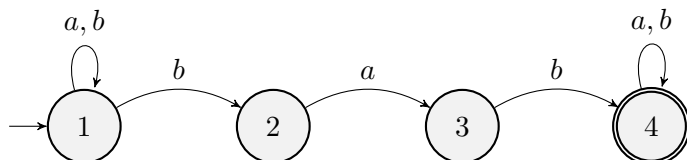
## 6.2 Návrh NFA a redukovaný DFA

Je dán jazyk  $L$  nad abecedou  $\{a, b\}$  takto:

$$L = \{w \mid w = ubabv; u, v \in \{a, b\}^*\},$$

tj.  $L$  se skládá ze všech slov, které obsahují slovo  $bab$  jako podslovo. Zkonstruuje nejprve NFA  $N$ , který přijímá  $L$ . Podmnožinovou konstrukcí k  $N$  zkonstruuje DFA a ten pak zredukujte.

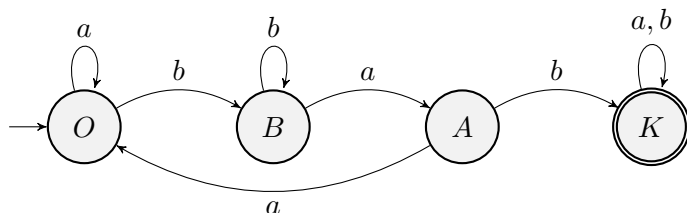
NFA:



Podmnožinová konstrukce:

		$a$	$b$	$\sim_0$	$a$	$b$	$\sim_1$	$a$	$b$	$\sim_2$	$a$	$b$	$\sim_3$
$\rightarrow$	$\{1\}$	$\{1\}$	$\{1, 2\}$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$B$	$O$
	$\{1, 2\}$	$\{1, 3\}$	$\{1, 2\}$	$O$	$O$	$O$	$O$	$A$	$O$	$B$	$A$	$B$	$B$
	$\{1, 3\}$	$\{1\}$	$\{1, 2, 4\}$	$O$	$O$	$K$	$A$	$O$	$K$	$A$	$O$	$K$	$A$
$\leftarrow$	$\{1, 2, 4\}$	$\{1, 3, 4\}$	$\{1, 2, 4\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$
$\leftarrow$	$\{1, 3, 4\}$	$\{1, 4\}$	$\{1, 2, 4\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$
$\leftarrow$	$\{1, 4\}$	$\{1, 4\}$	$\{1, 2, 4\}$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$	$K$

DFA:



## 6.3 Důkaz minimálního počtu stavů DFA pro jazyk

Zjistěte, jaký je minimální počet stavů DFA, který přijímá jazyk  $L_n = \{u1v \mid |v| = n - 1\}$  nad abecedou  $\Sigma = \{0, 1\}$ . Zdůvodněte. Jak by se změnil výsledek, kdyby bylo  $\Sigma = \{0, 1, 2\}$ ?

Jazyk  $L$  obsahuje všechny slova nad abecedou  $\Sigma$  takové, že slovo končí znakem 1 a za ním následuje přesně  $n - 1$  libovolných znaků z  $\Sigma$ . Automat tedy musí rozeznat, zda poslední symbol 1 padne přesně  $n$  pozic před koncem slova.

Tedy potřebujeme stav pro každý možný počet znaků přečtených od posledního 1, tj. od 0 do  $n$ . Pokud přečteme více než  $n$  znaků od posledního 1, můžeme tento stav sloučit do jednoho.

Proto minimální počet stavů pro abecedu  $\Sigma = \{0, 1\}$  je  $n + 1$ .

Rozšíření abecedy nemění zásadní strukturu automatu. DFA nad větší abecedou musí stále sledovat jen vzdálenost od posledního výskytu symbolu 1. Přítomnost dalších symbolů nemá vliv na logiku počítání.

Takže minimální počet stavů pro abecedu  $\Sigma = \{0, 1, 2\}$  je také  $n + 1$ .

## 6.4 Pumping lemma pro doplněk

Dokažte nebo vyvráťte toto tvrzení (Pumping lemma pro doplněk):

Pro každý regulární jazyk  $L$  nad abecedou  $\Sigma$  (tj. jazyk, který je přijímán nějakým DFA) existuje přirozené číslo  $n$  s touto vlastností:

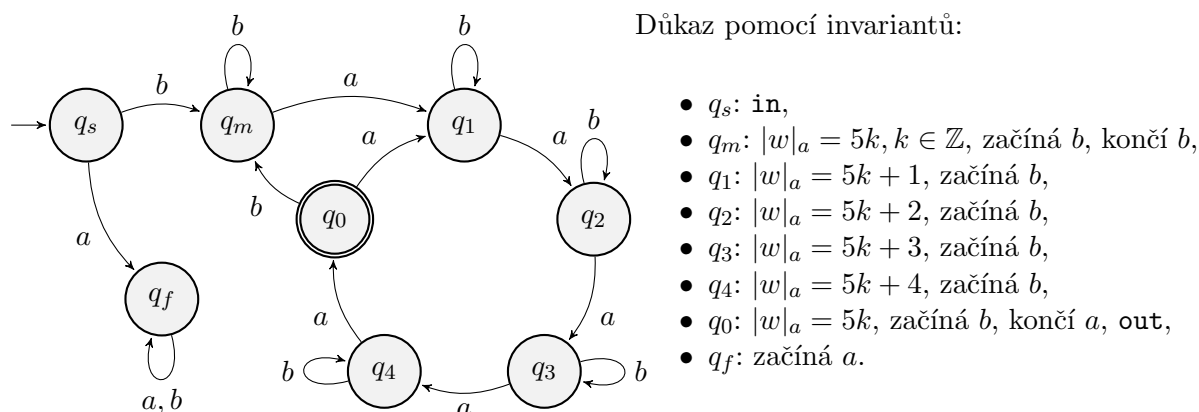
Každé slovo  $u \notin L$ , které je delší než  $n$  (tj.  $|u| > n$ ) lze rozdělit na tři slova  $u = xwy$ , tak, že

1.  $|xw| \leq n$ ,
2.  $w \neq \varepsilon$ ,
3. pro každé přirozené  $i = 0, 1, \dots$  platí  $xw^i y \notin L$ .

## 6.5 Návrh DFA dle jazyka

Navrhněte deterministický konečný automat (DFA), který přijímá jazyk  $L$  nad abecedou  $\{a, b\}$ , kde  $L$  obsahuje právě všechna slova  $w$  taková, že  $|w|_a$  je dělitelné 5,  $w$  začíná  $b$  a končí  $a$ .

O navrženém automatu ukažte, že opravdu přijímá daný jazyk.



## 6.6 Návrh DFA součinnou konstrukcí

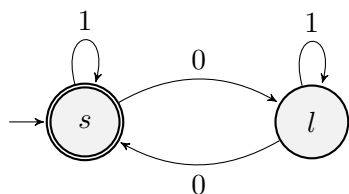
Navrhněte redukovaný DFA  $M$ , který přijímá jazyk  $L$  nad  $\Sigma = \{0, 1\}$ , kde

$$L = \{w \mid |w|_0 \text{ je sudé a za každým symbolem 1 je symbol 0}\}.$$

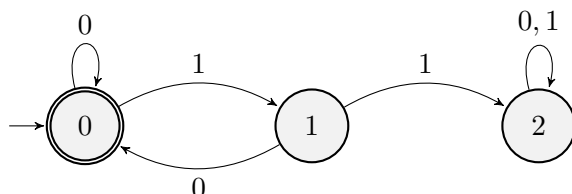
Postupujte buď součinnou konstrukcí nebo přímo. V druhém případě řádně zdůvodněte, proč  $M$  opravdu přijímá jazyk  $L$ .

Postup součinnou konstrukcí: Vytvoříme dva automaty, jeden pro pravidlo  $|w|_0$  je sudé, a druhý pro "za každým symbolem 1 je symbol 0".

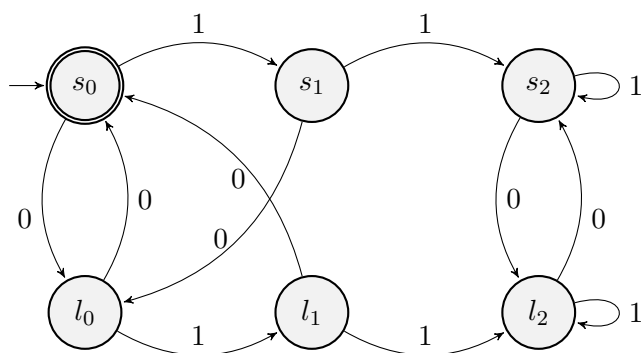
1.



2.



DFA sestavený součinnou konstrukcí dvou výše nakreslených automatů:

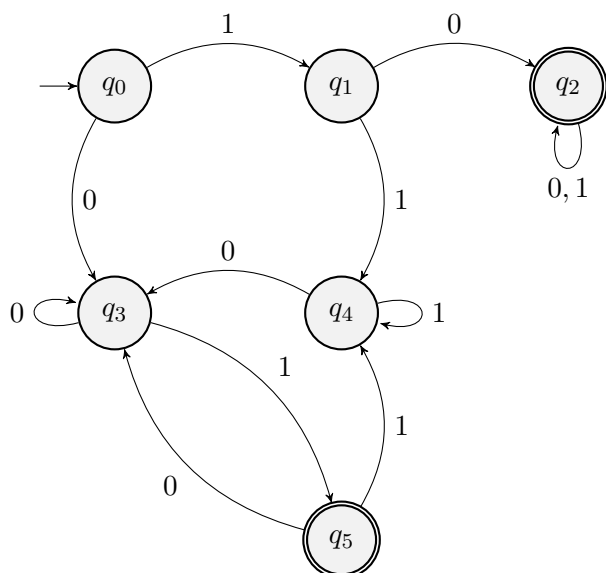


## 6.7 Návrh DFA

Navrhněte redukovaný DFA  $M$ , který přijímá jazyk  $L$  nad  $\Sigma = \{0, 1\}$ , kde

$$L = \{w \mid w \text{ začíná } 10 \text{ nebo končí } 01\}.$$

Zdůvodněte, proč  $M$  přijímá jazyk  $L$ .



Důkaz pomocí invariantů:

- $q_0$ : in,
- $q_1$ : začíná 1,
- $q_2$ : začíná 10, out,
- $q_3$ : nezačíná 10, končí 0,
- $q_4$ : nezačíná 10, končí 1,
- $q_5$ : nezačíná 10, končí 01, out.

Podmnožinová konstrukce:

		0	1	$\sim_0$	$a$	$b$	$\sim_1$	$a$	$b$	$\sim_2$
$\rightarrow$	$q_0$	$q_3$	$q_1$	$O$	$O$	$O$	$O$	$B$	$A$	$D$
	$q_1$	$q_2$	$q_4$	$O$	$K$	$O$	$A$	$K$	$O$	$A$
$\leftarrow$	$q_2$	$q_2$	$q_2$	$K$	$K$	$K$	$K$	$K$	$K$	$K$
	$q_3$	$q_3$	$q_5$	$O$	$O$	$K$	$B$	$B$	$C$	$B$
	$q_4$	$q_3$	$q_4$	$O$	$O$	$O$	$O$	$B$	$O$	$O$
$\leftarrow$	$q_5$	$q_3$	$q_4$	$K$	$O$	$O$	$C$	$B$	$O$	$C$

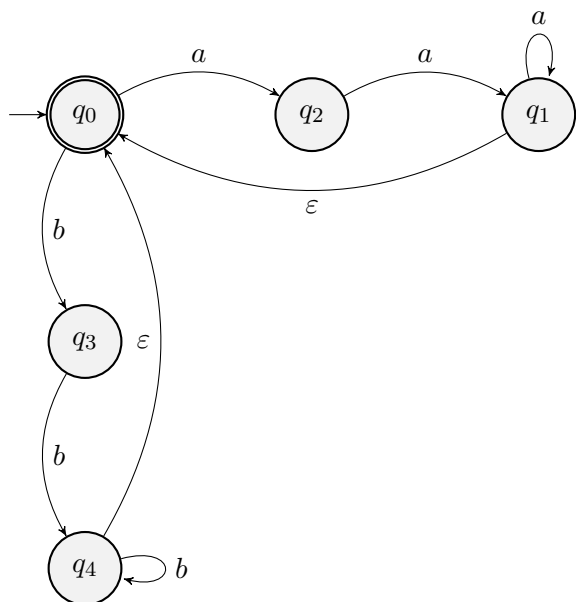
Protože má každý stav svou vlastní třídu, původní automat je již redukovaný.

