Езици, автомати, изчислимост

Стефан Вътев[1]

27 октомври 2015 г.

[1]ел. поща: stefanv@fmi.uni-sofia.bg

Съдържание

Глава 1

Увод

1.1 Съждително смятане

На англ. Propositional calculus

Съждителното смятане наподобява аритметичното смятане, като вместо аритметичните операции $+, -, \cdot, /$, имаме съждителни операции като $\neg, \wedge, \vee$. Например, $(p \vee q) \wedge \neg r$ е съждителна формула. Освен това, докато аритметичните променливи приемат стойности числа, то съждителните променливи приемат само стойности **истина (1)** или **неистина (0)**.

Съждителна формула наричаме съвкупността от съждителни променливи $p, q, r, \ldots$, свързани със знаците за логически операции $\neg, \vee, \wedge, \rightarrow, \leftrightarrow$ и скоби, определящи реда на операциите.

Съждителни операции

- Отрицание $\neg$
- Дизюнкция $\vee$
- Конюнкция $\wedge$
- Импликация $\rightarrow$
- Еквивалентност $\leftrightarrow$

Ще използваме таблица за истинност за да определим стойностите на основните съждителни операции при всички възможни набори на стойностите на променливите.

p	q	$\neg p$	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$\neg p \vee q$	$p \leftrightarrow q$	$(\neg p \wedge q) \vee (p \wedge \neg q)$
0	0	1	0	0	1	1	1	1
0	1	1	1	0	1	1	0	0
1	0	0	1	0	0	0	0	0
1	1	0	1	1	1	1	1	1

Съждително верен (валиден) е този логически израз, който има верностна стойност **1** при всички възможни набори на стойностите на съждителните променливи в израза, т.е. стълбът на израза в таблицата за истинност трябва да съдържа само стойности **1**.

Два съждителни израза φ и ψ са **еквивалентни**, което означаваме $\varphi \equiv \psi$, ако са съставени от едни и същи съждителни променливи и двата израза имат едни и същи верностни стойности при всички комбинации от верностни стойности на променливите. С други думи, колоните на двата израза в съответните им таблици за истинност трябва да съвпадат. Така например, от горната таблица се вижда, че $p \to q \equiv \neg p \vee q$ и $p \leftrightarrow q \equiv (\neg p \wedge q) \vee (p \wedge \neg q)$.

Съждителни закони

I) **Комутативен закон**

$$p \vee q \equiv q \vee p$$

$$p \wedge q \equiv q \wedge p$$

II) **Асоциативен закон**

$$(p \vee q) \vee r \equiv p \vee (q \vee r)$$

$$(p \wedge q) \wedge r \equiv p \wedge (q \wedge r)$$

III) **Дистрибутивен закон**

$$p \wedge (q \vee r) \equiv (p \wedge q) \vee (p \wedge r)$$

$$p \vee (q \wedge r) \equiv (p \vee q) \wedge (p \vee r)$$

IV) **Закони на де Морган**

$$\neg(p \wedge q) \equiv (\neg p \vee \neg q)$$

$$\neg(p \vee q) \equiv (\neg p \wedge \neg q)$$

V) **Закон за контрапозицията**

$$p \to q \equiv \neg q \to \neg p$$

VI) **Обобщен закон за контрапозицията**

$$(p \wedge q) \to r \equiv (p \wedge \neg r) \to \neg q$$

VII) **Закон за изключеното трето**

$$p \vee \neg p \equiv \mathbf{1}$$

VIII) **Закон за силогизма (транзитивност)**

$$[(p \to q) \wedge (q \to r)] \to (p \to r) \equiv \mathbf{1}$$

Лесно се проверява с таблиците за истинност, че законите са валидни.

1.2 Предикатно смятане

1.3 Доказателства на твърдения

Допускане на противното

Да приемем, че искаме да докажем, че свойството $P(x)$ е вярно за всяко естествено число. Един начин да направим това е следният:

- Допускаме, че съществува елемент n, за който $\neg P(n)$.
- Използвайки, че $\neg P(n)$ правим извод, от който следва факт, за който знаем, че винаги е лъжа. Това означава, че доказваме следното твърдение
$$\exists x \neg P(x) \to \mathbf{0}.$$
- Тогава можем да заключим, че $\forall x P(x)$, защото имаме следния извод:

$$\dfrac{\dfrac{\dfrac{\exists x \neg P(x) \to \mathbf{0}}{\mathbf{1} \to \neg \exists x \neg P(x)}}{\neg \exists x \neg P(x)}}{\forall x P(x)}$$

Ще илюстрираме този метод като решим няколко прости задачи.

Задача 1.1. За всяко $a \in \mathbb{Z}$, ако a^2 е четно, то a е четно.

Док. Ние искаме да докажем твърдението P, където:

$$P \equiv (\forall a \in \mathbb{Z})[a^2 \text{ е четно } \to \ a \text{ е четно}].$$

Да допуснем противното, т.е. изпълнено е $\neg P$. Лесно се вижда, че

$\neg(\forall x)(A(x) \to B(x))$ е еквивалентно на $(\exists x)(A(x) \wedge \neg B(x))$

$$\neg P \ \leftrightarrow \ (\exists a \in \mathbb{Z})[a^2 \text{ е четно } \wedge \ a \text{ не е четно}].$$

Да вземем едно такова нечетно a, за което a^2 е четно. Това означава, че $a = 2k + 1$, за някое $k \in \mathbb{Z}$, и

$$a^2 = (2k+1)^2 = 4k^2 + 4k + 1,$$

което очевидно е нечетно число. Но ние допуснахме, че a^2 е четно. Така достигаме до противоречие, следователно нашето допускане е грешно и

$$(\forall a \in \mathbb{Z})[a^2 \text{ е четно } \to \ a \text{ е четно}].$$

□

Задача 1.2. Докажете, $\sqrt{2}$ **не** е рационално число.

Док. Да допуснем, че $\sqrt{2}$ е рационално число. Тогава съществуват $a, b \in \mathbb{Z}$, такива че

$$\sqrt{2} = \frac{a}{b}.$$

Без ограничение, можем да приемем, че a и b са естествени числа, които нямат общи делители, т.е. не можем да съкратим дробта $\frac{a}{b}$. Получаваме, че

$$2b^2 = a^2.$$

Тогава a^2 е четно число и от Задача 1.1, a е четно число. Нека $a = 2k$. Получаваме, че

$$2b^2 = 4k^2,$$

от което следва, че

$$b^2 = 2k^2.$$

Това означава, че b също е четно число, $b = 2n$, за някое $n \in \mathbb{Z}$. Следователно, a и b са четни числа и имат общ делител 2, което е противоречие с нашето допускане, че a и b нямат общи делители. Така достигаме до противоречие. Накрая заключаваме, че $\sqrt{2}$ не е рационално число. □

Индукция върху естествените числа

Доказателството с индукция по $\mathbb{N}$ представлява следната схема:

Да напомним, че естествените числа са $\mathbb{N} = \{0, 1, 2, \dots\}$

$$\frac{P(0) \qquad (\forall x \in \mathbb{N})[P(x) \to P(x+1)]}{(\forall x \in \mathbb{N})P(x)}$$

Това означава, че ако искаме да докажем, че свойството $P(x)$ е вярно за всяко естествено число x, то трябва да докажем първо, че е изпълнено $P(0)$ и след това, за произволно естествено число x, ако $P(x)$ вярно, то също така е вярно $P(x+1)$.

Задача 1.3. Всяко естествено число $n \geq 2$ може да се запише като произведение на прости числа.

Док. Доказателството протича с индукция по $n \geq 2$.

а) За $n = 2$ е ясно.

б) Ако $n+1$ е просто число, то всичко е ясно. Ако $n+1$ е съставно, то

$$n + 1 = n_1 \cdot n_2.$$

Тогава $n_1 = p_1^{n_1} \cdots p_k^{n_k}$ и $n_2 = q_1^{m_1} \cdots q_r^{m_r}$, където $p_1, \dots, p_k$ и $q_1, \dots, q_r$ са прости числа. Тогава е ясно, че $n+1$ също е произведение на прости числа.

□

Задача 1.4. Докажете, че за всяко n,

$$\sum_{i=0}^{n} 2^i = 2^{n+1} - 1.$$

Док. Доказателството протича с индукция по n.

- За $n = 0$, $\sum_{i=0}^{0} 2^i = 1 = 2^1 - 1$.

- Нека твърдението е вярно за n. Ще докажем, че твърдението е вярно за $n+1$.

$$\begin{aligned}\sum_{i=0}^{n+1} 2^i &= \sum_{i=0}^{n} 2^i + 2^{n+1} \\ &= 2^{n+1} - 1 + 2^{n+1} && \text{(от И.П.)} \\ &= 2.2^{n+1} - 1 \\ &= 2^{(n+1)+1} - 1.\end{aligned}$$

□

1.4 Множества, релации, функции

Основни операции върху множества

Ще разгледаме няколко операции върху произволни множества A и B.

- **Сечение**
$$A \cap B = \{x \mid x \in A \wedge x \in B\}.$$
- **Обединение**
$$A \cup B = \{x \mid x \in A \vee x \in B\}.$$
- **Разлика**
$$A \setminus B = \{x \mid x \in A \wedge x \notin B\}.$$
- **Степенно множество**
$$\mathscr{P}(A) = \{x \mid x \subseteq A\}.$$
Примери:
  - $\mathscr{P}(\emptyset) = \{\emptyset\}$.
  - $\mathscr{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$.
  - $\mathscr{P}(\{\emptyset, \{\emptyset\}\}) = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$.
  - $\mathscr{P}(\{1, 2\}) = \{\emptyset, \{1\}, \{2\}, \{1, 2\}\}$.

Нека имаме редица от множества $\{A_1, A_2, \ldots, A_n\}$. Тогава имаме следните операции:

- **Обединение на редица от множества**
$$\bigcup_{i=1}^{n} A_i = \{x \mid \exists i(1 \leq i \leq n \;\&\; x \in A_i)\}.$$
- **Сечение на редица от множества**
$$\bigcap_{i=1}^{n} A_i = \{x \mid \forall i(1 \leq i \leq n \to x \in A_i)\}.$$

Задача 1.5. Проверете верни ли са свойствата:

а) $A \subseteq B \ \leftrightarrow \ A \setminus B = \emptyset \ \leftrightarrow \ A \cup B = B \ \leftrightarrow \ A \cap B = A$;

б) $A \setminus \emptyset = A$, $\emptyset \setminus A = \emptyset$, $A \setminus B = B \setminus A$.

в) $A \cap (B \cup A) = A \cap B$;

г) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$ и $A \cap (B \cup C) = (A \cup B) \cap (A \cup C)$;

д) $A \backslash B = A \ \leftrightarrow \ A \cap B = \emptyset$;

е) $A \backslash B = A \backslash (A \cap B)$ и $A \backslash B = A \backslash (A \cup B)$;

ж) $(A \cup B) \setminus C = (A \setminus C) \cup (B \setminus C)$;

з) $A \setminus (B \setminus C) = (A \setminus B) \setminus C$; Не е вярно!

и) $C \setminus (A \cup B) = (C \backslash A) \cap (C \backslash B)$ и $C \backslash (A \cap B) = (C \backslash A) \cup (C \backslash B)$ Закони на Де Морган

к) $C \backslash (\bigcup_{i=1}^{n} A_i) = \bigcap_{i=1}^{n} (C \backslash A_i)$ и $C \backslash (\bigcap_{i=1}^{n} A_i) = \bigcup_{i=1}^{n} (C \backslash A_i)$;

л) $(A \backslash B) \backslash C = (A \backslash C) \backslash (B \backslash C)$ и $A \backslash (B \backslash C) = (A \backslash B) \cup (A \cap C)$;

м) $A \subseteq B \Rightarrow \mathscr{P}(A) \subseteq \mathscr{P}(B)$;

н) $\mathscr{P}(A \cap B) = \mathscr{P}(A) \cap \mathscr{P}(B)$ и $\mathscr{P}(A \cup B) = \mathscr{P}(A) \cup \mathscr{P}(B)$; $X \subseteq A \cup B \stackrel{?}{\Rightarrow} X \subseteq A \vee X \subseteq B$

За да дадем определение на понятието релация, трябва първо да въведем понятието декартово произведение на множества, което пък от своя страна се основава на понятието наредена двойка.

Наредена двойка

За два елемента a и b въвеждаме операцията **наредена двойка** $\langle a, b \rangle$. Наредената двойка $\langle a, b \rangle$ има следното характеристичното свойство:

$$a_1 = a_2 \ \wedge \ b_1 = b_2 \quad \leftrightarrow \quad \langle a_1, b_1 \rangle = \langle a_2, b_2 \rangle.$$

Понятието наредена двойка може да се дефинира по много начини, стига да изпълнява характеристичното свойство. Ето примери как това може да стане:

1) Първото теоретико-множествено определение на понятието наредена двойка е дадено от Норберт Винер: Norbert Wiener (1914)

$$\langle a, b \rangle \stackrel{\text{деф}}{=} \{\{\{a\}, \emptyset\}, \{\{b\}\}\}.$$

2) Определението на Куратовски се приема за „стандартно” в наши дни: Kazimierz Kuratowski (1921)

$$\langle a, b \rangle \stackrel{\text{деф}}{=} \{\{a\}, \{a, b\}\}.$$

Задача 1.6. Докажете, че горните дефиниции наистина изпълняват характеристичното свойство за наредени двойки.

Определение 1.1. Сега можем, за всяко естествено число $n \geq 1$, да въведем понятието наредена n-орка $\langle a_1, \ldots, a_n \rangle$:

Пример за индуктивна (рекурсивна) дефиниция

$$\langle a_1 \rangle \stackrel{\text{деф}}{=} a_1,$$
$$\langle a_1, a_2, \ldots, a_n \rangle \stackrel{\text{деф}}{=} \langle a_1, \langle a_2, \ldots, a_n \rangle \rangle$$

Оттук нататък ще считаме, че имаме операцията наредена n-орка, без да се интересуваме от нейната формална дефиниция.

Декартово произведение

На англ. cartesian product

За две множества A и B, определяме тяхното декартово произведение като

Считаме, че $(A \times B) \times C = A \times (B \times C) = A \times B \times C$

$$A \times B = \{\langle a, b \rangle \mid a \in A \ \& \ b \in B\}.$$

За краен брой множества $A_1, A_2, \ldots, A_n$, определяме

$$A_1 \times A_2 \times \cdots \times A_n = \{\langle a_1, a_2, \ldots, a_n \rangle \mid a_1 \in A_1 \ \& \ a_2 \in A_2 \ \& \ \ldots \ \& \ a_n \in A_n\}.$$

Задача 1.7. Проверете, че:

а) $A \times (B \cup C) = (A \times B) \cup (A \times C)$.

б) $(A \cup B) \times C = (A \times C) \cup (B \times C)$.

в) $A \times (B \cap C) = (A \times B) \cap (A \times C)$.

г) $(A \cap B) \times C = (A \times C) \cap (B \times C)$.

д) $A \times (B \setminus C) = (A \times B) \setminus (A \times C)$.

е) $(A \setminus B) \times C = (A \times C) \setminus (B \times C)$.

