

Записки по ЕАИ

Александър Тангълов

3 януари 2023г.

Съдържание

1	Въведение	4
1.1	Азбуки и езици	4
1.1.1	Задачи	6
1.1.2	Решения	7
2	Крайни автомати и регулярни езици	10
2.1	Детерминирани крайни автомати	10
2.1.1	Задачи	12
2.1.2	Решения	14
2.2	Недетерминирани крайни автомати	15
2.2.1	Задачи	19
2.2.2	Решения	20
2.3	Затвореност относно регулярните операции	21
2.3.1	Задачи	24
2.3.2	Решения	26
2.4	Регулярни изрази и теорема на Клини	28
2.4.1	Задачи	31
2.4.2	Решения	31
2.5	Минимален автомат	31
2.5.1	Задачи	35
2.5.2	Решения	35
2.6	Критерии за нерегулярност	35
2.6.1	Задачи	38
2.6.2	Решения	39
3	Граматика и стекови автомати	41
3.1	Контекстно-свободни граматика	41
3.1.1	Задачи	43
3.1.2	Решения	44
3.2	Затвореност относно регулярните операции	44

3.2.1	Задачи	45
3.2.2	Решения	46
3.3	Коректност на контекстно-свободни граматики	46
3.3.1	Задачи	50
3.3.2	Решения	51
3.4	Синтактични дървета на извод	51
3.5	Езици, които не са контекстно-свободни	53
3.5.1	Задачи	55
3.5.2	Решения	56

Глава 1

Въведение

1.1 Азбуки и езици

Дефиниция 1. Азбука наричаме крайно множество от **символи**.

Пример 1. Ето два примера за азбуки:

1. Латинската азбука $\{a, \dots, z\}$.
2. Двоичната азбука $\{0, 1\}$.

Всяко крайно множество е азбука; формална дефиниция за понятието *символ* няма. За простота в курса ще използваме само латински букви, цифри и често срещани символи като \$, # и !. Освен това, вместо по-общото понятие *символ* ще използваме *буква* за елементите на една азбука.

Дефиниция 2. Дума над азбука ще наричаме всяка крайна редица от букви от азбуката.

За коя да е азбука, думата представляваща редица от 0 букви бележим с ϵ и наричаме **празната дума**.

Множеството от всички думи над дадена азбука Σ бележим със Σ^* .

Дължината на една дума е дължината ѝ като редица от букви. Бележим дължината на думата w с $|w|$.

Сега ще дадем няколко операции, които можем да прилагаме върху думите над дадена азбука.

Операции върху думи

1. **Конкатенацията** на думите w_1 и w_2 бележим с $w_1 \circ w_2$ или по-просто $w_1 w_2$. Неформално, конкатенацията на думите w_1 и w_2 представлява думата w_1 последвана непосредствено от думата w_2 .

Дума v наричаме **поддума** на думата w тогава и само тогава, когато съществуват думи w_1 и w_2 такива, че $w = w_1 v w_2$. Допълнително, ако $w = w_1 v$ за някоя дума w_1 , то v наричаме **суфикс** на w . Аналогично, ако $w = v w_2$ за някоя дума w_2 , то v наричаме **префикс** на w .

За всяка дума w и всяко естествено число i , думата w^i е дефинирана като

$$\begin{aligned} w^0 &= \epsilon, \text{ празната дума} \\ w^{i+1} &= w^i \circ w \text{ за всяко } i \geq 0 \end{aligned}$$

2. **Обръщането** на дума w , означаващо с w^{rev} , представлява думата w "написана на обратно".

Формалните дефиниции на *конкатенация* и *обръщане* по индукция са дадени в секцията със задачи (задачи 3 и 4).

Дефиниция 3. Всяко множество от думи над дадена азбука Σ — тоест, всяко подмножество на Σ^* — наричаме **език** над Σ .

Пример 2. Примери за езици са:

1. \emptyset е език над всяка азбука.
2. За всяка азбука Σ и всяка буква $a \in \Sigma$, $\{a\}$ е език над Σ .
3. Σ^* е език над Σ за всяка азбука Σ . Да забележим, че при $|\Sigma| \geq 1$, Σ^* е безкраен език.
4. $\{0, 01, 011, 0111, \dots\}$ е безкраен език над $\Sigma = \{0, 1\}$.
5. $\{w \in \{0, 1\}^* \mid w \text{ има равен брой 0-ли и 1-ци}\}$ е безкраен език над $\Sigma = \{0, 1\}$.
6. $\{w \in \Sigma^* \mid w = w^{rev}\}$ е безкраен език над всяка азбука Σ .

Предимно ще се вълнуваме от безкрайни езици.

Тъй като езиците са множества, върху тях можем да прилагаме стандартните операции на обединение, сечение и разлика. Ако от контекста се подразбира дадена азбука Σ , то пишейки \bar{L} за даден език L над Σ^* , ще имаме предвид разликата $\Sigma^* \setminus L$.

Операции върху езици

Освен стандартните такива, върху езици можем да прилагаме и следните две операции:

1. Конкатенация на езици. Ако L_1 и L_2 са езици над азбука Σ , тяхната **конкатенация** е езикът $L = L_1 \circ L_2$, или по-просто $L = L_1 L_2$, където

$$L = \{w \in \Sigma^* \mid w = w_1 \circ w_2 \text{ за някои } w_1 \in L_1 \text{ и } w_2 \in L_2\}.$$

Пример 3. Нека $L_1 = \{0, 00, 000, 0000, \dots\}$ и $L_2 = \{1, 11, 111, 1111, \dots\}$. Тогава

$$L_1 \circ L_2 = \{01, 011, 0111, \dots, 001, 0011, 00111, \dots, 0001, 00011, 000111, \dots\}.$$

2. Звезда на Клини. За даден език L над азбука Σ с L^* означаваме **звездата на Клини** на L . L^* е множеството от всички думи получени чрез прилагане на операцията конкатенация върху нула или повече думи от L . (Да забележим, че конкатенацията на нула думи би дала като резултат неутралния елемент на операцията конкатенация, а именно празната дума.) И така,

$$L^* = \{w \in \Sigma^* \mid w = w_1 \circ \dots \circ w_k \text{ за някое } k \geq 0 \text{ и някои } w_1, \dots, w_k \in L\}$$

Пример 4. Нека $L = \{0, 11\}$. Тогава

$$L^* = \{\epsilon, 0, 11, 00, 011, 110, 1111, 000, 0011, 0110, 01111, 1100, 11011, 11110, 111111, \dots\}.$$

Езикът LL^* означаваме с L^+ . Еквивалентно

$$L^+ = \{w \in \Sigma^* \mid w = w_1 \circ \dots \circ w_k \text{ за някое } k \geq 1 \text{ и някои } w_1, \dots, w_k \in L\}.$$

1.1.1 Задачи

Задача 1. Където е възможно дайте по един пример за дума във и дума извън всеки от следните езици над $\Sigma = \{a, b\}$.

- (а) $L = \{w \in \Sigma^* \mid \text{съществува } u \in \Sigma^2, w = uu^{rev}u\}$
- (б) $L = \{w \in \Sigma^* \mid ww = ww\}$
- (в) $L = \{w \in \Sigma^* \mid \text{съществуват } u, v \in \Sigma^*, uvw = wvu\}$
- (г) $L = \{w \in \Sigma^* \mid \text{съществува } u \in \Sigma^*, www = uu\}$

Задача 2. Докажете всяко от следните.

- (а) $\{\epsilon\}^* = \{\epsilon\}$
- (б) За всяка азбука Σ и всеки език $L \subseteq \Sigma^*$, $(L^*)^* = L^*$.
- (в) Ако a и b са различни букви, то $\{a, b\}^* = \{a\}^* (\{b\} \{a\}^*)^*$.
- (г) Ако Σ е произволна азбука, $\epsilon \in L_1 \subseteq \Sigma^*$ и $\epsilon \in L_2 \subseteq \Sigma^*$, то

$$(L_1 \Sigma L_2)^* = \Sigma^*.$$

(д) За всеки език L , $\emptyset L = L \emptyset = \emptyset$.

Задача 3. Да разгледаме формалната дефиниция на операцията *обръщане* с индукция по дължината на думата-операнд w :

(1) Ако $|w| = 0$, то $w^{rev} = w = \epsilon$.

(2) Ако $|w| = n + 1$ за някое $n \in \mathbb{N}$, то $w = ua$ за някое $a \in \Sigma$ и $w^{rev} = au^{rev}$.

Използвайте тази дефиниция, за да докажете всяко от следните.

(а) $(w^{rev})^{rev} = w$ за всяка дума w .

(б) Ако v е поддума на w , то v^{rev} е поддума на w^{rev} .

(в) $(w^i)^{rev} = (w^{rev})^i$ за всяка дума w и $i \geq 0$.

Задача 4. Да разгледаме формалната дефиниция на операцията *конкатенация* с индукция по дължината на втория операнд w_2 :

(1) Ако $|w_2| = 0$, то $w_1 \circ w_2 = w_1$.

(2) Ако $|w_2| = n + 1$ за някое $n \in \mathbb{N}$, то $w_2 = ua$ за някое $a \in \Sigma$ и $w_1 \circ w_2 = (w_1 \circ u)a$.

Използвайте тази дефиниция, за да докажете, че операцията конкатенация е асоциативна.

Задача 5. Кога е изпълнено, че $L^+ = L^* \setminus \{\epsilon\}$?

1.1.2 Решения

Задача 1. (а) $abbaab$ е въвв; $ababab$ е извън.

(б) ϵ е въвв; a е извън.

(в) Този език е Σ^* .

(г) ϵ е въвв; a е извън.

Задача 2. (а) Следва директно от равенството $\epsilon = \underbrace{\epsilon \circ \dots \circ \epsilon}_{k \text{ пъти}}$, където k е което и да е

естествено число, и дефиницията на *Звезда на Клини*.

(б) Имаме

$$w \in (L^*)^* \iff$$

$$w = w_1 \circ \dots \circ w_k \text{ за някое } k \geq 0 \text{ и някои } w_1, \dots, w_k \in L^* \iff$$

$$w = w_{1,1} \circ \dots \circ w_{1,l_1} \circ \dots \circ w_{k,1} \circ \dots \circ w_{k,l_k} \text{ за някои } l_1, \dots, l_k \geq 0 \text{ и } w_{i,j} \in L \text{ за } 1 \leq i \leq k \text{ и}$$

$$1 \leq j \leq l_i \iff$$

$w \in L^*$.

(в) Включването " \supseteq " е ясно. Ще докажем включването " \subseteq ". Нека

$w \in \{a, b\}^*$. Ако $w \in \{a\}^*$, то очевидно w принадлежи на езика в дясната страна. В противен случай можем да представим w във вида $w = w_1 w_2$, където $w_1 \in \{a\}^*$ и $w_2 \in \{b\}\{a, b\}^*$. (Тоест w_2 е суфиксът на w започващ от първото срещане на b .) На свой ред w_2 можем да запишем във вида $w_2 = w_{2,1}, \dots, w_{2,k}$, където $w_{2,i} \in \{b\}\{a\}^*$, за $1 \leq i \leq k$. (Тоест $\{w_{2,i}\}_{i=1}^k$ са максималните по включване поддуми на w_2 започващи с b и съдържащи точно едно b). Оттук следва, че $w_2 \in (\{b\}\{a\}^*)^*$. Общо имаме $w \in \{a\}^*(\{b\}\{a\}^*)^*$.

(г) Включването " \subseteq " е ясно. Ще докажем включването " \supseteq ". Нека

$w \in \Sigma^*$. Думата w има вида $w = a_1 \dots a_k$, където $|w| = k$ и $a_1, \dots, a_k \in \Sigma$. Мислейки за $\{a_i\}_{i=1}^k$ като за думи, а не букви, можем да презапишем w във вида $w = \epsilon a_1 \epsilon a_2 \epsilon \dots \epsilon a_k \epsilon$. Оттук $w = w_1 w_2 \dots w_k$, за $w_i \in L_1 \Sigma L_2$,

за $1 \leq i \leq k$. (Тук използваме, че $\epsilon \in L_1$ и $\epsilon \in L_2$.) Следователно

$w \in (L_1 \Sigma L_2)^*$.

(д) По дефиниция $\emptyset L = \{w \in \Sigma^* \mid w = w_1 \circ w_2 \text{ за някои } w_1 \in \emptyset \text{ и } w_2 \in L_2\} = \emptyset$.

Аналогично $L \emptyset = \emptyset$.

Задача 3.(а) Индукция по $|w|$. Ако $|w| = 0$, то $w = \epsilon$ и съгласно дефиницията на *обръщане*, $w^{rev} = \epsilon$. Тогава $(w^{rev})^{rev} = \epsilon^{rev} = \epsilon$. Ако $|w| = n + 1$ за някое $n \in \mathbb{N}$, то $w = ua$ за някое $u \in \Sigma^*$ и $a \in \Sigma$. Тогава по дефиниция $w^{rev} = au^{rev}$. Сега имаме $(w^{rev})^{rev} = (au^{rev})^{rev} = (u^{rev})^{rev} a \stackrel{\text{И.П.}}{=} ua = w$.

(б) С индукция по $|w|$ ще докажем, че

$$w = w_1 v w_2 \implies w^{rev} = w_2^{rev} v^{rev} w_1^{rev}.$$

Ако $|w| = 0$, то $w = \epsilon$ и значи $w_1 = v = w_2 = \epsilon$. От друга страна $w^{rev} = \epsilon$ и $w_2^{rev} = v^{rev} = w_1^{rev} = \epsilon$. Равенството следва непосредствено. Ако $|w| = n + 1$ за някое $n \in \mathbb{N}$, то $w = ua$ за някои $u \in \Sigma^*$ и $a \in \Sigma$. Имаме два случая.

1сл. v е поддума на u . Тогава $u = u_1 v u_2$ и от И.П. следва, че $u^{rev} = u_2^{rev} v^{rev} u_1^{rev}$. По дефиниция $w^{rev} = au^{rev} = au_2^{rev} v^{rev} u_1^{rev}$. Освен това от равенството $ua = u_1 v u_2 = w = w_1 v w_2$ можем да съобразим, че $w_1 = u_1$, $w_2 = u_2 a$. Тогава $w_1^{rev} = u_1^{rev}$ и $w_2^{rev} = au_2^{rev}$. Значи $w^{rev} = w_2^{rev} v^{rev} w_1^{rev}$, което искахме.

2сл. v не е поддума на u . Веднага следва, че $w = u_1 v$, където $u_1 \in \Sigma^*$ е такава, че $u = u_1 u_2$ за някоя $u_2 \in \Sigma^*$. От друга страна $w = w_1 v w_2$. Значи $w_1 = u_1$ и $w_2 = \epsilon$. Освен това от И.П. имаме, че $u^{rev} = u_2^{rev} u_1^{rev} = u_2^{rev} w_1^{rev}$. Тогава $w^{rev} = au_2^{rev} w_1^{rev} = v^{rev} w_1^{rev} = \epsilon v^{rev} w_1^{rev} = w_2^{rev} v^{rev} w_1^{rev}$, което искахме.

(в) Нека $w \in \Sigma^*$. Ще проведем индукция по i . Ако $i = 0$, то $(w^i)^{rev} = (w^0)^{rev} = \epsilon^{rev} = \epsilon = \epsilon^0 = (\epsilon^{rev})^0$. Ако $i = n + 1$ за някое $n \in \mathbb{N}$, то $(w^i)^{rev} = ((w^n w)^{rev})^{rev} = w^{rev} (w^n)^{rev} \stackrel{\text{И.П.}}{=} w^{rev} (w^{rev})^n = (w^{rev})^{n+1} = (w^{rev})^i$.

Задача 4. Искане да покажем, че за произволни думи w_1, w_2 и w_3 е изпълнено равенството $(w_1 \circ w_2) \circ w_3 = w_1 \circ (w_2 \circ w_3)$. За целта ще проведем индукция по $|w_3|$. Ако $|w_3| = 0$, то съгласно дефиницията на *конкатенация* имаме $(w_1 \circ w_2) \circ w_3 = w_1 \circ w_2 = w_1 \circ (w_2 \circ w_3)$. Ако $|w_3| = n + 1$ за някое $n \in \mathbb{N}$, то $w_3 = ua$ за някои $u \in \Sigma^*$ и $a \in \Sigma$. Тогава $(w_1 \circ w_2) \circ w_3 = ((w_1 \circ w_2) \circ u)a \stackrel{\text{И.П}}{=} (w_1 \circ (w_2 \circ u))a = w_1 \circ (w_2 \circ ua) = w_1 \circ (w_2 \circ w_3)$.

Задача 5. Твърдим, че $L^+ = L^* \setminus \{\epsilon\} \iff \epsilon \notin L$.

(\Rightarrow) Нека $L^+ = L^* \setminus \{\epsilon\}$. Тогава със сигурност $\epsilon \notin L^+$. Да допуснем, че $\epsilon \in L$. Директно следва, че $\epsilon \in L^+$, Тъй като $\epsilon = \underbrace{\epsilon \circ \dots \circ \epsilon}_{\text{произв. брой пъти}}$. Противоречие. Значи $\epsilon \notin L$.