## 7 Sedmé cvičení

### 7.1 Konstrukce redukovaného DFA k regulárnímu výrazu

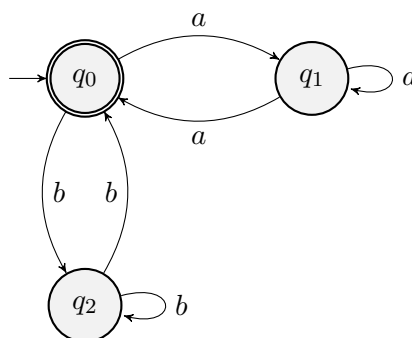
Je dán regulární výraz  $(aaa^* + bbb^*)^*$ . K danému výrazu sestrojte redukovaný DFA  $M$ , který přijímá jazyk reprezentovaný regulárním výrazem.

I.  $(aaa^* + bbb^*)^*$



zjednodušení: výraz se dá přepsat jako

II.  $(aa^*a + bb^*b)^*$



NFA:

		a	b
$\leftrightarrow$	$q_0$	$q_1$	$q_2$
	$q_1$	$\{q_0, q_1\}$	$\emptyset$
	$q_2$	$\emptyset$	$\{q_0, q_2\}$

Podmnožinová konstrukce pro II.

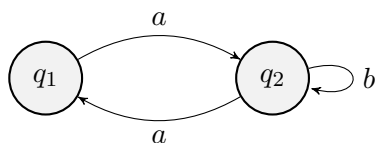
		a	b	$\sim_0$	a	b	$\sim_1$	a	b	$\sim_2$
$\leftrightarrow$	$q_0$	$q_1$	$q_2$	$K$	$O$	$O$	$A$	$B$	$C$	$A$
	$q_1$	$\{q_0, q_1\}$	$\emptyset$	$O$	$K$	$O$	$B$	$K$	$O$	$B$
$\leftarrow$	$q_2$	$\emptyset$	$\{q_0, q_2\}$	$O$	$O$	$K$	$C$	$O$	$D$	$C$
	$\{q_0, q_1\}$	$\{q_0, q_1\}$	$q_2$	$K$	$K$	$O$	$K$	$K$	$C$	$K$
	$\emptyset$	$\emptyset$	$\emptyset$	$O$	$O$	$O$	$O$	$O$	$O$	$O$
$\leftarrow$	$\{q_0, q_2\}$	$q_1$	$\{q_0, q_2\}$	$K$	$O$	$K$	$D$	$B$	$D$	$D$

### 7.2 Konstrukce redukovaného DFA k regulárnímu výrazu

Je dán regulární výraz  $(a + b(ab^*a)^*b)^*$ . K danému výrazu sestrojte redukovaný DFA  $M$ , který přijímá jazyk reprezentovaný regulárním výrazem.

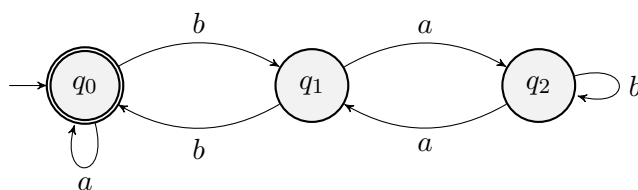
rozebereme si výraz:

veprostřed máme  $(ab^*a)^*$



a nabalujeme zbytek

$(a + b(ab^*a)^*b)^*$



Podmnožinová konstrukce.

	$a$	$b$	$\sim_0$	$a$	$b$	$\sim_1$
$\leftrightarrow$	$q_0$	$q_0 \ q_1$	$K$	$K$	$O$	$K$
	$q_1$	$q_2 \ q_0$	$O$	$O$	$K$	$A$
	$q_2$	$q_1 \ q_2$	$O$	$O$	$O$	$O$

Protože každý řádek má svou třídu, automat je již redukovaný.

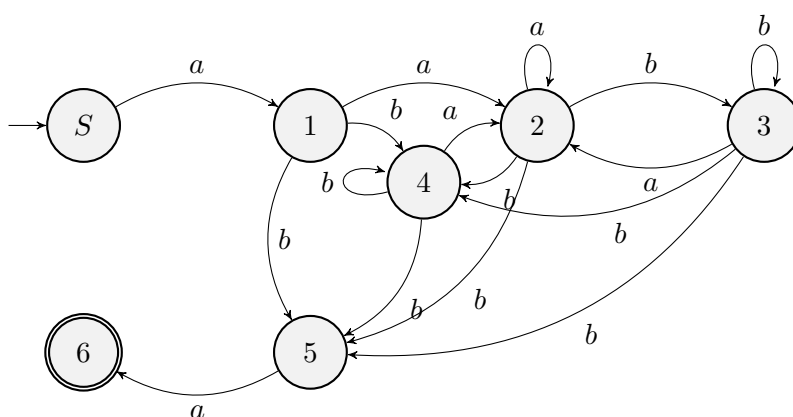
### 7.3 Konstrukce redukovaného DFA k regulárnímu výrazu

Je dán regulární výraz  $\mathbf{a(ab^* + b)^*ba}$ . K danému regulární výrazu sestrojte redukovaný DFA  $M$ , který přijímá jazyk reprezentovaný regulárním výrazem.

$\mathbf{a(ab^* + b)^*ba}$ :

$a_1(a_2b_3^* + b_4)^*b_5a_6$

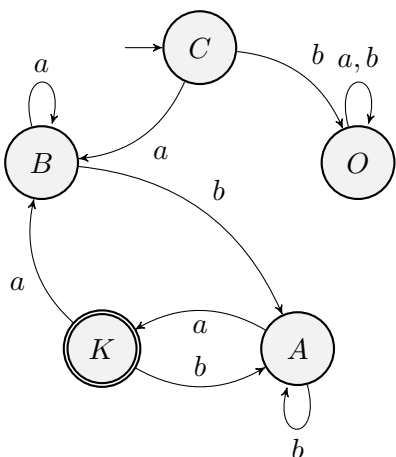
NFA	$a$	$b$
$\rightarrow$	$S$	1 —
	1	2 4, 5
	2	2 3, 4, 5
	3	2 3, 4, 5
	4	2 4, 5
	5	6 —
$\leftarrow$	6	— —



Podmnožinová konstrukce.

	$a$	$b$	$\sim_0$	$a$	$b$	$\sim_1$	$a$	$b$	$\sim_2$	$a$	$b$	$\sim_3$	$a$	$b$	$\sim_4$
$\rightarrow$	$\{S\}$	$\{1\}$	$\emptyset$	$O$	$O$	$O$	$O$	$O$	$O$	$B$	$O$	$C$	$B$	$O$	$C$
	$\{1\}$	$\{2\}$	$\{4, 5\}$	$O$	$O$	$O$	$O$	$O$	$A$	$B$	$B$	$A$	$B$	$B$	$A$
	$\emptyset$	$\emptyset$	$\emptyset$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$	$O$
	$\{2\}$	$\{2\}$	$\{3, 4, 5\}$	$O$	$O$	$O$	$O$	$O$	$A$	$B$	$B$	$A$	$B$	$B$	$A$
	$\{4, 5\}$	$\{2, 6\}$	$\{4, 5\}$	$O$	$K$	$O$	$A$	$K$	$A$	$A$	$K$	$A$	$A$	$K$	$A$
	$\{3, 4, 5\}$	$\{2, 6\}$	$\{3, 4, 5\}$	$O$	$K$	$O$	$A$	$K$	$A$	$A$	$K$	$A$	$A$	$K$	$A$
$\leftarrow$	$\{2, 6\}$	$\{2\}$	$\{3, 4, 5\}$	$K$	$O$	$O$	$K$	$O$	$A$	$K$	$B$	$A$	$K$	$B$	$A$

DFA:



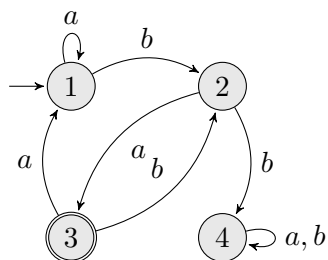
## 7.4 Tvorba regulárního výrazu z DFA

Pro daný DFA  $M$  vytvořte regulární výraz, který reprezentuje jazyk  $L(M)$ .

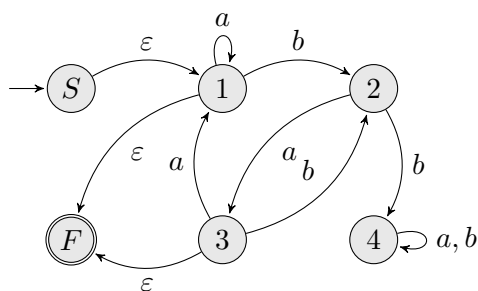
$M$ :

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

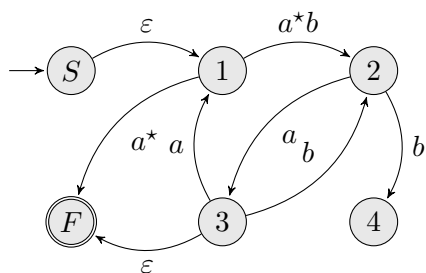
0. DFA:



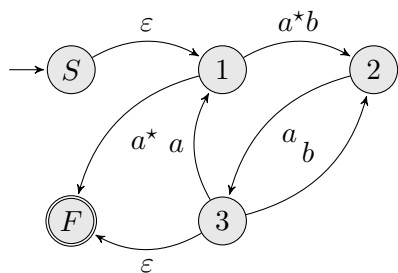
1. zavedu stavy  $S$ ,  $F$ :



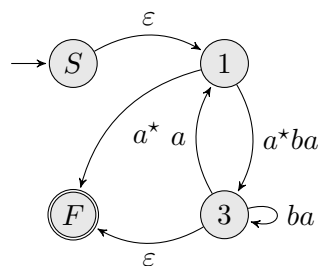
2. odstraňuji smyčky:



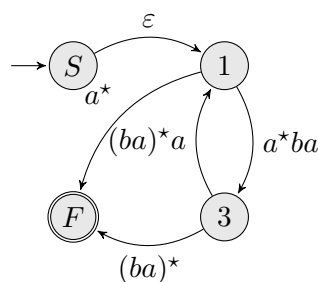
3. odstraňuji vrchol 4:



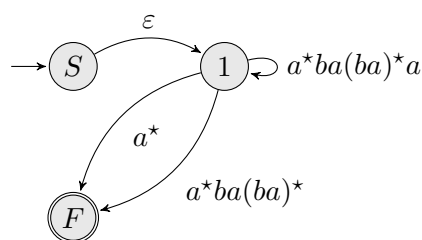
4. odstraňuji vrchol 2:



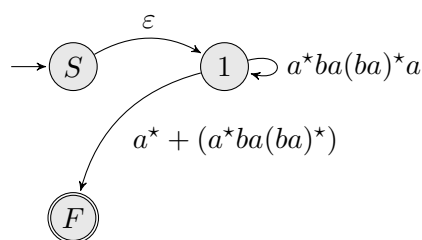
5. odstraňuji smyčky:



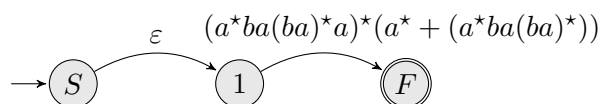
6. odstraňuji vrchol 3:



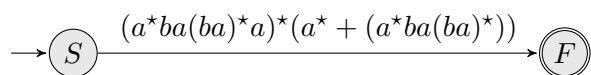
7. odstraňuji paralelní hrany:



8. odstraňuji smyčky:

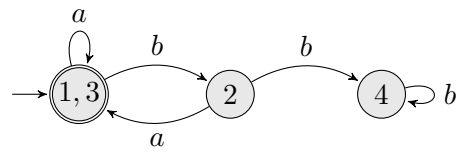


9. odstraňuji vrchol 1:

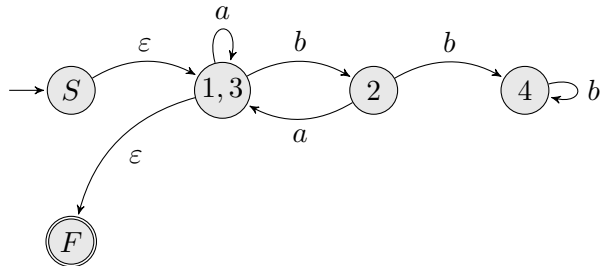


lifehack: dá se redukovat (stavy 1 a 3 jsou ekvivalentní):