Основни видове бинарни релации

Подмножествата R от вида $R \subseteq A \times A \times \cdots \times A$ се наричат релации. Релациите от вида $R \subseteq A \times A$ са важен клас, който ще срещаме често. Да разгледаме няколко основни видове релации от този клас:

I) **рефлексивна**, ако

$$(\forall x \in A)[\langle x, x \rangle \in R].$$

Например, релацията $\leq \subseteq \mathbb{N} \times \mathbb{N}$ е рефлексивна, защото

$$(\forall x \in \mathbb{N})[x \leq x].$$

II) **транзитивна**, ако

$$(\forall x, y, z \in A)[\langle x, y \rangle \in R \ \& \ \langle y, z \rangle \in R \to \langle x, z \rangle \in R].$$

Например, релацията $\leq \subseteq \mathbb{N} \times \mathbb{N}$ е транзитивна, защото

$$(\forall x, y, z \in A)[x \leq y \ \& \ y \leq z \ \to \ x \leq z].$$

III) **симетрична**, ако

$$(\forall x, y \in A)[\langle x, y\rangle \in R \to \langle y, x\rangle \in R].$$

Например, релацията $= \subseteq \mathbb{N} \times \mathbb{N}$ е рефлексивна, защото

$$(\forall x, y \in \mathbb{N})[x = y \ \to \ y = x].$$

IV) **антисиметрична**, ако

$$(\forall x, y \in A)[\langle x, y\rangle \in R \ \& \ \langle y, x\rangle \in R \to x = y].$$

Например, релацията $\leq \subseteq \mathbb{N} \times \mathbb{N}$ е антисиметрична, защото

$$(\forall x, y, z \in A)[x \leq y \ \& \ y \leq x \ \to \ x = y].$$

- Една бинарна релация R над множеството A се нарича **релация на еквивалентност**, ако R е рефлексивна, транзитивна и симетрична.
- За всеки елемент $a \in A$, определяме неговия **клас на еквивалентност** относно релацията на еквивалентност R по следния начин:

$$[a]_R \stackrel{\text{деф}}{=} \{b \in A \mid \langle a, b\rangle \in R\}.$$

Забележка. Лесно се съобразява, че за всеки два елемента $a, b \in A$,

$$\langle a, b\rangle \in R \ \leftrightarrow \ [a]_R = [b]_R.$$

Пример 1.1. За всяко естествено число $n \geq 2$, дефинираме релацията R_n като

$$\langle x, y\rangle \in R_n \ \leftrightarrow \ x \equiv y \pmod{n}.$$

Ясно е, че R_n са релации на еквивалентност.

Операции върху бинарни релации

I) **Композиция** на две релации $R \subseteq B \times C$ и $P \subseteq A \times B$ е релацията $R \circ P \subseteq A \times C$, определена като:

$$R \circ P \stackrel{\text{деф}}{=} \{\langle a, c\rangle \in A \times C \mid (\exists b \in B)[\langle a, b\rangle \in P \ \& \ \langle b, c\rangle \in R]\}.$$

II) **Обръщане** на релацията $R \subseteq A \times B$ е релацията $R^{-1} \subseteq B \times A$, определена като:

$$R^{-1} \stackrel{\text{деф}}{=} \{\langle x, y\rangle \in B \times A \mid \langle y, x\rangle \in R\}.$$

III) **Рефлексивно затваряне** на релацията $R \subseteq A \times A$ е релацията

$$P \stackrel{\text{деф}}{=} R \cup \{\langle a, a\rangle \mid a \in A\}.$$

Очевидно е, че P е рефлексивна релация, дори ако R не е.

IV) **Итерация** на релацията $R \subseteq A \times A$ дефинираме като за всяко естествено число n, дефинираме релацията R^n по следния начин:

$$R^0 \stackrel{\text{деф}}{=} \{\langle a, a\rangle \mid a \in A\}$$
$$R^{n+1} \stackrel{\text{деф}}{=} R^n \circ R.$$

Лесно се вижда, че $R^1 = R$

V) **Транзитивно затваряне** на $R \subseteq A \times A$ е релацията

✍ Проверете, че R^+ е транзитивна релация!

$$R^+ \stackrel{\text{деф}}{=} \bigcup_{n \geq 1} R^n.$$

За дадена релация R, с $R^\star$ ще означаваме нейното рефлексивно и транзитивно затваряне. От дефинициите е ясно, че

$$R^\star = \bigcup_{n \geq 0} R^n.$$

Видове функции

Функцията $f : A \to B$ е:

- **инекция**, ако (или f е **обратима**)

$$(\forall a_1, a_2 \in A)[a_1 \neq a_2 \to f(a_1) \neq f(a_2)],$$

или еквивалентно,

$$(\forall a_1, a_2 \in A)[f(a_1) = f(a_2) \to a_1 = a_2].$$

- **сюрекция**, ако (или f е **върху** B)

$$(\forall b \in B)(\exists a \in A)[f(a) = b].$$

- **биекция**, ако е инекция и сюрекция.

Задача 1.8. Докажете, че $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ е биекция, където

Канторово кодиране. Най-добре се вижда като се нарисува таблица

$$f(x, y) = \frac{(x + y)(x + y + 1)}{2} + x.$$

1.5 Азбуки, думи, езици

Основни понятия

- **Азбука** ще наричаме всяко крайно множество, като обикновено ще я означаваме със Σ. Елементите на азбуката Σ ще наричаме **букви** или символи.

Често ще използваме буквите a, b, c за да означаваме букви.

- **Дума** над азбуката Σ е произволна крайна редица от елементи на Σ. Например, за $\Sigma = \{a, b\}$, $aababba$ е дума над Σ с дължина 7. С $|\alpha|$ ще означаваме дължината на думата α.

Обикновено ще означаваме думите с α, β, γ, ω.

- Обърнете внимание, че имаме единствена дума с дължина 0. Тази дума ще означаваме с ε и ще я наричаме **празната дума**, т.е. $|\varepsilon| = 0$.

- С a^n ще означаваме думата съставена от n a-та. Формалната индуктивна дефиниция е следната:

$$a^0 \stackrel{\text{деф}}{=} \varepsilon,$$
$$a^{n+1} \stackrel{\text{деф}}{=} a^n a.$$

- Множеството от всички думи над азбуката Σ ще означаваме със $\Sigma^\star$. Например, за $\Sigma = \{a, b\}$,

$$\Sigma^\star = \{\varepsilon, a, b, aa, ab, ba, bb, aaa, aab, \dots\}.$$

  Обърнете внимание, че $\emptyset^\star = \{\varepsilon\}$.

Операции върху думи

- Операцията **конкатенация** взима две думи α и β и образува новата дума $\alpha \cdot \beta$ като слепва двете думи. Например $aba \cdot bb = ababb$. Обърнете внимание, че в общия случай $\alpha \cdot \beta \neq \beta \cdot \alpha$. Можем да дадем формална индуктивна дефиниция на операцията конкатенация по дължината на думата β.

  Често ще пишем $\alpha\beta$ вместо $\alpha \cdot \beta$

  - Ако $|\beta| = 0$, то $\beta = \varepsilon$. Тогава $\alpha \cdot \varepsilon \stackrel{\text{деф}}{=} \alpha$.
  - Ако $|\beta| = n + 1$, то $\beta = \gamma b$, $|\gamma| = n$. Тогава $\alpha \cdot \beta \stackrel{\text{деф}}{=} (\alpha \cdot \gamma)b$.

- Друга често срещана операция върху думи е **обръщането** на дума. Дефинираме думата α^R като обръщането на α по следния начин.

  - Ако $|\alpha| = 0$, то $\alpha = \varepsilon$ и $\alpha^R \stackrel{\text{деф}}{=} \varepsilon$.
  - Ако $|\alpha| = n + 1$, то $\alpha = a\beta$, където $|\beta| = n$. Тогава $\alpha^R \stackrel{\text{деф}}{=} (\beta^R)a$.

  Например, $reverse^R = esrever$.

- Казваме, че думата α е **префикс** на думата β, ако съществува дума γ, такава че $\beta = \alpha \cdot \gamma$. α е **суфикс** на β, ако $\beta = \gamma \cdot \alpha$, за някоя дума γ.

- Нека A и B са множества от думи. Дефинираме конкатенацията на A и B като

$$A \cdot B \stackrel{\text{деф}}{=} \{\alpha \cdot \beta \mid \alpha \in A \ \& \ \beta \in B\}.$$

  Обърнете внимание, че $\emptyset \cdot A = A \cdot \emptyset = \emptyset$
  Също така, $\{\varepsilon\} \cdot A = A \cdot \{\varepsilon\} = A$

- Сега за едно множество от думи A, дефинираме A^n индуктивно:

$$A^0 \stackrel{\text{деф}}{=} \{\varepsilon\},$$
$$A^{n+1} \stackrel{\text{деф}}{=} A^n \cdot A.$$

  Ако $A = \{ab, ba\}$, то $A^0 = \{\varepsilon\}$, $A^1 = A$, $A^2 = \{abab, abba, baba, baab\}$. Ако $A = \{a, b\}$, то $A^n = \{\alpha \in \{a, b\}^\star \mid |\alpha| = n\}$.

- За едно множеството от думи A, дефинираме:

  Операцията $\star$ е известна като звезда на Клини

$$A^\star \stackrel{\text{деф}}{=} \bigcup_{n \geq 0} A^n$$
$$= A^0 \cup A^1 \cup A^2 \cup A^3 \cup \dots$$
$$A^+ \stackrel{\text{деф}}{=} A \cdot A^\star.$$

Задача 1.9. Проверете:

а) операцията конкатенация е *асоциативна*, т.е. за всеки три думи α, β, γ,

$$(\alpha \cdot \beta) \cdot \gamma = \alpha \cdot (\beta \cdot \gamma);$$

б) за множествата от думи A, B и C,

$$(A \cdot B) \cdot C = A \cdot (B \cdot C);$$

в) $\{\varepsilon\}^\star = \varepsilon$;

г) за произволно множество от думи A, $A^\star = A^\star \cdot A^\star$ и $(A^\star)^\star = A^\star$;

д) за произволни букви a и b, $\{a, b\}^\star = \{a\}^\star \cdot (\{b\} \cdot \{a\}^\star)^\star$.

Задача 1.10. Докажете, че за всеки две думи α и β е изпълнено:

а) $(\alpha \cdot \beta)^R = \beta^R \cdot \alpha^R$;

б) α е префикс на β точно тогава, когато α^R е суфикс на β^R;

в) $(\alpha^R)^R = \alpha$;

г) $(\alpha^n)^R = (\alpha^R)^n$, за всяко $n \geq 0$.

Глава 2

Регулярни езици и автомати

2.1 Автоматни езици

Определение 2.1. Краен автомат е петорка $\mathcal{A} = \langle Q, \Sigma, s, \delta, F \rangle$, където

1) Q е крайно множество от състояния;

2) Σ е азбука;

3) $\delta : Q \times \Sigma \to Q$ е (частична) функция на преходите;

4) $s \in Q$ е начално състояние;

5) $F \subseteq Q$ е множеството от финални състояния, $F \neq \emptyset$.

Ако функцията на преходите δ е тотална функция, то казваме, че автоматът $\mathcal{A}$ е **тотален**. Това означава, че за всяка двойка $(q, a) \in \Sigma \times Q$, съществува $q' \in Q$, за което $\delta(q, a) = q'$.

Нека имаме една дума $\alpha \in \Sigma^\star$, $\alpha = a_1 a_2 \cdots a_n$. Казваме, че α се **разпознава** от автомата $\mathcal{A}$, ако съществува редица от състояния $q_0, q_1, q_2, \ldots, q_n$, такива че:

- $q_0 = s$, началното състояние на автомата;
- $\delta(q_i, a_{i+1}) = q_{i+1}$, за всяко $i = 0, \ldots, n-1$;
- $q_n \in F$.

Казваме, че $\mathcal{A}$ **разпознава** езика L, ако $\mathcal{A}$ разпознава точно думите от L, т.е. $L = \{\alpha \in \Sigma^\star \mid \mathcal{A} \text{ разпознава } \alpha\}$. Обикновено означаваме езика, който се разпознава от даден автомат $\mathcal{A}$ с $\mathcal{L}(\mathcal{A})$. В такъв случай ще казваме, че езикът L е **автоматен**.

При дадена (частична) функция на преходите δ, често е удобно да разглеждаме (частичната) функция $\delta^\star : Q \times \Sigma^\star \to Q$, като е дефинирана по следния начин:

Това е пример за индуктивна (рекурсивна) дефиниция по дължината на думата α

- $\delta^\star(q, \varepsilon) = q$, за всяко $q \in Q$;
- $\delta^\star(q, \beta a) = \delta(\delta^\star(q, \beta), a)$, за всяко $q \in Q$, всяко $a \in \Sigma$ и $\beta \in \Sigma^\star$.

Тогава една дума α се *разпознава* от автомата $\mathcal{A}$ точно тогава, когато $\delta^\star(s,\alpha) \in F$. Оттук следва, че

$$\mathcal{L}(\mathcal{A}) = \{\alpha \in \Sigma^\star \mid \delta^\star(s,\alpha) \in F\}.$$

Твърдение 2.1. $(\forall q \in Q)(\forall \alpha, \beta \in \Sigma^\star)[\delta^\star(q, \alpha\beta) = \delta^\star(\delta^\star(q,\alpha),\beta)]$.

Упътване. Индукция по дължината на β. □

✍ Напишете доказателството!

Моментното описание на изчисление с краен автомат представлява двойка от вида $(q,\alpha) \in Q \times \Sigma^\star$, т.е. автоматът се намира в състояние q, а думата, която остава да се прочете е α. Удобно е да въведем бинарната релация $\vdash_{\mathcal{A}}$ над $Q \times \Sigma^\star$, която ще ни казва как моментното описание на автомата $\mathcal{A}$ се променя след изпълнение на една стъпка:

$$(q, x\alpha) \vdash_{\mathcal{A}} (p,\alpha), \text{ ако } \delta(q,x) = p.$$

Рефлексивното и транзитивно затваряне на $\vdash_{\mathcal{A}}$ ще означаваме с $\vdash^\star_{\mathcal{A}}$. Получаваме, че

$$\mathcal{L}(\mathcal{A}) = \{\alpha \in \Sigma^\star \mid (s,\alpha) \vdash^\star_{\mathcal{A}} (p,\varepsilon) \ \&\ p \in F\}.$$

Нашата дефиниция на автомат позволява δ да бъде частична функция, т.е. може да има $q \in Q$ и $a \in \Sigma$, за които $\delta(q,a)$ не е дефинирана. Следващото твърдение ни казва, че ние съвсем спокойно можем да разглеждаме автомати само с тотални функции на преходите δ.

Твърдение 2.2. За всеки краен автомат $\mathcal{A}$, съществува *тотален* краен автомат $\mathcal{A}'$, за който $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$.

Док. Нека $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$. Дефинираме тоталния автомат

$$\mathcal{A}' = \langle Q \cup \{q_e\}, \Sigma, \delta', s, F\rangle,$$

като за всеки преход (q,a), за който δ не е дефинирана, дефинираме δ' да отива в новото състояние q_e. Ето и цялата дефиниция на новата функция на преходите δ':

- $\delta'(q_e, a) = q_e$, за всяко $a \in \Sigma$;
- За всяко $q \in Q$, $a \in \Sigma$, ако $\delta(q,a) = p$, то $\delta'(q,a) = p$;
- За всяко $q \in Q$, $a \in \Sigma$, ако $\delta(q,a)$ не е дефинирано, то $\delta'(q,a) = q_e$.

q_e - error състояние

$\mathcal{A}'$ симулира $\mathcal{A}$

Сега лесно може да се докаже, че $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}')$. □

✍ Довършете доказателството!

Твърдение 2.3. Класът на автоматните езици е затворен относно операцията **обединение**. Това означава, че ако L_1 и L_2 са два произволни автоматни езика над азбуката Σ, то $L_1 \cup L_2$ също е автоматен език.

Док. Нека $L_1 = \mathcal{L}(\mathcal{A}_1)$ и $L_2 = \mathcal{L}(\mathcal{A}_2)$, където $\mathcal{A}_1 = \langle Q_1, \Sigma, s_1, \delta_1, F_1\rangle$ и $\mathcal{A}_2 = \langle Q_2, \Sigma, s_2, \delta_2, F_2\rangle$ са **тотални**. Определяме автомата $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$, който разпознава $L_1 \cup L_2$ по следния начин:

Защо изискваме $\mathcal{A}_1$ и $\mathcal{A}_2$ да са тотални?