(\Leftarrow) Обратно нека $\epsilon \notin L$. Сега от дефиницията на L^+ следва, че $\epsilon \notin L^+$. Освен това очевидно $L^+ \subseteq L^*$. Тогава $L^+ \subseteq L^* \setminus \{\epsilon\}$. Сега нека $w \in L^* \setminus \{\epsilon\}$. Тогава $w = w_1 \circ \dots \circ w_k$ за някои $w_1, \dots, w_k \in L$ и $k \geq 0$. От факта, че $w \neq \epsilon$ следва, че $k \geq 1$. Значи $w \in L^+$. Следователно $L^* \setminus \{\epsilon\} \subseteq L^+$. Общо имаме $L^+ = L^* \setminus \{\epsilon\}$.

Глава 2

Крайни автомати и регулярни езици

2.1 Детерминирани крайни автомати

Крайният автомат представлява абстрактна машина снабдена с множество от състояния, азбука и правила за преход между състоянията с буквите от азбуката. Някои от състоянията на автомата са *крайни*, а някои — при *детерминираните крайни автомати*, точно едно — *начални*. Работата на автомата при подадена дума на вход е да прочете думата и в процеса на което, да извърши зададените преходи спрямо снабдените му правила. След прочитането на подадената дума, автоматът има единствена функционалност да върне отговор *да* или *не* на въпроса дали четенето е приключило в крайно състояние.

Дефиниция 1. Детерминиран краен автомат (ДКА) е наредена петорка $A = (Q, \Sigma, \delta, s, F)$, където

- Q е **крайно** множество от **състояния**,
- Σ е азбука,
- $s \in Q$ е **началното състояние**,
- $F \subseteq Q$ е множеството от **крайни състояния** и
- δ , **функцията на преходите**, е функция от $Q \times \Sigma$ към Q .

Неформално, за една дума $w \in \Sigma^*$ казваме, че автоматът A разпознава w , ако четенето на w по автомата приключи в някое заключително състояние $q \in F$.

За да стигнем до формалната дефиниция на това, какво означава един автомат да разпознава дума w , трябва първо да дефинираме така наречената *разширена функция на преходите* $\hat{\delta} : Q \times \Sigma^* \rightarrow Q$. При аргументи състояние q и дума w , функцията $\hat{\delta}$

върща състоянието, в което ще попаднем, тръгвайки от q и четейки думата w . Дефиницията е с индукция по дължината на w .

База: $\hat{\delta}(q, \epsilon) = q$, за всяко $q \in Q$.

Стъпка: Сега да предположим, че $w = ua$ за някои $u \in \Sigma^*$ и $a \in \Sigma$. Тогава

$$\hat{\delta}(q, w) = \delta(\hat{\delta}(q, u), a).$$

Сега вече формално, казваме, че автоматът A **разпознава** думата w , ако просто $\hat{\delta}(s, w) \in F$.

С $L(A)$ означаваме множеството от всички думи $w \in \Sigma^*$, които A разпознава. Тоест, $L(A) = \{w \in \Sigma^* \mid \hat{\delta}(s, w) \in F\}$.

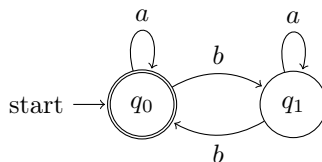
Пример 1. Нека A е детерминираният краен автомат $(Q, \Sigma, \delta, s, F)$, където

$$\begin{aligned} Q &= \{q_0, q_1\}, \\ \Sigma &= \{a, b\}, \\ s &= q_0, \\ F &= \{q_0\}, \end{aligned}$$

и δ е функцията, представена чрез следната таблица.

q	σ	$\delta(q, \sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_1
q_1	b	q_0

Можем да онагледим вида на A чрез следната фигура.



Състоянието q_0 , бивайки крайно, е илюстрирано с двойно оградено кръгче.

Вече лесно се вижда, че $L(A) = \{w \in \{a, b\}^* \mid w \text{ има четен брой } b\text{-та}\}$.

Пример 2. Сега ще построим ДКА A , който разпознава езика

$L = \{w \in \{a, b\}^* \mid w \text{ не съдържа три последователни } b\text{-та}\}$. Нека

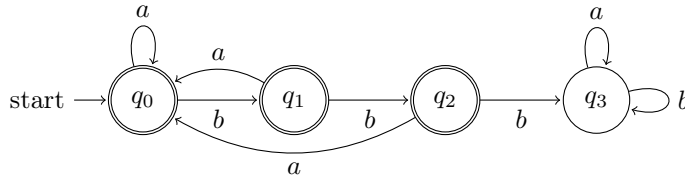
$A = (Q, \Sigma, \delta, s, F)$, където

$$\begin{aligned}
Q &= \{q_0, q_1, q_2, q_3\}, \\
\Sigma &= \{a, b\}, \\
s &= q_0, \\
F &= \{q_0, q_1, q_2\},
\end{aligned}$$

и δ е функцията, представена чрез следната талбита.

q	σ	$\delta(q, \sigma)$
q_0	a	q_0
q_0	b	q_1
q_1	a	q_0
q_1	b	q_2
q_2	a	q_0
q_2	b	q_3
q_3	a	q_3
q_3	b	q_3

Можем да онагледим вида на A чрез следната фигура.



Можем да мислим за състоянието q_3 като за "мъртво състояние"—влезем ли веднъж в него в процеса на четене, не можем да излезем.

2.1.1 Задачи

Задача 1. Нека A е детерминиран краен автомат. Кога е изпълнено, че $\epsilon \in L(A)$? Докажете отговора си.

Задача 2. Постройте детерминирани крайни автомати разпознаващи всеки от следните езици.

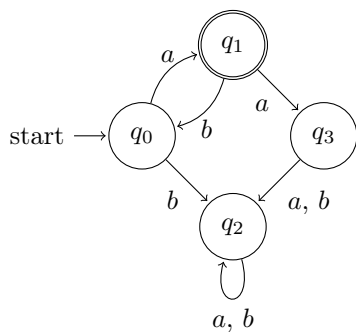
- (а) $\{w \in \{a, b\}^* \mid \text{всяко } a \text{ в } w \text{ е последвано непосредствено от } b\}$.
- (б) $\{w \in \{a, b\}^* \mid w \text{ има } abab \text{ като поддума}\}$.
- (в) $\{w \in \{a, b\}^* \mid w \text{ няма нито } aa, \text{ нито } bb \text{ като поддума}\}$.

(г) $\{w \in \{a, b\}^* \mid w \text{ има нечетен брой } a\text{-та и четен брой } b\text{-та}\}.$

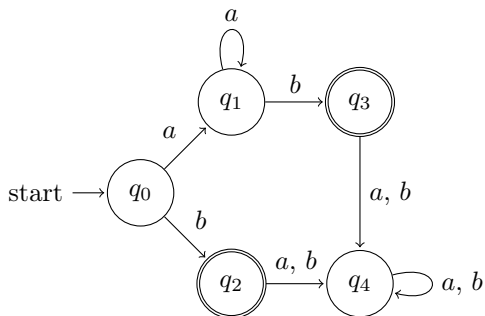
(д) $\{w \in \{a, b\}^* \mid w \text{ има } ab \text{ и } ba \text{ като поддуми}\}.$

Задача 3. За всеки от дадените по-долу автомати посочете езиците, които разпознават.

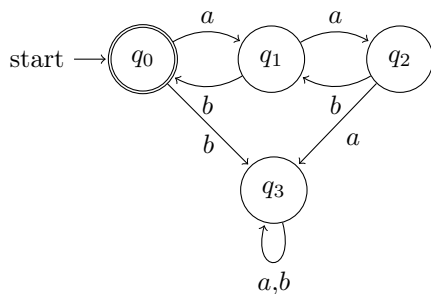
(а)



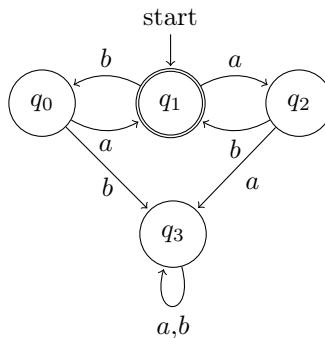
(б)



(в)



(г)



Задача 4. Докажете, че за произволни думи w_1 и w_2 и произволно състояние q , $\hat{\delta}(q, w_1 w_2) = \hat{\delta}(\hat{\delta}(q, w_1), w_2).$

2.1.2 Решения

Задача 1. Нека s е началното състояние на A , а F — множеството от крайните му състояния. Твърдим, че $\epsilon \in L(A) \iff s \in F$.

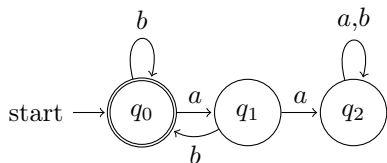
Имаме $\epsilon \in L(A) \iff$

$\hat{\delta}(s, \epsilon) \in F \iff$

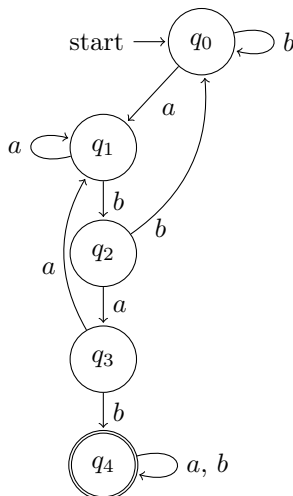
$s \in F$.

Задача 2.

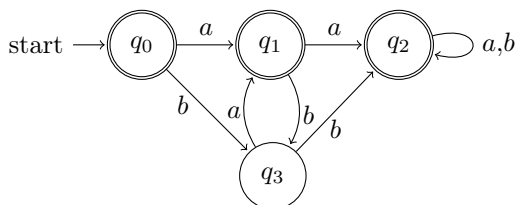
(а)



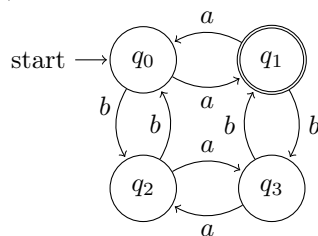
(б)



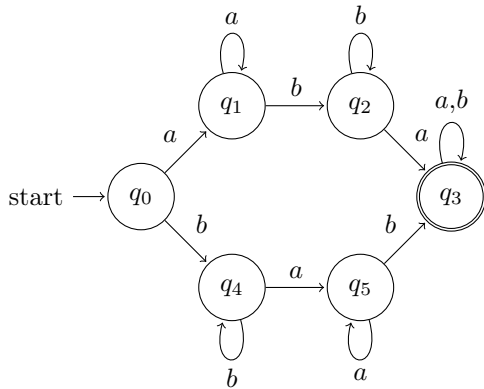
(в)



(г)



(д)



Задача 3. (a) $a \circ \{ba\}^*$
 (б) $\{a\}^* \circ \{b\}$
 (в) $(\{a\} \circ \{ab\}^* \circ \{b\})^*$
 (г) $(\{ab\}^* \circ \{ba\}^*)^*$.

Задача 4. Ще докажем твърдението с индукция по $|w_2|$.

База: Ако $w_2 = \epsilon$, то $\hat{\delta}(q, w_1 w_2) = \hat{\delta}(q, w_1) = \hat{\delta}(\hat{\delta}(q, w_1), \epsilon) = \hat{\delta}(\hat{\delta}(q, w_1), w_2)$.

Стъпка: Ако $w_2 = ua$ за някои $u \in \Sigma^*$ и $a \in \Sigma$, то

$$\hat{\delta}(q, w_1 w_2) =$$

$$\hat{\delta}(q, w_1 ua) =$$

$$\delta(\hat{\delta}(q, w_1 u), a) \stackrel{\text{И.П}}{=} \delta(\hat{\delta}(q, w_1), a)$$

$$\stackrel{\text{деф. } \delta}{=} \delta(\hat{\delta}(\hat{\delta}(q, w_1), u), a)$$

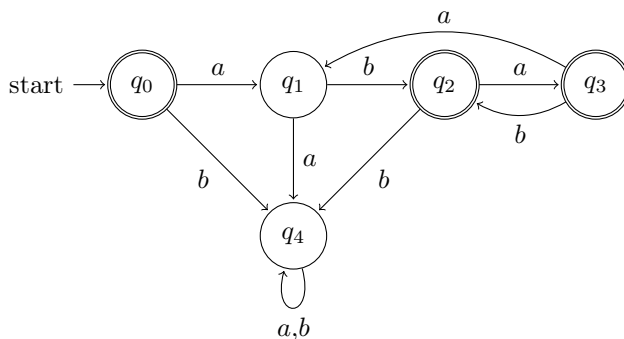
$$\stackrel{\text{деф. } \delta}{=} \hat{\delta}(\hat{\delta}(q, w_1), ua) =$$

$$\hat{\delta}(\hat{\delta}(q, w_1), w_2).$$

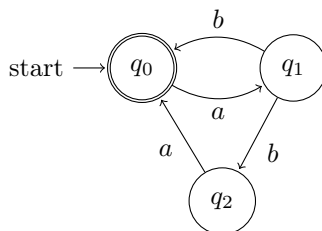
2.2 Недетерминирани крайни автомати

Детерминистичността при детерминираните крайни автомати се изразява в това, че във всеки един момент от четенето на подадената му дума, детерминираният автомат има едно и точно едно състояние, в което може да отиде с току-що прочетената буква. По тази причина, измислянето на ДКА за даден език може да бъде тро-

маво в някои случаи. Например, нека разгледаме следния ДКА, разпознаващ езика $L = (\{ab\} \cup \{aba\})^*$.



Може да се покаже, че не съществува ДКА с по-малко от пет състояния за този език, така че този автомат е минимален по отношение на броя на състоянията му. Сега нека за момент забравим за това, че функцията на преходите е *функция* по същество (тоест за всяка двойка от състояние и буква от домейна има точно едно състояние, което ѝ съответства) и да се опитаме да измислим по-прост "автомат" за дадения език. Най-естествена е следната идея.



Наистина, езика на този "автомат" е точно исканият. Броят на състоянията е само три, а идеята — съвършено проста. Формално обаче, тази илюстрация не съответства на това, което досега сме наричали *автомат*. За да разширим понятието, нека въведем следната дефиниция.

Дефиниция 1. Недетерминиран краен автомат (НКА) е наредена петорка $N = (Q, \Sigma, \Delta, s, F)$, където

- Q е *крайно* множество от **състояния**,
- Σ е азбука,
- $s \in Q$ е **началното състояние**,
- $F \subseteq Q$ е множеството от **крайни състояния** и
- Δ , **функцията на преходите**, е функция от $Q \times \Sigma$ в 2^Q .

По тази дефиниция, автоматът на последната картинка вече точно съответства на конкретен НКА.

Следва да дефинираме **разширената функция на преходите**

$\hat{\Delta} : Q \times \Sigma^* \rightarrow 2^Q$ за НКА. Отново ще сторим това с индукция по думата-аргумент w .

База: $\hat{\Delta}(q, \epsilon) = \{q\}$, за всяко $q \in Q$.

Стъпка: Сега да предположим, че $w = ua$ за някои $u \in \Sigma^*$ и $a \in \Sigma$. Нека също предположим, че $\hat{\Delta}(q, u) = \{q_1, \dots, q_k\}$. Тогава

$$\hat{\Delta}(q, w) = \bigcup_{i=1}^k \Delta(q_i, a).$$

Казваме, че недетерминираният автомат N **разпознава** думата w , ако просто $\hat{\Delta}(s, w) \cap F \neq \emptyset$.

С $L(N)$ означаваме множеството от всички думи $w \in \Sigma^*$, които N разпознава. Тоест, $L(N) = \{w \in \Sigma^* \mid \hat{\Delta}(s, w) \cap F \neq \emptyset\}$.

Вече демонстрирахме удобството, което недетерминираните крайни автомати предоставят. Сега ни вълнува въпроса, дали те са неотменна част от теорията на крайните автомати. Тоест, дали съществуват езици, които се разпознават от някой НКА, но не се разпознават от нито един ДКА. Следната теорема ни дава отговора на този въпрос, а именно — не, всичко, което можем да сторим с недетерминиран автомат, можем да сторим и с детерминиран такъв.

Теорема 1 (на Рабин-Скот). За всеки НКА N съществува ДКА A , такъв че $L(N) = L(A)$.

Доказателството на тази теорема, макар и да не е изложено тук, е *конструктивно*.

С други думи, то предоставя *алгоритъм*, по който от даден НКА N да конструираме *еквивалентен* на него ДКА A (тоест такъв, че $L(N) = L(A)$). Силно неформално, идеята е следната:

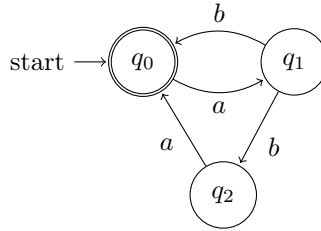
Нека ни е даден НКА $N = (Q_N, \Sigma, \Delta, s, F_N)$. Ще конструираме еквивалентен на него ДКА $A = (Q_A, \Sigma, \delta, s, F_A)$.