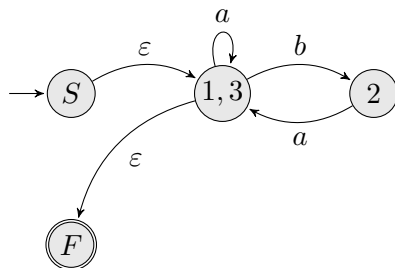
	$a$	$b$
$\leftrightarrow$	1	1 2
	2	3 4
$\leftarrow$	3	1 2
	4	4 4



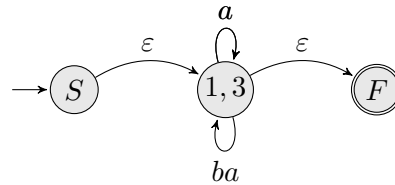
1. přidám  $S$ ,  $F$ :



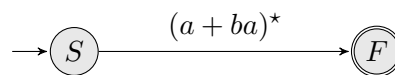
2. zbavím se nodu č. 4:



3. odstraním node č. 2:



4. spojím obsah smyček, odstraním smyčku a node:





## 8 Osmé cvičení

### 8.1 Konstrukce regulární gramatiky k automatu

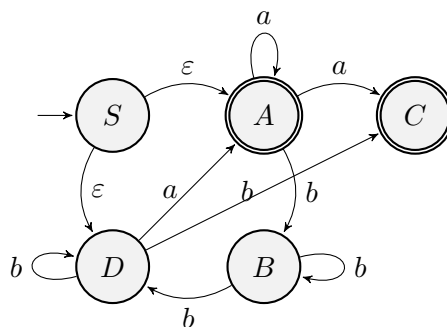
K automatu  $M$ , který je dán následující tabulkou, zkonstruuje regulární gramatiku  $\mathcal{G}$ , která generuje jazyk  $L = L(M)$ .

$M$ :

	$a$	$b$
$\leftrightarrow$ $A$	$A, C$	$B$
$B$	$\emptyset$	$B, D$
$\leftarrow$ $C$	$\emptyset$	$\emptyset$
$\rightarrow$ $D$	$A$	$C, D$

$$\begin{aligned}\mathcal{G} &= (N, \Sigma, S, P) \\ N &= \{S, A, B, C, D\} \\ \Sigma &= \{a, b\}\end{aligned}$$

mám více vstupů  $\rightarrow$  přidám si  $S$



$$\begin{aligned}P : & S \rightarrow A \mid D \\ & A \rightarrow aA \mid aC \mid bB \mid \varepsilon \\ & B \rightarrow bB \mid bD \\ & C \rightarrow \varepsilon \\ & D \rightarrow aA \mid bC \mid bD\end{aligned}$$

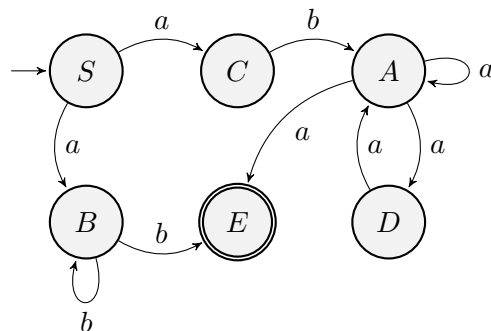
### 8.2 Tvorba DFA ke gramatice 3. typu

Ke gramatice  $\mathcal{G}$  typu 3 zkonstruuje konečný automat, který přijímá jazyk  $L(\mathcal{G})$ . Gramatika  $\mathcal{G} = (N, \{a, b\}, S, P)$ , kde  $N = \{S, A, B\}$  a pravidla jsou

$$\begin{aligned}P : & S \rightarrow abA \mid aB \\ & A \rightarrow aA \mid aaA \mid a \\ & B \rightarrow bB \mid b\end{aligned}$$

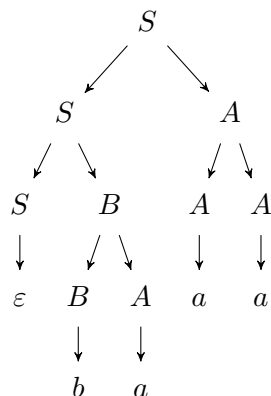
sestavím automat podle nových pravidel

$$\begin{aligned}P : & S \rightarrow abA \mid aB && \text{překáží mi } abA \\ & C \rightarrow bA && \\ & S \rightarrow aC \mid aB && \text{nové pravidlo } S \\ & A \rightarrow aA \mid aaA \mid a && \text{zase nám vadí } aaA \\ & D \rightarrow aA && \\ & E \rightarrow \varepsilon && \\ & A \rightarrow aA \mid aD \mid aE && \text{nové pravidlo } A \\ & B \rightarrow bB \mid bE && \text{nové pravidlo } B\end{aligned}$$



### 8.3 Práce s derivačním stromem bezkontextové gramatiky

Je dán derivační strom v bezkontextové gramatice:



- Napište pravidla minimální CF gramatiky, ve které je to derivační strom.
- Napište levou derivaci odpovídající tomuto derivačnímu stromu.
- Rozhodněte, zda je gramatika víceznačná.

a)

$$P : \begin{aligned} S &\rightarrow SA \mid SB \mid \varepsilon \\ A &\rightarrow AA \mid a \\ B &\rightarrow BA \mid b \end{aligned}$$

c) je víceznačná - tj. je možné alespoň jedno slovo vygenerovat dvěma způsoby. Například díky pravidlu  $A \rightarrow AA$ .

b)  $S \xRightarrow{S \rightarrow SA} SA \xRightarrow{S \rightarrow SB} SBA \xRightarrow{S \rightarrow \varepsilon} BA \xRightarrow{B \rightarrow BA} BAA \xRightarrow{B \rightarrow b} bAA \xRightarrow{A \rightarrow a} baA \xRightarrow{A \rightarrow AA} baAA \xRightarrow{A \rightarrow a^{(2)}} baaa.$

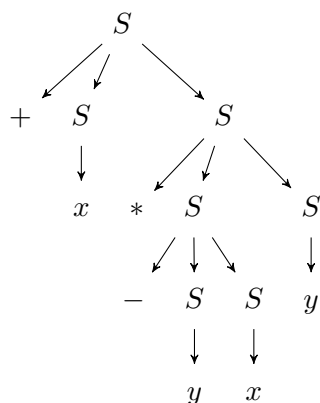
### 8.4 Tvorba derivačního stromu k bezkontextové gramatice

Je dána bezkontextová gramatika  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S\}$ ,  $\Sigma = \{+, \star, -, x, y\}$ , s pravidly

$$S \rightarrow +SS \mid \star SS \mid -SS \mid x \mid y$$

- Nakreslete derivační strom, který má za výsledek slovo  $w = +x \star -yxy$ .
- Zkonstruuje levou derivaci slova  $w$  odpovídající derivačnímu stromu z části a).

1.



2.

$$\begin{aligned} S &\xRightarrow{S \rightarrow +SS} +SS \xRightarrow{S \rightarrow x} +xS \xRightarrow{S \rightarrow \star SS} +x \star SS \xRightarrow{S \rightarrow -SS} +x \star -SSS \xRightarrow{S \rightarrow y} +x \star -ySS \xRightarrow{S \rightarrow x} +x \star -yxy \end{aligned}$$

## 8.5 Návrh bezkontextové gramatiky pro jazyk

Navrhněte bezkontextovou gramatiku  $\mathcal{G}$ , která generuje jazyk  $L = \{0^i 1^i 2^j; i, j \geq 0\}$ . Zdůvodněte, proč gramatika  $\mathcal{G}$  jazyk  $L$  generuje.

$$\begin{aligned} P: S &\rightarrow XY \\ X &\rightarrow 0X1 \mid \varepsilon \\ Y &\rightarrow Y2 \mid \varepsilon \end{aligned}$$

1.  $L \subseteq L(\mathcal{G})$  (gramatika vygeneruje vše):

$$S \xrightarrow{S \rightarrow XY} XY \xrightarrow{X \rightarrow 0X1(i)} 0^i X 1^i Y \xrightarrow{Y \rightarrow 2Y(j)} 0^i X 1^i Y 2^j \xrightarrow{X \rightarrow \varepsilon} 0^i 1^i Y 2^j \xrightarrow{Y \rightarrow \varepsilon} 0^i 1^i 2^j.$$

2.  $L(\mathcal{G}) \subseteq L$  (gramatika nevygeneruje nic navíc):

Uvažujme derivaci  $S \Rightarrow \star w$ . Pak poslední použité pravidlo musí být  $X \rightarrow \varepsilon$  nebo  $Y \rightarrow \varepsilon$ . Proto v derivaci musí být použito pravidlo  $S \rightarrow XY$ . Mezi tím může být použit nějaký počet pravidel  $X \rightarrow 0X1$  a  $Y \rightarrow Y2$ . Jinak pravidla být použita nemohou. Tedy drivace má tvar  $S \xrightarrow{S \rightarrow XY} XY \xrightarrow{X \rightarrow 0X1(i)} 0^i X 1^i Y \xrightarrow{Y \rightarrow 2Y(j)} 0^i X 1^i Y 2^j \xrightarrow{X \rightarrow \varepsilon} 0^i 1^i Y 2^j \xrightarrow{Y \rightarrow \varepsilon} 0^i 1^i 2^j$ .

## 8.6 Tvorba nevypouštěcí gramatiky

Ke gramatice  $\mathcal{G}$  zkostruujte nevypouštěcí gramatiku  $\mathcal{G}_1$ , pro kterou  $L(\mathcal{G}_1) = L(\mathcal{G}) - \{\varepsilon\}$ .

$$\begin{aligned} P: S &\rightarrow aSbA \mid \varepsilon \\ A &\rightarrow aBbA \mid bCB \mid CD \\ B &\rightarrow bbBa \mid aS \\ C &\rightarrow aAaA \mid \varepsilon \\ D &\rightarrow SC \mid aABa \end{aligned}$$

obecný formální zápis u nevypouštěcích gramatik:

$$\begin{aligned} V &= \{A \mid A \Rightarrow \star \varepsilon\} \\ V_1 &= \{A \mid A \rightarrow \varepsilon \in P\} \\ V_2 &= V_1 \cup \{A \mid A \rightarrow \alpha \in P, \alpha \in V_1^*\} \\ V_{i+1} &= V_i \cup \{A \mid A \rightarrow \alpha \in P, \alpha \in V_i^*\} \end{aligned}$$

příklad:

$$\begin{aligned} V &= \{A \mid A \Rightarrow \star \varepsilon\} \\ V_1 &= \{A \mid A \rightarrow \varepsilon \in P\} = \{S, C\} \\ V_2 &= V_1 \cup \{A \mid A \rightarrow \alpha \in P, \alpha \in V_1^*\} = V_1 \cup \{D\} = \\ &= \{S, C, D\} \\ V_3 &= V_2 \cup \{A\} = \{S, A, C, D\} \end{aligned}$$

$\mathcal{G}_1$ :

$$\begin{aligned} P: S &\rightarrow aSbA \mid abA \mid aSb \mid ab \\ A &\rightarrow aBbA \mid aBb \mid bCB \mid bB \mid CD \mid C \mid D \\ B &\rightarrow bbBa \mid aS \mid a \\ C &\rightarrow aAaA \mid aAa \mid aaA \mid aa \\ D &\rightarrow SC \mid S \mid C \mid aABa \mid aBa \end{aligned}$$

## 8.7 Konstrukce gramatiky 3. typu k automatu

K automatu  $M$  zkonstruujte gramatiku typu 3, která generuje jazyk  $L(M)$ , kde  $M$  je dán tabulkou

		$a$	$b$
$\rightarrow$	$A$	$\{A, B\}$	$\{C\}$
	$B$	$\{B\}$	$\{C\}$
$\leftrightarrow$	$C$	$\emptyset$	$\{D\}$
$\leftarrow$	$D$	$\{B\}$	$\{D\}$

$\mathcal{G} = (N, \Sigma, S, P)$

$N = \{S, A, B, C, D\}$

$\Sigma = \{a, b\}$

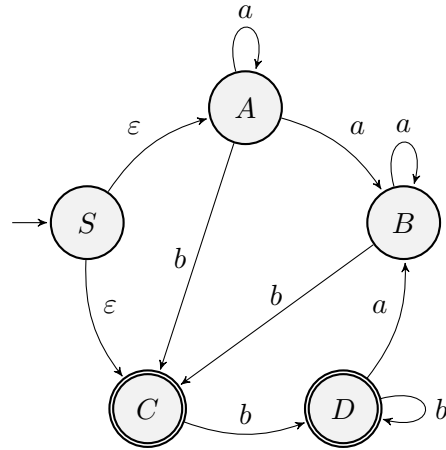
$P : S \rightarrow A \mid C$

$A \rightarrow aA \mid aB \mid bC$

$B \rightarrow aB \mid bC$

$C \rightarrow bD \mid \varepsilon$

$D \rightarrow aB \mid bD \mid \varepsilon$



## 8.8 Návrh bezkontextové gramatiky

Navrhněte bezkontextovou gramatiku  $\mathcal{G}$ , která generuje jazyk  $L = \{0^i 1^j; 0 \leq i \leq j\}$ . Zdůvodněte, proč gramatika  $\mathcal{G}$  jazyk  $L$  generuje.