- $Q = Q_1 \times Q_2$;
- Определяме за всяко $\langle r_1, r_2\rangle \in Q$ и всяко $a \in \Sigma$,

$$\delta(\langle r_1, r_2\rangle, a) = \langle \delta_1(r_1,a), \delta_2(r_2,a)\rangle;$$

Едновременно симулираме изчисление и по двата автомата

- $s = \langle s_1, s_2 \rangle$;
- $F = \{\langle r_1, r_2 \rangle \mid r_1 \in F_1 \vee r_2 \in F_2\} = (F_1 \times Q_2) \cup (Q_1 \times F_2)$.

□

По-нататък ще дадем друга конструкция за обединение, която ще бъде по-ефективна.

✍ Проверете, че $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$

Следствие 2.1. Класът на автоматните езици е затворен относно операцията **сечение**. Това означава, че ако L_1 и L_2 са два произволни автоматни езика над азбуката Σ, то $L_1 \cap L_2$ също е автоматен език.

Док. Използвайте конструкцията на автомата $\mathcal{A} = \langle Q, \Sigma, s, \delta, F \rangle$ от *Твърдение 2.3*, с единствената разлика, че тук избираме финалните състояния да бъдат елементите на множеството

✍ Докажете, че така построения автомат $\mathcal{A}$ разпознава $L_1 \cap L_2$!

$$F = \{\langle q_1, q_2 \rangle \mid q_1 \in F_1 \ \& \ q_2 \in F_2\} = F_1 \times F_2.$$

□

Твърдение 2.4. Нека L е автоматен език. Тогава $\Sigma^\star \setminus L$ също е автоматен език.

Док. Нека $L = L(\mathcal{A})$, където $\mathcal{A} = \langle Q, \Sigma, s, \delta, F \rangle$ е **тотален**. Да вземем автомата $\mathcal{A}' = \langle Q, \Sigma, s, \delta, Q \setminus F \rangle$, т.е. $\mathcal{A}'$ е същият като $\mathcal{A}$, с единствената разлика, че финалните състояния на $\mathcal{A}'$ са тези състояния, които **не** са финални в $\mathcal{A}$. □

Защо искаме $\mathcal{A}$ да бъде тотален ?

✍ Проверете, че $\Sigma^\star \setminus L = \mathcal{L}(\mathcal{A}')$

Задача 2.1. За всеки от следните езици L, постройте автомат $\mathcal{A}$, който разпознава езика L.

а) $L = \{a^n b \mid n \geq 0\}$;

б) $L = \{\varepsilon, a, b\}$;

в) $L = \emptyset$;

г) $L = \{a, b\}^\star \setminus \{\varepsilon\}$;

д) $L = \{a^n b^m \mid n, m \geq 0\}$;

е) $L = \{a^n b^m \mid n, m \geq 1\}$;

ж) $L = \{a, b\}^\star \setminus \{a\}$;

з) $L = \{w \in \{a, b\}^\star \mid$ съдържа поне две $a\}$;

и) $L = \{w \in \{a, b\}^\star \mid$ съдържа поне две a и поне едно $b\}$;

к) $L = \{w \in \{a, b\}^\star \mid$ на всяка нечетна позиция на w е буквата $a\}$;

л) $L = \{w \in \{a, b\}^\star \mid w$ съдържа четен брой a и най-много едно $b\}$;

м) $L = \{w \in \{a, b\}^\star \mid |w| \leq 3\}$;

н) $L = \{w \in \{a, b\}^\star \mid w$ не започва с $ab\}$;

о) $L = \{w \in \{a, b\}^\star \mid w$ завършва с $ab\}$;

п) $L = \{w \in \{a, b\}^\star \mid w$ съдържа $bab\}$;

р) $L = \{w \in \{a,b\}^\star \mid w$ не съдържа $bab\}$;

с) $L = \{w \in \{a,b\}^\star \mid w$ няма две последователни $a\}$; (решена е по-долу)

т) $L = \{w \in \{a,b\}^\star \mid w$ започва и завършва с буквата $a\}$;

у) $L = \{w \in \{a,b\}^\star \mid w$ започва и завършва с една и съща буква$\}$;

ф) $L = \{w \in \{a,b\}^\star \mid |w| \equiv 0 \pmod 2$ & w съдържа точно едно $a\}$;

х) $L = \{w \in \{a,b\}^\star \mid$ всяко a в w се следва от поне едно $b\}$;

ц) $L = \{w \in \{a,b\}^\star \mid |w| \equiv 0 \bmod 3\}$;

ч) $L = \{w \in \{a,b\}^\star \mid N_a(w) \equiv 1 \bmod 3\}$; $N_a(w)$ - броят на срещанията на буквата a в думата w

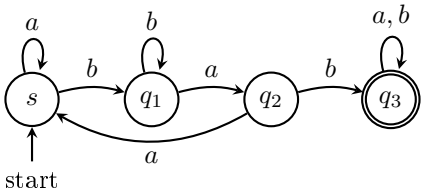
ш) $L = \{\omega \in \{a,b\}^\star \mid N_a(\omega) \equiv 0 \bmod 3$ & $N_b(\omega) \equiv 1 \bmod 2\}$;

щ) $L = \{\omega \in \{a,b\}^\star \mid N_a(\omega) \equiv 0 \bmod 2 \ \vee \ \omega$ съдържа точно две $b\}$;

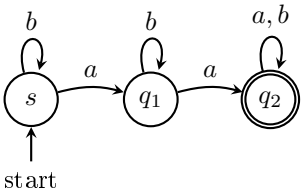
ю) $L = \{\omega \in \{a,b\}^\star \mid \omega$ съдържа равен брой срещания на ab и на $ba\}$.

я) $L = \{\omega_1 \sharp \omega_2 \sharp \omega_3 \mid \forall i \in [1,3](\omega_i \in \{a,b\}^\star$ & $|\omega_i| \geq i+1)\}$;

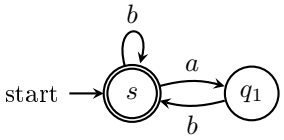
✍ За всички тези автомати, дефинирайте функциите на преходите им!



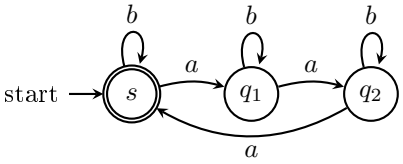
(а) $\{\omega \in \{a,b\}^\star \mid \omega$ съдържа $bab\}$



(б) $\{\omega \in \{a,b\}^\star \mid N_a(\omega) \geq 2\}$



(в) $\{\omega \in \{a,b\}^\star \mid$ всяко a в ω се следва от поне едно $b\}$



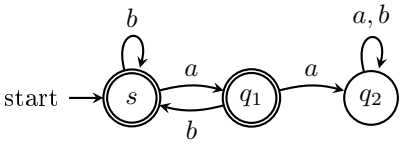
(г) $\{\omega \in \{a,b\}^\star \mid N_a(\omega) \equiv 0 \pmod 3\}$

В повечето от горните задачи е лесно да се съобрази, че построения автомат разпознава желания език. При по-сложни задачи обаче, ще се наложи да дадем доказателство, като обикновено се прилага *метода на математическата индукция* върху дължината на думите. Ще разгледаме няколко такива примера.

Задача 2.2. Докажете, че езикът L е автоматен, където

$$L = \{\alpha \in \{a,b\}^\star \ \mid \ \alpha \text{ не съдържа две поредни срещания на } a\}.$$

Док. Да разгледаме $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$ с функция на преходите



Ще докажем, че $L = \mathcal{L}(\mathcal{A})$. Първо ще се концентрираме върху доказателството на $\mathcal{L}(\mathcal{A}) \subseteq L$. Ще докажем с индукция по дължината на думата α, че:

Озн. $|\alpha|$ - дължината на думата α

(1) ако $\delta^\star(s, \alpha) = s$, то α не съдържа две поредни срещания на a и ако $|\alpha| > 0$, то α завършва на b;

(2) ако $\delta^\star(s, \alpha) = q_1$, то α не съдържа две поредни срещания на a и завършва на a.

За $|\alpha| = 0$, то твърденията (1) и (2) са ясни (Защо?). Да приемем, че твърденията (1) и (2) са верни за произволни думи α с дължина n. Нека $|\alpha| = n + 1$, т.е. $\alpha = \beta x$, където $|\beta| = n$ и $x \in \Sigma$. Ще докажем (1) и (2) за α.

- Нека $\delta^\star(s, \beta x) = s = \delta(\delta^\star(s, \beta), x)$. Според дефиницията на функцията δ, $x = b$ и $\delta^\star(s, \beta) \in \{s, q_1\}$. Тогава по **И.П.** за (1) и (2), β не съдържа две поредни срещания на a. Тогава е очевидно, че βx също не съдържа две поредни срещания на a.
- Нека $\delta^\star(s, \beta x) = q_1 = \delta(\delta^\star(s, \beta), x)$. Според дефиницията на δ, $x = a$ и $\delta^\star(s, \beta) = s$. Тогава по **И.П.** за (1), β не съдържа две поредни срещания на a и ако $|\beta| > 0$, то завършва на b. Тогава е очевидно, че βx също не съдържа две поредни срещания на a.

Така доказахме с индукция по дължината на думата, че за всяка дума α са изпълнени твърденията (1) и (2). По дефиниция, ако $\alpha \in \mathcal{L}(\mathcal{A})$, то $\delta^\star(s, \alpha) \in \{s, q_1\}$ и от (1) и (2) следва, че и в двата случа α не съдържа две поредни срещания на буквата a, т.е. $\alpha \in L$. С други думи, доказахме, че

$$\mathcal{L}(\mathcal{A}) \subseteq L.$$

Сега ще докажем другата посока, т.е. $L \subseteq \mathcal{L}(\mathcal{A})$. Това означава да докажем, че

$$(\forall \alpha \in \Sigma^\star)[\alpha \in L \;\Rightarrow\; \delta^\star(s, \alpha) \in F],$$

което е еквивалентно на

Да напомним, че $p \Rightarrow q \equiv \neg q \Rightarrow \neg p$

$$(\forall \alpha \in \Sigma^\star)[\delta^\star(s, \alpha) \notin F \;\Rightarrow\; \alpha \notin L]. \tag{2.1}$$

Това е лесно да се съобрази. Щом $\delta^\star(s, \alpha) \notin F$, то $\delta^\star(s, \alpha) = q_2$ и думата α може да се представи по следния начин:

$$\alpha = \beta a \gamma \;\&\; \delta^\star(s, \beta) = q_1.$$

Използвайки свойство (2) от по-горе, понеже $\delta^\star(s, \beta) = q_1$, то β не съдържа две поредни срещания на a, но завършва на a. Сега е очевидно,

че βa съдържа две поредни срещания на a и щом βa е префикс на α, то думата $\alpha \notin L$. С това доказахме Свойство 2.1, а следователно и посоката $L \subseteq \mathcal{L}(\mathcal{A})$. $\square$

За една дума $\alpha \in \{0,1\}^\star$, нека с $\alpha_{(2)}$ да означим числото в десетична бройна система, което се представя в двоична бройна система като α. Например, $1101_{(2)} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = 13$. Тогава имаме следните свойства:

- $\varepsilon_{(2)} = 0$,
- $(\alpha 0)_{(2)} = 2 \cdot (\alpha)_{(2)}$,
- $(\alpha 1)_{(2)} = 2 \cdot (\alpha)_{(2)} + 1$.

Да отбележим, че за всяко число n има безкрайно много думи α, за които $\alpha_{(2)} = n$. Например, $10_{(2)} = 010_{(2)} = 0010_{(2)} = \cdots$

Задача 2.3. Докажете, че $L = \{\omega \in \{0,1\}^\star \mid \omega_{(2)} \equiv 2 \,(\mod 3)\}$ е автоматен.

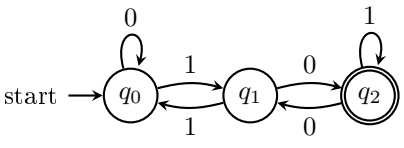
Док. Нашият автомат ще има три състояния $\{q_0, q_1, q_2\}$, като началното състояние ще бъде q_0. Целта ни е да дефинираме така автомата, че да имаме следното свойство:

$$(\forall \alpha \in \Sigma^\star)(\forall i < 3)[\alpha_{(2)} \equiv i \pmod 3 \;\Leftrightarrow\; \delta^\star(q_0, \alpha) = q_i], \tag{2.2}$$

т.е. всяко състояние отговаря на определен остатък при деление на три. Понеже искаме нашия автомат да разпознава тези думи α, за които $\alpha_{(2)} \equiv 2 \mod 3$, финалното състояние ще бъде q_2. Дефинираме функцията δ следвайки следните свойства:

- $\alpha_{(2)} \equiv 0 \mod 3 \;\Rightarrow\; (\alpha 0)_{(2)} \equiv 0 \mod 3$; $\delta(q_0, 0) = q_0$
- $\alpha_{(2)} \equiv 0 \mod 3 \;\Rightarrow\; (\alpha 1)_{(2)} \equiv 1 \mod 3$; $\delta(q_0, 1) = q_1$
- $\alpha_{(2)} \equiv 1 \mod 3 \;\Rightarrow\; (\alpha 0)_{(2)} \equiv 2 \mod 3$; $\delta(q_1, 0) = q_2$
- $\alpha_{(2)} \equiv 1 \mod 3 \;\Rightarrow\; (\alpha 1)_{(2)} \equiv 0 \mod 3$; $\delta(q_1, 1) = q_0$
- $\alpha_{(2)} \equiv 2 \mod 3 \;\Rightarrow\; (\alpha 0)_{(2)} \equiv 1 \mod 3$; $\delta(q_2, 0) = q_1$
- $\alpha_{(2)} \equiv 2 \mod 3 \;\Rightarrow\; (\alpha 1)_{(2)} \equiv 2 \mod 3$. $\delta(q_2, 1) = q_2$

Ето и картинка на автомата $\mathcal{A}$:



Фигура 2.2: $\mathcal{L}(\mathcal{A}) \stackrel{?}{=} \{\omega \in \{0,1\}^\star \mid \alpha_{(2)} \equiv 2 \pmod 3\}$

Да разгледаме твърденията:

(1) $\delta^\star(q_0, \alpha) = q_0 \;\Rightarrow\; \alpha_{(2)} \equiv 0 \mod 3$;

(2) $\delta^\star(q_0, \alpha) = q_1 \;\Rightarrow\; \alpha_{(2)} \equiv 1 \mod 3$;

(3) $\delta^\star(q_0, \alpha) = q_2 \;\Rightarrow\; \alpha_{(2)} \equiv 2 \mod 3$.