1. Инициализираме Q_A , δ и F_A съответно с \emptyset .
2. Добавяме $\{s\}$ към Q_A .
3. Разглеждаме състояние $q \in Q_A$, което не е маркирано като *обходено*. За всяка буква $a \in \Sigma$ добавяме множеството $B = \bigcup_{q_i \in q} \Delta(q_i, a)$ към Q_A и в последствие добавяме двойката $((q, a), B)$ към δ . Маркираме състоянието q като *обходено* и

прилагаме стъпка 3 отново, ако още има необходими състояния.

4. След като необходимите състояния се изчерпат "сканираме" множеството Q_A и добавяме към F_A тези състояния B , за които $B \cap F_N \neq \emptyset$.

Пример 1. Да приложим алгоритъма за детерминиране върху автомата с три състояния, който предложихме за езика $L = (ab \cup aba)^*$.



Първо инициализираме Q , δ и F с \emptyset . След това добавяме състоянието $\{q_0\}$ към Q и преминаваме към стъпка 3.

Единственото състояние в Q е $\{q_0\}$ и то не е маркирано като обходено. Добавяме $\{q_1\}$ и \emptyset към Q съответно заради преходите от q_0 с буквите a и b . Добавяме в δ преходите $((\{q_0\}, a), \{q_1\})$ и $((\{q_0\}, b), \emptyset)$.

Сега разглеждаме необходимото състояние $\{q_1\}$. Добавяме $\{q_0, q_2\}$ в Q заради преходите от q_1 с b . Към δ добавяме преходите $((\{q_1\}, a), \emptyset)$ и $((\{q_1\}, b), \{q_0, q_2\})$.

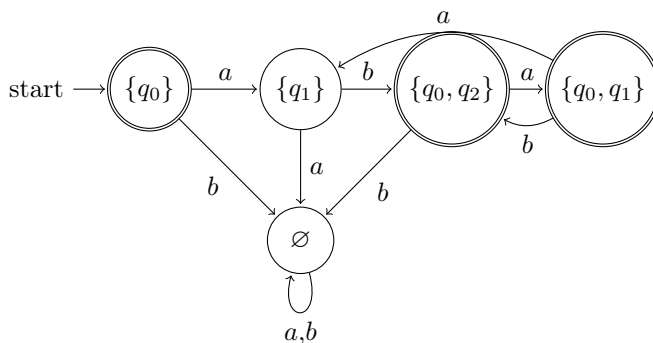
Следва да разгледаме необходимото състояние \emptyset . Добавяме преходите $((\emptyset, a), \emptyset)$ и $((\emptyset, b), \emptyset)$.

Разглеждаме необходимото състояние $\{q_0, q_2\}$. Добавяме $\{q_0, q_1\}$ към Q заради преходите от q_0 и q_1 с буквата a . Добавяме и преходите $((\{q_0, q_2\}, a), \{q_0, q_1\})$ и $((\{q_0, q_2\}, b), \emptyset)$ към δ .

Преминаваме към необходимото състояние $\{q_0, q_1\}$. Добавяме към δ преходите $((\{q_0, q_1\}, a), \{q_1\})$ и $((\{q_0, q_1\}, b), \{q_0, q_2\})$.

С това се изчерпаха необходимите състояния. Сканираме Q и заключаваме, че $F = \{\{q_0\}, \{q_0, q_2\}, \{q_0, q_1\}\}$.

Резултатът от детерминирането е изобразен отдолу.

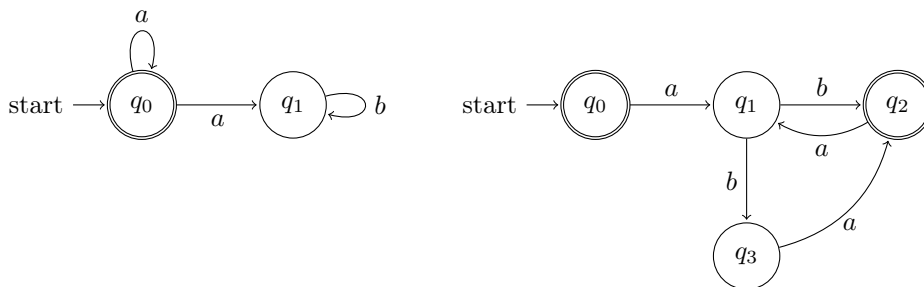


Ясно се вижда, че това всъщност е детерминираният автомат от началото на упражнението, но с променени "имена" на състоянията.

2.2.1 Задачи

Задача 1. (а) Кой от следните думи се разпознават от автомата показан на картинката долу вляво?

- (i) a
- (ii) aa
- (iii) aab
- (iv) ϵ



(б) Кой от следните думи се разпознават от автомата показан на картинката горе вдясно?

- (i) ϵ
- (ii) ab
- (iii) $abab$
- (iv) aba
- (v) $abaa$

Задача 2. Намерете НКА за всеки от следните езици.

- (а) $(\{ab\}^*)(\{ba\}^*) \cup \{a\}\{a\}^*$
- (б) $((\{ab\} \cup \{aab\})^*\{a\}^*)^*$
- (в) $((\{a\}^*\{b\}^*\{a\}^*)^*\{b\})^*$
- (г) $(\{ba\} \cup \{b\})^* \cup (\{bb\} \cup \{a\})^*$

Задача 3. Намерете НКА за всеки от следните езици. След това детерминирайте посочените от вас автомати.

- (а) $(\{ab\} \cup \{aab\} \cup \{aba\})^*$
- (б) $(\{a\} \cup \{b\})^*\{aabab\}$
- (в) $(\{a\} \cup \{b\})^*\{a\}(\{a\} \cup \{b\})(\{a\} \cup \{b\})(\{a\} \cup \{b\})(\{a\} \cup \{b\})$

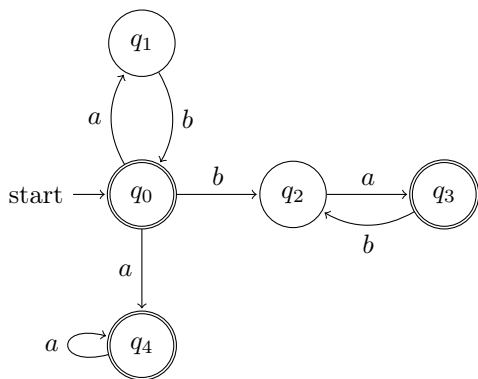
2.2.2 Решения

Задача 1.(а) Всички освен aab .

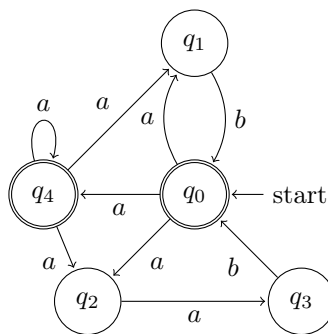
(б) Всички освен $abaa$.

Задача 2.

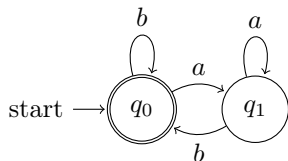
(а)



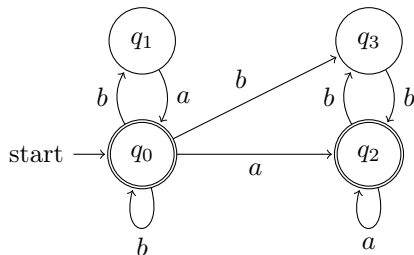
(б)



(в)



(г)



2.3 Затвореност относно регулярните операции

В това упражнение целим да покажем, че класът на *автоматните* езици е затворен относно някои от операциите, които въведохме върху езици — *обединение*, *конкатенация* и *звезда на Клини*.

Дефиниция 1. Казваме, че един език L е **автоматен**, ако съществува ДКА A , такъв че $L(A) = L$.

Теорема 1. Класът на автоматните езици е затворен относно операциите

- (а) *обединение*;
- (б) *конкатенация*;
- (в) *звезда на Клини*;

Доказателството на горната теорема би се състояло в това да се предложат конструкции, които по подадени два детерминирани автомата A_1 и A_2 или единствен ДКА A строят нови детерминирани автомати, разпознаващи съответно езиците $L(A_1) \cup L(A_2)$, $L(A_1) \circ L(A_2)$ и $L(A)^*$, след което да се докаже коректността на предложените конструкции. Ние ще се ограничим само до това да предложим конструкции.

(а) *Обединение.* Нека $A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ и $A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ са детерминирани крайни автомати. Ще конструираме нов НКА N , такъв че $L(N) = L(A_1) \cup L(A_2)$. Идеята е да добавим ново начално състояние, което недетерминистично да извършва всички преходи, които s_1 и s_2 извършват, ефективно опитвайки се да *познаем* дали входната дума принадлежи на $L(A_1)$ или на $L(A_2)$.

Формално, ако Б.О.О. допуснем, че $Q_1 \cap Q_2 = \emptyset$, то N дефинираме както следва: $N = (Q, \Sigma, \Delta, s, F)$, където $s \notin Q_1 \cup Q_2$,

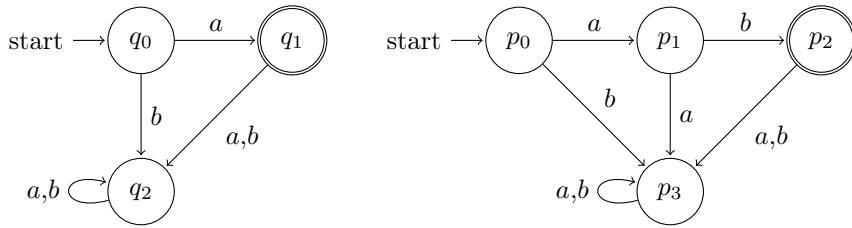
$$Q = Q_1 \cup Q_2 \cup \{s\},$$

$$F = \begin{cases} F_1 \cup F_2 \cup \{s\}, & \text{ако } s_1 \in F_1 \text{ или } s_2 \in F_2, \\ F_1 \cup F_2, & \text{иначе,} \end{cases}$$

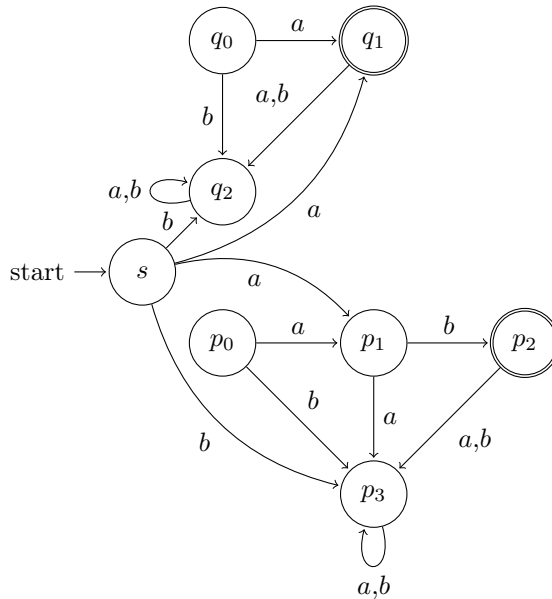
а Δ дефинираме по следния начин

$$\Delta(q, a) = \begin{cases} \{\delta_1(q, a)\}, & \text{ако } q \in Q_1, \\ \{\delta_2(q, a)\}, & \text{ако } q \in Q_2, \\ \{\delta_1(q, a)\} \cup \{\delta_2(q, a)\}, & \text{ако } q = s. \end{cases}$$

Пример 1. Ще построим НКА за езика $\{a\} \cup \{ab\}$ използвайки горната конструкция върху следните два автомата за езиците $\{a\}$ и $\{ab\}$ съответно.



Резултатния НКА е следният.



(б) *Конкатенация.* Отново, нека $A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ и $A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$, са детерминирани крайни автомати. Конструираме НКА N такъв че $L(N) = L(A_1) \circ L(A_2)$. Идеята този път е N да симулира A_1 докато стигне до някое негово крайно състояние, след което да направи недетерминистичен "скок" към A_2 чрез имитация на преходите на s_2 със следващата прочетена буква.

Формално, ако Б.О.О. допуснем, че $Q_1 \cap Q_2 = \emptyset$, то N дефинираме както следва:
 $N = (Q, \Sigma, \Delta, s_1, F)$,

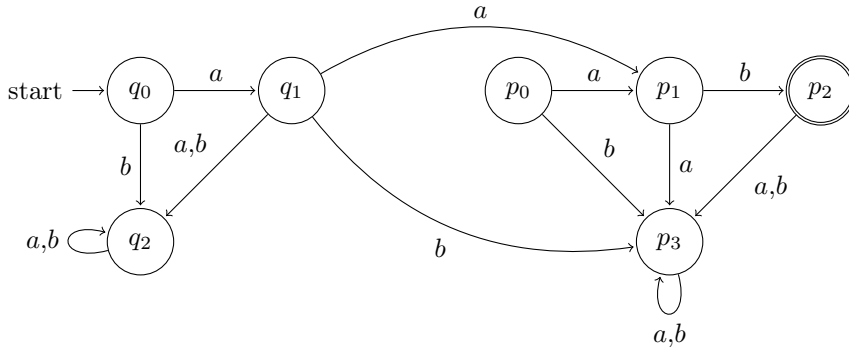
$$Q = Q_1 \cup Q_2$$

$$F = \begin{cases} F_1 \cup F_2, & \text{ако } s_2 \in F_2, \\ F_2, & \text{иначе,} \end{cases}$$

а Δ дефинираме по следния начин

$$\Delta(q, a) = \begin{cases} \{\delta_1(q, a)\}, & \text{ако } q \in Q_1 \setminus F_1, \\ \{\delta_1(q, a)\} \cup \{\delta_2(s_2, a)\}, & \text{ако } q \in F_1, \\ \{\delta_2(q, a)\}, & \text{ако } q \in Q_2. \end{cases}$$

Пример 2. Ще построим НКА за езика $\{a\} \circ \{ab\}$ използвайки горната конструкция върху автоматите от **Пример 1**.



(в) *Звезда на Клини.* Нека $A = (Q, \Sigma, \delta, s, F)$ е ДКА. Ще конструираме НКА N , такъв че $L(N) = L(A)^*$. Естествената идея би била да конкатенираме A към самия себе си без да създаваме копие и да направим началното състояние крайно, ако вече не е такова. Този подход обаче е грешен, защото може да вкара нови думи в езика в случаите, в които A има преходи обратно към началното състояние. За да получим автомат

за $L(A)^*$ трябва да добавим ново начално и заключително състояние, което да имитира преходите на оригиналното начално състояние вместо да правим оригиналното начално състояние крайно.

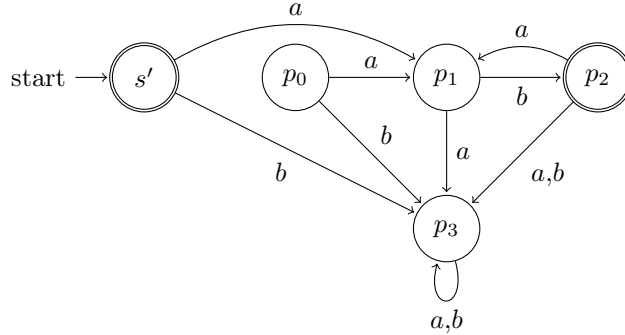
Формално N дефинираме както следва: $N = (Q', \Sigma, \Delta, s', F')$,

$$\begin{aligned} s' &\notin Q \\ Q' &= Q \cup \{s'\} \\ F' &= F \cup \{s'\}, \end{aligned}$$

а Δ дефинираме по следния начин

$$\Delta(q, a) = \begin{cases} \{\delta(q, a)\}, & \text{ако } q \in Q \setminus F_1, \\ \{\delta(s, a)\} \cup \{\delta(q, a)\}, & \text{ако } q \in F_1, \\ \{\delta(s, a)\}, & \text{ако } q = s'. \end{cases}$$

Пример 3. Ще построим НКА за езика $\{ab\}^*$ използвайки горната конструкция върху автомата за $\{ab\}$ от **Пример 1**.



Разбира се, предложените конструкции строят недетерминирани автомати, а нашата цел беше да построим детерминирани такива, за да покажем затвореността на класа на автоматните езици. Както показахме на миналото упражнение, резултатните недетерминирани автомати винаги могат да се детерминират като последна стъпка в предложените конструкции.

2.3.1 Задачи

Задача 1. Дайте *директни* конструкции за затвореността относно следните две операции.

- (а) *Допълнение.*
 (б) *Сечение.*

Забележка. Директни ще рече да не използвате, че сечението може да се представи чрез допълнение и обединение.

Задача 2. Използвайки конструкциите от **Теорема 1**, постройте недетерминирани крайни автомати разпознаващи следните езици.