$S \rightarrow XY$

$X \rightarrow 0X1 \mid \varepsilon$

$Y \rightarrow Y1 \mid \varepsilon$

Zdůvodnění:

1. Dvě možnosti:  $i = j$ , a  $i < j$ , kde  $j = i + n$ ,  $n > 0$ .

$$S \xrightarrow{S \rightarrow XY} XY \xrightarrow{X \rightarrow 0X1(i)} 0^i X 1^i Y \Rightarrow \begin{cases} i < j: & 0^i X 1^i Y \xrightarrow{Y \rightarrow Y1(n)} 0^i X 1^i Y 1^n \xrightarrow{X \rightarrow \varepsilon} 0^i 1^i Y 1^n \xrightarrow{Y \rightarrow Y1} 0^i 1^{i+n=j} \\ i = j: & 0^i X 1^i Y \xrightarrow{Y \rightarrow \varepsilon} 0^i X 1^i \xrightarrow{X \rightarrow \varepsilon} 0^i 1^i \end{cases}$$

2. (fancy důkaz, doslova převzato z autorského řešení paní doc. Demlové)

Uvažujme derivaci  $S \Rightarrow^* w$ . Poslední pravidlo musí být  $S \rightarrow \varepsilon$ .

Provedeme indukci podle počtu kroků derivace  $n$ :

$$S \Rightarrow^n 0^i S 1^j, \quad \text{kde } i \leq j.$$

**Základní krok** ( $n = 1$ ): Pro  $n = 1$ :

$$S \rightarrow 0S1 \quad \text{nebo} \quad S \rightarrow S1, \quad \text{a tedy } 0^i S 1^j, \text{ kde } i \leq j.$$

**Indukční krok:** Předpokládejme, že každá derivace o  $n$  krocích generuje:

$$S \Rightarrow^n 0^i S 1^j, \quad i \leq j.$$

Pak derivace o  $n + 1$  krocích bude:

$$S \rightarrow 0S1 \Rightarrow^n 0^{i+1} S 1^{j+1}, \quad \text{a tedy } i + 1 \leq j + 1.$$

Nebo:

$$S \Rightarrow^n 0^i S 1^j \Rightarrow 0^i 1^j.$$

**Závěr:** Z  $S$  je možné odvodit právě slova  $0^i 1^j$ , kde  $0 \leq i \leq j$ , a nic jiného.

## 9 Deváté cvičení

### 9.1 Návrh bezkontextové gramatiky

Navrhněte bezkontextové gramatiky generující následující jazyky

- a)  $L_1 = \{0^{m+n}1^n0^m \mid 0 \leq n, m\}$ .  
b)  $L_2 = \{0^i1^j \mid 0 \leq i < j\}$ .

Zdůvodněte, proč gramatika  $\mathcal{G}$  jazyk  $L$  generuje.

a)

$$P: S \rightarrow 0S0 \mid A \\ A \rightarrow 0A1 \mid \varepsilon$$

$$(a) \quad S \xrightarrow{S \rightarrow 0S0(m)} 0^m S 0^m \xrightarrow{S \rightarrow A} 0^m A 0^m \xrightarrow{A \rightarrow 0A1(n)} 0^m 0^n A 1^n 0^m \xrightarrow{A \rightarrow \varepsilon} 0^m 1^m 0^n 1^n 0^m = 0^{m+n} 1^n 0^m \\ (b) \quad S \Rightarrow^* w, S \Rightarrow 0^i S 0^i, S \Rightarrow A, A \Rightarrow^* 0^j 1^j, 0^i 0^j 1^j 0^i \in L(\mathcal{G})$$

b)

$$P: S \rightarrow 0S1 \mid S1 \mid 1$$

Důkazy mě těžce nebaví, všude jsou cca stejné.

### 9.2 Konstrukce nevypouštěcí gramatiky

Ke gramatice  $\mathcal{G}$  zkonstruujte nevypouštěcí gramatiku  $\mathcal{G}_1$ , pro kterou  $L(\mathcal{G}_1) = L(\mathcal{G}) - \{\varepsilon\}$ . Gramatiku  $\mathcal{G}_1$  zredukujte.

$$P: S \rightarrow AB \mid \varepsilon \\ A \rightarrow aAAb \mid bS \mid CA \\ B \rightarrow BbA \mid CaC \mid \varepsilon \\ C \rightarrow aBB \mid bS$$

$$V_1 = \{A \mid A \rightarrow \varepsilon \in P\} = \{S, B\} \\ V_2 = V_1 \cup \{A \mid A \rightarrow \alpha \in P, \alpha \in V_1^*\} = V_1 \cup \emptyset = \{S, B\}$$

$$\mathcal{G}_1: S \rightarrow AB \mid A \\ A \rightarrow aAAb \mid bS \mid b \mid CA \\ B \rightarrow BbA \mid bA \mid CaC \\ C \rightarrow aBB \mid aB \mid a \mid bS \mid b$$

Gramatika  $\mathcal{G}_1$  už je redukována.

#### Obecný postup pro redukci

$$V = \{A \mid A \in N, A \Rightarrow_{\mathcal{G}}^* w, w \in \Sigma^*\} \\ V_1 = \{A \mid A \Rightarrow_{\star} w \in P, w \in \Sigma^*\} \\ V_2 = V_1 \cup \{A \mid A \rightarrow \alpha \in P, \alpha \in (\Sigma \cup V_1)^*\} \\ U = \{A \mid A \in V, \exists \alpha, \beta \in (V \cup \Sigma)^* \text{ tak, že } S \Rightarrow_{\mathcal{G}}^* \alpha A \beta\}$$

Jazyk není prázdný právě tehdy, kdy  $S \in V$ .

### 9.3 Redukce gramatiky

Zredukujte gramatiku  $\mathcal{G}$ , která je dána pravidly

$$\begin{aligned} P: S &\rightarrow SA \mid SB \mid \varepsilon \\ A &\rightarrow bSA \mid baS \\ B &\rightarrow aB \mid Ba \mid DA \\ C &\rightarrow aCB \mid bA \\ D &\rightarrow AB \end{aligned}$$

redukce tldr:

$V_1 \dots$  to, co se promítne na  $\varepsilon$  nebo na terminály

$V_2 \dots$  to, co se promítne na terminály a na to, co už je ve  $V_1$

$U_0 \dots \{S\}$

$U_1 \dots$  neterminály, do kterých se dostanu z  $S$ , pak z  $U_1$  atd.

$$V_1 = \{S\}$$

$$V_2 = V_1 \cup \{A\} = \{S, A\}$$

$$V_3 = V_2 \cup \{C\} = \{S, A, C\}$$

$$V_4 = V_3 \cup \emptyset = \{S, A, C\}$$

Zde nechám jenom neterminály z  $V$  a z pravé strany vyškrtám pravidla obsahující neterminály  $\notin V$ :

$$\begin{aligned} P: S &\rightarrow SA \mid \varepsilon \\ A &\rightarrow bSA \mid baS \\ C &\rightarrow bA \end{aligned}$$

Sem přidávám neterminály, do kterých se dostanu z počátečního stavu  $S$ , pak ze stavů v odpovídajícím  $U_i$ :

$$U_0 = \{S\}$$

$$U_{i+1} = U_i \cup \left\{ A \mid \text{existují } B \in U_i, \alpha, \beta \in (V \cup \Sigma)^* \text{ tak, že } B \rightarrow \alpha A \beta \in P' \right\}$$

$$U_1 = U_0 \cup \{A\} = \{S, A\}$$

$$U_2 = U_1 \cup \emptyset = \{S, A\}$$

A ponechám jen pravidla, která nám zbyla v  $U$ :

$$\begin{aligned} P: S &\rightarrow SA \mid \varepsilon \\ A &\rightarrow bSA \mid baS \end{aligned}$$

### 9.4 Důkaz, že gramatika generuje alespoň jedno slovo

Rozhodněte, zda gramatika  $\mathcal{G}$  generuje alespoň jedno slovo, tj. zda  $L(\mathcal{G}) \neq \emptyset$ , kde  $\mathcal{G}$  je dána pravidly

$$\begin{aligned} P: S &\rightarrow aS \mid AB \mid CD \\ A &\rightarrow aDb \mid AD \mid BC \\ B &\rightarrow bSb \mid BB \\ C &\rightarrow BA \mid ASb \\ D &\rightarrow ABCD \mid \varepsilon \end{aligned}$$

$$V_1 = \{D\}$$

$$V_2 = V_1 \cup \{A\} = \{D, A\}$$

$$V_3 = V_2 \cup \emptyset = \{D, A\} = V$$

$S \notin V \rightarrow L(\mathcal{G}) = \emptyset$ . Tedy gramatika  $\mathcal{G}$  negeneruje ani jedno slovo.

## Chomského normální tvar polopatě

Všechna pravidla na pravé straně mají buď přesně 2 neterminály nebo přesně 1 terminál.

### 9.5 Převod gramatiky do Chomského normálního tvaru

Je dána gramatika  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, A, B\}$ ,  $\Sigma = \{0, 1\}$  a  $P$  je

$$\begin{aligned}P: S &\rightarrow A \mid 0SA \mid \varepsilon \\ A &\rightarrow 1A \mid B1 \mid 1 \\ B &\rightarrow 0B \mid 0\end{aligned}$$

Převeďte  $\mathcal{G}$  do Chomského normálního tvaru.

**1. krok** Vytvořit nevypouštěcí gramatiku.

$$V = \{S\}$$

$$\begin{aligned}P: S &\rightarrow A \mid 0SA \mid 0A \\ A &\rightarrow 1A \mid B1 \mid 1 \\ B &\rightarrow 0B \mid 0\end{aligned}$$

**2. krok** Nahrazení právě jednoho neterminálu pravých stran jeho pravidly.

zde:  $S \rightarrow A$ , máme zde  $A \rightarrow 1A \mid B1 \mid 1$

$$\begin{aligned}P: S &\rightarrow 1A \mid B1 \mid 1 \mid 0SA \mid 0A \\ A &\rightarrow 1A \mid B1 \mid 1 \\ B &\rightarrow 0B \mid 0\end{aligned}$$

vytvořím pomocná pravidla pro terminály, které "nezůstaly samy"

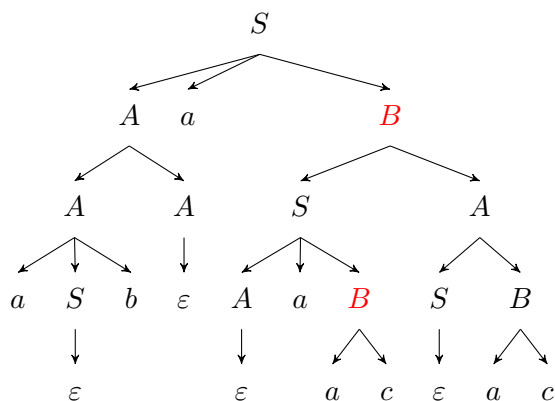
$$\begin{aligned}P: S &\rightarrow X_1A \mid BX_1 \mid 1 \mid X_0SA \mid X_0A \\ A &\rightarrow X_1A \mid BX_1 \mid 1 \\ B &\rightarrow X_0B \mid 0 \\ X_0 &\rightarrow 0 \\ X_1 &\rightarrow 1\end{aligned}$$

**3. krok** Zbavím se dlouhých slov ( $\geq 3$ ) (opět vytvořím pomocná pravidla).

$$\begin{aligned}P: S &\rightarrow X_1A \mid BX_1 \mid 1 \mid X_0Y \mid X_0A \\ Y &\rightarrow SA \\ A &\rightarrow X_1A \mid BX_1 \mid 1 \\ B &\rightarrow X_0B \mid 0 \\ X_0 &\rightarrow 0 \\ X_1 &\rightarrow 1\end{aligned}$$