Ще докажем (1), (2) и (3) *едновременно* с индукция по дължината на думата α. За $|\alpha| = 0$, всички условия са изпълнени. (Защо?) Да приемем, че (1), (2) и (3) са изпълнени за думи с дължина n. Нека $|\alpha| = n + 1$, т.е. $\alpha = \beta x$, $|\beta| = n$. За да приложим индукционното предположение, ще използваме следното свойство:

$$\delta^\star(q_0, \beta x) = \delta(\delta^\star(q_0, \beta), x).$$

Ще докажем подробно само (3) понеже другите твърдения се доказват по сходен начин. Нека $\delta^\star(q_0, \beta x) = q_2$. Имаме два случая:

Обърнете внимание, че в доказателството на (3) използваме И.П. не само за (3), но и за (2)

- $x = 0$. Тогава, по дефиницията на δ, $\delta(q_1, 0) = q_2$ и следователно, $\delta^\star(q_0, \beta) = q_1$. По **И.П.** за (2) с β,

  $$\delta^\star(q_0, \beta) = q_1 \;\Rightarrow\; \beta_{(2)} \equiv 1 \bmod 3$$

  Тогава, $(\beta 0)_{(2)} \equiv 2 \mod 3$. Така доказахме, че

  $$\delta^\star(q_0, \beta 0) = q_2 \;\Rightarrow\; (\beta 0)_{(2)} \equiv 2 \bmod 3.$$

- $x = 1$. Тогава, по дефиницията на δ, $\delta(q_2, 1) = q_2$ и следователно, $\delta^\star(q_0, \beta) = q_2$. По **И.П.** за (3) с β,

  $$\delta^\star(q_0, \beta) = q_2 \;\Rightarrow\; \beta_{(2)} \equiv 2 \mod 3.$$

  Тогава, $(\beta 1)_{(2)} \equiv 2 \mod 3$. Така доказахме, че

  $$\delta^\star(q_0, \beta 1) = q_2 \;\Rightarrow\; (\beta 1)_{(2)} \equiv 2 \mod 3.$$

За да докажем (1), нека $\delta^\star(q_0, \beta x) = q_0$.

- $x = 0$. Разсъжденията са аналогични, като използваме **И.П.** за (1).
- $x = 1$. Разсъжденията са аналогични, като използваме **И.П.** за (2).

По същия начин доказваме и (2). Нека $\delta^\star(q_0, \beta x) = q_1$.

- При $x = 0$, използваме **И.П.** за (3).
- При $x = 1$, използваме **И.П.** за (1).

От (1), (2) и (3) следва директно, че $\mathcal{L}(\mathcal{A}) \subseteq L$.

За другата посока, нека $\alpha \in L$, т.е. $(\alpha)_{(2)} \equiv 2 \bmod 3$. Ако допуснем, че $\alpha \notin \mathcal{L}(\mathcal{A})$, то това означава, че $\delta^\star(q_0, \alpha) \in \{q_0, q_1\}$. Но в тези случаи получаваме от твърдения (1) и (2), че $(\alpha)_{(2)} \equiv 0 \bmod 3$ или $(\alpha)_{(2)} \equiv 1 \bmod 3$. Това е противоречие с избора на $\alpha \in L$. Следователно, ако $\alpha \in L$, то $\delta(q_0, \alpha) = q_2$. Така доказахме и посоката $L \subseteq \mathcal{L}(\mathcal{A})$. $\square$

2.2 Регулярни езици

Определение 2.2. Нека е дадена азбука Σ. Дефинираме множеството от *регулярни езици* над азбуката Σ и едновременно с това множеството от *регулярни изрази*, които разпознават тези езици.

Това е друг пример за индуктивна (рекурсивна) дефиниция.

1) за всеки символ $a \in \Sigma$, $\{a\}$ е регулярен език, който се разпознава от регулярния израз a;

2) $\{\varepsilon\}$ е регулярен език, който се разпознава от регулярния израз ε;

3) $\emptyset$ е регулярен език, който се разпознава от регулярния израз $\emptyset$;

4) $L_1 \cup L_2$, където L_1 и L_2 са регулярни езици, който се разпознава от регулярния израз $(r_1 + r_2)$, където r_1 и r_2 са регулярните изрази за L_1 и L_2. Записваме, че $\mathcal{L}(r_1) \cup \mathcal{L}(r_2) = \mathcal{L}(r_1 + r_2)$.

5) $L_1 \cdot L_2 = \{uw \mid u \in L_1 \ \& \ w \in L_2\}$, където L_1 и L_2 са регулярни езици, който се разпознава от регулярния израз $(r_1 \cdot r_2)$, където r_1 и r_2 са регулярните изрази за L_1 и L_2. Записваме, че $\mathcal{L}(r_1) \cdot \mathcal{L}(r_2) = \mathcal{L}(r_1 \cdot r_2)$.

Тази операция се нарича конкатенация. Обикновено изпускаме знака $\cdot$

6) $L^\star = \{w_1 w_2 \cdots w_n \mid n \in \mathbb{N} \ \& \ w_i \in L \text{ за всяко } i \leq n\}$, където L е регулярен език, който се разпознава от регулярния израз $(r^\star)$, където r е регулярния израз за L. Записваме, че $\mathcal{L}(r)^\star = \mathcal{L}(r^\star)$. Можем да запишем, че $L^\star = \bigcup_n L^n$, където $L^0 = \{\varepsilon\}$ и $L^{n+1} = L^n \cdot L$.

Звезда на Клини

Пример 2.1. Нека да разгледаме няколко примера какво точно представлява прилагането на операцията звезда на Клини върху един език.

- Нека $L = \{0, 11\}$. Тогава:
  - $L^0 = \{\varepsilon\}$, $L^1 = L$,
  - $L^2 = L^1 \cdot L^1 = \{00, 011, 110, 1111\}$,
  - $L^3 = L^1 \cdot L^2 = \{000, 0011, 0110, 01111, 1100, 11011, 11110, 111111\}$.
- Нека $L = \emptyset$. Тогава:
  - $L^0 = \{\varepsilon\}$,
  - $L^1 = \emptyset$,
  - $L^2 = L^1 \cdot L^1 = \emptyset$.

  Получаваме, че $L^\star = \{\varepsilon\}$, т.е. *краен* език
- Нека $L = \{0^i \mid i \in \mathbb{N}\} = \{\varepsilon, 0, 00, 000, \dots\}$. Тогава лесно може да се види, че $L = L^\star$.

Задача 2.4. За произволни регулярни изрази r и s, проверете:

а) $r + s = s + r$;

б) $(\varepsilon + r)^\star = r^\star$;

в) $\emptyset^\star = \varepsilon$;

г) $(r^\star s^\star) = (r + s)^\star$;

д) $(r^\star)^\star = r^\star$;

е) $(rs + r)^\star r = r(sr + r)^\star$;

ж) $s(rs + s)^\star r = rr^\star s(rr^\star s)^\star$;

з) $(r + s)^\star = r^\star + s^\star$;

и) $\emptyset^\star = \varepsilon^\star$;

Теорема 2.1 (Клини)**.** Всеки автоматен език се описва с регулярен израз.

Док. Нека $L = \mathcal{L}(\mathcal{A})$, за някой краен детерминиран автомат $\mathcal{A}$. Да фиксираме едно изброяване на състоянията $Q = \{q_1, \ldots, q_n\}$, като началното състояние е q_1. Ще означаваме с $L(i, j, k)$ множеството от тези думи, които могат да се разпознаят от автомата по път, който започва от q_i, завършва в q_j, и междинните състояния имат индекси $\leq k$. Например, за думата $\alpha = a_1 a_2 \cdots a_n$ имаме, че $\alpha \in L(i, j, k)$ точно тогава, когато съществуват състояния $q_{l_1}, \ldots, q_{l_{n-1}}$, като $l_1, \ldots, l_{n-1} \leq k$ и

стр. 79 от [PL98], стр. 33 от [HU79]

$$q_i \overset{a_1}{\to} q_{l_1} \overset{a_2}{\to} q_{l_2} \overset{a_3}{\to} \cdots \overset{a_{n-1}}{\to} q_{l_{n-1}} \overset{a_n}{\to} q_j.$$

Тогава за $n = |Q|$,

$$L(i, j, n) = \{\alpha \in \Sigma^\star \mid \delta^\star(q_i, \alpha) = q_j\}.$$

Така получаваме, че

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

Ще докажем с **индукция по** k, че за всяко i, j, k, множествата от думи $L(i, j, k)$ се описват с регулярен израз $r^k_{i,j}$

а) Нека $k = 0$. Ще докажем, че за всяко i, j, $L(i, j, 0)$ се описва с регулярен израз. Имаме да разгледаме два случая.

Ако $i = j$, то

$$L(i, j, 0) = \{\varepsilon\} \cup \{a \in \Sigma \mid \delta(q_i, a) = q_j\}.$$

Ако $i \neq j$, то

$$L(i, j, 0) = \{a \in \Sigma \mid \delta(q_i, a) = q_j\}.$$

б) Да предположим, че $k > 0$ и за всяко i, j, можем да намерим регулярните изрази, съответстващи на $L(i, j, k - 1)$. Тогава

$$L(i, j, k) = L(i, j, k - 1) \ \cup \ L(i, k, k - 1) \cdot (L(k, k, k - 1)^\star) \cdot L(k, j, k - 1).$$

Тогава по **И.П.** следва, че $L(i, j, k)$ може да се опише с регулярен израз, който е

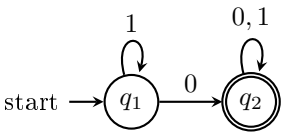
$$r^{k-1}_{i,j} + r^{k-1}_{i,k} \cdot (r^{k-1}_{k,k})^\star \cdot r^{k-1}_{k,j}.$$

Заключаваме, че за всяко i, j, k, $L(i, j, k)$ може да се опише с регулярен израз $r^k_{i,j}$. Тогава, ако $F = \{q_{i_1}, \ldots, q_{i_k}\}$, то $\mathcal{L}(\mathcal{A})$ се описва с регулярния израз

$$r^n_{1,i_1} + r^n_{1,i_2} + \cdots + r^n_{1,i_k}.$$

□

Пример 2.2. Да разгледаме следния автомат:



За да намерим регулярния език за автомата от Пример 2.2, трябва да намерим $r^2_{1,2}$, защото началното състояние е q_1, финалното е q_2 и броят на състоянията в автомата е 2.

$$\begin{aligned}
r^0_{1,1} &= \varepsilon + 1,\\
r^0_{1,2} &= 0,\\
r^0_{2,1} &= \emptyset,\\
r^0_{2,2} &= \varepsilon + 0 + 1,\\
r^1_{1,2} &= r^0_{1,2} + r^0_{1,1} \cdot (r^0_{1,1})^\star \cdot r^0_{1,2} = 0 + (\varepsilon + 1)(\varepsilon + 1)^\star 0 = 1^\star 0,\\
r^1_{2,2} &= r^0_{2,2} + r^0_{2,1} \cdot (r^0_{1,1})^\star \cdot r^0_{1,2} = \varepsilon + 0 + 1 + \emptyset(\varepsilon + 1)^\star 0 = \varepsilon + 0 + 1\\
r^2_{1,2} &= r^1_{1,2} + r^1_{1,2}(r^1_{2,2})^\star r^1_{2,2}\\
&= 1^\star 0 + 1^\star 0(\varepsilon + 0 + 1)^\star(\varepsilon + 0 + 1) = 1^\star 0(0 + 1)^\star.
\end{aligned}$$

Ясно е, че L_1 се описва с регулярния израз $r^2_{1,2} = 1^\star 0(0 + 1)^\star$.

Следващата ни цел е да видим, че имаме и обратната посока на горната лема. Ще докажем, че всеки регулярен език е автоматен. За тази цел първо ще въведем едно обобщение на понятието краен детерминиран автомат.

2.3 Недетерминирани крайни автомати

Определение 2.3. Недетерминиран краен автомат представлява

$$\mathcal{N} = \langle Q, \Sigma, s, \Delta, F\rangle,$$

Въведени от Рабин и Скот [RS59]

За яснота, често ще означаваме с $\mathcal{N}$ недетерминирани автомати, а с $\mathcal{A}$ детерминирани автомати

- Q е крайно множество от състояния;
- Σ е крайна азбука;
- $\Delta : Q \times \Sigma \to \mathscr{P}(Q)$ е функцията на преходите. Обърнете внимание, че тя е тотална.

Да напомним, че $\mathscr{P}(Q) = \{R \mid R \subseteq Q\}$, $|\mathscr{P}(Q)| = 2^{|Q|}$

- $s \in Q$ е началното състояние;
- $F \subseteq Q$ е множеството от финални състояния.

Сипсър [Sip97] позволява ε-преходи

Теорема 2.2. За всеки НКА $\mathcal{N}$ съществува еквивалентен на него КДА $\mathcal{D}$, т.е. $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{D})$.

Док. Нека $\mathcal{N} = \langle Q, \Sigma, s, \Delta, F\rangle$. Ще построим КДА $\mathcal{D} = (Q', \Sigma, \delta, s', F')$. Конструкцията е следната:

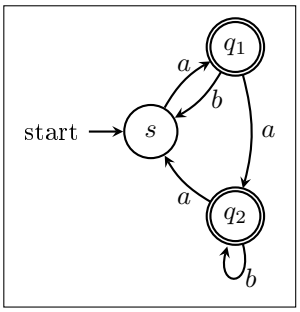
Да отбележим, че детерминираният автомат $\mathcal{D}$ има не повече от $2^{|Q|}$ на брой състояния

- $Q' = \mathscr{P}(Q)$;
- $\delta(R, a) = \{q \in Q \mid (\exists r \in R)[q \in \Delta(r, a)]\} = \bigcup_{r \in R} \Delta(r, a)$;
- $s' = \{s\}$;
- $F' = \{R \subseteq Q \mid R \cap F \neq \emptyset\}$.

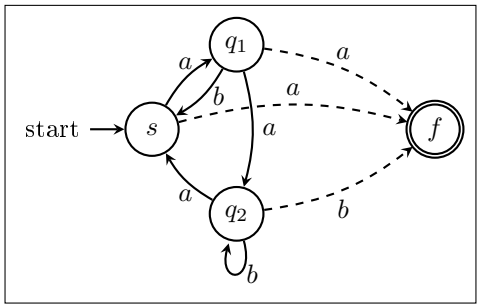
□

Задача 2.5. За всеки КНА $\mathcal{N}$ съществува КНА $\mathcal{N}'$ с едно финално състояние, за който $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{N}')$.

Упътване. Вместо формална конструкция, да разгледаме един пример, който илюстрира идеята.



(а) автомат $\mathcal{N}$



(б) автомат $\mathcal{N}'$, $\mathcal{L}(\mathcal{N}') = \mathcal{L}(\mathcal{N})$

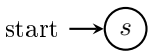
За произволен автомат $\mathcal{N}$, формулирайте точно конструкцията на $\mathcal{N}'$ с едно финално състояние и докажете, че наистина $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{N}')$. Обърнете внимание, че примера показва, че е възможно $\mathcal{N}$ да е детерминиран автомат, но полученият $\mathcal{N}'$ да бъде недетерминиран. □

Задача 2.6. Докажете, че ако L е автоматен език, то $L^R = \{\omega^R \mid \omega \in L\}$ също е автоматен.

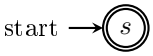
Нека $\mathcal{A}$, $L = \mathcal{L}(\mathcal{A})$, е само с едно финално състояние.

Лема 2.1. Съществува КНА $\mathcal{N} = \langle Q, \Sigma, s, \Delta, F\rangle$, който разпознава езика $L(r)$, където $r = \emptyset$, $r = \varepsilon$ или $r = a$, за $a \in \Sigma$.

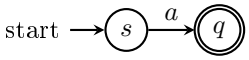
Док.



(а) $L(\emptyset)$



(б) $L(\varepsilon)$



(в) $L(a)$

□

Лема 2.2. Класът на автоматните езици е затворен относно операцията **конкатенация**. Това означава, че ако L_1 и L_2 са два произволни автоматни езика, то $L_1 \cdot L_2$ също е автоматен език.

Док. Нека са дадени автоматите:

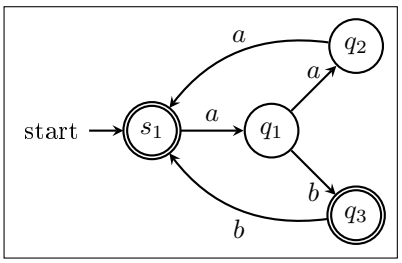
- $\mathcal{N}_1 = \langle Q_1, \Sigma, s_1, \Delta_1, F_1 \rangle$, като $\mathcal{L}(\mathcal{N}_1) = L_1$;
- $\mathcal{N}_2 = \langle Q_2, \Sigma, s_2, \Delta_2, F_2 \rangle$, като $\mathcal{L}(\mathcal{N}_2) = L_2$.