- (а) $\{a\}^* \circ (\{ab\} \cup \{ba\} \cup \{\epsilon\}) \circ \{b\}^*$
 (б) $(\{a\} \cup \{b\})^* \circ (\{\epsilon\} \cup \{c\})$
 (в) $(\{a\}^* \cup \{b\}^*) \circ \{c\}$

Забележка. В **Задача 2** можете неформално да приложите конструкциите върху недетерминирани автомати, вместо да започнете от детерминирани такива и да детерминирате на всяка стъпка. Ако човек е схванал конструкциите за детерминирани автомати, не би трябвало да се затрудни да ги прилага и за недетерминирани такива, въпреки липсата на формално изложение.

Задача 3. Нека $L \subseteq \Sigma^*$. Дефинираме следните езици.

1. $Pref(L) = \{w \in \Sigma^* \mid x = wy \text{ за някои } x \in L, y \in \Sigma^*\}$ (множеството от **префиксите** на L).
2. $Suff(L) = \{w \in \Sigma^* \mid x = yw \text{ за някои } x \in L, y \in \Sigma^*\}$ (множеството от **суфиксите** на L).
3. $Subseq(L) = \{w_1w_2\dots w_k \mid k \in \mathbb{N}, w_i \in \Sigma^* \text{ за } i = 1, \dots, k, \text{ и има дума } x = x_0w_1x_1w_2x_2\dots w_kx_k \in L\}$. (множеството от **подредиците** на L).
4. $Max(L) = \{w \in L \mid \text{ако } x \neq \epsilon, \text{ то } wx \notin L\}$.
5. $L^{rev} = \{w^{rev} \mid w \in L\}$.

Покажете че ако L е автоматен, то всеки от следните езици също е автоматен.

- (а) $Pref(L)$
 (б) $Suff(L)$
 (в) $Subseq(L)$
 (г) $Max(L)$
 (д) L^{rev}

Задача 4. За два езика L_1 и L_2 над азбука Σ , дефинираме *частното* на L_1 по отношение L_2 да бъде езикът

$$L_1/L_2 = \{w \in \Sigma^* \mid wx \in L_1 \text{ за някое } x \in L_2\}.$$

Докажете, че ако L_1 е автоматен, то L_1/L_2 е автоматен език.

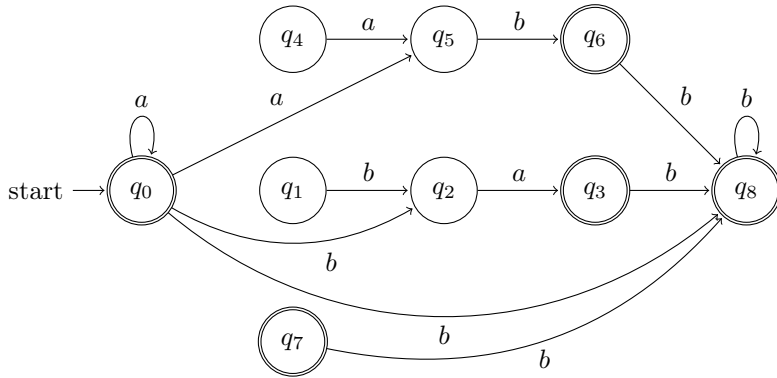
2.3.2 Решения

Задача 1.(а) Достатъчно е да обърнем крайността на всяко състояние. Тоест $F' = Q \setminus F$. Тук обаче е ключово да отбележим, че входният автомат трябва да бъде детерминиран.

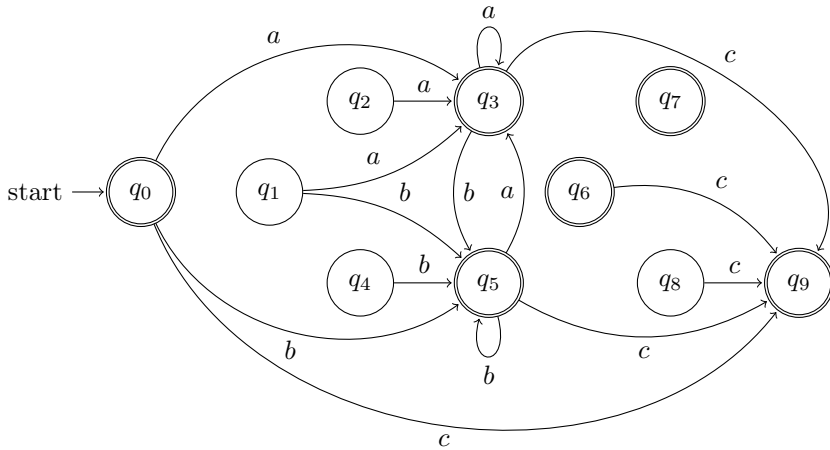
(б) Нека $A_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ и $A_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$. Предлагаме НКА $N = (Q_1 \times Q_2, \Sigma, \Delta, (s_1, s_2), F_1 \times F_2)$, където Δ е дефинирана както следва.

$$\Delta((q_1, q_2), a) = (\delta_1(q_1, a), \delta_2(q_2, a))$$

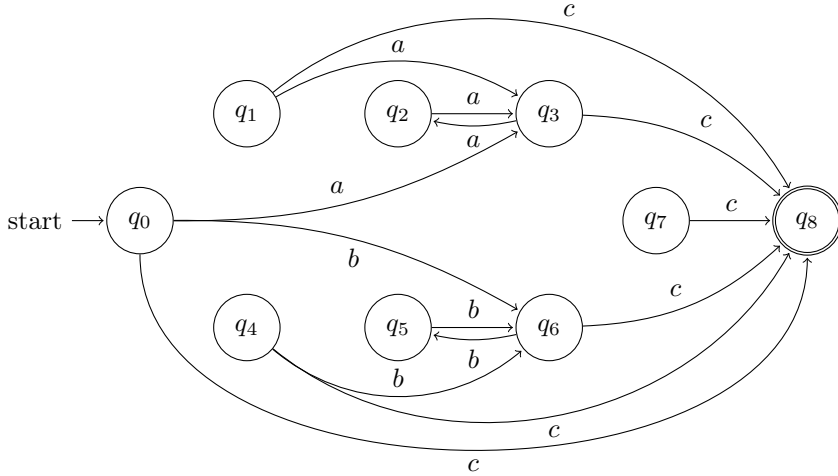
Задача 2. (а)



(б)



(в)



Задача 3. (а) Просто променяме множеството от крайните състояния по следния начин:

$$F' = \{q \in Q \mid (\exists w \in \Sigma^*)[\hat{\delta}(q, w) \in F]\}.$$

(б) Едно състояние $q \in Q$ ще наричаме *достижимо от s* , ако съществува дума $w \in \Sigma^*$, такава че $\hat{\delta}(s, w) = q$. Нека \mathbb{Q} е множеството от всички достижими от s състояния на входния автомат (този, разпознаващ L). Добавяме ново начално състояние $s \notin Q$. Добавяме s към F . Ясно е как работи Δ върху състояния, които не са s . Нека $a \in \Sigma$. Имаме

$$\Delta(s, a) = \{\delta(q, a) \mid q \in \mathbb{Q}\}.$$

(в) Нека $Q = \{q_1, \dots, q_n\}$. За $i = 1, \dots, n$ нека Q_i е множеството от достижимите от q_i състояния. Нека $a \in \Sigma$. Тогава

$$\Delta(q_i, a) = \bigcup_{q \in Q_i} \{\delta(q, a)\}$$

(г) Променяме множеството от крайните състояния по следния начин:

$$F' = \{q \in F \mid (\forall w \in \Sigma^+)[\hat{\delta}(q, w) \notin F]\}.$$

(д) Първо обръщаме преходите на автомата:

$$\Delta(q, a) = \{p \in Q \mid \delta(p, a) = q\}.$$

След това добавяме ново начално състояние $s \notin Q$, което е крайно тстк началното състояние на входния автомат е крайно. Към Δ добавяме преходите

$$\Delta(s, a) = \bigcup_{q \in F} \Delta(q, a),$$

за всяко $a \in \Sigma$.

2.4 Регулярни изрази и теорема на Клини

Регулярният израз е още едно средство за описване на формалните езици над дадена азбука Σ . Грубо казано, регулярните изрази използват само буквите от азбуката и *символите* \emptyset , ϵ , $+$, $*$, $($ и $)$, за да дадат представяне на езиците над Σ . Формалната дефиниция за *синтаксиса* на регулярните изрази е следната.

Дефиниция 1. Нека е дадена азбука Σ . **Регулярен израз** над Σ дефинираме индуктивно, както следва.

База: \emptyset е регулярен израз, ϵ е регулярен израз и всеки елемент на Σ е регулярен израз.

Стъпка: (1) Ако α и β са регулярни изрази, то и $(\alpha\beta)$ е регулярен израз.
 (2) Ако α и β са регулярни изрази, то и $(\alpha + \beta)$ е регулярен израз.
 (3) Ако α е регулярен израз, то и α^* е регулярен израз.

Всеки регулярен израз съответства на един и точно един език над Σ . Формално, релацията между регулярните изрази и езиците, които те представят, се описва чрез функцията \mathcal{L} , такава че ако α е регулярен израз, то $\mathcal{L}(\alpha)$ е езикът, който α представя. Функцията \mathcal{L} ще наричаме

семантика, а стойността ѝ в α — **семантика на α** .

За да дефинираме формално \mathcal{L} ще трябва да кажем еднозначно какво тя връща за всеки даден регулярен израз α над Σ . Това естествено ще сторим с индукция по конструкцията на α .

База: $\mathcal{L}(\emptyset) = \emptyset$, $\mathcal{L}(\epsilon) = \{\epsilon\}$ и $\mathcal{L}(a) = \{a\}$ за всяко $a \in \Sigma$.

Стъпка: (1) Ако α и β са регулярни изрази, то $\mathcal{L}((\alpha\beta)) = \mathcal{L}(\alpha)\mathcal{L}(\beta)$.
 (2) Ако α и β са регулярни изрази, то $\mathcal{L}((\alpha + \beta)) = \mathcal{L}(\alpha) \cup \mathcal{L}(\beta)$.
 (3) Ако α е регулярен израз, то $\mathcal{L}(\alpha^*) = \mathcal{L}(\alpha)^*$.

Пример 1. Да намерим $\mathcal{L}((a + b)^*a)$. Имаме следното.

$$\mathcal{L}((a + b)^*a) = \mathcal{L}((a + b)^*)\mathcal{L}(a) \quad // \text{ съгласно (2)}$$

$$\begin{aligned}
&= \mathcal{L}((a+b)^*)\{a\} && // \text{ съгласно (1)} \\
&= \mathcal{L}((a+b))^*\{a\} && // \text{ съгласно (4)} \\
&= (\mathcal{L}(a) \cup \mathcal{L}(b))^*\{a\} && // \text{ съгласно (3)} \\
&= (\{a\} \cup \{b\})^*\{a\} && // \text{ съгласно (1) два пъти} \\
&= (\{a, b\})^*\{a\} \\
&= \{w \in \Sigma^* \mid w \text{ завършва на } a\}
\end{aligned}$$

Класът на **регулярните езици** над дадена азбука Σ се дефинира да бъде множеството от всички езици L над Σ , такива че $L = \mathcal{L}(\alpha)$ за някой регулярен израз α над Σ . Алтернативно, можем да мислим за класа на регулярните езици като *затварянето* на множеството от езици

$$\{\{\sigma\} \mid \sigma \in \Sigma\} \cup \{\{\epsilon\}\} \cup \{\emptyset\}$$

относно операциите обединение, конкатенация и звезда на Клини.

Основен резултат в теорията на формалните езици е следната теорема за еквивалентност.

Теорема 1(на Клини). Един език L е регулярен тогава и само тогава, когато е автоматен.

Едната посока на доказателството на тази теорема вече на практика сме направили, а именно правата; трябва само да дадем автомати за базовите регулярно езици — \emptyset , $\{\epsilon\}$ и $\{a\}$ за $a \in \Sigma$. От там нататък затвореността на автоматните езици относно операциите обединение, конкатенация и звезда на Клини ни дават именно, че регулярността на един език влече неговата автоматност.

Обратната посока ще бъде от интерес за нас за целите на това упражнение. Как от детерминиран автомат $A = (Q, \Sigma, \delta, s, F)$ да получим регулярен израз α , такъв че $L(A) = \mathcal{L}(\alpha)$? За начало, да номерираме състоянията на автомата; нека $Q = \{q_1, \dots, q_n\}$, като $s = q_1$. За $i, j = 1, \dots, n$ и $k = 0, \dots, n$, дефинираме $R(i, j, k)$ да бъде множеството от всички думи в Σ^* , с които A може да направи преход от състояние q_i до състояние q_j без да премине през *вътрешни* състояния с индекси по-големи от k — крайните състояния за този преход q_i и q_j могат да имат индекси по-големи от k . За $k = n$, имаме че

$$R(i, j, n) = \{w \in \Sigma^* \mid \hat{\delta}(q_i, w) = q_j\}.$$

Следователно

$$L(A) = \bigcup \{R(1, j, n) \mid q_j \in F\}.$$

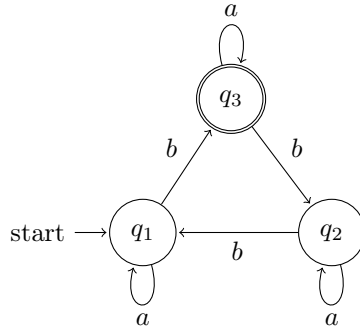
За целите на доказателството следва да се покаже, че езиците $R(i, j, k)$ по всевъзможните i, j и k са регулярни, и следователно такъв е и $L(A)$. Доказателството на този факт се извършва с индукция по k . За нас съществено е изразяването на езика $R(i, j, k)$ чрез езици с по-малка стойност на параметъра k , а именно

$$R(i, j, k) = R(i, j, k-1) \cup R(i, k, k-1)R(k, k, k-1)^*R(k, j, k-1).$$

Това уравнение просто ни казва, че за да стигне от състояние q_i до състояние q_j без да минава през състояние с номер по-голям от k , автоматът A може да стори едно от следните две неща

- (1) да отиде от състояние q_i до състояние q_j , минавайки през вътрешни състояния с номера, по-малки от $k-1$; или
- (2) да премине първо от q_i до q_k ; след което да премине от q_k до q_k нула или повече пъти; след което да премине от q_k до q_j ; като през цялото време не минава през *вътрешни* състояния с индекси по-големи от $k-1$.

Пример 2. Да построим регулярен израз за езика $L = \{w \in \Sigma^* \mid w \text{ има } 3k+1 \text{ } b\text{-та за някое } k \in \mathbb{N}\}$, разпознат от следния автомат.



Ако се опитаме подробно да проследим конструкцията от доказателството на обратната посока на **Теорема 1** ще трябва да построим всички езици $R(i, j, k)$ по всевъзможните стойности за i, j и k , които са $3 * 3 * 4 = 36$ на брой. Ясно е, че $R(i, j, 0) = \{\sigma \in \Sigma \mid \delta(q_i, \sigma) = q_j\}$ за $i, j = 1, 2, 3$.

Нататък за $k = 1$ имаме $R(1, 1, 1) = \mathcal{L}(a^*)$,
 $R(1, 2, 1) = \emptyset$, $R(1, 3, 1) = \mathcal{L}(a^*b)$, $R(2, 1, 1) = R(2, 1, 0)$,
 $R(2, 2, 1) = R(2, 2, 0)$, $R(2, 3, 1) = \mathcal{L}(ba^*b)$, а $R(3, j, 1) = R(3, j, 0)$,
за $j = 1, 2, 3$.

След това за $k = 2$ имаме $R(1, j, 2) = R(1, j, 1)$, за $j = 1, 2, 3$,
 $R(2, 1, 2) = \mathcal{L}(a^*ba^*)$, $R(2, 2, 2) = \mathcal{L}(a^*)$, $R(2, 3, 2) = \mathcal{L}(a^*ba^*b)$,

$$R(3, 1, 2) = \mathcal{L}(ba^*ba^*), R(3, 2, 2) = \mathcal{L}(ba^*), R(3, 3, 2) = \mathcal{L}(a + ba^*ba^*b).$$

Накрая за $k = 3$ е достатъчно да намерим $R(1, 3, 3)$, защото съгласно **Теорема 1**, $L(A) = R(1, 3, 3)$. Имаме

$$\begin{aligned} R(1, 3, 3) &= R(1, 3, 2) \cup R(1, 3, 2)R(3, 3, 2)^*R(3, 3, 2) = \\ &= \mathcal{L}(a^*b) \cup \mathcal{L}(a^*b)\mathcal{L}(ba^*ba^*b)^*\mathcal{L}(ba^*ba^*b) = \mathcal{L}(a^*b + a^*b(a + ba^*ba^*b)^*(a + ba^*ba^*b)) = \\ &= \mathcal{L}(a^*b(\epsilon + (a + ba^*ba^*b)^*(a + ba^*ba^*b))) = \mathcal{L}(a^*b(a + ba^*ba^*b)^*). \end{aligned}$$

2.4.1 Задачи

2.4.2 Решения

2.5 Минимален автомат

Дефиниция 1. Нека $L \subseteq \Sigma^*$ е език, и нека $x, y \in \Sigma^*$. Казваме, че x и y са **еквивалентни по отношение на L** , означено $x \approx_L y$, ако за всяка дума $z \in \Sigma^*$, следното е вярно: $xz \in L \iff yz \in L$. Тази релация наричаме **релация на Нероуд за езика L** .