### 9.6 Práce s derivačním stromem bezkontextové gramatiky

Je dán derivační strom v bezkontextové gramatice:



- Napište pravidla minimální CF gramatiky, ve které je to derivační strom.
- Napište levou derivaci odpovídající tomuto derivačnímu stromu.
- Rozložte výsledek derivačního stromu  $w$  na pět částí  $w = w_1w_2w_3w_4w_5$  tak, že  $w_2w_4 \neq \varepsilon$  a slovo  $w_1w_2^2w_3w_4^2w_5$  je také generované gramatikou z bodu a).
- Rozhodněte, zda je gramatika víceznačná.

a)  $P: S \rightarrow AaB \mid \varepsilon$

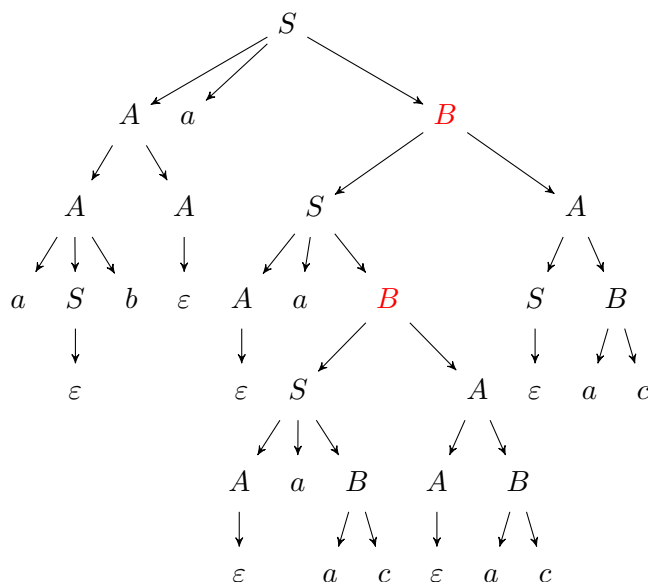
$A \rightarrow AA \mid SB \mid aSb \mid \varepsilon$

$B \rightarrow SA \mid ac$

b)  $S \xrightarrow{S \rightarrow AaB} AaB \xrightarrow{A \rightarrow AA} AAaB \xrightarrow{A \rightarrow aSb} aSbAaB \xrightarrow{S \rightarrow \varepsilon} abAaB \xrightarrow{A \rightarrow \varepsilon} abaB \xrightarrow{B \rightarrow SA} abaSA \xrightarrow{A \rightarrow Aab} abaAaBA \xrightarrow{A \rightarrow \varepsilon} abaaBA \xrightarrow{B \rightarrow ac} abaaacA \xrightarrow{A \rightarrow SB} abaaacSB \xrightarrow{S \rightarrow \varepsilon} abaaacB \xrightarrow{B \rightarrow ac} abaaacac.$

c) basically pumping lemma pro gramatiky

$w_1 = aba, w_2 = a, w_3 = ac, w_4 = ac, w_5 = \varepsilon.$



Jak toho docílím: jdu odspoda stromu a najdu dva stejné neterminály (různě v grafu) a v podstatě ten strom nafouknu (zkopíruju nějaký podstrom z vyššího do nižšího neterminálu, v ukázce červený node  $B$  v nejvýš vpravo kopíruju do červeného  $B$ čka vlevo od něj).

d) je víceznačná, přepisujeme  $A \rightarrow AA$ .



## 10 Desáté cvičení

### 10.1 Algoritmus CYK

Je dána gramatika  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, A, B, C, D\}$ ,  $\Sigma = \{a, b\}$  a pravidla  $P$  jsou dána:

Přepis pravidel:

$AB \leftarrow S$   
 $AC \leftarrow A$   
 $AD \leftarrow S, A$   
 $BA \leftarrow D$   
 $BC \leftarrow B$   
 $CS \leftarrow S$   
 $DS \leftarrow C$   
 $SC \leftarrow C$

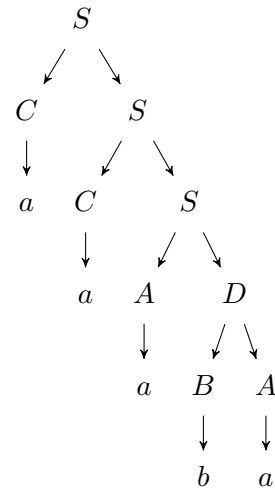
$P: S \rightarrow AB \mid CS \mid AD$   
 $A \rightarrow AC \mid AD \mid a$   
 $B \rightarrow BC \mid b$   
 $C \rightarrow DS \mid SC \mid a$   
 $D \rightarrow BA \mid b$

Algoritmicky rozhodněte, zda gramatika  $\mathcal{G}$  generuje slova  $w_1$  a  $w_2$ , kde  $w_1 = aaaba$  a  $w_2 = abbaa$ . Pokud ano, nakreslete derivační strom a napište jemu odpovídající levou derivaci.

slovo  $w_1$ :

$C, A, S$				
$S, A$	$C, A, S$			
$A$	$S, A$	$S, A, C$		
$A$	$A$	$S, A$	$D, B$	
$A, C$	$A, C$	$A, C$	$B, D$	$A, C$
$a$	$a$	$a$	$b$	$a$

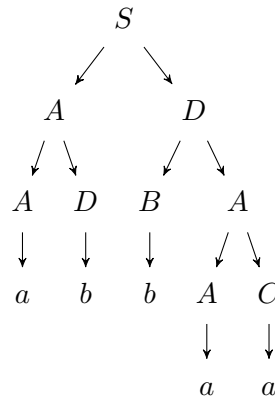
$S \xrightarrow{CS} CS \xrightarrow{C \rightarrow a} aS \xrightarrow{S \rightarrow CS} aCS \xrightarrow{C \rightarrow a} aaS \xrightarrow{S \rightarrow AD} aaAD \xrightarrow{A \rightarrow a} aaaD \xrightarrow{D \rightarrow BA} aaaBA \xrightarrow{B \rightarrow b} aaabA \xrightarrow{A \rightarrow a} aaaba.$



slovo  $w_2$ :

$C, A, S$				
$C, A, S$	—			
$S, A$	—	$D, B$		
$S, A$	—	$D, B$	$A$	
$A, C$	$B, D$	$B, D$	$A, C$	$A, C$
$a$	$b$	$b$	$a$	$a$

$S \xrightarrow{AD} AD \xrightarrow{A \rightarrow AD} ADD \xrightarrow{A \rightarrow a} aDD \xrightarrow{D \rightarrow b} abD \xrightarrow{D \rightarrow BA} abBA \xrightarrow{B \rightarrow b} abbA \xrightarrow{A \rightarrow AC} abbAC \xrightarrow{A \rightarrow a} abbaC \xrightarrow{C \rightarrow a} abbaa.$



## 10.2 Algoritmus CYK

Je dána gramatika  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, A, B, C, D\}$ ,  $\Sigma = \{a, b\}$  a pravidla  $P$  jsou dána:

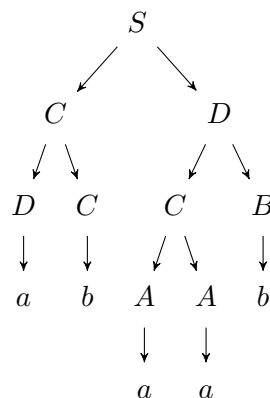
$$\begin{aligned} P: S &\rightarrow BD \mid CD \mid DA \\ A &\rightarrow CA \mid a \\ B &\rightarrow CB \mid b \\ C &\rightarrow AA \mid BC \mid DC \mid b \\ D &\rightarrow AC \mid BB \mid CB \mid a \end{aligned}$$

Algoritmem CYK rozhodněte, zda slovo  $w_1 = abaab$  je touto gramatikou generováno. Pokud ano, nakreslete derivační strom a napište levou derivaci.

$w_1 = abaab$ :

$D, C, B, S$				
$D, C, A, S$	$S, D, C, B$			
$A, S, C$	$C$	$D, B, C$		
$D, C$	$S, A$	$C, S$	$D, C$	
$A, D$	$B, C$	$A, D$	$A, D$	$B, C$
$a$	$b$	$a$	$a$	$b$

$$\begin{aligned} S &\xrightarrow{S \rightarrow CD} CD \xrightarrow{C \rightarrow DC} DCD \xrightarrow{D \rightarrow a} aCD \xrightarrow{C \rightarrow b} abD \xrightarrow{D \rightarrow CB} abCB \xrightarrow{C \rightarrow AA} abAAB \xrightarrow{A \rightarrow a} abaAB \xrightarrow{A \rightarrow a} abaaB \xrightarrow{B \rightarrow b} abaab. \end{aligned}$$



## 10.3 Bezkontextové Pumping lemma

**Tento důkaz se neobjeví u písemné ani ústní zkoušky**

S využitím Pumping Lemmatu ukažte, že následující jazyk není bezkontextový, kde

$$L = \{ww; w \in \{a, b\}^*\}$$

**Pumping Lemma.** Pro každý CF jazyk  $L$  existuje přirozené číslo  $m \geq 1$  takové, že každé slovo  $z \in L$  délky alespoň  $m$  lze rozdělit na pět částí  $z = uvwxy$  tak, že:

- $|vwx| \leq m$ , (tj. prostřední část není příliš dlouhá),
- $vx \neq \varepsilon$  (tj. alespoň jedno ze slov  $v$ ,  $x$  není prázdné),
- pro všechna  $i \geq 0$  platí  $uv^iwx^iy \in L$ , (tj.  $v$  a  $x$  se dají do slova „napumpovat“ a stále dostaneme slovo z jazyka  $L$ ).

spoiler alert: nedoděláno

Zvolíme  $z = a^m b^m a^m b^m \in L$ ,  $|z| = 4m > m$ .

...

Máme 7 možností: Takže to dělat nebudeme.

## 10.4 Důkaz generování slova matematickou indukcí

Je dána CF gramatika  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, A, B, C\}$ ,  $\Sigma = \{a, b\}$  a  $P$  je:

$$\begin{aligned} P: S &\rightarrow SA \mid aSb \mid Cb \\ A &\rightarrow SC \mid \varepsilon \\ B &\rightarrow bAB \mid bS \mid AA \\ C &\rightarrow CB \mid bA \mid a \end{aligned}$$

Pomocí matematické indukce dokažte, že:

$$A \Rightarrow_{\mathcal{G}}^* S^i AC^i$$

pro všechna  $i \geq 0$ . Toho využijte k důkazu, že  $(ab)^{i+1}(ab^3)^i$  jsou generována gramatikou  $\mathcal{G}$  pro každé  $i \geq 0$ .

1) Základní krok:  $i = 0$

Pro  $i = 0$  platí:

$$A \Rightarrow^* A$$

což odpovídá:

$$S^0 AC^0 = A \quad \checkmark$$

2) Indukční krok:

předp.  $A \Rightarrow^* S^n AC^n$ , chceme dokázat  $A \Rightarrow^* S^{n+1} AC^{n+1}$ .

$$A \xRightarrow{A \rightarrow SC} SC \xRightarrow{S \rightarrow SA} SAC \xRightarrow{I.P.}^* SS^n AC^n C$$

nebo

$$A \xRightarrow{I.P.} S^n AC^n \xRightarrow{A \rightarrow SC} S^n SCC^n \xRightarrow{S \rightarrow SA} S^n SACC^n.$$

## 10.5 Algoritmus CYK

Je dána gramatika  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, A, B, C, D\}$ ,  $\Sigma = \{a, b, c\}$  a pravidla  $P$  jsou dána

$$\begin{aligned} P: S &\rightarrow AB \mid CD \mid AC \\ A &\rightarrow AC \mid a \\ B &\rightarrow BD \mid b \\ C &\rightarrow AD \mid a \\ D &\rightarrow BA \mid b \end{aligned}$$

Algoritmem CYK rozhodněte, zda gramatika  $\mathcal{G}$  generuje slova  $w_1$  a  $w_2$ , kde  $w_1 = baaba$  a  $w_2 = abaaa$ . Pokud ano, nakreslete derivační strom a napište jemu odpovídající levou derivaci.