Ще дефинираме автомата $\mathcal{N} = \langle Q, \Sigma, s, \Delta, F \rangle$ като

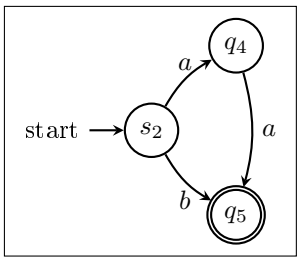
$$\mathcal{L}(\mathcal{N}) = L_1 \cdot L_2 = \mathcal{L}(\mathcal{N}_1) \cdot \mathcal{L}(\mathcal{N}_2).$$

- $Q = Q_1 \cup Q_2$;
- $s = s_1$;
- $F = \begin{cases} F_1 \cup F_2, & \text{ако } s_2 \in F_2 \\ F_2, & \text{иначе.} \end{cases}$
- $\Delta(q,a) = \begin{cases} \Delta_1(q,a), & \text{ако } q \in Q_1 \setminus F_1 \ \&\ a \in \Sigma \\ \Delta_2(q,a), & \text{ако } q \in Q_2 \ \&\ a \in \Sigma \\ \Delta_1(q,a) \cup \Delta_2(s_2,a), & \text{ако } q \in F_1 \ \&\ a \in \Sigma. \end{cases}$

□

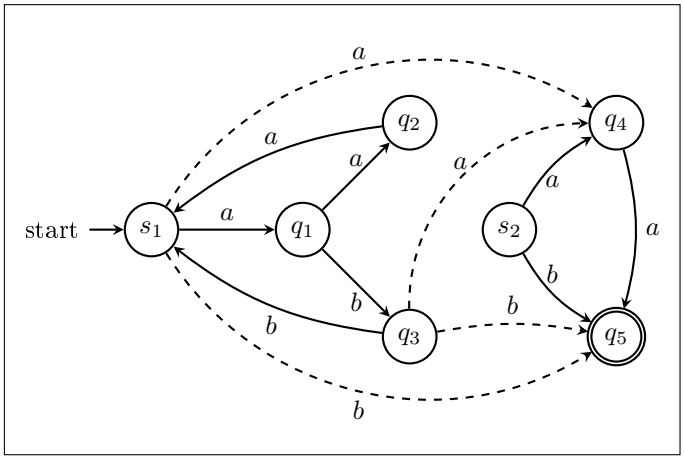


(а) автомат $\mathcal{N}_1$



(б) автомат $\mathcal{N}_2$

Пример 2.3. За да построим автомат, който разпознава конкатенацията на $\mathcal{L}(\mathcal{N}_1)$ и $\mathcal{L}(\mathcal{N}_2)$, трябва да свържем финалните състояния на $\mathcal{N}_1$ с изходящите от s_2 състояния на $\mathcal{N}_2$.



Фигура 2.6: $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{N}_1) \cdot \mathcal{L}(\mathcal{N}_2)$

Обърнете внимание, че $\mathcal{N}_1$ и $\mathcal{N}_2$ са детерминирани автомати, но $\mathcal{N}$ е недетерминиран. Също така, в този пример се оказва, че вече s_2 е недостижимо състояние, но в общия случай не можем да го премахнем, защото може да има преходи влизащи в s_2.

Лема 2.3. Класът от автоматните езици е затворен относно операцията **обединение**.

Док. Нека са дадени автоматите:

- $\mathcal{N}_1 = \langle Q_1, \Sigma, s_1, \Delta_1, F_1 \rangle$, като $L(\mathcal{N}_1) = L_1$;
- $\mathcal{N}_2 = \langle Q_2, \Sigma, s_2, \Delta_2, F_2 \rangle$, като $L(\mathcal{N}_2) = L_2$.

Ще дефинираме автомата $\mathcal{N} = \langle Q, \Sigma, s, \Delta, F \rangle$, така че

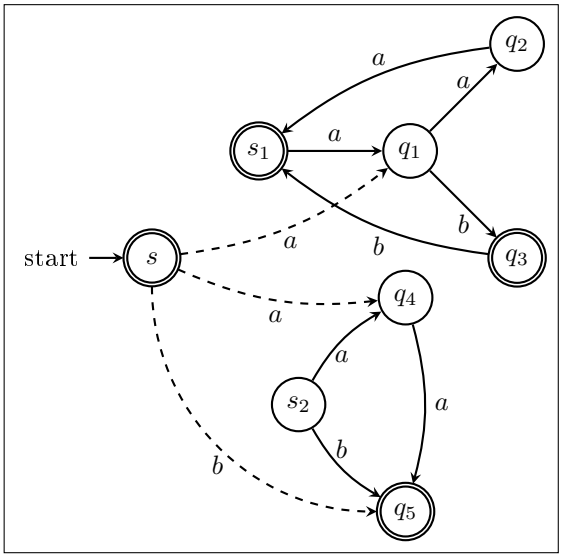
$$L(\mathcal{N}) = L(\mathcal{N}_1) \cup L(\mathcal{N}_2).$$

- $Q = Q_1 \cup Q_2 \cup \{s\}$;
- $F = \begin{cases} F_1 \cup F_2 \cup \{s\}, & \text{ако } s_1 \in F_1 \vee s_2 \in F_2 \\ F_1 \cup F_2, & \text{иначе} \end{cases}$
- $\Delta(q,a) = \begin{cases} \Delta_1(q,a), & \text{ако } q \in Q_1 \ \& \ a \in \Sigma \\ \Delta_2(q,a), & \text{ако } q \in Q_2 \ \& \ a \in \Sigma \\ \Delta_1(s_1,a) \cup \Delta_2(s_2,a), & \text{ако } q = s \ \& \ a \in \Sigma. \end{cases}$

□

Забележка. В началното състояние на новопостроения автомат $\mathcal{N}$ не влизат ребра.

Пример 2.4. За да построим автомат, който разпознава обединението на $\mathcal{L}(\mathcal{N}_1)$ и $\mathcal{L}(\mathcal{N}_2)$, трябва да свържем финалните състояния на $\mathcal{N}_1$ с изходящите от s_2 състояния на $\mathcal{N}_2$.



Фигура 2.7: $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{N}_1) \cup \mathcal{L}(\mathcal{N}_2)$

Обърнете внимание, че $\mathcal{N}_1$ и $\mathcal{N}_2$ са детерминирани автомати, но $\mathcal{N}$ е недетерминиран. Освен това, новото състояние s трябва да бъде маркирано като финално, защото s_1 е финално.

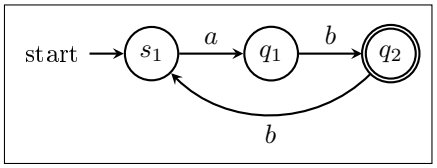
Лема 2.4. Класът от автоматните езици е затворен относно операцията **звезда на Клини**.

Док. Нека е даден автомата $\mathcal{N} = \langle Q, \Sigma, s, \Delta, F \rangle$, за който е изпълнено, че $L(\mathcal{N}) = L(r)$. Първата стъпка е да построим $\mathcal{N}_1 = \langle Q_1, \Sigma, s_1, \Delta_1, F_1 \rangle$, такъв че

$$L(\mathcal{N}_1) = \bigcup_{n \geq 1} (L(\mathcal{N}))^n = \bigcup_{n \geq 1} (L(r))^n = L(r^+).$$

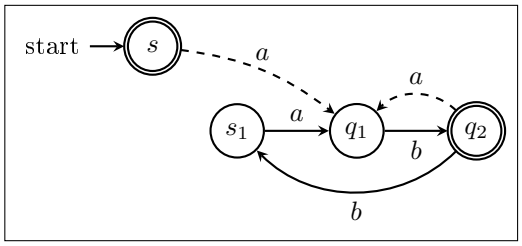
- $Q_1 = Q$;
- $s_1 = s$;
- $F_1 = F$;
- $\Delta_1(q, a) = \begin{cases} \Delta(q, a), & \text{ако } q \in Q \setminus F, a \in \Sigma \\ \Delta(q, a) \cup \Delta(s, a), & \text{ако } q \in F, a \in \Sigma. \end{cases}$

Накрая строим автомат $\mathcal{N}_2$, за който $L(\mathcal{N}_2) = \{\varepsilon\} \cup L(\mathcal{N}_1)$. □



Фигура 2.8: автомат $\mathcal{N}_3$

Пример 2.5. Нека да приложим конструкцията за да намерим автомат разпознаващ $\mathcal{L}(\mathcal{N}_3)^\star$.



Фигура 2.9: $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{N}_3)^\star = \mathcal{L}(\mathcal{N}_3)^+ \cup \{\varepsilon\}$

Лесно се вижда, че $\mathcal{L}(\mathcal{N}_1) = \{(abb)^n ab \mid n \in \mathbb{N}\}$. Формално погледнато, след като построим автомат за езика $\mathcal{L}(\mathcal{N}_1)^+$, трябва да приложим конструкцията за обединение на автомата за езика $\mathcal{L}(\mathcal{N}_1)^+$ с автомата за езика $\{\varepsilon\}$. Защо трябва да добавим ново начално състояние s? Да допуснем, че вместо това сме направили s_1 финално. Тогава има опасност да разпознаем повече думи. Например, думата abb би се разпознала от този автомат, но $abb \notin \mathcal{L}(\mathcal{N}_1)^\star$.

Задача 2.7. Да фиксираме една дума α над дадена азбука Σ. Опишете алгоритъм, който за вход произволен текстов файл β, отговаря дали думата α се среща в β. Каква е сложността на този алгоритъм относно дължините на α и β ?

(текстовият файл $\beta \in \Sigma^\star$)

2.4 Езици, които не са регулярни

Лема 2.5 (за покачването (регулярни езици))**.** Нека L да бъде регулярен език. Съществува число $p \geq 1$, зависещо само от L, за което за всяка дума $\alpha \in L, |\alpha| \geq p$ може да бъде записана във вида $\alpha = xyz$ и

На англ. се нарича Pumping Lemma

Има подобна лема и за безконтекстни езици

Обърнете внимание, че $0 \in \mathbb{N}$ и $xy^0z = xz$

1) $|y| \geq 1$;

2) $|xy| \leq p$;

3) $(\forall i \in \mathbb{N})[xy^iz \in L]$.

Док. Понеже L е регулярен, той се разпознава от $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$. Да положим $p = |Q|$ и нека $\alpha = a_1a_2\cdots a_k$ е дума, за която $k \geq p$. Да разгледаме първите p стъпки от изпълнението на α върху $\mathcal{A}$:

стр. 88 от [PL98], стр. 78 от [Sip97]

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} \cdots \xrightarrow{a_p} q_p.$$

Тъй като $|Q| = p$, а по този път участват $p+1$ състояния $q_0, q_1, \ldots, q_p$, то съществуват числа i, j, за които $0 \leq i < j \leq p$ и $q_i = q_j$. Нека разделим думата α на три части по следния начин:

$$x = a_1 \cdots a_i, \quad y = a_{i+1} \cdots a_j, \quad z = a_{j+1} \cdots a_k.$$

Ясно е, че $|y| \geq 1$ и $|xy| = j \leq p$. Освен това, лесно се съобразява, че

✍ Докажете!

за всяко $i \in \mathbb{N}$, $xy^iz \in L$. Да разгледаме само случая за $i = 0$. Думата $xy^0z = xz \in L$, защото имаме следното изчисление:

$$q_0 \overset{a_1}{\to} \cdots \overset{a_i}{\to} q_i \overset{a_{j+1}}{\to} q_{j+1} \cdots \overset{a_p}{\to} q_p \in F,$$

защото $q_i = q_j$. □

Практически е по-полезно да разглеждаме следната еквивалентна формулировка на лемата за покачването.

Контрапозиция на твърдението $p \to q$ е твърдението $\neg q \to \neg p$

Ясно е, че всеки краен език е регулярен. Нали?

Следствие 2.2 (Контрапозиция на лемата за покачването)**.** Нека L е произволен **безкраен** език. Нека също така е изпълнено, че за всяко естествено число $p \geq 1$ можем да намерим дума $\alpha \in L$, $|\alpha| \geq p$, такава че за всяко разбиване на думата на три части, $\alpha = xyz$, със свойствата $|y| \geq 1$ и $|xy| \leq p$, е изпълнено, че $(\exists i)[xy^iz \notin L]$. Тогава L **не** е регулярен език.

Док. *Лема 2.5* гласи, че ако L е регулярен език, то

$$(\exists p \geq 1)(\forall \alpha \in L)[|\alpha| \geq p \Rightarrow (\exists x, y, z \in \Sigma^\star)[\alpha = xyz \ \wedge \ |y| \geq 1 \ \wedge \ |xy| \leq p \ \wedge \ (\forall i \in \mathbb{N})[xy^iz \in L]]].$$

Отрицанието на горното твърдение гласи, че ако

$$(\forall p \geq 1)(\exists \alpha \in L)[|\alpha| \geq p \ \wedge (\forall x, y, z \in \Sigma^\star)[\alpha \neq xyz \ \vee \ |y| \ngeq 1 \ \vee \ |xy| \nleq p \ \vee \ (\exists i \in \mathbb{N})[xy^iz \notin L]]],$$

то L **не** е регулярен език. Горната формула е еквивалентна на:

$$(\forall p \geq 1)(\exists \alpha \in L)[|\alpha| \geq p \ \wedge \ (\forall x, y, z \in \Sigma^\star)[(\alpha = xyz \wedge |y| \geq 1 \wedge |xy| \leq p) \Rightarrow (\exists i \in \mathbb{N})[xy^iz \notin L]]].$$

□

Пример 2.6. Езикът $L = \{a^nb^n \mid n \in \mathbb{N}\}$ **не** е регулярен.

Док. Ще докажем, че

Това е важен пример. По-късно ще видим, че този език е безконтекстен

$$(\forall p \geq 1)(\exists \alpha \in L)[|\alpha| \geq p \ \wedge \ (\forall x, y, z \in \Sigma^\star)[(\alpha = xyz \wedge |y| \geq 1 \wedge |xy| \leq p) \Rightarrow (\exists i \in \mathbb{N})[xy^iz \notin L]].$$

Доказателството следва стъпките:

- Разглеждаме произволно число $p \geq 1$ (нямаме власт над избора на p).
- Избираме дума $\alpha \in L$, за която $|\alpha| \geq p$. Имаме свободата да изберем каквато дума α си харесаме, стига тя да принадлежи на L и да има дължина поне p. Щом имаме тази свобода, нека да изберем думата $\alpha = a^pb^p \in L$. Очевидно е, че $|\alpha| \geq p$.

  Няма общо правило, което да ни казва как избираме думата α. Възможно е да пробваме с няколко думи α, докато намерим такава, която върши работа
- Разглеждаме произволно разбиване на α на три части, $\alpha = xyz$, за които изискваме свойствата $|xy| \leq p$ и $|y| \geq 1$ (не знаем нищо друго за x, y и z освен тези две свойства).

  Обърнете внимание, че думата α зависи от константата p
- Ще намерим $i \in \mathbb{N}$, за което $xy^iz \notin L$. Понеже $|xy| \leq p$, то $y = a^k$, за $1 \leq k \leq p$. Тогава, ако вземем $i = 0$, получаваме $xy^0z = a^{p-k}b^p$. Ясно е, че $xz \notin L$, защото $p - k < p$.

Тогава от *Следствие 2.2* следва, че L не е регулярен език. □

Пример 2.7. Езикът $L = \{a^mb^n \mid m, n \in \mathbb{N} \ \& \ m < n\}$ **не** е регулярен.