Забележете, че \approx_L е релация на еквивалентност над Σ^* за всеки език L над Σ . Класовете на еквивалентност на тази релация, неформално казано, са множествата от думи, такива че слепването на коя да е фиксирана дума към края на кои да е две от тях води до получаването на две думи, за които или и двете са в L , или и двете са извън L . С $[x]_L$ означаваме класът на думата x по отношение на релацията на Нероуд за L .

Пример 1. Да намерим класовете на еквивалентност за езика

$L = (ab + ba)^*$. Не е трудно да се съобрази, че този език има точно *четири* класа на еквивалентност по отношение на релацията на Нероуд, а именно:

- (1) $[\epsilon]_L = L$,
- (2) $[a]_L = La$,
- (3) $[b]_L = Lb$,
- (4) $[aa]_L = L(aa + bb)\Sigma^*$.

За (1) можем да отбележим, че не всеки език има свойството, че $[\epsilon]_L = L$. Разгледайте например езика $(ab)^*(a + b)$. (2) и (3) са класовете на думите, които представляват конкатенация на дума от L с a или b съответно. Тези думи могат да се продължат до дума в L *единствено* с представител на bL^* и aL^* съответно. (4)

е класът на думите с дължина ≥ 2 , които не са в L . Каквото и да конкатенираме към произволни два представителя на този клас, резултатната дума ще бъде извън езика.

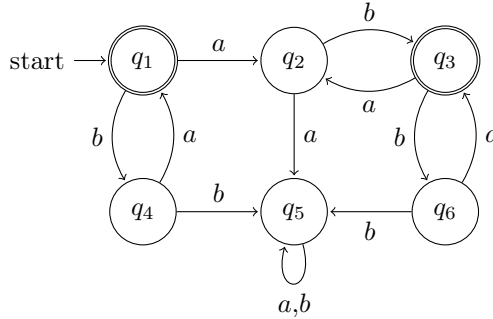
Дефиниция 2. Нека $A = (Q, \Sigma, \delta, s, F)$ е ДКА. Казваме, че две думи $x, y \in \Sigma^*$ са **еквивалентни по отношение на A** , означено с $x \sim_A y$, ако

$$\hat{\delta}(s, x) = \hat{\delta}(s, y).$$

Тази релация наричаме **релация на Нероуд по автомата A** .

Отново, \sim_L е релация на еквивалентност над Σ^* за всеки език L над Σ . Класовете ѝ на еквивалентност се идентифицират с достижимите от s състояния в A . Означаваме съответстващия на състоянието $q \in Q$ клас с E_q .

Пример 2. Да разгледаме отново езика $L = (ab + ba)^*$. Следният автомат A разпознава точно L .



Класовете на еквивалентност на \sim_A са

- (1) $E_{q_1} = (ba)^*$,
- (2) $E_{q_2} = La$,
- (3) $E_{q_3} = (ba)^*abL$,
- (4) $E_{q_4} = b(ab)^*$,
- (5) $E_{q_5} = L(bb + aa)\Sigma^*$,
- (6) $E_{q_6} = (ba)^*abLb$.

Връзката между двете релации, които дефинирахме е следната.

Теорема 1. За всеки ДКА $A = (Q, \Sigma, \delta, s, F)$ и всеки две думи $x, y \in \Sigma^*$, ако $x \sim_A y$, то $x \approx_{L(A)} y$.

Друг начин да изкажем тази връзка е да кажем, че релацията \sim_L *прецизира* релацията $\approx_{L(A)}$. В общия случай, за две релации на еквивалентност R и R' казваме, че R **прецизира** R' , ако за всеки x, y xRy влече $xR'y$. Обратно, ще казваме, че R' **апроксимира** R . Ако релацията на еквивалентност \sim прецизира релацията на еквивалентност \approx , то всеки от класовете на еквивалентност на \sim се съдържа в някой от класовете на \approx . С други думи, всеки клас на еквивалентност на \approx е обединение на един или повече класове на еквивалентност на \sim .

Теорема 1 ни дава, че всеки автомат разпознаващ език L има поне толкова състояния, колкото са класовете на еквивалентност на \approx_L . Тоест, броят на класовете на еквивалентност на тази релация е *долна граница* за броя на състоянията на който да е автомат, разпознаващ езика L . Но дали тази долна граница е достижима? Следната теорема ни дава отговор на този въпрос.

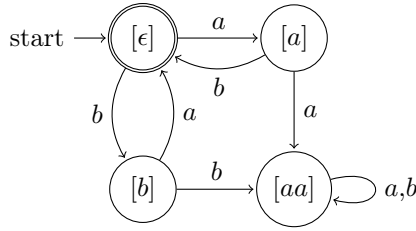
Теорема 2(на Майхил-Нероуд). Нека $L \subseteq \Sigma^*$ е регулярен език. Тогава съществува ДКА с точно толкова на брой състояния, колкото е броят на класовете на еквивалентност на \approx_L , който разпознава L .

Идеята е най-естествена. Използвайки само релацията \approx_L ще конструираме ДКА $A = (Q, \Sigma, \delta, s, F)$, такъв че $L(A) = L$. A е дефиниран както следва:

$$\begin{aligned} Q &= \{[w] \mid w \in \Sigma^*\}, \text{ множеството от класовете на еквивалентност на } \approx_L. \\ s &= [\epsilon], \text{ класът на еквивалентност на } \epsilon \text{ спрямо } \approx_L. \\ F &= \{[w] \mid w \in L\}, \text{ множеството от класовете на еквивалентност на думите в } L \\ &\quad \text{спрямо } \approx_L. \\ \text{Накрая, за всеки клас } [w] \in Q \text{ и всяка буква } a \in \Sigma, \text{ дефинираме } \delta([w], a) &= [wa]. \end{aligned}$$

Разбира се, такъв автомат може да се конструира само ако \approx_L има краен брой класове на еквивалентност. Как сме сигурни, че това е така? Езикът L е регулярен. Значи съществува ДКА A' , такъв че $L(A') = L$. Според **Теорема 2** конструираният от нас автомат A също разпознава L . Но броя на състоянията на A е точно броят на класовете на еквивалентност на \approx_L . Този брой на свой ред е по-малък от броя на класовете на еквивалентност на $\sim_{A'}$, съгласно **Теорема 1**. Сега е достатъчно да забележим, че броят на класовете на еквивалентност на $\sim_{A'}$ е точно броя на състоянията на A' , който е краен. Значи броят на състоянията на A е ограничен отгоре от крайно число.

Пример 3. Минималния автомат за езика $L = (ab + ba)^*$ можем да получим от **Пример 1** и **Пример 2**.



Изведеното до тук не предоставя алгоритъм с който по даден ДКА A да конструираме минималния автомат за $L(A)$. Следва да опишем един такъв алгоритъм. Първо да дефинираме релация $\equiv_A \subseteq Q \times Q$ по следния начин. За всеки две състояния $q, p \in Q$

$$q \equiv_A p \iff (\forall w \in \Sigma^*) [\hat{\delta}(q, w) \in F \iff \hat{\delta}(p, w) \in F].$$

Лесно се забелязва, че тази релация е релация на еквивалентност. Нейните класове на еквивалентност са точно тези множества от състояния, които трябва да се "сляят" в A , за да получим минималния автомат за $L(A)$.

Алгоритъмът ни ще трябва да изчислява класовете на еквивалентност на \equiv_A . За целта дефинираме следната редица от апроксимации на релацията \equiv_A . За всеки две състояния $q, p \in Q$

$$q \equiv_A^n p \iff (\forall w \in \Sigma^*) [|w| \leq n \implies (\hat{\delta}(q, w) \in F \iff \hat{\delta}(p, w) \in F)].$$

Очевидно, всяка от релациите $\equiv_A^0, \equiv_A^1, \equiv_A^2, \dots$ е апроксимация на предшественика си в редицата. Освен това, $q \equiv_A^0 p$ е вярно тстк q и p са едновременно крайни или едновременно некрайни състояния. Тоест \equiv_A^0 има точно два класа на еквивалентност: F и $Q \setminus F$. Сега остана само да покажем как \equiv_A^n зависи от \equiv_A^{n-1} за всяко $n \geq 1$. Връзката е следната

$$(\forall q \in Q)(\forall p \in Q)[q \equiv_A^n p \iff q \equiv_A^{n-1} p \ \& \ (\forall a \in \Sigma)[\delta(q, a) \equiv_A^{n-1} \delta(p, a)]].$$

Алгоритъмът за изчисление на \equiv_A има следния вид.

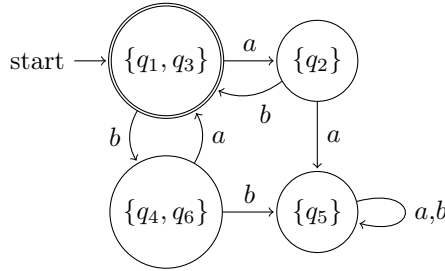
- (1) По начало класовете на \equiv_A^0 са F и $Q \setminus F$;
- (2) За всяко $n = 1, 2, \dots$ изчисли класовете на \equiv_A^n чрез класовете на \equiv_A^{n-1} докато тези на \equiv_A^n не станат същите като тези на \equiv_A^{n-1} .
- (3) Върни \equiv_A^n спрямо текущото достигнатото n .

Пример 4. Да приложим алгоритъма върху автомата от **Пример 2**. Очакваме естествено да получим като резултат автомата от **Пример 3**.

По начало, класовете на \equiv_A^0 са $\{q_1, q_3\}$ и $\{q_2, q_4, q_5, q_6\}$.

След първата итерация на (2), класовете на \equiv_A^1 са $\{q_1, q_3\}$, $\{q_2\}$, $\{q_4, q_6\}$ и $\{q_5\}$.

След втората итерация на (2), класовете не се разбиват допълнително. Алгоритъмът терминира и връща горните четири множества като класовете на еквивалентност на \equiv_A . Това са именно състоянията на минималния автомат за $L = (ab + ba)^*$. Крайни са тези от тях, които са получени от разбиването на класа на крайните състояния (тоест тези, които съдържат само крайни състояния). Преход от състояние Q с буквата σ има към състоянието $\{p \in Q \mid \delta(q, \sigma) = p, \text{ за някое } q \in Q\}$. Резултатният автомат е *изоморфен* на този от **Пример 3**.



2.5.1 Задачи

Задача 1. (а) Намерете класовете на еквивалентност спрямо \approx_L за всеки от следните езици:

- (i) $L = (aab + ab)^*$.
- (ii) $L = \{w \mid w \text{ има поддумата } aababa\}$.
- (iii) $L = \{ww^{rev} \mid w \in \{a, b\}^*\}$.
- (iv) $L = \{ww \mid w \in \{a, b\}^*\}$.
- (v) $L_n = (a + b)^* a(a + b)^n$, където $n \in N^+$.

(б) За тези езици от (а), за които отговорът е краен, дайте минимален ДКА, разпознаващ съответния език.

2.5.2 Решения

2.6 Критерии за нерегулярност

Вече разгледахме два метода, чрез които показваме регулярността на даден език. Регулярните езици над дадена азбука Σ обаче са изброимо много (защото регулярните изрази над Σ са изброимо много). От друга страна, множеството на всички езици над Σ , $\mathcal{P}(\Sigma^*)$, е неизброимо безкрайно като степенно множество на изброимо безкрайно множество, Σ^* . Тези съображения показват съществуването на нерегулярни езици

над Σ . Ясно е, че каквато и да е Σ и каквито и да са тези езици, те най-малкото ще бъдат безкрайни. Но какво откроява безкрайните нерегулярни езици от безкрайните регулярни такива? Както знаем, всеки регулярен език се разпознава от ДКА. Ако L е безкраен регулярен език, а A е ДКА с n състояния, такъв че $L(A) = L$, то безкрайността на L влече съществуването на думи с дължина поне n . Нека w е една такава дума. Тогава пътят на w по A със сигурност минава през поне $n + 1$ състояния. По принципа на Дирихле, някое от състоянията по този път ще се повтаря; тоест пътят на w по A ще съдържа цикъл. Естествено движейки се по A ние можем да изберем да се "завъртим" по този цикъл произволен брой пъти, след което наличието на w в L ни гарантира, че ще можем да приключим този път в крайно състояние, ефективно вкарвайки безкрайно много думи (за всяко количество "завъртания"), зависими по някакъв начин от w , в L . Всичко това можем да изкажем чрез следната теорема.

Теорема 1 (лема за разрастването/the pumping lemma). Нека L е регулярен език. Тогава съществува естествено число $p \geq 1$, такова че всяка дума $w \in L$ с $|w| \geq p$ може да се презапише във вида $w = xyz$, като

- (1) $|y| \geq 1$, тоест $y \neq \epsilon$;
- (2) $|xy| \leq p$;
- (3) $xy^iz \in L$ за всяко $i \in \mathbb{N}$.

Първия (най-малкия) свидетел за съществуването на числото p е именно броят на състоянията на минималния автомат за L . Такъв автомат знаем че съществува, заради регулярността на L . Мислейки за минималния автомат A за L , всяка дума $w \in L$ с $|w| \geq p$ ще съдържа цикъл в пътя си по A . Представянето $w = xyz$ заедно със свойствата си (1), (2) и (3) точно съответства на този факт. Наистина, x е думата прочетена от началното състояние до началото на *първия* цикъл в пътя на w по A ; y е думата прочетена по този цикъл (затова $|y| \geq 0$), а z е думата прочетена от края на цикъла до крайното състояние, в което w приключва пътя си по A . Свойството (2) следва от избора ни y да е думата прочетена по първия цикъл в пътя на w по A — първото повторение на състояние ще се случи след прочитането на y , тоест $|xy|$ няма как да бъде по-голяма от броя на състоянията на A . Свойството (3) е именно свойството, което дава и името на теоремата; щом можем да се въртим по така фиксирания първи цикъл в пътя на w по A произволен брой пъти, четейки по него думата y , то за всяко $i \in \mathbb{N}$, думата xy^iz ще да бъде в $L(A)$, а значи и в L .

Теорема 1 по същество е импликация от вида

$$L \text{ е регулярен } \implies P(L).$$

Разбира се в сила е и нейната контрапозиция

$$\neg P(L) \implies L \text{ не е регулярен.}$$

Това е първият критерии за нерегулярност, който ще разгледаме. Отрицанието на свойството P трябва да се разгледа по-подробно. Първо, да запишем свойството P формално.

$$P(L) \iff (\exists p \in \mathbb{N}^+)(\forall w \in L)[|w| \geq p \implies (\exists x \in \Sigma^*)(\exists y \in \Sigma^*)(\exists z \in \Sigma^*)[w = xyz \ \& \ |y| \geq 1 \ \& \ |xy| \leq p \ \& \ (\forall i \in \mathbb{N})[xy^iz \in L]]]$$

Сега можем да съобразим, че отрицанието на P е точно

$$\neg P(L) \iff (\forall p \in \mathbb{N}^+)(\exists w \in L)[|w| \geq p \ \& \ (\forall x \in \Sigma^*)(\forall y \in \Sigma^*)(\forall z \in \Sigma^*)[w = xyz \ \& \ |y| \geq 0 \ \& \ |xy| \leq p \implies (\exists i \in \mathbb{N})[xy^iz \notin L]]].$$

Ако успеем да покажем, че даден език притежава горното свойство, то контрапозицията на лемата за разрастването ни дава, че този език не е регулярен.

Първо трябва да удовлетворим кванторите в началото на формулата

$$(\forall p \in \mathbb{N}^+) \text{ и } (\exists w \in L), \text{ както и първия конюнкт в областта им на действие } |w| \geq p.$$

За целта избираме в ролята на w подходяща дума с дължина по-голяма от и зависеща от p в L . Зависимостта на дължината на w от p е това, което удовлетворява първия квантор.

След това удовлетворяваме втория конюнкт в областта на действие на $(\exists w \in L)$.

За целта разглеждаме произволно представяне на така избраната w във вида $w = xyz$ със свойствата (1) и (2) и показваме такова естествено число i , че $xy^iz \notin L$.

Пример 1. Ще докажем посредством **Теорема 1**, че езикът

$L = \{a^n b^n \mid n \in \mathbb{N}\}$ не е регулярен, като покажем, че L не притежава свойството P (тоест, че L притежава свойството $\neg P$). Нека $p \in \mathbb{N}^+$. Да разгледаме думата $w = a^p b^p \in L$. Очевидно $|w| \geq p$. Сега да разгледаме някое представяне $w = xyz$ на w със свойствата (1) и (2). Щом $|xy| \leq p$ и $|y| \geq 1$, то $y = a^k$ за някое k , такова че $1 \leq k \leq p$. Тогава за $i = 2$ имаме, че $xy^2z = a^{p+k} b^p \notin L$.