Levá derivace:

$$\begin{aligned} AB &\leftarrow S \\ AC &\leftarrow S, A \\ AD &\leftarrow C \\ BA &\leftarrow D \\ BD &\leftarrow B \\ CD &\leftarrow S \end{aligned} \quad \begin{aligned} S &\xrightarrow{S \rightarrow CD} CD \xrightarrow{C \rightarrow a} aD \xrightarrow{D \rightarrow BA} aBA \xrightarrow{B \rightarrow b} abA \Rightarrow \\ &\xrightarrow{A \rightarrow AC} abAC \xrightarrow{A \rightarrow AC} abACC \xrightarrow{A \rightarrow a} abaCC \Rightarrow \\ &\xrightarrow{C \rightarrow a} abaaC \xrightarrow{C \rightarrow a} abaaa \end{aligned}$$

Derivační strom pro slovo  $w_2$ :

slovo  $w_1 = baaba$ :

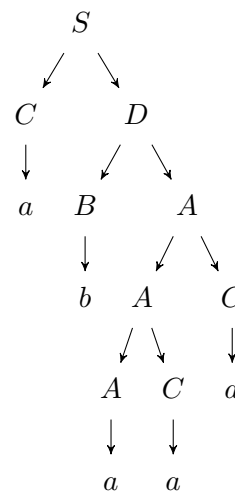
$D$				
$D$	$S, A, C$			
$D$	$S, C, A$	$C, S$		
$D$	$S, A$	$S, C$	$D$	
$B, D$	$A, C$	$A, C$	$B, D$	$A, C$
$b$	$a$	$a$	$b$	$a$

Gramatika  $\mathcal{G}$  negeneruje slovo  $w_1$ .

slovo  $w_2 = abaaa$ :

$C, S$				
$C, S$	$D$			
$C, S$	$D$	$S, A$		
$S, C$	$D$	$S, A$	$S, A$	
$A, C$	$B, D$	$A, C$	$A, C$	$A, C$
$a$	$b$	$a$	$a$	$a$

Gramatika  $\mathcal{G}$  generuje slovo  $w_2$ .



## 11 Jedenácté cvičení

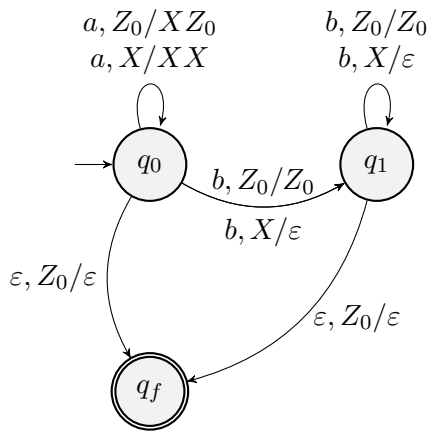
### 11.1 Stavový diagram zásobníkového automatu, práce na automatu

Je dán zásobníkový automat  $A = (Q, \Sigma, \Gamma, \delta, q_0, Z_0, F)$ , kde jednotlivé části jsou  $Q = \{q_0, q_1, q_2, q_f\}$ ,  $\Sigma = \{a, b\}$ ,  $\Gamma = \{Z_0, X\}$  a přechodová funkce je daná tabulkou

	$(a, Z_0)$	$(a, X)$	$(b, Z_0)$	$(b, X)$	$(\varepsilon, Z_0)$	$(\varepsilon, X)$
$\rightarrow q_0$	$(q_0, XZ_0)$	$(q_0, XX)$	$(q_1, Z_0)$	$(q_1, \varepsilon)$	$(q_f, \varepsilon)$	—
$q_1$	—	—	$(q_1, Z_0)$	$(q_1, \varepsilon)$	$(q_f, \varepsilon)$	—
$\leftarrow q_f$	—	—	—	—	—	—

- Nakreslete stavový diagram zásobníkového automatu  $A$ .
- Ukažte práci zásobníkového automatu nad slovem  $aabba$  a slovem  $abbbb$ .
- Charakterizujte jazyk  $L$ , který tento zásobníkový automat přijímá. Tvrzení zdůvodněte.

Stavový diagram automatu  $A$ .



Práce nad slovem  $w_1 = aabba$ .

$(q_0, aabba, Z_0) \vdash (q_0, abba, XZ_0) \vdash (q_0, bba, XXZ_0) \vdash (q_1, ba, XZ_0) \vdash (q_1, a, Z_0)$ . Konec, neúspěch.

Práce nad slovem  $w_2 = abbbb$ .

$(q_0, abbbb, Z_0) \vdash (q_0, bbbb, XZ_0) \vdash (q_1, bbb, Z_0) \vdash (q_1, bb, Z_0) \vdash (q_1, b, Z_0) \vdash (q_1, \varepsilon, Z_0) \vdash (q_f, \varepsilon, \varepsilon)$ . Konec, úspěch.

$$L(A) \stackrel{?}{=} \overbrace{\{a^i b^j \mid 0 \leq i \leq j\}}^L$$

Důkaz.

a)  $L \subseteq L(A)$

- $i = j = 0$ :  $(q_0, \varepsilon, Z_0) \vdash (q_f, \varepsilon, \varepsilon)$
- $0 = i < j$ :  $(q_0, b^j, Z_0) \vdash (q_1, b^{j-1}, Z_0) \vdash^{(j-1)} (q_1, \varepsilon, Z_0) \vdash (q_f, \varepsilon, \varepsilon)$
- $0 < i \leq j$ :  $(q_0, a^i b^j, Z_0) \vdash^i (q_0, b^{i+k}, X^i Z_0) \vdash (q_0, b^{i+k-1}, X^{i-1} Z_0) \vdash^{(i-1)} (q_1, b^k, Z_0) \vdash^k (q_1, \varepsilon, Z_0) \vdash (q_f, \varepsilon, \varepsilon)$ ,  
 $i + k = j, k = j - i \geq 0$ .

b)  $L(A) = N(A) \subseteq L$

Dokazujeme, že pokud  $w \in L(A)$ , pak  $w$  má tvar  $a^i b^j$ , kde  $0 \leq i \leq j$ . Pro  $w \in L(A)$  musí zásobníkový automat  $A$  skončit ve stavu  $q_f$  s prázdným vstupním řetězcem i zásobníkem.

- Přidávání  $a$ : Každý symbol  $a$  způsobí, že do zásobníku přidáme jeden symbol  $X$ . Zásobník tak obsahuje  $i$  symbolů  $X$  po zpracování  $a^i$ .
- Zpracování  $b$ : Každý symbol  $b$  odstraní jeden symbol  $X$  ze zásobníku, pokud je tam ještě přítomen. Pokud už jsou všechny  $X$  odstraněny, symbol  $b$  pouze projde automatem, aniž by zásobník změnil svůj stav.

- Prázdný zásobník: Stav  $q_f$  je přístupný pouze tehdy, když je zásobník prázdný. To znamená, že každý přidaný  $X$  odpovídá odstraněnému  $X$ , což nastane právě tehdy, když  $i \leq j$ .

Z toho plyne, že  $w = a^i b^j$  s  $0 \leq i \leq j$ .

## 12 Dvanácté cvičení

### 12.1 Tvorba zásobníkového automatu podle jazyka

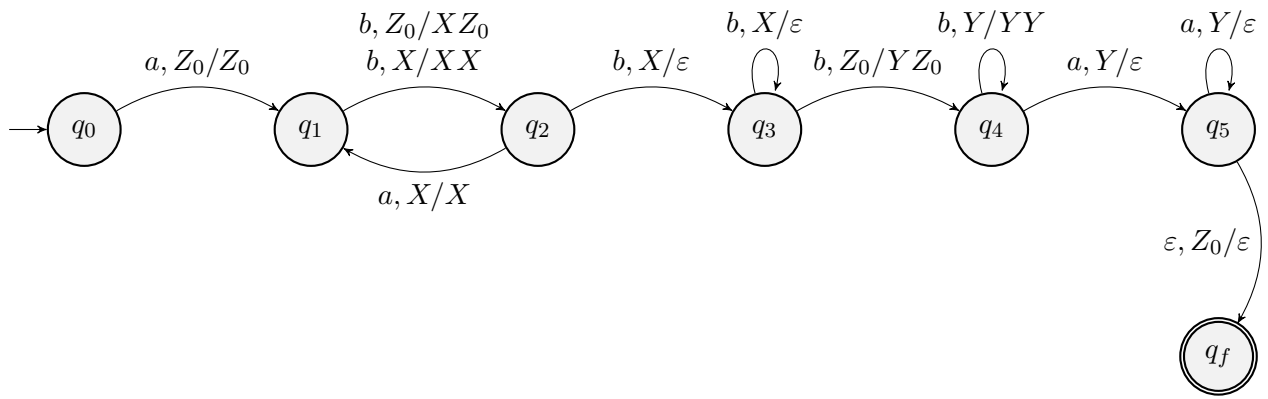
Je dán jazyk  $L$  nad abecedou  $\Sigma = \{a, b\}$ . Sestrojte zásobníkové automaty  $A, B$  tak, že  $L = N(A)$  a  $L = L(B)$  (tj.  $A$  přijímá  $L$  prázdným zásobníkem,  $B$  přijímá  $L$  koncovým stavem), kde

$$L = \{(ab)^i b^j a^{j-i} \mid 0 < i < j\}.$$

$$\Rightarrow L = \{(ab)^i b^{i+k} a^k \mid i > 0, k > 0\} \Rightarrow N(A) = \underbrace{\{(ab)^i b^i \mid i > 0\}}_X, L(B) = \underbrace{\{b^k a^k \mid k > 0\}}_Y.$$

Dva způsoby řešení:

a) Přímě.



b) Přes gramatiku.

	Prázdným zásobníkem:	Koncovým stavem:
$\mathcal{G} : S \rightarrow AB$	$\delta(q, \varepsilon, S) = \{(q, AB)\}$	$\delta(q_0, \varepsilon, Z_0) = \{(q, SZ_0)\}$
$A \rightarrow abAb \mid abb$	$\delta(q, \varepsilon, A) = \{(q, abAb), (q, abb)\}$	$\delta(q, \varepsilon, S) = \{(q, AB)\}$
$B \rightarrow bBa \mid ba$	$\delta(q, \varepsilon, B) = \{(q, bBa), (q, ba)\}$	$\delta(q, \varepsilon, A) = \{(q, abAb), (q, abb)\}$
	$\delta(q, a, a) = \{(q, \varepsilon)\}$	$\delta(q, \varepsilon, B) = \{(q, bBa), (q, ba)\}$
	$\delta(q, b, b) = \{(q, \varepsilon)\}$	$\delta(q, a, a) = \{(q, \varepsilon)\}$
		$\delta(q, b, b) = \{(q, \varepsilon)\}$
		$\delta(q, \varepsilon, Z_0) = \{(q_f, \varepsilon)\}$

1.  $L \subseteq L(G)$

$$S \xrightarrow{S \rightarrow AB} AB \xrightarrow{A \rightarrow abAb}^{(i-1)} (ab)^{i-1} Ab^{i-1} B \xrightarrow{A \rightarrow abb} (ab)^i b^i B \xrightarrow{B \rightarrow bBa}^{(k-1)} (ab)^i b^i b^{k-1} Ba^{k-1} \xrightarrow{B \rightarrow ba} (ab)^i b^i b^k a^k.$$

2.  $L(G) \subseteq L$

$$S \rightarrow^* w$$

$$S \rightarrow AB$$

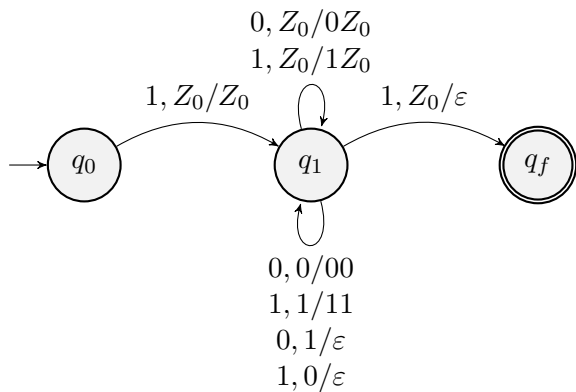
$$A \xrightarrow{A \rightarrow abAb}^j (ab)^j Ab^j \xrightarrow{A \rightarrow abb} (ab)^j abb^j$$

## 12.2 Tvorba zásobníkového automatu podle jazyka

Je dán jazyk  $L$  nad abecedou  $\Sigma = \{a, b\}$ . Sestrojte zásobníkové automaty  $A, B$  tak, že  $L = N(A)$  a  $L = L(B)$  (tj.  $A$  přijímá  $L$  prázdným zásobníkem,  $B$  přijímá  $L$  koncovým stavem), kde

$$L = \{w \mid w \text{ začíná a končí symbolem } 1 \text{ a obsahuje o dvě } 1 \text{ více než } 0\}.$$

$$L = \{1u1 \mid |u|_0 = |u|_1, i = |u|_0\}$$



1)  $L \subseteq N(A)$

$$(q_0, 1u1, Z_0) \vdash (q_1, u1, Z_0) \vdash^* (q_1, 1, Z_0) \vdash (q_f, \varepsilon, \varepsilon).$$

2)  $L(B) \subseteq L$

Musíme ukázat, že pokud zásobníkový automat  $B$  přijme slovo  $w$ , pak  $w \in L$ . To znamená, že  $w$  začíná a končí symbolem 1 a obsahuje o dvě více symbolů 1 než 0.