Док. Доказателството следва стъпките:

- Разглеждаме произволно число $p \geq 1$.
- Избираме дума $\alpha \in L$, за която $|\alpha| \geq p$. Имаме свободата да изберем каквато дума α си харесаме, стига тя да принадлежи на L и да има дължина поне p. Щом имаме тази свобода, нека да изберем думата $\alpha = a^p b^{p+1} \in L$. Очевидно е, че $|\alpha| \geq p$.
- Разглеждаме произволно разбиване на α на три части, $\alpha = xyz$, за които изискваме свойствата $|xy| \leq p$ и $|y| \geq 1$ (не знаем нищо друго за x, y и z освен тези две свойства).
- Ще намерим $i \in \mathbb{N}$, за което $xy^i z \notin L$. Понеже $|xy| \leq p$, то $y = a^k$, за $1 \leq k \leq p$. Тогава ако вземем $i = 2$, получаваме

  $$xy^2z = a^{p-k}a^{2k}b^{p+1} = a^{p+k}b^{p+1}.$$

  Ясно е, че $xy^2z \notin L$, защото $p + k \geq p + 1$.

Тогава от *Следствие 2.2* следва, че L не е регулярен език. □

Пример 2.8. Езикът $L = \{a^n \mid n \text{ е просто число}\}$ не е регулярен.

Док. Доказателството следва стъпките:

- Разглеждаме произволно число $p \geq 1$.
- Избираме дума $w \in L$, за която $|w| \geq p$. Можем да изберем каквото w си харесаме, стига то да принадлежи на L и да има дължина поне p. Нека да изберем думата $w \in L$, такава че $|w| > p+1$. Знаем, че такава дума съществува, защото L е безкраен език. По-долу ще видим защо този избор е важен за нашите разсъждения.
- Разглеждаме произволно разбиване на w на три части, $w = xyz$, за които изискваме свойствата $|xy| \leq p$ и $|y| \geq 1$.
- Ще намерим i, за което $xy^i z \notin L$, т.е. ще намерим i, за което $|xy^i z| = |xz| + i \cdot |y|$ е *съставно число*. Понеже $|xy| \leq p$ и $|xyz| > p+1$, то $|z| > 1$. Да изберем $i = |xz| > 1$. Тогава:

  $$|xy^i z| = |xz| + i.|y| = |xz| + |xz|.|y| = (1 + |y|)|xz|$$

  е съставно число, следователно $xy^i z \notin L$.

Тогава от *Следствие 2.2* следва, че L не е регулярен език. □

Задача 2.8. Докажете, че езикът $L = \{a^{n^2} \mid n \in \mathbb{N}\}$ не е регулярен.

Док. Доказателството следва стъпките:

- Разглеждаме произволно число $p \geq 1$.
- Избираме достатъчно дълга дума, която принадлежи на езика L. Например, нека $w = a^{p^2}$.
- Разглеждаме произволно разбиване на w на три части, $w = xyz$, като $|xy| \leq p$ и $|y| \geq 1$.

- Ще намерим i, за което $xy^iz \notin L$. В нашия случай това означава, че $|xz| + i \cdot |y|$ не е точен квадрат. Тогава за $i = 2$,

$$p^2 = |xyz| < |xy^2z| = |xz| + 2|y| \leq p^2 + 2p < (p+1)^2.$$

  Получаваме, че $p^2 < |xy^2z| < (p+1)^2$, откъдето следва, че $|xy^2z|$ не е точен квадрат. Следователно, $xy^2z \notin L$.

Тогава от *Следствие 2.2* следва, че L не е регулярен език. □

Задача 2.9. Докажете, че езикът $L = \{a^{n!} \mid n \in \mathbb{N}\}$ не е регулярен.

Док. Доказателството следва стъпките:

- Разглеждаме произволно число $p \geq 1$.
- Избираме достатъчно дълга дума, която принадлежи на езика L. Например, нека $\omega = a^{(p+2)!}$.
- Разглеждаме произволно разбиване на ω на три части, $\omega = xyz$, като $|xy| \leq p$ и $|y| \geq 1$. Да обърнем внимание, че $1 \leq |y| \leq p$
- Ще намерим i, за което $xy^iz \notin L$. В нашия случай това означава, че $|xz| + i \cdot |y|$ не е от вида $n!$. Възможно ли е $xy^0z \in L$? Понеже $|xyz| = (p+2)!$, това означава, че $|xz| = k!$, за някое $k \leq p+1$. Тогава

  $$|y| = |xyz| - |xz| = (p+2)! - k! \geq (p+2)! - (p+1)! = (p+1).(p+1)! > p.$$

  Достигнахме до противоречие.

Тогава от *Следствие 2.2* следва, че L не е регулярен език. □

Следствия от лемата за покачването

Твърдение 2.5. Нека е даден автомата $\mathcal{A} = \langle Q, \Sigma, s, \delta, F \rangle$. Езикът $\mathcal{L}(\mathcal{A})$ е **непразен** точно тогава, когато съдържа дума $\alpha, |\alpha| < |Q|$.

Док. Ще разгледаме двете посоки на твърдението.

($\Rightarrow$) Нека $L = \mathcal{L}(\mathcal{A})$ е непразен език и нека $m = \min\{|\alpha| \mid \alpha \in L\}$. Ще докажем, че $m < |Q|$. За целта, да допуснем, че $m \geq |Q|$ и да изберем $\alpha \in L$, за която $|\alpha| = m$. Според *Лема 2.5*, съществува разбиване $xyz = \alpha$, такова че $xz \in L$. При положение, че $|y| \geq 1$, то $|xz| < m$, което е противоречие с минималността на m. Заключаваме, че нашето допускане е грешно. Тогава $m < |Q|$, откъдето следва, че съществува дума $\alpha \in L$ с $|\alpha| < |Q|$.

($\Leftarrow$) Тази посока е тривиална. Ако L съдържа дума α, за която $|\alpha| < |Q|$, то е очевидно, че L е непразен език.

□

Следствие 2.3. Съществува алгоритъм, който проверява дали даден автомат разпознава непразен език.

Следствие 2.4. Съществува алгоритъм, който определя дали два автомата $\mathcal{A}_1$ и $\mathcal{A}_2$ разпознават един и същ език.

Упътване. Използвайте наблюдението, че

$$\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2) \;\leftrightarrow\; (\mathcal{L}(\mathcal{A}_1) \setminus \mathcal{L}(\mathcal{A}_2)) \cup (\mathcal{L}(\mathcal{A}_2) \setminus \mathcal{L}(\mathcal{A}_1)) = \emptyset.$$

□

Твърдение 2.6. Регулярният език L, разпознаван от КДА $\mathcal{A}$, е **безкраен** точно тогава, когато съдържа дума $\alpha, |Q| \leq |\alpha| < 2|Q|$.

Док. Да разгледаме двете посоки на твърдението.

($\Leftarrow$) Нека L е регулярен език, за който съществува дума α, такава че $|Q| \leq |\alpha| < 2|Q|$. Тогава от *Лема 2.5* следва, че съществува разбиване $\alpha = xyz$ със свойството, че за всяко $i \in \mathbb{N}$, $xy^iz \in L$. Следователно, L е безкраен, защото $|y| \geq 1$.

($\Rightarrow$) Нека L е безкраен език и да вземем *най-късата* дума $\alpha \in L$, за която $|\alpha| \geq 2|Q|$. Понеже L е безкраен, знаем, че такава дума съществува. Тогава отново по *Лема 2.5*, имаме следното разбиване на α:

$$\alpha = xyz,\ |xy| \leq |Q|,\ 1 \leq |y|,\ xz \in L.$$

Но понеже $|xyz| \geq 2|Q|$, а $1 \leq |y| \leq |Q|$, то $|xyz| > |xz| \geq |Q|$ и понеже избрахме $\alpha = xyz$ да бъде най-късата дума с дължина поне $2|Q|$, заключаваме, че $|Q| \leq |xz| < 2|Q|$ и $xz \in L$.

□

Следствие 2.5. Съществува алгоритъм, който проверява дали даден регулярен език е безкраен.

Примери, за които лемата не е приложима

Пример 2.9. Езикът $L = \{c^k a^n b^m \mid k, n, m \in \mathbb{N} \;\&\; k = 1 \implies m = n\}$ **не** е регулярен, но условието за покачване от *Лема 2.5* е изпълнено за него.

Док. Да допуснем, че L е регулярен. Тогава ще следва, че

$$L_1 = L \cap ca^\star b^\star = \{ca^n b^n \mid n \in \mathbb{N}\}$$

е регулярен, но с лемата за разрастването лесно се вижда, че L_1 не е.

Сега да проверим, че условието за покачване от *Лема 2.5* е изпълнено за L. Да изберем константа $p = 2$. Сега трябва да разгледаме всички думи $\alpha \in L$, $|\alpha| \geq 2$ и за всяка α да посочим разбиване $xyz = \alpha$, за което са изпълнени трите свойства от лемата.

Условията за x, y, z са:

$|xy| \leq 2$
$|y| \geq 1$
$(\forall i \in \mathbb{N})(xy^iz \in L)$

- Ако $\alpha = a^n$ или $\alpha = b^n$, $n \geq 2$, то е очевидно, че можем да намерим такова разбиване.
- $\alpha = a^n b^m$ и $n + m \geq 2$, $n \geq 1$. Избираме $x = \varepsilon$, $y = a$, $z = a^{n-1}b^m$.
- $\alpha = ca^n b^n$, $n \geq 1$. Избираме $x = \varepsilon$, $y = c$, $z = a^n b^n$.
- $\alpha = c^2 a^n b^m$. Избираме $x = \varepsilon$, $y = c^2$, $z = a^n b^m$.
- $\alpha = c^k a^n b^m$, $k \geq 3$. Избираме $x = \varepsilon$, $y = c$, $z = c^{k-1} a^n b^m$.

□

2.5 Минимизация на КДА

- Нека $L \subseteq \Sigma^\star$ е език и нека $x, y \in \Sigma^\star$. Казваме, че x и y са **еквивалентни относно** L, което записваме като $x \approx_L y$, ако е изпълнено:

  $\approx_L$ е известна като релация на Майхил-Нероуд

  $$(\forall z \in \Sigma^\star)[xz \in L \ \leftrightarrow \ yz \in L].$$

  Това означава, че $x \approx_L y$, ако или и двете думи са в L или и двете не са в L и освен това, като прибавим произволна дума на края на x и y, новополучените думи са или и двете в L или и двете не са в L.

- Нека $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$ е КДА. Казваме, че две думи $\alpha, \beta \in \Sigma^\star$ са **еквивалентни относно** $\mathcal{A}$, което означаваме с $\alpha \sim_{\mathcal{A}} \beta$, ако

  Трябва ли $\mathcal{A}$ да е тотален?

  $$\delta^\star(s, \alpha) = \delta^\star(s, \beta).$$

- Проверете, че $\approx_L$ и $\sim_{\mathcal{A}}$ са **релации на еквивалентност**, т.е. те са рефлексивни, транзитивни и симетрични.

- Класът на еквивалентност на думата α относно релацията $\approx_L$ означаваме като
  $$[\alpha]_L = \{\beta \in \Sigma^\star \mid \alpha \approx_L \beta\}.$$
  С $|\approx_L|$ ще означаваме броя на класовете на еквивалентност на релацията $\approx_L$.

- Класът на еквивалентност на думата α относно релацията $\sim_{\mathcal{A}}$ означаваме като
  $$[\alpha]_{\mathcal{A}} = \{\beta \in \Sigma^\star \mid \alpha \sim_{\mathcal{A}} \beta\}.$$
  С $|\sim_{\mathcal{A}}|$ ще означаваме броя на класовете на еквивалентност на релацията $\sim_{\mathcal{A}}$.

  $|\sim_{\mathcal{A}}|$ също се нарича и **индекс** на релацията $\sim_{\mathcal{A}}$

- Казваме, че едно състояние q е **достижимо** в автомата $\mathcal{A}$, ако съществува дума $\alpha \in \Sigma^\star$, за която $\delta^\star_{\mathcal{A}}(s, \alpha) = q$. Премахването на недостижимите състояния от един автомат запазва разпознавания език.

- Съобразете, че всяко състояние на $\mathcal{A}$, което е достижимо от началното състояние, определя клас на еквивалентност относно релацията $\sim_{\mathcal{A}}$. Това означава, че ако за всяка дума означим $q_\alpha = \delta^\star_{\mathcal{A}}(s, \alpha)$, то $\alpha \sim_{\mathcal{A}} \beta$ точно тогава, когато $q_\alpha = q_\beta$. Заключаваме, че броят на класовете на еквивалентност на $\sim_{\mathcal{A}}$ е равен на броя на достижимите от s състояния.

- Релациите $\approx_{\mathcal{L}}$ и $\sim_{\mathcal{A}}$ са *дясно-инвариантни*, т.е. за всеки две думи α и β е изпълнено:

  ✍ Проверете!

  $$\alpha \sim_{\mathcal{A}} \beta \implies (\forall \gamma \in \Sigma^\star)[\alpha\gamma \sim_{\mathcal{A}} \beta\gamma],$$
  $$\alpha \approx_{\mathcal{L}} \beta \implies (\forall \gamma \in \Sigma^\star)[\alpha\gamma \approx_{\mathcal{L}} \beta\gamma].$$

Теорема 2.3. За всеки КДА $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$ е изпълнено:

$$(\forall \alpha, \beta \in \Sigma^\star)[\alpha \sim_{\mathcal{A}} \beta \implies \alpha \approx_{\mathcal{L}(\mathcal{A})} \beta].$$

С други думи, $[\alpha]_{\mathcal{A}} \subseteq [\alpha]_{\mathcal{L}(\mathcal{A})}$, за всяка дума $\alpha \in \Sigma^\star$.

Док. Да означим за всяка дума α, $q_\alpha = \delta^\star_{\mathcal{A}}(s, \alpha)$. Лесно се съобразява, че за всеки две думи α и β имаме

$$\begin{aligned} \alpha \sim_{\mathcal{A}} \beta &\leftrightarrow \delta^\star(s,\alpha) = \delta^\star(s,\beta) && \text{(по деф. на } \sim_{\mathcal{A}}) \\ &\leftrightarrow q_\alpha = q_\beta. \end{aligned}$$

Нека $\alpha \sim_{\mathcal{A}} \beta$. Ще проверим, че $\alpha \approx_{\mathcal{L}(\mathcal{A})} \beta$. За произволно $\gamma \in \Sigma^\star$ имаме:

$$\begin{aligned} \alpha\gamma \in \mathcal{L}(\mathcal{A}) &\leftrightarrow \delta^\star(s,\alpha\gamma) \in F && \text{(по деф. на } \mathcal{L}(\mathcal{A})) \\ &\leftrightarrow \delta^\star(\delta^\star(s,\alpha),\gamma) \in F && \text{(по деф. на } \delta^\star) \\ &\leftrightarrow \delta^\star(q_\alpha,\gamma) \in F && (q_\alpha = \delta^\star(s,\alpha)) \\ &\leftrightarrow \delta^\star(q_\beta,\gamma) \in F && (q_\alpha = q_\beta, \text{ защото } \alpha \sim_{\mathcal{A}} \beta) \\ &\leftrightarrow \delta^\star(\delta^\star(s,\beta),\gamma) \in F && (q_\beta = \delta^\star(s,\beta)) \\ &\leftrightarrow \delta^\star(s,\beta\gamma) \in F && \text{(по деф. на } \delta^\star) \\ &\leftrightarrow \beta\gamma \in \mathcal{L}(\mathcal{A}) && \text{(по деф. на } \mathcal{L}(\mathcal{A})). \end{aligned}$$

Заключаваме, че

$$(\forall \alpha, \beta \in \Sigma^\star)[\alpha \sim_{\mathcal{A}} \beta \implies \alpha \approx_{\mathcal{L}(\mathcal{A})} \beta].$$

□

Следствие 2.6. За всеки тотален КДА $\mathcal{A}$ е изпълнено, че

$$|\approx_{\mathcal{L}(\mathcal{A})}| \leq |\sim_{\mathcal{A}}|.$$

Док. Нека $A = \{[\alpha]_{\mathcal{L}(\mathcal{A})} \mid \alpha \in \Sigma^\star\}$ и $B = \{[\alpha]_{\mathcal{A}} \mid \alpha \in \Sigma^\star\}$. Да разгледаме изображението $f : B \to A$, определено като $f([\alpha]_{\mathcal{A}}) = [\alpha]_{\mathcal{L}(\mathcal{A})}$.