Сега да си припомним, че **теоремата на Майхил-Нероуд** също беше импликация от вида

$$L \text{ е регулярен } \implies N(L),$$

където $N(L)$ е следното свойство:

$$N(L) \iff \text{съществува ДКА за } L \text{ с толкова на брой състояния, колкото са класовете на еквивалентност на } \approx_L.$$

Това свойство влече, че броят на класовете на еквивалентност на \approx_L трябва да бъде краен. Тоест имаме импликацията

$$N(L) \implies \approx_L \text{ има краен брой класове на еквивалентност.}$$

Двете импликации дотук ни дават общо

$$L \text{ е регулярен } \implies \approx_L \text{ има краен брой класове на еквивалентност.}$$

Значи в сила е и контрапозитивното

$$\approx_L \text{ има безкраен брой класове на еквивалентност } \implies L \text{ не е регулярен.}$$

Това е вторият критерий за нерегулярност, който ще разгледаме. Но как можем да покажем, че даден език има безкраен брой класове на еквивалентност по релацията си на Нероуд? Идеята е да намерим безкрайна редица от думи w_1, w_2, \dots , такава че всеки две думи в нея са в различен клас на еквивалентност по \approx_L . Най-простият начин е да създадем подходяща зависимост между дължината на думата w_i и индекса ѝ — числото i .

Пример 2. Ще покажем отново, че езикът $L = \{a^n b^n \mid n \in \mathbb{N}\}$ не е регулярен; този път използвайки новия критерий. Да разгледаме редицата w_1, w_2, \dots , за която

$$w_i = a^i, \text{ за всяко } i \in \mathbb{N}.$$

Нека $i, j \in \mathbb{N}$ са такива, че $i \neq j$. Ще покажем, че $[w_i]_L \neq [w_j]_L$. За целта е достатъчно да покажем, че $w_i \notin [w_j]_L$, тъй като $w_i \in [w_i]_L$. Еквивалентно на това, трябва да покажем, че $w_i \not\approx_L w_j$. Да разгледаме думата $b^i \in \Sigma^*$. От една страна $a^i b^i \in L$. От друга страна $a^j b^i \notin L$, тъй като $i \neq j$. Значи действително $w_i \not\approx_L w_j$. Така показахме, че \approx_L има безкраен брой класове на еквивалентност. Следователно L не е регулярен.

Пример 3. Ще покажем и трети начин, по който можем да показваме, че даден език е нерегулярен. Този начин е да се възползваме от регулярните операции и езици, за които вече сме показали, че са нерегулярни. Например да разгледаме езика

$$L = \{w \in \{a, b\}^* \mid w \text{ има равен брой } a\text{-та и } b\text{-та}\}.$$

L не е регулярен, защото ако беше, то такъв би бил и $L \cap a^* b^*$ — по затвореност на регулярните езици относно сечение. Обаче, $L \cap a^* b^* = \{a^n b^n \mid n \in \mathbb{N}\}$, за който вече два пъти показахме, че не е регулярен.

2.6.1 Задачи

Задача 1. Използвайте лемата за разрастването или теоремата на Майхил-Нероуд, за да покажете, че следните езици не са регулярни.

- (а) $\{ww \mid w \in \{a, b\}^*\}$
- (б) $\{ww^{rev} \mid w \in \{a, b\}^*\}$

- (в) $\{w\bar{w} \mid w \in \{a, b\}^*\}$, където \bar{w} е думата получена от w , като сме заменили всяко срещане на a с b и обратно
 (г) $\{w^{|w|} \mid w \in \{a, b\}^*\}$.

Задача 2. Докажете, че езикът $\{a^n b a^m b a^{n+m} \mid n, m \geq 1\}$ не е регулярен.

Задача 3. Една дума w над $\{a, b\}$ наричаме *балансирана*, ако следните две неща са изпълнени:

- (1) във всеки префикс на w , броят на a -тата е не по-малък от броя на b -тата;
 (2) броят на a -тата в w е равен на броя на b -тата.

Докажете, че езикът $\{w \in \{a, b\}^* \mid w \text{ е балансирана}\}$ не е регулярен.

Задача 4. Кои от следните твърдения са верни? Аргументирайте се.

- (а) Всяко подмножество на регулярен език е регулярен език.
 (б) Всеки регулярен език си има собствено подмножество, което е регулярен език.
 (в) Ако L е регулярен, то регулярен е и $\{xy \mid x \in L \text{ \& } y \notin L\}$.
 (г) $\{w \mid w = w^{rev}\}$ е регулярен.
 (д) Ако L е регулярен език, то такъв е и $\{w \mid w \in L \text{ \& } w^{rev} \in L\}$.
 (е) Ако R е множество от регулярни езици, то $\bigcup R$ е регулярен език.
 (ж) $\{xyx^{rev} \mid x, y \in \Sigma^*\}$ е регулярен.

2.6.2 Решения

Задача 1. Ако използвате PL едни възможни избори са:

- (а) $w = a^p b a^p b$, $i = 2$;
 (б) $w = a^p b b a^p$, $i = 2$;
 (в) $w = a^p b^p$, $i = 2$;
 (г) $w = (a^p b)^{p+1}$, $i = 2$.

Ако използвате Майхил-Нероуд:

- (а) $w_i = a^i b$, за $i \in \mathbb{N}$; конкатенирате $a^i b$ към w_i и w_j за $i \neq j$;
 (б) $w_i = a^i b$, за $i \in \mathbb{N}$; конкатенирате $b a^i$ към w_i и w_j за $i \neq j$;
 (в) $w_i = a^i$, за $i \in \mathbb{N}$; конкатенирате b^i към w_i и w_j за $i \neq j$;
 (г) $w_i = (a^i b)^i$, за $i \in \mathbb{N}$; конкатенирате $a^i b$ към w_i и w_j за $i \neq j$;

Алтернативен подход за (в) е да пресечем езика с $a^* b^*$, получавайки $\{a^n b^n \mid n \in \mathbb{N}\}$.

Задача 2. С PL: $w = a^p b a^p b a^{2p}$, $i = 2$. С Майхил-Нероуд: $w_i = a^{i+1} b a^{i+1} b$, за $i \in \mathbb{N}$ и конкатенирате $a^{2(i+1)}$ към w_i и w_j за $i \neq j$.

Задача 3. Пресичаме езика с a^*b^* и получаваме $\{a^n b^n \mid n \in \mathbb{N}\}$.

Задача 4. (а) Не е вярно. Най-просто, Σ^* е регулярен език и всеки език над Σ е подмножество на Σ^* . Но както показахме в това упражнение, над Σ съществуват и нерегулярни езици.

(б) Не е вярно. Тривиално, \emptyset е регулярен език, който няма собствени подмножества.

(в) Вярно е. Имаме

$$\begin{aligned} \{xy \mid x \in L \ \& \ y \notin L\} &= \\ \{xy \mid x \in L \ \& \ y \in \overline{L}\} &= \\ \{x \mid x \in L\} \circ \{y \mid y \in \overline{L}\} &= L\overline{L}. \end{aligned}$$

Но ние вече показахме, че регулярните езици са затворени относно операциите допълнение и конкатенация.

(г) Не е вярно. Ако пресечем този език с езика на думите с четна дължина, $(aa + ab + ba + bb)^*$, получаваме езика от (б) на **Задача 1**.

(д) Вярно е. Имаме

$$\begin{aligned} \{w \mid w \in L \ \& \ w^{rev} \in L\} &= \\ \{w \mid w \in L \ \& \ w \in L^{rev}\} &= L \cap L^{rev}. \end{aligned}$$

Но ние вече показахме, че регулярните езици са затворени относно операциите обръщане и сечение.

(е) Не е вярно. Въпреки че обединението, приложено краен брой пъти, запазва регулярността, ако го приложим върху подходящо безкрайно множество от регулярни езици, можем да получим нерегулярен език. Конкретно, да разгледаме в ролята на R множеството от крайни и следователно регулярни езици $\{\{\epsilon\}, \{ab\}, \{aabb\}, \{aaabbb\}, \dots\}$.

Очевидно $\bigcup R = \{a^n b^n \mid n \in \mathbb{N}\}$.

(ж) Вярно е. Този език е точно Σ^* .

Глава 3

Граматики и стекови автомати

3.1 Контекстно-свободни граматики

Автоматите, които разглеждахме досега бяха **разпознаватели на езици**. *Граматиката*, подобно на регулярния израз е **генератор на езици**. Тя представлява множество от правила и променливи, посредством които строим думи над дадена азбука.

Дефиниция 1. Контекстно-свободна граматика е наредена четворка $G = (V, \Sigma, R, S)$, където

- V е *крайно* множество от **нетерминали(променливи)**,
- Σ е азбука, чиито елементи ще наричаме **терминали**,
- R , **множеството от правилата**, е крайно подмножество на $V \times (\Sigma \cup V)^*$ и
- $S \in V$ е **началната променлива**.

Вместо $(X, y) \in R$ ще пишем $X \rightarrow_G y$. За всеки две думи $u, v \in (\Sigma \cup V)^*$ ще пишем $u \Rightarrow_G v$ тогава и само тогава, когато съществуват думи $x, y \in (\Sigma \cup V)^*$ и нетерминал $A \in V$, такива че $u = xAy, v = xv'y$ и $A \rightarrow_G v'$. Релацията \Rightarrow_G^* е рефлексивното и транзитивно затваряне на \Rightarrow_G . Накрая, **езика генериран от G** е езикът

$$L(G) = \{w \in \Sigma^* \mid S \Rightarrow_G^* w\}.$$

Един език L наричаме **контекстно-свободен**, ако $L = L(G)$ за някоя контекстно-свободна граматика G . Ако от контекста се подразбира, за коя граматика става въпрос, ще пишем \rightarrow вместо \rightarrow_G и \Rightarrow вместо \Rightarrow_G .

Всяка редица от вида

$$w_0 \Rightarrow_G w_1 \Rightarrow_G \dots \Rightarrow_G w_n$$

наричаме **извод** по G на w_n от w_0 , за $w_0, \dots, w_n \in (\Sigma \cup V)^*$ и $n \in \mathbb{N}$. Дължината на един извод е броя на срещанията на символа $\boxed{\Rightarrow_G}$ в него. Фактът, че съществува извод с дължина n на v от u по G ще записваме $u \Rightarrow_G^n v$.

Пример 1. Да разгледаме контекстно-свободната граматика

$G = (V, \Sigma, R, S)$, където $V = S$, $\Sigma = \{a, b\}$ и R се състои от правилата $S \rightarrow aSb$ и $S \rightarrow \epsilon$. Възможен извод по G е например

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aabb.$$

На първите две стъпки използвахме правилото $S \rightarrow aSb$, а на последната използвахме $S \rightarrow \epsilon$. Лесно е да се съобрази, че $L(G) = \{a^n b^n \mid n \in \mathbb{N}\}$. Следователно, някои контекстно-свободни езици не са регулярни.

Пример 2. Следната граматика генерира всички думи от балансирани леви и десни скоби: всяка лява скоба може да се съчетае с уникална дясна скоба след нея, и всяка дясна скоба може да се съчетае с уникална лява скоба преди нея. Нека $G = (V, \Sigma, R, S)$, където

$$\begin{aligned} V &= \{S\}, \\ \Sigma &= \{(\,,)\}, \\ R &= \{S \rightarrow \epsilon, S \rightarrow SS, S \rightarrow (S)\}. \end{aligned}$$

Два възможни извода по тази граматика са например

$$S \Rightarrow SS \Rightarrow S(S) \Rightarrow S((S)) \Rightarrow S(()) \Rightarrow (S)(()) \Rightarrow ()(())$$

и

$$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ()S \Rightarrow ()(S) \Rightarrow ()((S)) \Rightarrow ()(()).$$

Значи една и съща дума може да има два различни извода от S по една и съща граматика.

Сега ще покажем, че всички регулярни езици са контекстно-свободни. За целта ще дадем директна конструкция, която по подаден ДКА

$A = (Q, \Sigma, \delta, s, F)$ генерира контекстно-свободна граматика

$G = (V, \Sigma, R, S)$, такава че $L(A) = L(G)$. Конструкцията е следната.

$$\begin{aligned} V &= Q, \\ S &= s, \\ R &= \{q \rightarrow \sigma p \mid \delta(q, \sigma) = p\} \cup \{q \rightarrow \epsilon \mid q \in F\}. \end{aligned}$$

Има и други начини да покажем, че регулярните езици са подмножество на контекстно-свободните, но тях ще покажем по-нататък.

3.1.1 Задачи

Задача 1. Да разгледаме граматиката $G = (V, \Sigma, R, S)$, където

$$\begin{aligned} V &= \{S, A\}, \\ \Sigma &= \{a, b\}, \\ R &= \{S \rightarrow AA, A \rightarrow AAA, A \rightarrow a, A \rightarrow bA, A \rightarrow Ab\}. \end{aligned}$$

- (а) Кои думи от $L(G)$ могат да се изведът с извод с дължина не повече от четири?
- (б) Дайте поне четири различни извода на думата *babbab*.
- (в) За всеки $m, n, p > 0$, опишете извод по G на думата $b^m a b^n a b^p$.

Задача 2. Да разгледаме граматиката $G = (V, \Sigma, R, S)$, където

$$\begin{aligned} V &= \{S, A\}, \\ \Sigma &= \{a, b\}, \\ R &= \{S \rightarrow aAa, S \rightarrow bAb, S \rightarrow \epsilon, A \rightarrow SS\}. \end{aligned}$$

Дайте извод на думата *baabbb* по G .

Задача 3. Дайте контекстно-свободни граматики за всеки от следните езици.

- (а) $\{w \# w^{rev} \mid w \in \{a, b\}^*\}$
- (б) $\{ww^{rev} \mid w \in \{a, b\}^*\}$
- (в) $\{w \in \{a, b\}^* \mid w = w^{rev}\}$

Задача 4. Да фиксираме азбука $\Sigma = \{a, b, (,), +, *, \epsilon, \oslash\}$. Дайте контекстно-свободна граматика, която генерира всички думи над Σ^* , които са регулярни изрази над $\{a, b\}$.

Задача 5. Дайте контекстно-свободни граматики за всеки от следните езици.

- (а) $\{a^m b^n \mid m \geq n\}$
- (б) $\{a^m b^n c^p d^q \mid m + n = p + q\}$
- (в) $\{w \in \{a, b\}^* \mid \text{броят на } b\text{-тата в } w \text{ е равен на два пъти броя на } a\text{-тата}\}$
- (г) $\{uawb \mid u, w \in \{a, b\}^* \text{ \& } |u| = |w|\}$
- (д) $\{w_1 \# w_2 \# \dots \# w_k \# \# w_j^{rev} \mid k \geq 1 \text{ \& } 1 \leq j \leq k \text{ \& } w_i \in \{a, b\}^+ \text{ за } i = 1, \dots, k\}$
- (е) $\{a^m b^n \mid m \leq 2n\}$

3.1.2 Решения

Задача 1. (а) aa, baa, aba, aab ;

(б) 1. $S \rightarrow AA \rightarrow bAA \rightarrow bAbA \rightarrow bAbbA \rightarrow bAbbAb \rightarrow babbAb \rightarrow babbab$;

2. $S \rightarrow AA \rightarrow AbA \rightarrow AbAb \rightarrow bAbAb \rightarrow bAbbAb \rightarrow babbAb \rightarrow babbab$;

3. $S \rightarrow AA \rightarrow bAA \rightarrow bAAb \rightarrow bAbAb \rightarrow bAbbAb \rightarrow babbAb \rightarrow babbab$;

4. $S \rightarrow AA \rightarrow AbA \rightarrow AbbA \rightarrow bAbbA \rightarrow bAbbAb \rightarrow babbAb \rightarrow babbab$.

(в) $S \rightarrow AA \rightarrow \underbrace{bAA \rightarrow bbAA \rightarrow \dots \rightarrow b^m AA}_{m \text{ пъти}} \rightarrow \underbrace{b^m AbA \rightarrow b^m AbbA \rightarrow \dots \rightarrow b^m Ab^n A}_{n \text{ пъти}} \rightarrow \underbrace{b^m ab^n A \rightarrow b^m ab^n Ab \rightarrow b^m ab^n Abb \rightarrow \dots \rightarrow b^m ab^n Ab^p}_{p \text{ пъти}} \rightarrow b^m ab^n ab^p$.

Задача 2. $S \rightarrow bAb \rightarrow bSSb \rightarrow baAaSb \rightarrow baAabAbb \rightarrow baabAbb \rightarrow baabbb$.

Задача 3. (а) $S \rightarrow aSa \mid bSb \mid \#$;

(б) $S \rightarrow aSa \mid bSb \mid \epsilon$;

(в) $S \rightarrow aSa \mid bSb \mid a \mid b \mid \epsilon$.