Vlastnosti přechodů:

1. Automat přechází z počátečního stavu  $q_0$  do  $q_1$ , pokud první symbol je 1. Tuto 1 nezapočítáme.
2. Ve stavu  $q_1$  se zásobník automat udržuje informaci o rozdílu mezi počtem symbolů 1 a 0:
  - Při čtení symbolů 0 a 1 přidává, respektive odstraní (pokud existuje) odpovídající značku do zásobníku, což udržuje informaci o tom, zda je stejný počet obou znaků.
3. Pokud zpracování skončí symbolem 1 a zásobník je prázdný, automat přejde do koncového stavu  $q_f$ , což zajišťuje, že rozdíl mezi počtem symbolů 1 a 0 je přesně dvě.

Každé slovo přijaté automatem  $B$  splňuje podmínky:

- Začíná symbolem 1 (nutný přechod z  $q_0$  do  $q_1$ ).
- Končí symbolem 1 (nutný přechod z  $q_1$  do  $q_f$ ).
- Obsahuje přesně o dvě více symbolů 1 než 0 (zásobník na konci zajišťuje prázdný stav).

Z toho plyne, že  $L(B) \subseteq L$ .

## 12.3 Tvorba zásobníkového automatu podle jazyka

Je dán jazyk  $L$ . Sestrojte zásobníkové automaty  $A, B$  tak, že  $L = N(A)$  a  $L = L(B)$  (tj.  $A$  přijímá  $L$  prázdným zásobníkem,  $B$  přijímá  $L$  koncovým stavem), kde

$$L = \{a^i b^i c^{i+j}\}$$

Ukažte práci nad slovem  $w = abcc$ .



Konstrukce pomocí gramatiky:

$$S \rightarrow aSc \mid A$$

$$A \rightarrow bAc \mid \varepsilon$$

Prázdným zásobníkem:

$$\delta(q, \varepsilon, S) = \{(q, aSc), (q, A)\}$$

$$\delta(q, \varepsilon, A) = \{(q, bAc), (q, \varepsilon)\}$$

$$\delta(q, a, a) = \{(q, \varepsilon)\}$$

$$\delta(q, b, b) = \{(q, \varepsilon)\}$$

$$\delta(q, c, c) = \{(q, \varepsilon)\}$$

Koncovým stavem:

$$\delta(q_0, \varepsilon, Z_0) = \{(q, SZ_0)\}$$

$$\delta(q, \varepsilon, S) = \{(q, AB)\}$$

$$\delta(q, \varepsilon, A) = \{(q, abAb), (q, abb)\}$$

$$\delta(q, \varepsilon, B) = \{(q, bBa), (q, ba)\}$$

$$\delta(q, \varepsilon, a) = \{(q, \varepsilon)\}$$

$$\delta(q, \varepsilon, b) = \{(q, \varepsilon)\}$$

$$\delta(q, \varepsilon, Z_0) = \{(q_f, \varepsilon)\}$$

Práce nad slovem  $w = abcc$ :

$$(q, abbc, S) \vdash (q, abcc, aSc) \vdash (q, bcc, Sc) \vdash$$

$$(q, bcc, Ac) \vdash (q, bcc, bAcc) \vdash (q, cc, Acc) \vdash$$

$$(q, cc, cc) \vdash (q, c, c) \vdash (q, \varepsilon, \varepsilon)$$

Práce nad slovem  $w = abcc$ :

$$(q_0, abc, Z_0) \vdash (q, abbc, SZ_0) \vdash (q, abcc, aScZ_0) \vdash$$

$$(q, bcc, SZ_0c) \vdash (q, bcc, AcZ_0) \vdash (q, bcc, bAccZ_0) \vdash$$

$$(q, cc, AccZ_0) \vdash (q, cc, ccZ_0) \vdash (q, c, cZ_0) \vdash$$

$$(q, \varepsilon, Z_0) \vdash (q_f, \varepsilon, \varepsilon)$$

## 12.4 Důkaz bezkontextovosti jazyka

Je dán jazyk  $L = \{0^n 1^m; 0 \leq n \leq m \leq 2n\}$ . Rozhodněte, zda jazyk  $L$  je bezkontextový.

V případě, že je bezkontextový, najděte buď bezkontextovou gramatiku, která ho generuje, nebo zásobníkový automat, který ho přijímá.

V případě, že není bezkontextový, tvrzení dokažte.

Například  $\mathcal{G}$  s pravidly  $P$ :

$$S \rightarrow 0S11 \mid 0S1 \mid \varepsilon$$

Důkaz.

1)  $L \subseteq L(\mathcal{G})$  –  $\mathcal{G}$  generuje celé  $L$

- $0 < n \leq m \leq 2n \rightarrow k = 2n - m \geq 0$ :

$$S \xrightarrow{S \rightarrow 0S1} 0^k S 1^k \xrightarrow{S \rightarrow 0S11} 0^k 0^{n-1} S 1^{n-1} 1^{n-1} 1^k \xrightarrow{S \rightarrow \varepsilon} 0^n 1^{2n-k} = 0^n 1^{2n-2n+m} = 0^n 1^m.$$

- $0 = n = m$ :  $0^0 1^0 = \varepsilon \in L$ .

2)  $L(\mathcal{G}) \subseteq L$  –  $\mathcal{G}$  negeneruje nic navíc

Nechť  $w \in L(\mathcal{G})$ , kde  $w = 0^n 1^m$ . Sledujeme pravidla gramatiky  $\mathcal{G}$ :

(0) základní krok: Pokud  $S \rightarrow \varepsilon$ , pak  $w = \varepsilon = 0^0 1^0$ . Zřejmě  $\varepsilon \in L$ .

(1) indukční krok:

- Pokud  $S \rightarrow 0S1$ , přidáváme jeden symbol 0 na začátek a jeden symbol 1 na konec. Po aplikaci tohoto pravidla se počet 0 zvýší o jeden a počet 1 o jedna, tedy platí  $n \leq m \leq 2n$ .
- Pokud  $S \rightarrow 0S11$ , přidáváme jeden symbol 0 na začátek a dva symboly 1 na konec. Po aplikaci tohoto pravidla se počet 0 zvýší o jeden a počet 1 o dva, tedy stále platí  $n \leq m \leq 2n$ .

## 13 Třinácté cvičení

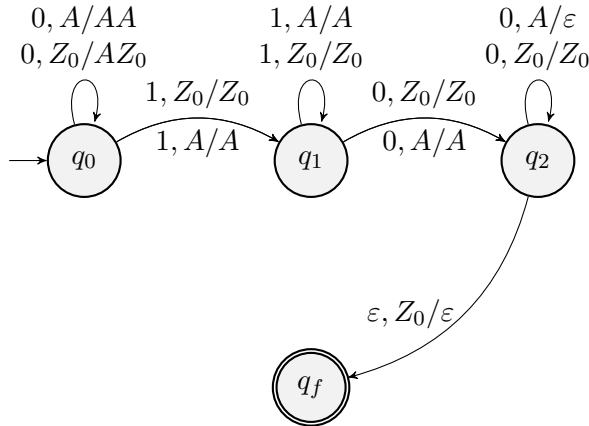
### 13.1 Tvorba a práce zásobníkového automatu podle jazyka

Je dán jazyk  $L$  nad abecedou  $\Sigma = \{0, 1\}$ . Sestrojte zásobníkové automaty  $A, B$  tak, že  $L = N(A)$  a  $L = L(B)$  (tj.  $A$  přijímá  $L$  prázdným zásobníkem,  $B$  přijímá  $L$  koncovým stavem), kde

$$L = \left\{ 0^i 1^j 0^k \mid 0 \leq i < k, j > 0 \right\}.$$

Ukažte práci jednoho ze zásobníkových automatů nad slovem 011000 a nad slovem 001110.

Přímou metodou:



Práce nad slovem  $w_1 = 011000$ .

$(q_0, 011000, Z_0) \vdash (q_0, 11000, AZ_0) \vdash (q_1, 1000, AZ_0)$   
 $\vdash (q_1, 000, AZ_0) \vdash (q_2, 00, AZ_0) \vdash (q_f, 0, AZ_0) \times$   
 $\vdash (q_2, 0, Z_0)$

$(q_2, 0, Z_0) \vdash (q_f, 0, Z_0) \times$   
 $\vdash (q_2, \varepsilon, Z_0) \vdash (q_f, \varepsilon, \varepsilon) \checkmark$  Konec, úspěch.

Práce nad slovem  $w_2 = 001110$ .

$(q_0, 001110, Z_0) \vdash^{(2)} (q_0, 1110, AAZ_0) \vdash (q_1, 110, AAZ_0)$   
 $\vdash^{(2)} (q_1, 0, AAZ_0) \vdash (q_2, \varepsilon, AAZ_0) \times$  Konec, neúspěch.

Přes gramatiku:

$\mathcal{G}: S \rightarrow S0 \mid 0S0 \mid A0$

$A \rightarrow 1A \mid 1$

Důkaz.

1)  $L \subseteq L(\mathcal{G})$

$S \Rightarrow^* 0^i S 0^k \xrightarrow{S \rightarrow A0} 0^i A 0^{k+1} \xrightarrow{A \rightarrow 1A}^{(j)} 0^i 1^j A 0^{k+1} \xrightarrow{A \rightarrow 1} 0^i 1^{j+1} 0^{k+1}, i \leq k, j > 0.$

2)  $L(\mathcal{G}) \subseteq L$

### 13.2 Tvorba nevypouštěcí gramatiky

Je dána bezkontextová gramatika  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, A, B, C\}$ ,  $\Sigma = \{0, 1\}$  a  $P$  je dáno

$S \rightarrow SA \mid 0$

$A \rightarrow BAB \mid 1$

$B \rightarrow CB \mid \varepsilon$

$C \rightarrow AS \mid 0 \mid \varepsilon$

Ke gramatice  $\mathcal{G}$  vytvořte nevypouštěcí gramatiku  $\mathcal{G}_1$ . V gramatice  $\mathcal{G}_1$  odstraňte levou rekurzi.