- Първо ще проверим, че f е **функция**, т.е. трябва да проверим, че

  $$(\forall \alpha, \beta \in \Sigma^\star)[\alpha \sim_{\mathcal{A}} \beta \implies f([\alpha]_{\mathcal{A}}) = f([\beta]_{\mathcal{A}})].$$

  Да допуснем, че съществуват думи α и β, такива че $[\alpha]_{\mathcal{A}} = [\beta]_{\mathcal{A}}$, но $f([\alpha]_{\mathcal{A}}) = [\alpha]_{\mathcal{L}(\mathcal{A})} \neq [\beta]_{\mathcal{L}(\mathcal{A})} = f([\beta]_{\mathcal{A}})$. Понеже $\sim_{\mathcal{A}}$ релация на еквивалентност, от $[\alpha]_{\mathcal{L}(\mathcal{A})} \neq [\beta]_{\mathcal{L}(\mathcal{A})}$ следва, че $[\alpha]_{\mathcal{L}(\mathcal{A})} \cap [\beta]_{\mathcal{L}(\mathcal{A})} = \emptyset$. От *Теорема 2.3* следва веднага, че това е невъзможно, защото

  $$\emptyset \neq [\alpha]_{\mathcal{A}} = [\beta]_{\mathcal{A}} \subseteq [\alpha]_{\mathcal{L}(\mathcal{A})} \cap [\beta]_{\mathcal{L}(\mathcal{A})}.$$

- Очевидно е, че f е **сюрекция**, защото на всеки клас $[\alpha]_{\mathcal{L}(\mathcal{A})}$ съответства класа $[\alpha]_{\mathcal{A}}$. $(\forall a \in A)(\exists b \in B)(f(b) = a)$

- От това, че $f : B \to A$ е сюректива функция следва, че $|B| \leq |A|$. Защо? ✍ Обяснете!

□

Следствие 2.7. Нека L е произволен регулярен език L. Всеки тотален КДА $\mathcal{A}$, който разпознава L има свойството

$$|Q| \geq |\approx_L|.$$

Док. Да изберем $\mathcal{A}$, който разпознава L, бъде такъв, че да **няма недостижими състояния**. Тъй като всяко достижимо състояние определя клас на еквивалентност относно $\sim_{\mathcal{A}}$, то получаваме, че $|Q| = |\sim_{\mathcal{A}}|$. Комбинирайки със *Следствие 2.6*,

$$|Q| = |\sim_{\mathcal{A}}| \geq |\approx_L|.$$

□

Така получаваме *долна граница* за броя на състоянията в тотален автомат разпознаващ езика L. Този брой е не по-малък от броя на класовете на еквивалентност на $\approx_L$.

Теорема за съществуване на МКДА

Определение 2.4. Нека $\mathcal{A}$ а тотален КДА, за който $L = \mathcal{L}(\mathcal{A})$. Казваме, че $\mathcal{A}$ е **минимален** за езика L, ако $|Q_{\mathcal{A}}| = |\approx_L|$.

Теорема 2.4 (Майхил-Нероуд)**.** Нека $L \subseteq \Sigma^\star$ е регулярен език. Тогава съществува КДА $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$, който разпознава L, с точно толкова състояния, колкото са класовете на еквивалентност на релацията $\approx_L$, т.е. $|Q| = |\approx_L|$.

Myhill-Nerode (1958)

Тази теорема не е доказана в [Sip97]. Тук следваме [PL98].

Док. Да фиксираме регулярния език L. Ще дефинираме тотален КДА $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$, разпознаващ L, като:

- $Q = \{[\alpha]_L \mid \alpha \in \Sigma^\star\}$;
- $s = [\varepsilon]_L$;
- $F = \{[\alpha]_L \mid \alpha \in L\} = \{[\alpha]_L \mid [\alpha]_L \cap L \neq \emptyset\}$;
- Определяме изображението δ като за всяка буква $x \in \Sigma$ и всяко състояние $[\alpha]_L \in Q$,
$$\delta([\alpha]_L, x) = [\alpha x]_L.$$

Първо, трябва да се уверим, че множеството от състояния Q е крайно, т.е. релацията $\approx_{\mathcal{L}}$ има крайно много класове на еквивалентност. И така, тъй като $\mathcal{L}$ е регулярен език, то той се разпознава от някой тотален КДА $\mathcal{A}'$. От *Следствие 2.7* имаме, че $|Q^{\mathcal{A}'}| \geq |\approx_L|$. Понеже $Q^{\mathcal{A}'}$ е крайно множество, то $\approx_L$ има крайно много класове и следователно Q също е крайно множество.

Второ, трябва да се уверим, че изображението δ задава функция, т.е. да проверим, че за всеки две думи α, β и всяка буква x,

$$[\alpha]_L = [\beta]_L \implies \delta([\alpha]_L, x) = \delta([\beta]_L, x).$$

Но това се вижда веднага, защото от определението на релацията $\approx_L$ следва, че ако $\alpha \approx_L \beta$, то за всяка буква x, $\alpha x \approx_L \beta x$, т.е. $[\alpha x]_L = [\beta x]_L$ и

$$\begin{aligned} [\alpha]_L = [\beta]_L &\implies [\alpha x]_L = [\beta x]_L && (\text{свойство на } \approx_L) \\ &\implies \delta([\alpha]_L, x) = [\alpha x]_L = [\beta x]_L = \delta([\beta]_L, x) && (\text{деф. на } \delta) \end{aligned}$$

Така вече сме показали, че $\mathcal{A}$ е коректно зададен тотален КДА. Остава да покажем, че $\mathcal{A}$ разпознава езика L, т.е. $\mathcal{L}(\mathcal{A}) = L$. За целта, първо ще докажем две помощни твърдения.

Твърдение 2.7. За всеки две думи α и β, $\delta^\star([\alpha]_L, \beta) = [\alpha\beta]_L$.

Док. Ще докажем това свойство с индукция по дължината на β.

- За $\beta = \varepsilon$ свойството следва директно от дефиницията на $\delta^\star$ като рефлексивно и транзитивно затваряне на δ, защото $\delta^\star([\alpha]_L, \varepsilon) = [\alpha]_L$.
- Нека $|\beta| = n + 1$ и да приемем, че сме доказали твърдението за думи с дължина n. Тогава $\beta = \gamma a$, където $|\gamma| = n$. Свойството следва от следните равенства:

$$\begin{aligned}
\delta^\star([\alpha]_L, \gamma a) &= \delta(\delta^\star([\alpha]_L, \gamma), a) && (\text{ от деф. на } \delta^\star)\\
&= \delta([\alpha\gamma]_L, a) && (\text{от } \textbf{И.П.} \text{ за } \gamma)\\
&= [\alpha\gamma a]_L && (\text{от деф. на } \delta)\\
&= [\alpha\beta]_L && (\beta = \gamma a).
\end{aligned}$$

□

Твърдение 2.8. За всяка дума α, $[\alpha]_L \cap L = \emptyset$ точно тогава, когато $\alpha \in L$.

Док. За посоката ($\Rightarrow$), нека $\beta \in [\alpha]_L \cap L$. Понеже $\beta \in [\alpha]_L$, имаме по дефиниция, че

$$(\forall \gamma \in \Sigma^\star)[\alpha\gamma \in L \ \leftrightarrow \ \beta\gamma \in L].$$

В частност при $\gamma = \varepsilon$, получаваме $\alpha \in L \ \leftrightarrow \ \beta \in L$. Тогава щом $\beta \in L$, то $\alpha \in L$. Посоката ($\Leftarrow$) е очевидна, защото $\alpha \in [\alpha]_L$. □

За да се убедим, че $L = \mathcal{L}(\mathcal{A})$ е достатъчно да проследим еквивалентностите:

$$\begin{aligned}
\alpha \in \mathcal{L}(\mathcal{A}) &\leftrightarrow \delta^\star(s, \alpha) \in F && (\text{от деф. на } \mathcal{L}(\mathcal{A}))\\
&\leftrightarrow \delta^\star([\varepsilon]_L, \alpha) \in F && (\text{по деф. } s = [\varepsilon]_L)\\
&\leftrightarrow \delta^\star([\varepsilon]_L, \alpha) = [\alpha]_L \cap L \neq \emptyset && (\text{от деф. на } F)\\
&\leftrightarrow \delta^\star([\varepsilon]_L, \alpha) = [\alpha]_L \ \&\ \alpha \in L && (\text{от } \textit{Твърдение 2.8})\\
&\leftrightarrow \alpha \in L && (\text{от } \textit{Твърдение 2.7}).
\end{aligned}$$

□

Определение 2.5. Нека $\mathcal{A}_1 = \langle Q_1, \Sigma, s_1, \delta_1, F_1\rangle$ и $\mathcal{A}_2 = \langle Q_2, \Sigma, s_2, \delta_2, F_2\rangle$. Казваме, че $\mathcal{A}_1$ и $\mathcal{A}_2$ са **изоморфни**, което означаваме с $\mathcal{A}_1 \cong \mathcal{A}_2$, ако съществува биекция $f : Q_1 \to Q_2$, за която:

- $f(s_1) = s_2$;
- $f[F_1] = \{f(q) \mid q \in F_1\} = F_2$;
- $(\forall a \in \Sigma)(\forall q \in Q_1)[f(\delta_1(q, a)) = \delta_2(f(q), a)]$.

Ще казваме, че f задава изоморфизъм на $\mathcal{A}_1$ върху $\mathcal{A}_2$.

Това означава, че два автомата $\mathcal{A}_1$ и $\mathcal{A}_2$ са изоморфни, ако можем да получим $\mathcal{A}_2$ като преименуваме състоянията на $\mathcal{A}_1$.

Следствие 2.8. Нека е даден регулярният език L. Всички минимални автомати за L са изоморфни на $\mathcal{A}_0$, автоматът построен в теоремата на Майхил-Нерод.

Док. Нека $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$ е произволен тотален автомат, за който $\mathcal{L}(\mathcal{A}) = L$ и $|Q| = |\approx_L|$. Съобразете, че $\mathcal{A}$ е *свързан*, т.е. всяко състояние на $\mathcal{A}$ е достижимо от началното. Искаме да докажем, че $\mathcal{A} \cong \mathcal{A}_0$. Понеже $\mathcal{A}$ е свързан, за всяко състояние q можем да намерим дума ω_q, за която $\delta^\star(s, \omega_q) = q$. Да дефинираме изображението $f : Q \to [\approx_L]$ като

$$f(q) = [\omega_q]_L.$$

Ще докажем, че f задава изоморфизъм на $\mathcal{A}$ върху $\mathcal{A}_0$.

- Първо да съобразим, че ако $\delta^\star_\mathcal{A}(s, \alpha) = q$, то $[\omega_q]_L = [\alpha]_L$. Понеже $\delta^\star_\mathcal{A}(s, \alpha) = q = \delta^\star_\mathcal{A}(s, \omega_q)$, то $\omega_q \sim_\mathcal{A} \alpha$. От *Теорема 2.3* имаме, че

  $$\omega_q \sim_\mathcal{A} \alpha \implies \omega_q \approx_L \alpha.$$

  Това означава, $[\omega_q]_L = [\alpha]_L$ и следователно f е определена коректно, т.е. f е **функция**.

- Ще проверим, че f е **инективна**, т.е.

  $$(\forall q_1, q_2 \in Q)[q_1 \neq q_2 \implies f(q_1) \neq f(q_2)].$$

  Да допуснем, че има състояния $q_1 \neq q_2$, за които

  $$f(q_1) = [\omega_{q_1}]_L = [\omega_{q_2}]_L = f(q_2).$$

  Тогава $\omega_{q_1} \not\sim_\mathcal{A} \omega_{q_2}$ и $\omega_{q_1} \approx_L \omega_{q_2}$. Но тогава от *Следствие 2.7* получаваме, че $|\sim_\mathcal{A}| > |\approx_L|$, което противоречи на минималността на $\mathcal{A}$. ✍ Обяснете!

- За да бъде f **сюрективна** трябва за всеки клас $[\beta]_L$ да съществува състояние q, за което $f(q) = [\beta]_L$. Понеже $\mathcal{A}$ е свързан, съществува състояние q, за което $\delta^\star_\mathcal{A}(s, \beta) = q$. Вече се убедихме, че в този случай $\beta \approx_L \omega_q$, защото $\beta \sim_\mathcal{A} \omega_q$. Тогава $f(q) = [\omega_q]_L = [\beta]_L$.

- За последно оставихме проверката, че f наистина е **изоморфизъм**:

$$\begin{aligned}
f(\delta_\mathcal{A}(q, a)) &= f(\delta_\mathcal{A}(\delta^\star_\mathcal{A}(s, \omega_q), a)) && \text{(от избора на } \omega_q) \\
&= f(\delta^\star_\mathcal{A}(s, \omega_q a)) && \text{(от деф. на } \delta^\star_\mathcal{A}) \\
&= [\omega_q a]_L && \text{(от деф. на } f) \\
&= \delta^\star_{\mathcal{A}_0}([\varepsilon]_L, \omega_q a) && \text{(от деф. на } \mathcal{A}_0) \\
&= \delta_{\mathcal{A}_0}(\delta^\star_{\mathcal{A}_0}([\varepsilon]_L, \omega_q), a) && \text{(от деф. на } \delta^\star_{\mathcal{A}_0}) \\
&= \delta_{\mathcal{A}_0}([\omega_q]_L, a) && \text{(от } \textit{Твърдение 2.7}) \\
&= \delta_{\mathcal{A}_0}(f(q), a) && (f(q) = [\omega_q]_L).
\end{aligned}$$

□

Проверка за регулярност на език

Твърдение 2.9. Езикът L е регулярен точно тогава, когато релацията $\approx_L$ има *крайно много* класове на еквивалентност.

Док. Ако L е регулярен, то той се разпознава от някой КДА $\mathcal{A}$, който има крайно много състояния и следователно крайно много класове на еквивалентност относно $\sim_{\mathcal{A}}$. Релацията $\approx_L$ е по-груба от $\sim_{\mathcal{A}}$ и има по-малко класове на еквивалентност. Следователно, $\approx_L$ има крайно много класове на еквивалентност.

За другата посока, ако $\approx_L$ има крайно много класове на еквивалентност, то можем да построим КДА $\mathcal{A}$ както в доказателството на *Теорема 2.4*, който разпознава L. □

Това следствие ни дава още един начин за проверка дали даден език е регулярен. За разлика от *Лема 2.5*, сега имаме **необходимо и достатъчно условие**. При даден език L, ние разглеждаме неговата релация $\approx_L$. Ако тя има крайно много класове, то езикът L е регулярен. В противен случай, езикът L не е регулярен.

Пример 2.10. За езика $L = \{a^n b^n \mid n \in \mathbb{N}\}$ имаме, че $|\approx_L| = \infty$, защото

$$(\forall k, j \in \mathbb{N})[k \neq j \implies [a^k b]_L \neq [a^j b]_L].$$

Проверете, че $[a^k b]_L = \{a^k b, a^{k+1}b^2, \ldots, a^{k+l}b^{l+1}, \ldots\}$. Така получаваме, че релацията $\approx_L$ има безкрайно много класове на еквивалентност. Заключаваме, че този език **не** е регулярен.