Задача 4. $S \rightarrow a \mid b \mid \oslash \mid \epsilon \mid (SS) \mid (S + S) \mid S^*$.

Задача 5.(а) $S \rightarrow aS \mid aSb \mid \epsilon$;

(б) $S \rightarrow aSd \mid A \mid B \mid \epsilon$

$A \rightarrow aAc \mid C \mid \epsilon$

$B \rightarrow bBd \mid C \mid \epsilon$

$C \rightarrow bCc \mid \epsilon$;

(в) $S \rightarrow aSbSbS \mid bSaSbS \mid bSbSaS \mid \epsilon$;

(г) $S \rightarrow Ab$

$A \rightarrow aAb \mid bAa \mid aAa \mid bAb \mid a$;

(д) $S \rightarrow AB \mid B$

$B \rightarrow aBa \mid bBb \mid \#A\# \mid \#\#$

$A \rightarrow aA \mid bA \mid AA \mid \#$;

(е) $S \rightarrow aSbb \mid Sb \mid \epsilon$;

3.2 Затвореност относно регулярните операции

В духа на това, което направихме при регулярните езици, сега ще покажем някои свойства на затвореност на контекстно-свободните езици относно операции върху езици.

Теорема 1. Контекстно-свободните езици са затворени относно обединение, конкатенация и звезда на Клини.

Нека $G_1 = (V_1, \Sigma, R_1, S_1)$ и $G_2 = (V_2, \Sigma, R_2, S_2)$ са контекстно-свободни граматички и без ограничение на общността да допуснем, че $V_1 \cap V_2 = \emptyset$. Конструкциите са следните.

Обединение. Нека S е символ, който не принадлежи на $V_1 \cup V_2$. Езикът $L(G_1) \cup L(G_2)$ се генерира от граматика

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{S \rightarrow S_1, S \rightarrow S_2\}, S).$$

Конкатенация. Подобно, $L(G_1) \circ L(G_2)$ се генерира от граматиката

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma, R_1 \cup R_2 \cup \{S \rightarrow S_1 S_2\}, S).$$

Звезда на Клини. $L(G_1)^*$ се генерира от

$$G = (V_1 \cup \{S\}, \Sigma, R_1 \cup \{S \rightarrow \epsilon, S \rightarrow S S_1\}, S).$$

Теорема 1 влече вече доказан факт, че класът на регулярните езици се съдържа в класа на контекстно-свободните такива.

3.2.1 Задачи

Задача 1. Използвайте затвореността относно обединение, за да покажете, че следните езици са контекстно-свободни.

- (а) $\{a^m b^n \mid m \neq n\}$
- (б) $\{a, b\}^* \setminus \{a^n b^n \mid n \in \mathbb{N}\}$
- (в) $\{a^m b^n c^p d^q \mid n = q, \text{ или } m \leq p \text{ или } m + n = p + q\}$
- (г) $\{w \in \{a, b\}^* \mid w = w^R\}$

Задача 2. Покажете, че езикът $L = \{a^n b^{n+m} a^n \mid n, m \in \mathbb{N}\}$ е контекстно свободен, използвайки затвореността относно конкатенация.

Задача 3. Покажете, че езикът $L = \{uw^{rev}v \mid w = vu\}$ е контекстно свободен, използвайки затвореността относно конкатенация.

Задача 4. Покажете, че е контекстно-свободен следният език

$$L = \{a^{n_1} \# a^{n_1+n_2} \# a^{n_2+n_3} \# \dots \# a^{n_{k-1}+n_k} \# a^{n_k} \mid k, n_1, \dots, n_k \in \mathbb{N}\}$$

3.2.2 Решения

Задача 1.(а) $\{a^m b^n \mid n < m\} \cup \{a^m b^n \mid n > m\}$;

(б) Можем да представим този език като обединението на следните два езика:

(1) $\{a, b\}^* \setminus \mathcal{L}(a^* b^*)$ (думи, в които има поне едно b преди някое a)

(2) $\{a^n b^m \mid n \neq m\}$

За (1) имаме следната граматика:

$$S \rightarrow AbAaA$$

$$A \rightarrow aA \mid bA \mid \epsilon.$$

(в) $\{a^m b^n c^p d^q \mid n = q\} \cup \{a^m b^n c^p d^q \mid m \leq p\} \cup \{a^m b^n c^p d^q \mid m + n = p + q\}$. Граматиките са съответно следните:

$$(1) S \rightarrow aS \mid A$$

$$A \rightarrow bAd \mid B$$

$$B \rightarrow cB \mid \epsilon;$$

$$(2) S \rightarrow Sd \mid A$$

$$A \rightarrow aAc \mid Ac \mid B$$

$$B \rightarrow bB \mid \epsilon;$$

(3) правена на предишното упражнение.

$$(г) \{ww^{rev} \mid w \in \Sigma^*\} \cup \{waw^{rev} \mid w \in \Sigma^*\} \cup \{wbw^{rev} \mid w \in \Sigma^*\}$$

Задача 2. $L = \{a^n b^n \mid n \in \mathbb{N}\} \circ \{b^n a^n \mid n \in \mathbb{N}\}$.

Задача 3. $L = \{ww^{rev} \mid w \in \Sigma^*\}^2$.

Задача 4. $L = \{a^n \# a^n \mid n \in \mathbb{N}\}^*$.

3.3 Коректност на контекстно-свободни граматики

В тази секция ще покажем как се доказва формално коректността на една контекстно-свободна граматика.

Пример 1. Ще докажем коректността на следната граматика G , генерираща езика $L = \{a^n b^n \mid n \in \mathbb{N}\}$.

$$G : \boxed{S \rightarrow aSb \mid \epsilon}$$

Твърдението, което се опитваме да докажем е следното:

$$(\forall w \in \Sigma^*)[w \in L(G) \iff w \in L].$$

За целите на доказателството му, първо ще докажем следната помощна Лема за релацията \Rightarrow^* в контекста на дадената граматика.

Лема 1. За всяка дума $w \in (\Sigma \cup V)^*$, ако $S \Rightarrow^* w$, то w е в един от следните два вида:

- (1) $w = a^n S b^n$, за някое $n \in \mathbb{N}$.
- (2) $w = a^n b^n$, за някое $n \in \mathbb{N}$.

Доказателство: Еквивалентно, искаме да докажем, че

$$(\forall n \in \mathbb{N})[(\forall w \in (\Sigma \cup V)^*)[S \Rightarrow^n w \implies w \text{ е от вид (1) или (2)}]].$$

Това естествено ще сторим с индукция относно n .

База: Ако $n = 0$ и w е такава поредица от терминали и нетерминали, че $S \Rightarrow^0 w$, то съществува извод с дължина 0 на w от S . С други думи $S = w$. Тогава $w = a^0 S b^0$, тоест w е от вид (1).

Стъпка: Ако $n > 0$ и w е такава поредица от терминали и нетерминали, че $S \Rightarrow^n w$, то съществува извод с дължина n на w от S . Да фиксираме един такъв извод

$$S = w_0 \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_{n-1} \Rightarrow w_n = w.$$

От този извод можем да извлечем, че $S \Rightarrow^{n-1} w_{n-1}$. Съгласно И.П. това означава, че w_{n-1} е от вид (1) или (2). Но $w_{n-1} \Rightarrow w_n$ и значи няма как w_{n-1} да е от вид (1) (тя трябва да съдържа нетерминали, съгласно дефиницията на \Rightarrow). Тогава $w_{n-1} = a^k S b^k$ за някое $k \in \mathbb{N}$. Сега, щом $w_{n-1} \Rightarrow w_n$ и в w_{n-1} има един единствен нетерминал, S , то имаме, че $w_n = a^k a S b b^k$ или $w_n = a^k \epsilon b^k$. Тоест $w_n = a^{k+1} S b^{k+1}$ или $w_n = a^k b^k$. Следователно w_n е от вид (1) или (2). Остана само да си припомним, че $w_n = w$.

Сега преминаваме към доказателството на същинското твърдение. Нека $w \in \Sigma^*$.

(\Rightarrow) Нека $w \in L(G)$. Тогава $S \Rightarrow^* w$ и $w \in \Sigma^*$. Съгласно **Лема 1** това означава, че $w = a^n b^n$ за някое $n \in \mathbb{N}$. Значи $w \in L$.

(\Leftarrow) Обратно с индукция относно n ще покажем, че

$$(\forall n \in \mathbb{N})[S \Rightarrow^{n+1} a^n b^n],$$

което е еквивалентно на обратната посока на твърдението.

База: $n = 0$. Имаме извода $S \Rightarrow \epsilon = a^0 b^0$. Значи $S \Rightarrow^1 a^0 b^0$.

Стъпка: Да допуснем, че за някое $n \in \mathbb{N}$ $S \Rightarrow^{n+1} a^n b^n$. Приемаме за очевидно, че това означава, че $aSb \Rightarrow^{n+1} aa^n b^n b = a^{n+1} b^{n+1}$. От друга страна $S \Rightarrow aSb$ и значи общо имаме "извода"

$$S \Rightarrow aSb \Rightarrow^{n+1} a^{n+1} b^{n+1}.$$

Тоест $S \Rightarrow^{n+2} a^{n+1} b^{n+1}$, което искахме да покажем.

Продължаваме с доказателство на коректността на к-св. граматика с две променливи.

Пример 1. Ще докажем коректността на следната граматика G , генерираща езика $L = \{a^m b^n \mid m > n\}$:

$$G: \begin{array}{l} S \rightarrow aA \\ A \rightarrow aA \mid aAb \mid \epsilon \end{array}$$

Отново, твърдението което се опитваме да докажем е следното:

$$(\forall w \in \Sigma^*) [w \in L(G) \iff w \in L].$$

Следната Лема късаеща релацията \Rightarrow^* ще ни бъде нужна.

Лема 1. За всяка дума $w \in (\Sigma \cup V)^*$, ако $S \Rightarrow^* w$, то w е в един от следните четири вида:

- (1) S ;
- (2) $a^m Ab^n$, за някои $m, n \in \mathbb{N}$ и $m > n$;
- (3) $a^m b^n$, за някои $m, n \in \mathbb{N}$ и $m > n$.

Доказателство: Еквивалентно, искаме да покажем, че

$$(\forall n \in \mathbb{N}) [(\forall w \in (\Sigma \cup V)^*) [S \Rightarrow^n w \implies w \text{ е в някои от видовете (1)-(3)}]].$$

За целта ще проведем индукция относно n .

База: Ако $n = 0$ и w е такава редица от терминали и нетерминали, че $S \Rightarrow^0 w$, то $w = S$. Следователно, w е от вид (1).

Стъпка: Ако $n > 0$ и w е такава редица от терминали и нетерминали, че $S \Rightarrow^n w$, то съществува извод с дължина n на w от S . Да фиксираме един такъв извод

$$S = w_0 \Rightarrow w_1 \Rightarrow \dots \Rightarrow w_{n-1} \Rightarrow w_n = w.$$

От този извод можем в частност да заключим, че $S \Rightarrow^{n-1} w_{n-1}$. Съгласно И.П. това означава, че w_{n-1} е в някои от видовете (1)-(3). Веднага отхвърляме възможността

w_{n-1} да е от вид (3), тъй като тя трябва да съдържа нетерминали, съгласно дефиницията на \Rightarrow . Значи имаме следните два случая:

1 сл. $w_{n-1} = S$. Тогава единствената възможност е $w_n = aA$, съгласно правилата на G . Следователно w_n е от вид (2), за $m = 1$.

2 сл. $w_{n-1} = a^m Ab^n$, за някои $m, n \in \mathbb{N}$, такива че $m > n$. Тук имаме три възможности за това, кое правило ще приложим на следваща стъпка (върху единствения нетерминал в редицата, A).

2.1 сл. $w_n = a^m aAb^n$, тоест приложили сме правилото $\boxed{A \rightarrow aA}$. Тогава $w_n = a^{m+1} Ab^n$ е очевидно от вид (2), тъй като $m > n \implies m+1 > n$.

2.2 сл. $w_n = a^m aAbb^n$, тоест приложили сме правилото $\boxed{A \rightarrow aAb}$. Тогава $w_n = a^{m+1} Ab^{n+1}$ е очевидно от вид (2), тъй като $m > n \implies m+1 > n+1$.

2.3 сл. $w_n = a^m \epsilon b^n$, тоест приложили сме правилото $\boxed{A \rightarrow \epsilon}$. Тогава $w_n = a^m b^n$ е очевидно от вид (3).

С това случаите се изчерпаха и индукцията приключи. Лемата е доказана.

Преминаваме към доказателството на същинското твърдение. Нека $w \in \Sigma^*$.

(\Rightarrow) Нека $w \in L(G)$. Тогава $S \Rightarrow^* w$ и $w \in \Sigma^*$. Съгласно **Лема 1**, $w = a^m b^n$, за някои $m, n \in \mathbb{N}$, такива че $m > n$. Следователно $w \in L$.

(\Leftarrow) Обратно, с индукция относно m ще докажем, че

$$(\forall m \in \mathbb{N})[(\forall n \in \mathbb{N})[m > n \implies S \Rightarrow^{m+1} a^m b^n]]$$

База: $m = 0$. Тогава със сигурност $m \not> n$. Значи импликацията е тривиално изпълнена.

Стъпка: $m > 0$. Нека $m = m_1 > 0$. Искаме да покажем следното:

$$(\forall n \in \mathbb{N})[m_1 > n \implies S \Rightarrow^{m_1+1} a^{m_1} b^n]$$

За целта ще проведем индукция по n .

База: $n = 0$. Съгласно И.П. на външната индукция, $S \Rightarrow^{m_1-1+1} a^{m_1-1} b^0$. Тоест $S \Rightarrow^{m_1} a^{m_1-1}$. Следователно $aA \Rightarrow^{m_1-1} a^{m_1-1}$. От тук имаме, че $aaA \Rightarrow^{m_1-1} aa^{m_1-1}$. Тоест $aaA \Rightarrow^{m_1-1} a^{m_1}$. От друга страна $S \Rightarrow aA$ и $aA \Rightarrow aaA$. Общо имаме $S \Rightarrow^{m_1-1+2} a^m$. Тоест $S \Rightarrow^{m_1+1} a^{m_1}$.

Стъпка: $n > 0$. Нека $m_1 > n$. Тогава $m_1 - 1 > n - 1$. По И.П. на външната индукция имаме, че $S \Rightarrow^{m_1-1+1} a^{m_1-1} b^{n-1}$. Тоест $S \Rightarrow^{m-1} a^{m_1-1} b^{n-1}$. Тогава $aA \Rightarrow^{m_1-1} a^{m_1-1} b^{n-1}$. Следователно $aaAb \Rightarrow^{m_1-1} aa^{m_1-1} b^{n-1} b$. Тоест $aaAb \Rightarrow^{m_1-1} a^{m_1} b^n$. От друга страна $S \Rightarrow aA$ и $aA \Rightarrow aaAb$. Общо имаме, че $S \Rightarrow^{m_1-1+2} a^{m_1} b^n$. Тоест $S \Rightarrow^{m_1+1} a^{m_1} b^n$, което искахме да покажем.

3.3.1 Задачи

Задача 1. Докажете формално коректностите на граматиките, предложени на следното упражнение за следните езици.

(а) $\{ww^{rev} \mid w \in \{a, b\}^*\}$

(б) $\{a^m b^n \mid m \geq n\}$

(в) $\{w \in \{a, b\}^* \mid \text{броят на } a\text{-тата в } w \text{ е равен на два пъти броя на } a\text{-тата}\}$

Задача 2. Докажете коректността на следната регулярна граматика, генерираща езика $\mathcal{L}(a^*b)$:

$$G: \begin{array}{l} S \rightarrow aS \mid A \\ A \rightarrow b \end{array}$$

Задача 3. Докажете коректността на следната регулярна граматика, генерираща езика $L = \{w \in \{a, b\}^* \mid w \text{ има четен брой } a\text{-та}\}$:

$$G: \begin{array}{l} S \rightarrow bS \mid aA \mid \epsilon \\ A \rightarrow bA \mid aS \end{array}$$

Задача 4. Докажете коректността на следната граматика, генерираща езика $\mathcal{L}(aa^*bb^*)$:

$$G: \begin{array}{l} S \rightarrow AB \\ A \rightarrow aA \mid a \\ B \rightarrow bB \mid b. \end{array}$$

Задача 5. Докажете коректността на следната граматика, генерираща езика $L = \{a^n b^m a^k \mid n + k = m\}$:

$$G: \begin{array}{l} S \rightarrow AB \\ A \rightarrow aAb \mid \epsilon \\ B \rightarrow bBa \mid \epsilon \end{array}$$

3.3.2 Решения

Задача 2. Възможните видове на редиците, генерирани от G са:

- (1) $a^n S$;
- (2) $a^n A$;
- (3) $a^n b$.

Задача 3. Възможните видове на редиците, генерирани от G са:

- (1) $(b^n a b^k a)^m b^l a b^s A$;
- (2) $(b^n a b^k a)^m b^l S$;
- (3) $(b^n a b^k a)^m b^l$.