**1. krok** Vytvoření nevypouštěcí gramatiky  $\mathcal{G}_1$ .

$V = \{x \mid x \Rightarrow^* \varepsilon\}$

$V_1 = \{x \mid x \rightarrow \varepsilon \in P\} = \{B, C\}$

$V_2 = V_1 \cup \{x \mid x \rightarrow \alpha \in P, \alpha \in V_1^+\} = V_1 \cup \emptyset = V_1 = V.$

$P': S \rightarrow SA \mid 0$

$A \rightarrow BAB \mid AB \mid BA \mid A \mid 1$

$B \rightarrow CB \mid C$

$C \rightarrow AS \mid 0$

**2. krok** odstranění levé rekurze. (postup shora dolů)

$$S \rightarrow 0 \mid 0S'$$

$$S' \rightarrow A \mid AS'$$

$$A \rightarrow BAB \mid BA \mid 1 \mid BABA' \mid BAA' \mid 1A'$$

$$A' \rightarrow B \mid BA'$$

$$B \rightarrow CB \mid C$$

$$C \rightarrow AS \mid 0$$

### 13.3 Převod gramatiky do Greibachové normální formy

Do Greibachové normální formy převeďte gramatiku  $\mathcal{G}$ , kde  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, E, F\}$ ,  $\Sigma = \{a, *, +, ()\}$  a  $P$  je dáno

$$S \rightarrow (E)$$

$$E \rightarrow F * F \mid F + F$$

$$F \rightarrow a \mid S$$

**1. krok** oindexování neterminálů.

$$A_1 = S$$

$$A_2 = E$$

$$A_3 = F$$

**2. krok** odstranění levých rekurzí. Kontrola správného pořadí indexů (na pravé straně vždy neterminál s větším indexem), jinak sloučit pravidla. (postup shora dolů)

$$S \rightarrow (E)$$

$$E \rightarrow F * F \mid F + F$$

$$F \rightarrow a \mid (E)$$

**3. krok** nahrazení prvních neterminálů pravých stran, které neterminálem začínají, pravidly. (postup zespoda nahoru)

$$S \rightarrow (E)$$

$$E \rightarrow a * F \mid a + F \mid (E) * F \mid (E) + F$$

$$F \rightarrow a \mid (E)$$

**4. krok** za prvním terminálem pravé strany vždy následují pouze neterminály.

$$S \rightarrow (EX$$

$$E \rightarrow aYF \mid aZF \mid (EXYF \mid (EXZF$$

$$F \rightarrow a \mid (EX$$

$$X \rightarrow )$$

$$Y \rightarrow *$$

$$Z \rightarrow +$$

### 13.4 Převod gramatiky do Greibachové normální formy

Do Greibachové normální formy převěďte gramatiku  $\mathcal{G}$ , kde  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, A, B\}$ ,  $\Sigma = \{a, b, c\}$  a  $P$  je dáno

$$\begin{aligned} S &\rightarrow Ab \mid B \\ A &\rightarrow Aba \mid Bcc \\ B &\rightarrow Sa \mid b \end{aligned}$$

**1. krok** oindexování neterminálů.

$$\begin{aligned} A_1 &= S \\ A_2 &= A \\ A_3 &= B \end{aligned}$$

**2. krok** odstranění levých rekurzí. Kontrola správného pořadí indexů (na pravé straně vždy neterminál s větším indexem), jinak sloučit pravidla. (postup shora dolů)

$$\begin{aligned} S &\rightarrow Ab \mid B \\ A &\rightarrow Ba \mid BaA' \\ A' &\rightarrow ba \mid baA' \\ B &\rightarrow Aba \mid Ba \mid b \\ B &\rightarrow Bab a \mid BaA'ba \mid Ba \mid b \\ B &\rightarrow b \mid bB' \\ B' &\rightarrow aba \mid aA'ba \mid a \mid abaB' \mid aA'baB' \mid aB' \end{aligned}$$

**3. krok** nahrazení prvních neterminálů pravých stran, které neterminálem začínají, pravidly. (postup zespoda nahoru)

$$\begin{aligned} S &\rightarrow Bab \mid BaA' \mid b \mid bB' \\ A &\rightarrow ba \mid bB'a \mid baA' \mid bB'aA' \\ A' &\rightarrow ba \mid baA' \\ B &\rightarrow b \mid bB' \\ B' &\rightarrow aba \mid aA'ba \mid a \mid abaB' \mid aA'baB' \mid aB' \end{aligned}$$

**4. krok** za prvním terminálem pravé strany vždy následují pouze neterminály.

$$\begin{aligned} S &\rightarrow BaY \mid BaA' \mid b \mid bB' \\ A &\rightarrow bX \mid bB'X \mid bXA' \mid bB'aA' \\ A' &\rightarrow bX \mid bXA' \\ B &\rightarrow b \mid bB' \\ B' &\rightarrow aXY \mid aA'YX \mid a \mid aYXB' \mid aA'YXB' \mid aB' \\ X &\rightarrow a \\ Y &\rightarrow b \end{aligned}$$

## 14 Čtrnácté cvičení

### 14.1 Převod gramatiky do Greibachové normální formy

Do Greibachové normální formy převed'te gramatiku  $\mathcal{G}$ , kde  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, A\}$ ,  $\Sigma = \{0, 1\}$  a  $P$  je dáno

$$\begin{aligned} S &\rightarrow SA \mid 0 \\ A &\rightarrow AS \mid 1 \end{aligned}$$

**1. krok** oindexování neterminálů.

$$\begin{aligned} X_1 &= S \\ X_2 &= A \\ X_3 &= S' \\ X_4 &= A' \end{aligned}$$

**2. krok** odstranění levých rekurzí.

$$\begin{aligned} S &\rightarrow 0 \mid 0S' \\ S' &\rightarrow A \mid AS' \\ A &\rightarrow 1 \mid 1A' \\ A' &\rightarrow S \mid SA' \end{aligned}$$

**3. krok** Kontrola správného pořadí indexů (na pravé straně vždy neterminál s větším indexem), jinak sloučit pravidla. (postup shora dolů)

$$\begin{aligned} S &\rightarrow 0 \mid 0S' \\ A &\rightarrow 1 \mid 1A' \\ S' &\rightarrow 1 \mid 1A' \mid 1S' \mid 1A'S' \\ A' &\rightarrow 0 \mid 0S' \mid 0A' \mid 0S'A' \end{aligned}$$

**4. krok** za prvním terminálem pravé strany vždy následují pouze neterminály. ✓

### 14.2 Zkoušková ukázka práce na bezkontextové gramatice

Je dána bezkontextová gramatika  $\mathcal{G} = (N, \Sigma, S, P)$ , kde  $N = \{S, A, B, C\}$ ,  $\Sigma = \{a, b\}$  a  $P$  je dáno

$$\begin{aligned} S &\rightarrow Sa \mid Sb \mid bC \\ A &\rightarrow CBA \mid BC \mid b \\ B &\rightarrow aB \mid \varepsilon \\ C &\rightarrow AA \mid bBb \mid \varepsilon \end{aligned}$$

1. Ke gramatice  $\mathcal{G}$  najděte nevypouštěcí gramatiku  $\mathcal{G}_1$ . Kroky převodu popište.
2. Ke gramatice  $\mathcal{G}_1$  najděte gramatiku  $\mathcal{G}_2$  v Chomského normálním tvaru, která generuje stejný jazyk jako gramatika  $\mathcal{G}_1$ . Jednotlivé kroky popište, gramatiku v Chomského normálním tvaru definujte.
3. Pomocí matematické indukce dokažte, že platí  $A \Rightarrow_{\mathcal{G}}^* A^i C (BA)^{i+1}$  pro každé  $i \geq 0$ . Toho využijte k důkazu, že  $b^{i+2}(ab)^{i+1}$  je generováno gramatikou  $\mathcal{G}$  pro každé  $i \geq 0$ .
4. Je gramatika  $\mathcal{G}$  víceznačná? Víceznačnou gramatiku definujte.
5. V gramatice  $\mathcal{G}_1$  odstraňte levou rekurzi u symbolu  $S$ . Postup popište.

**1.** Nevypouštěcí gramatika  $\mathcal{G}_1$ .

$$\begin{aligned}
V &= \{A \mid A \Rightarrow^* \varepsilon\} \\
V_1 &= \{A \mid A \rightarrow \varepsilon \in P\} = \{B, C\} \\
V_2 &= V_1 \cup \{A \mid A \rightarrow \alpha \in P, \alpha \in V_1^*\} = V_1 \cup \{A\} = \{A, B, C\} \\
V_3 &= V_2 \cup \{A \mid A \rightarrow \alpha \in P, \alpha \in V_2^*\} = V_2 \cup \varepsilon = V_2 = V
\end{aligned}$$

$$\begin{aligned}
\mathcal{G}_1 : S &\rightarrow Sa \mid Sb \mid bC \mid b \\
&A \rightarrow CBA \mid CA \mid CB \mid BA \mid BC \mid B \mid C \mid b \\
&B \rightarrow aB \mid a \\
&C \rightarrow AA \mid A \mid bBb \mid bb
\end{aligned}$$

2. Chomského normální tvar gramatiky  $\mathcal{G}_1$ .

1. **krok** nahrazení samostatných terminálů pravidly. (pokud nastane např.  $A \rightarrow A$ , tak vynechat.)

$$\begin{aligned}
S &\rightarrow Sa \mid Sb \mid bC \mid b \\
A &\rightarrow CBA \mid CA \mid CB \mid BA \mid BC \mid \underbrace{aB \mid a}_B \mid \underbrace{AA \mid bBb \mid bb}_C \mid b \\
B &\rightarrow aB \mid a \\
C &\rightarrow AA \mid \underbrace{CBA \mid CA \mid CB \mid BA \mid BC \mid B \mid C \mid b}_A \mid bBb \mid bb
\end{aligned}$$

2. **krok** nahrazení terminálů neterminály pokud nejsou samotné.

$$\begin{aligned}
S &\rightarrow SX_1 \mid SX_2 \mid X_2C \mid b \\
A &\rightarrow CBA \mid CA \mid CB \mid BA \mid BC \mid X_1B \mid X_1 \mid AA \mid X_2BX_2 \mid X_2X_2 \mid X_2 \\
B &\rightarrow X_1B \mid X_1 \\
C &\rightarrow AA \mid CBA \mid CA \mid CB \mid BA \mid BC \mid B \mid C \mid X_2 \mid X_2BX_2 \mid X_2X_2 \\
X_1 &\rightarrow a \\
X_2 &\rightarrow b
\end{aligned}$$

3. **krok** nahrazení pravých stran, která mají délku  $\geq 3$ .

$$\begin{aligned}
S &\rightarrow SX_1 \mid SX_2 \mid X_2C \mid b \\
A &\rightarrow Y \mid CA \mid CB \mid BA \mid BC \mid X_1B \mid X_1 \mid AA \mid Z \mid X_2X_2 \mid X_2 \\
Y &\rightarrow CBA \\
Z &\rightarrow X_2BX_2 \\
B &\rightarrow X_1B \mid X_1 \\
C &\rightarrow AA \mid Y \mid CA \mid CB \mid BA \mid BC \mid B \mid C \mid X_2 \mid Z \mid X_2X_2 \\
X_1 &\rightarrow a \\
X_2 &\rightarrow b
\end{aligned}$$

3. Důkaz.

$$A \Rightarrow_{\mathcal{G}}^* A^i C(BA)^{i+1}$$

$$\text{Základní krok: } i = 0: A^0 C(BA)^1 = CBA \checkmark A \Rightarrow_{\mathcal{G}}^* CBA.$$

$$\text{Indukční krok: } i \geq 0: \text{ indukční předpoklad: } A \Rightarrow^* A^i C(BA)^{i+1}.$$

$$A \xrightarrow{A \rightarrow CBA} CBA \xrightarrow{C \rightarrow AA} AA_{IP}(BA) \xrightarrow{IP}^* AA^i C(BA)^{i+1}(BA) = A^{i+1} C(BA)^{i+2}. \checkmark$$

$$\text{A tedy, } b^{i+2}(ab)^{i+1} \in L(\mathcal{G})?$$

$$S \xrightarrow{S \rightarrow bC} bC \xrightarrow{C \rightarrow AA} bAA \xrightarrow{A \rightarrow b} b^2A \Rightarrow \text{[dle důkazu výše]} \Rightarrow^* b^2 A^i C(BA)^{i+1} \xrightarrow{A \rightarrow b} b^{i+2} C(BA)^{i+1} \xrightarrow{C \rightarrow \varepsilon}$$

$$b^{i+2}(BA)^{i+1} \xrightarrow{B \rightarrow aB} b^{i+2}(aBA)^{i+1} \xrightarrow{B \rightarrow \varepsilon} b^{i+2}(aA)^{i+1} \xrightarrow{A \rightarrow b} b^{i+2}(ab)^{i+1}. \checkmark$$

4. Je gramatika  $\mathcal{G}$  víceznačná?

Víceznačnost = existují alespoň 2 derivační stromy / 2 levé derivace pro jedno libovolné slovo z  $\mathcal{G}$ .

Například mějme slovo  $w = bbb$ .

První způsob vygenerování slova  $w$ :  $S \xrightarrow{S \rightarrow bC} bC \xrightarrow{C \rightarrow AA} bAA \xrightarrow{A \rightarrow b}^2 bbb$ .

Druhý způsob vygenerování slova  $w$ :  $S \xrightarrow{S \rightarrow bC} bC \xrightarrow{C \rightarrow bBb} bbBb \xrightarrow{B \rightarrow \varepsilon} bbb$ .

A tedy gramatika  $\mathcal{G}$  je víceznačná.

5. Odstranění levých rekurzí.

Levá rekurze se vyskytuje pouze v pravidlu  $S \rightarrow Sa \mid Sb \mid bC \mid b$ .

Je potřeba přidat pouze jeden neterminál. Pokud by se jich přidalo více, nová gramatika by generovala méně slov, než původní.

$$S \rightarrow bC \mid bCS' \mid b \mid S'$$

$$S' \rightarrow a \mid aS' \mid b \mid bS'$$