Задача 4. Езикът по дефиниция е конкатенация на $\mathcal{L}(aa^*)$ и $\mathcal{L}(bb^*)$. Ако използваме конструкцията за конкатенация, е достатъчно да покажем, че граматиката G_1 : $\boxed{S_1 \rightarrow aS_1 \mid a}$ генерира $\mathcal{L}(aa^*)$ и аналогично, че граматиката G_2 : $\boxed{S_2 \rightarrow bS_2 \mid b}$ генерира $\mathcal{L}(bb^*)$.

Задача 5. Езикът може да се представи като конкатенация на $\{a^n b^n \mid n \in \mathbb{N}\}$ и $\{b^n a^n \mid n \in \mathbb{N}\}$. Ако използваме конструкцията за конкатенация, е достатъчно да покажем, че граматиката G_1 : $\boxed{S_1 \rightarrow aS_1 b \mid \epsilon}$ генерира $\{a^n b^n \mid n \in \mathbb{N}\}$ и аналогично, че граматиката G_2 : $\boxed{S_2 \rightarrow bS_2 a \mid \epsilon}$ генерира $\{b^n a^n \mid n \in \mathbb{N}\}$.

3.4 Синтактични дървета на извод

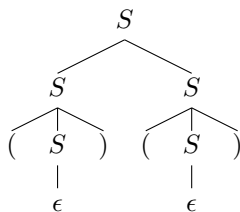
Нека G е контекстно-свободна граматика. Както вече видяхме, една дума $w \in L(G)$ може да има няколко извода по G . Да разгледаме по-прост пример отново използващ граматиката, генерираща думите от балансирани леви и десни скоби — думата $()()$ има два различни извода, именно,

$$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ()S \Rightarrow ()(S) \Rightarrow ()()$$

и

$$S \Rightarrow SS \Rightarrow S(S) \Rightarrow (S)(S) \Rightarrow (S)S \Rightarrow ()()$$

В действителност, тези два извода са в някакъв смисъл "еднакви". Използваните правила са едни и същи и се прилагат на едни и същи позиции в междинните низове. Единствената разлика е в *реда*, в който тези правила се прилагат. Интуитивно, и двата извода могат да се получат чрез обхождане на следното дърво.



Неформално, една такава картинка наричаме **синтактично дърво на извод**. Точките наричаме **върхове**; всеки връх има етикет, който е елемент на $\Sigma \cup V$. Най-горния връх наричаме **корен**, а най-долните върхове наричаме **листа**. Всички листа са етикетирани с терминали или с ϵ . Конкатенирайки етикетите на листата в ред от ляво надясно, получаваме изведената дума, която още ще наричаме **продукт** на дървото.

По-формално, за дадена контекстно-свободна граматика $G = (V, \Sigma, R, S)$, дефинираме синтактично дърво на извод заедно с неговите корен, листа и продукт индуктивно, както следва.

1. $\circ \sigma$

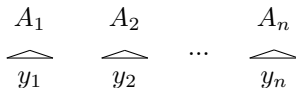
Това е синтактично дърво за всяко $\sigma \in \Sigma$. Единственият връх на това дърво е едновременно негов корен и единствено листо. Продукта на дървото е σ .

2. Ако $A \rightarrow \epsilon$ е правило в R , то

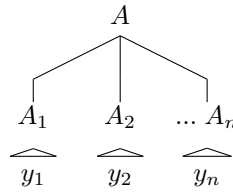


е синтактично дърво на извод; корена му е върхът с етикет A , единственото му листо е върхът с етикет ϵ , и продукта му е ϵ .

3. Ако



са синтактични дървета, за $n \geq 1$, с корени етикетирани A_1, \dots, A_n съответно, и продукти y_1, \dots, y_n , и $A \rightarrow A_1 \dots A_n$ е правило в R , то



е синтактично дърво на извод. Корена му е новият връх етикетиран A , листата му са листата на съставлящите го синтактични дървета, и продукта му е $y_1 \dots y_n$.

3.5 Езици, които не са контекстно-свободни

Множеството на контекстно-свободните езици над дадено множество от терминали и нетерминали $\Sigma \cup V$ също е изброимо безкрайно. За жалост нямаме толкова подходящо описание на контекстно-свободните езици, каквито бяха регулярните изрази за регулярните езици, с помощта на което да съобразим този факт. Достатъчно е обаче да забележим, че множеството на всички възможни правила над $\Sigma \cup V$ е $V \times (\Sigma \cup V)^*$, което е изброимо безкрайно, по следните съображения:

- V е крайно;
- Σ е крайно;
- $\Sigma \cup V$ е крайно м-во, като обединение на крайни м-ва;
- $(\Sigma \cup V)^*$ е изброимо безкрайно;
- декартовото произведение на крайно множество с изброимо безкрайно такова е изброимо безкрайно.

Правилата на една граматика са *крайно* подмножество на изброимо безкрайно множество $V \times (\Sigma \cup V)^*$. Тоест, множеството от всички възможни множества от правила е множеството от всички *крайни* подмножества на изброимо безкрайно множество. Като такова, то е изброимо безкрайно (всяко крайно подмножество може да се кодира в разлагане на прости множители на някое естествено число; това кодиране е съобразено с даденото ни изброяване). Възможните множества от правила образуват изброимо безкрайно множество. Остана само да съобразим, че възможните избори на начална променлива образуват крайно множество и можем да заключим, че има изброимо много контекстно-свободни граматики при фиксирано множество от терминали и нетерминали, а от тук и контекстно-свободните езици са изброимо много.

Щом множеството на контекстно-свободните езици е изброимо безкрайно, а множеството на всички езици е неизброимо безкрайно, то съществуват езици, които не са контекстно-свободни. За да открием критерий за това, кога един език не е контекстно-свободен отново е смислено да търсим свойство, което всички безкрайни контекстно-свободни езици притежават.

Нека $G = (V, \Sigma, R, S)$ е контекстно-свободна граматика. С $\phi(G)$ означаваме най-

големият брой символи в дясната страна на кое да е правило в R . **Път** в синтактично дърво на извод е редица от върхове свързани със линия в дървото на извод; първия връх е коренът, а посления е листо. **Дължината** на този път е броят на свързващите линии в него. **Височината** на едно синтактично дърво е дължината на най-дългия път в него.

Лема 1. Продуктът на дадено синтактично дърво на G с височина h има дължина не повече от $\phi(G)^h$.

Доказателството на горната лема е проста индукция по $h \geq 1$. Лемата ни казва, че синтактичното дърво на коя да е дума $w \in L(G)$ с $|w| > \phi(G)^h$ трябва да има път по-дълъг от h . Това е ключово за доказателството на следната теорема, даваща ни търсеното свойство, притежавано от всички контекстно-свободни езици.

Теорема 1 (лема за разрастването за контекстно-свободни езици). Нека L е контекстно-свободен език. Тогава съществува естествено число $p \geq 1$, такова че всяка дума $w \in L$ с $|w| \geq p$ може да се презапише във вида $w = vwx yz$, като

- (1) $|wy| \geq 1$,
- (2) $|wxy| \leq p$ и
- (3) $(\forall i \in \mathbb{N})[vw^i xy^i z \in L]$.

Нека $G = (V, \Sigma, R, S)$ е такава контекстно-свободна граматика, че $L(G) = L$. Ясно е, че един възможен избор за p е именно $\phi(G)^{|V|} + 1$ — ако w е дума от $L(G)$, такава че $|w| \geq \phi(G)^{|V|} + 1$, то всяко синтактично дърво за w ще има продукт с дължина поне $\phi(G)^{|V|} + 1$, тоест по-голяма от $\phi(G)^{|V|}$. Съгласно **Лема 1**, всяко такова дърво трябва да има път по-дълъг от $|V|$, тоест с дължина поне $|V| + 1$. Този път ще съдържа $|V| + 2$ върха, от които точно един — листото — е етикетиран с терминал, а останалите $|V| + 1$ са етикетиран с нетерминали. Тогава е ясно, че поне два върха по пътя ще са етикетиран с един и същ нетерминал. Ако v_1 и v_2 са два такива върха, то заменяйки "поддървото, вкоренено във v_1 " с "поддървото, вкоренено във v_2 " получаваме синтактично дърво на извод за $vw^0 xy^0 z$. Заменяйки i на брой пъти последователно "поддървото, вкоренено във v_2 " с "поддървото, вкоренено във v_1 " получаваме синтактично дърво на извод за $vw^{i+1} xy^{i+1} z$.

Пример 1. $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$ не е контекстно-свободен. За да докажем това, да допуснем, че L контекстно-свободен и нека p е въпросното число от **лемата за разрастването**. Съгласно лемата, думата $w = a^p b^p c^p$ може да се презапише във вида $w = vwx yz$ за $wy \neq \epsilon$ и $vw^i xy^i z \in L$ за всяко $i \in \mathbb{N}$. Има два случая, всеки от които води до противоречие.

1сл. wy съдържа появи на всяка от трите букви a, b и c . Тогава поне една измежду w и y трябва да съдържа появи на поне две от тези букви. Веднага се вижда, че редът

на буквите в vw^2xy^2z е развален — има b преди a , или c преди a или b .

2сл. wy съдържа появи само на някои, но не всяка от буквите a, b и c . Тогава веднага можем да съобразим, че vw^2xy^2z ще има неравен брой a -та, b -та и c -та.

Пример 2. $L = \{a^n \mid n \geq 1 \text{ е просто число}\}$ не е контекстно-свободен. За да покажем това, да допуснем, че L е контекстно-свободен и нека p е въпросното число от **лемата за разрастването**. Нека p' е първото просто число по-голямо от p . Тогава думата $a^{p'}$ може да се презапише във вида $w = vwx yz$, където $wy \neq \epsilon$. Тогава $wy = a^q$ и $v x z = a^r$, където q и r са естествени числа и $q > 0$. Съгласно лемата, $vw^i xy^i z \in L$ за всяко $i \in \mathbb{N}$. Тоест, $r + iq$ е просто число за всяко $i \in \mathbb{N}$. За $i = q + r + 1$ обаче имаме, че

$$\begin{aligned} r + iq &= \\ r + (q + r + 1)q &= \\ r + q^2 + rq + q &= \\ r(q + 1) + q(q + 1) &= \\ (q + 1)(r + q), \end{aligned}$$

което е произведение на две числа, всяко от които е по-голямо от нула и значи няма как да е просто число. Противоречие, породено от допускането ни, че L е контекстно-свободен език.

3.5.1 Задачи

Задача 1. Използвайте лемата за разрастването за да докажете, че следните езици не са контекстно-свободни.

- (а) $\{a^{n^2} \mid n \in \mathbb{N}\}$
- (б) $\{www \mid w \in \{a, b\}^*\}$
- (в) $\{w \in \{a, b, c\}^* \mid w \text{ съдържа равен брой } a\text{-та, } b\text{-та и } c\text{-та}\}$

Задача 2. Използвайте лемата за разрастването за да докажете, че езикът $L = \{babaabaaab \dots ba^{n-1}ba^n b \mid n \geq 1\}$ не е контекстно-свободен.

Задача 3. Кои от следните езици са контекстно-свободни? Обосновете се.

- (а) $\{a^m b^n c^p \mid m = n \text{ или } n = p \text{ или } m = p\}$
- (б) $\{a^m b^n c^p \mid m \neq n \text{ или } n \neq p \text{ или } m \neq p\}$
- (в) $\{a^m b^n c^p \mid m = n \text{ \& } n = p \text{ \& } m = p\}$

3.5.2 Решения

Задача 1. (а) Да допуснем, че L е контекстно-свободен. Разглеждаме думата $a^{p^2} \in L$. Съгласно лемата за разрастването $a^{p^2} = vwxyz$ за някои $v, w, x, y, z \in \{a\}^*$. Нека $w = a^k$ и $y = a^l$ съгласно свойства (1) и (2) от лемата, $1 \leq k + l \leq p$. От тук имаме, че $p^2 + 1 \leq p^2 + k + l \leq p^2 + p$. Тоест, $p^2 < p^2 + k + l \leq p^2 + p < p^2 + 2p + 1$. Значи $p^2 < p^2 + k + l < (p + 1)^2$. Тоест $p^2 < |vw^2xy^2z| < (p + 1)^2$, откъдето следва, че $|vw^2xy^2z|$ няма как да бъде точен квадрат. Следователно $vw^2xy^2z \notin L$. Намерихме число, за което прилагайки свойство (3) стигаме до дума извън L . Противоречие с лемата. Значи L не е контекстно-свободен.

(б) Разглеждаме думата $(a^p b)^3 \in L$. За wxy имаме следните два случая:

(1) $wxy = a^k$, аз някое $1 \leq k \leq p$. Тогава лесно може да се съобрази, че покачвайки нагоре ще излезем от езика.

(2) $wxy = a^k b a^l$ за някои $1 \leq k + l + 1 \leq p$. Имаме следните подслучаи:

(2.1) Буквата b е в w или в y . Тогава думата vw^0xy^0z ще има две b -та и няма как да е в L .

(2.2) Буквата b е в x . Тогава както и да изберем да се възползваме от свойство (3) на лемата, ще стигнем до дума извън L , защото броят на a -тата няма да е един и същ преди всяко от трите b -та, а е ясно, че всяка разбивка на покачената дума на три еднакви части www ще има свойството, че w завършва на b .

(в) Разглеждаме думата $a^p b^p c^p \in L$. За wxy имаме следните два случая:

(1) $wxy = \sigma^k$ за някои $\sigma \in \{a, b, c\}$ и $1 \leq k \leq p$. Тук е ясно, че както и да покачим, излизаме от езика.

(2) wxy попада в интервал, застъпващ *точно* две различни букви. Независимо как са разпределени тези две букви между w и y , тъй като $|wy| \geq 1$, то покачвайки в коя да е посока ще получим дума, в която бройките на трите различни букви не са равни.

Задача 2. Да разгледаме думата $babaabaaab...ba^{p-1}ba^p b \in L$. Нека $vwxyz$ е нейно разбиване със свойствата от лемата за разрастването. Случаите, в които wxy попада изцяло или частично преди $a^p b$ са ясни — покачвайки в която и да е посока, линейно нарастващата структура на думата се нарушава и попадаме извън езика. Да разгледаме случая, в който wxy попада изцяло в $a^p b$. Имаме следните подслучаи:

(1) $wxy = a^k$ за някое $1 \leq k \leq p$. Тук нещата пак са ясни — ако се опитаме да приложим св-во (3) веднага разваляме линейно нарастващата структура на a -тата.

(2) $wxy = a^k b$ за някое $1 \leq k \leq p$. Ако w и y са едновременно непразни, то в частност y съдържа b -то, а $w = a^l$, за някое $l \leq k$ и покачвайки нагоре добавяме a -та преди последното b в оригиналната дума, което разваля структурата. Ако само y е празна, то x съдържа b -то и отново имаме същия проблем. Ако и x и y са празни, то $w = a^k b$ и покачвайки нагоре получаваме думата $babaabaaab...ba^{p-1}ba^p ba^k b$, която не е в L , тъй като $k \leq p$, тоест $k \neq p + 1$. Ако w е празна, а y е непразна, то $y = a^l b$, за някое $l \leq k$ и покачвайки нагоре получаваме същия проблем. С това случаите се изчерпах.

Задача 3. (а) Този език е $\{a^m b^m c^n \mid m, n \in \mathbb{N}\} \cup \{a^m b^n c^n \mid m, n \in \mathbb{N}\} \cup \{a^m b^n c^m \mid m, n \in \mathbb{N}\}$ и следователно е контекстно-свободен. Граматики ще дадем за първия и третия от тези операнди.

$$(1) \begin{array}{l} S \rightarrow Sc \mid A \\ A \rightarrow aAb \mid \epsilon. \end{array}$$

$$(2) \begin{array}{l} S \rightarrow aSc \mid B \\ B \rightarrow bB \mid \epsilon. \end{array}$$

(б) Този език е $\{a^m b^n c^p \mid m \neq n\} \cup \{a^m b^n c^p \mid n \neq p\} \cup \{a^m b^n c^p \mid m \neq p\}$. На свой ред той е равен на $[\{a^m b^n \mid m \neq n\} \circ \mathcal{L}(c^*)] \cup [\mathcal{L}(a^*) \circ \{b^n c^p \mid n \neq p\}] \cup \{a^m b^n c^p \mid m \neq p\}$. Всеки от тези езици знаем вече, че е контекстно-свободен, освен $\{a^m b^n c^p \mid m \neq p\}$.

Контекстно-свободна граматика за този език е например:

$$\begin{array}{l} S \rightarrow aA \mid Cc \\ A \rightarrow aA \mid aAc \mid B \\ C \rightarrow Cc \mid aCc \mid B \\ B \rightarrow bB \mid \epsilon \end{array}$$

Следователно и целият език е контекстно-свободен.

(в) Това е езикът $\{a^n b^n c^n \mid n \in \mathbb{N}\}$, за който вече показахме, че не е контекстно свободен.