Пример 2.11. За езика $L = \{a^{n^2} \mid n \in \mathbb{N}\}$ имаме, че $|\approx_L| = \infty$, защото

$$(\forall m, n \in \mathbb{N})[m \neq n \implies [a^{n^2}]_L \neq [a^{m^2}]_L].$$

Без ограничение на общността, да разгледаме $n < m$ и думата $\gamma = a^{2n+1}$. Тогава $a^{n^2}\gamma = a^{(n+1)^2} \in L$, но $m^2 < m^2 + 2n + 1 < (m+1)^2$ и следователно $a^{m^2}\gamma = a^{m^2+2n+1} \notin L$.

Пример 2.12. За езика $L = \{a^{n!} \mid n \in \mathbb{N}\}$ имаме, че $|\approx_L| = \infty$, защото

$$(\forall m, n \in \mathbb{N})[m \neq n \implies [a^{n!}]_L \neq [a^{m!}]_L].$$

Без ограничение на общността, да разгледаме $n < m$ и думата $\gamma = a^{(n!)n}$. Тогава $a^{n!}\gamma = a^{(n+1)!} \in L$, но $m! < m! + (n!)n < m! + (m!)m = (m+1)!$ и следователно $a^{m!}\gamma = a^{m!+(n!)n} \notin L$.

Задача 2.10. Да разгледаме редицата от числа $\{f_n\}$ зададена по следния начин:

$$f_0 = f_1 = 1$$
$$f_{n+2} = f_n + f_{n+1}.$$

Докажете, че $|\approx_L| = \infty$, където

$$L = \{a^{f_n} \mid n \in \mathbb{N}\}.$$

Алгоритъм за намиране на минимален КДА.

- Ако имаме даден регулярен език L, то ние знаем как да построим минимален автомат разпознаващ L.

- Сега да фиксираме произволен КДА $\mathcal{A} = \langle Q, \Sigma, s, \delta, F\rangle$. Ще видим как можем да намерим минимален автомат $\mathcal{A}'$ разпознаващ $\mathcal{L}(\mathcal{A})$. Това означава, че броят на състоянията на $\mathcal{A}'$ трябва да е равен на $|\approx_{\mathcal{L}(\mathcal{A})}|$.

- Казваме, че две състояния p, q на автомата са **еквивалентни**, означаваме $p \equiv_{\mathcal{A}} q$, ако

$$p \equiv_{\mathcal{A}} q \overset{\text{деф}}{\leftrightarrow} (\forall \gamma \in \Sigma^\star)[\delta^\star(p, \gamma) \in F \quad \leftrightarrow \quad \delta^\star(q, \gamma) \in F].$$

- Релацията $\equiv_{\mathcal{A}}$ между състояния на автомата $\mathcal{A}$ е релация на еквивалентност.

- Нека q_α е състоянието, което съответства на думата α в $\mathcal{A}$, т.е. $\delta^\star_{\mathcal{A}}(s, \alpha) = q_\alpha$. Тогава лесно се вижда, че: ✍ Проверете!

$$q_\alpha \equiv_{\mathcal{A}} q_\beta \quad \leftrightarrow \quad \alpha \approx_{\mathcal{L}(\mathcal{A})} \beta.$$

  Това означава, че ако в $\mathcal{A}$ **няма недостижими състояния**, то всяко състояние на $\mathcal{A}$ е от вида q_α, за някое α, и

$$|\equiv_{\mathcal{A}}| = |\approx_{\mathcal{L}(\mathcal{A})}|.$$

Проблемът с намирането на класовете на еквивалентност на релацията $\equiv_{\mathcal{A}}$ е кванторът $\forall \gamma \in \Sigma^\star$ в нейната дефиницията. Алгоритъмът представлява намирането на релации $\equiv_n$, където

$$p \equiv_n q \overset{\text{деф}}{\leftrightarrow} (\forall \gamma \in \Sigma^{\leq n})[\delta^\star(p, \gamma) \in F \quad \leftrightarrow \quad \delta^\star(q, \gamma) \in F],$$

където сме положили

$$\Sigma^{\leq n} = \{\alpha \in \Sigma^\star \mid |\alpha| \leq n\}.$$

Релациите $\equiv_n$ представляват апроксимации на релацията $\equiv_{\mathcal{A}}$. Обърнете внимание, че за всяко n, $\equiv_n$ е *по-груба* релация от $\equiv_{n+1}$, която на свой ред е по-груба от $\equiv_{\mathcal{A}}$.

Алгоритъмът строи $\equiv_n$ докато не срещнем n, за което $\equiv_n = \equiv_{n+1}$. Тъй като броят на класовете на еквивалентност на $\equiv_{\mathcal{A}}$ е краен (не надминава $|Q|$), то със сигурност ще намерим такова n, за което $\equiv_n = \equiv_{n+1}$. Тогава заключаваме, че $\equiv_{\mathcal{A}} = \equiv_n$.

Понеже единствената дума с дължина 0 е ε и по определение $\delta^\star(p, \varepsilon) = p$, лесно се съобразява, че $\equiv_0$ има два класа на еквивалентност. Единият е F, а другият е $Q \setminus F$.

Твърдение 2.10. За всеки две състояния $p, q \in Q$, и всяко n, $p \equiv_{n+1} q$ точно тогава, когато

а) $p \equiv_n q$ и

б) $(\forall a \in \Sigma)[\delta(q, a) \equiv_n \delta(p, a)]$.

Док. Да разгледаме следната последователност от еквивалентни преобразувания:

$$\begin{aligned}
p \equiv_{n+1} q &\leftrightarrow (\forall \gamma \in \Sigma^{\leq n+1})[\delta^\star(p,\gamma) \in F \leftrightarrow \delta^\star(q,\gamma) \in F] && \text{(от деф.)}\\
&\leftrightarrow (\forall \gamma \in \Sigma^{\leq n})[\delta^\star(p,\gamma) \in F \leftrightarrow \delta^\star(q,\gamma) \in F] \;\&\\
&\quad (\forall a \in \Sigma)(\forall \gamma \in \Sigma^{\leq n})[\delta^\star(p,a\gamma) \in F \leftrightarrow \delta^\star(q,a\gamma) \in F]\\
&\leftrightarrow p \equiv_n q \;\& && \text{(от деф.)}\\
&\quad (\forall \gamma \in \Sigma^{\leq n})[\delta^\star(\delta(p,a),\gamma) \in F \leftrightarrow \delta^\star(\delta(q,a),\gamma) \in F] && (\textit{Твърдение 2.1})\\
&\leftrightarrow p \equiv_n q \;\& \; (\forall a \in \Sigma)[\delta(p,a) \equiv_n \delta(q,a)].
\end{aligned}$$

□

Следствие 2.9. Ако $\equiv_n = \equiv_{n+1}$, то за всяко $k \geq n$, $\equiv_n = \equiv_k$.

✍ Докажете!

Следствие 2.10. Ако $\equiv_n = \equiv_{n+1}$, то $\equiv_n = \equiv_{\mathcal{A}}$.

✍ Докажете!

Следствие 2.11. Ако $p \equiv_{\mathcal{A}} q$, то $\delta(p,a) \equiv_{\mathcal{A}} \delta(q,a)$.

✍ Докажете!

Нека е даден автомата $A = \langle Q, \Sigma, s, \delta, F\rangle$ и нека сме намерили n, за което $\equiv_n = \equiv_{n+1}$. Тогава $\equiv_{\mathcal{A}} = \equiv_n$. Строим автомата $\mathcal{A}' = \langle Q', \Sigma, s', \delta', F'\rangle$ по следния начин:

- $Q' = \{[q]_{\equiv_{\mathcal{A}}} \mid q \in Q\}$;
- $s' = [s]_{\equiv_{\mathcal{A}}}$;
- $\delta'([q]_{\equiv_{\mathcal{A}}}, a) = [\delta(q,a)]_{\equiv_{\mathcal{A}}}$;
- $F' = \{[q]_{\equiv_{\mathcal{A}}} \mid [q]_{\equiv_{\mathcal{A}}} \subseteq F\}$;

Лесно се вижда, че δ' е функция, т.е. горната дефиниция наистина задава автомат. Също така, ако $\mathcal{A}$ е тотален автомат без недостижими състояния, то $\mathcal{A}'$ е тотален автомат без недостижими състояния. Така от направените по-горе разсъждения следва, че $|\equiv_{\mathcal{A}}| = |\approx_{\mathcal{L}(\mathcal{A})}|$.

✍ Обяснете защо δ' е функция!

Твърдение 2.11. $(\forall q \in Q)(\forall \alpha \in \Sigma^\star)[\delta'^\star([q]_{\equiv_{\mathcal{A}}}, \alpha) = [\delta^\star(q,\alpha)]_{\equiv_{\mathcal{A}}}]$.

✍ Докажете с индукция по дължината на α!

Твърдение 2.12. Едно състояние $q \in F$ точно тогава, когато $[q]_{\equiv_{\mathcal{A}}} \subseteq F$.

✍ Обяснете!

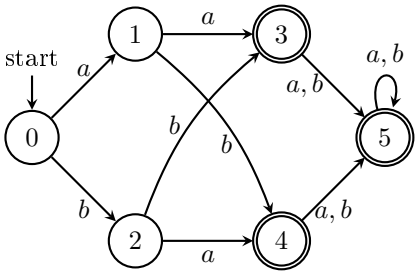
Твърдение 2.13. Автоматът $\mathcal{A}'$ е минимален автомат разпознаващ $\mathcal{L}(\mathcal{A})$.

Док. От направените по-горе разсъждения знаем, че $|\equiv_{\mathcal{A}}| = |\approx_{\mathcal{L}(\mathcal{A})}|$. Затова е достатъчно е да проверим, че $\mathcal{L}(\mathcal{A}') = \mathcal{L}(\mathcal{A})$. Достатъчно да се уверим, че следните еквивалентности са изпълнени:

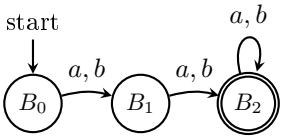
$$\begin{aligned}
\alpha \in \mathcal{L}(\mathcal{A}) &\leftrightarrow \delta^\star(s,\alpha) = q \in F && \text{(от деф. на } \mathcal{L}(\mathcal{A}))\\
&\leftrightarrow [\delta^\star(s,\alpha)]_{\equiv_{\mathcal{A}}} = [q]_{\equiv_{\mathcal{A}}} \subseteq F && (\textit{Твърдение 2.12})\\
&\leftrightarrow \delta'^\star([s]_{\equiv_{\mathcal{A}}}, \alpha) = [q]_{\equiv_{\mathcal{A}}} \subseteq F && (\textit{Твърдение 2.11})\\
&\leftrightarrow \delta'^\star([s]_{\equiv_{\mathcal{A}}}, \alpha) = [q]_{\equiv_{\mathcal{A}}} \in F' && \text{(от деф. на } F')\\
&\leftrightarrow \alpha \in \mathcal{L}(\mathcal{A}') && \text{(от деф. на } \mathcal{L}(\mathcal{A}')).
\end{aligned}$$

□

Пример 2.13. Да разгледаме следния краен детерминиран автомат $\mathcal{A}$.



(а) Ще построим минимален автомат, разпознаващ $\mathcal{L}(\mathcal{A})$



(б) Получаваме следния минимален автомат $\mathcal{A}_0$, $\mathcal{L}(\mathcal{A}_0) = \mathcal{L}(\mathcal{A})$

Съобразете, че $\mathcal{L}(\mathcal{A}) = \{\alpha \in \{a,b\}^\star \mid |\alpha| \geq 2\}$.

Ще приложим алгоритъма за минимизация за да получим минималния автомат за езика L. За всяко $n = 0, 1, 2, \ldots$, ще намерим класовете на еквивалентност на $\equiv_n$, докато не намерим n, за което $\equiv_n = \equiv_{n+1}$.

- Класовете на еквивалентност на $\equiv_0$ са два. Те са $A_0 = Q \setminus F = \{0, 1, 2\}$ и $A_1 = F = \{3, 4, 5\}$.
- Сега да видим дали можем да разбием някои от класовете на еквивалентност на $\equiv_0$.

Q	0	1	2	$3^\star$	$4^\star$	$5^\star$
$\equiv_0$	A_0	A_0	A_0	A_1	A_1	A_1
a	A_0	A_1	A_1	A_1	A_1	A_1
b	A_0	A_1	A_1	A_1	A_1	A_1

  Виждаме, че $0 \not\equiv_1 1$ и $1 \equiv_1 2$. Класовете на еквивалентност на $\equiv_1$ са $B_0 = \{0\}$, $B_1 = \{1, 2\}$, $B_2 = \{3, 4, 5\}$.
- Сега да видим дали можем да разбием някои от класовете на еквивалентност на $\equiv_1$.

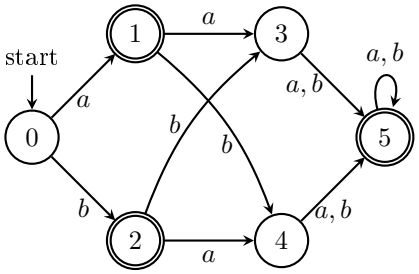
Q	0	1	2	$3^\star$	$4^\star$	$5^\star$
$\equiv_1$	B_0	B_1	B_1	B_2	B_2	B_2
a	B_1	B_2	B_2	B_2	B_2	B_2
b	B_1	B_2	B_2	B_2	B_2	B_2

  Виждаме, че $\equiv_1 = \equiv_2$. Следователно, минималният автомат има три състояния. Той е изобразен на Фигура 2.10б. Минималният автомат може да се представи и таблично:

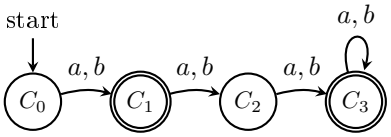
Получаваме, че $\equiv_{\mathcal{A}} = \equiv_1$

δ	B_0	B_1	B_2
a	B_1	B_2	B_2
b	B_1	B_2	B_2

Пример 2.14. Да разгледаме следния краен детерминиран автомат $\mathcal{A}$.



(а) Ще построим минимален автомат, разпознаващ $\mathcal{L}(\mathcal{A})$



(б) Получаваме следния минимален автомат $\mathcal{A}_0$, $\mathcal{L}(\mathcal{A}_0) = \mathcal{L}(\mathcal{A})$

Съобразете, че $\mathcal{L}(\mathcal{A}) = \{a, b\} \cup \{\alpha \in \{a, b\}^\star \mid |\alpha| \geq 3\}$.

Отново следваме същата процедура за минимизация. Ще намерим класовете на еквивалентност на $\equiv_n$, докато не намерим n, за което $\equiv_n = \equiv_{n+1}$.

- Класовете на екиваленост на $\equiv_0$ са $A_0 = Q \setminus F = \{0, 3, 4\}$ и $A_1 = F = \{1, 2, 5\}$.
- Разбиваме класовете на еквивалентност на $\equiv_0$.

Q	0	$1^\star$	$2^\star$	3	4	$5^\star$
$\equiv_0$	A_0	A_1	A_1	A_0	A_0	A_1
a	A_1	A_0	A_0	A_1	A_1	A_1
b	A_1	A_0	A_0	A_1	A_1	A_1

  Виждаме, че $1 \not\equiv_1 5$ и $1 \equiv_0 5$. Следователно, $\equiv_0 \neq \equiv_1$. Класовете на еквивалентност на $\equiv_1$ са $B_0 = \{0, 3, 4\}$, $B_1 = \{1, 2\}$, $B_2 = \{5\}$.
- Сега се опитваме да разбием класовете на еквивалентност на $\equiv_1$.

Q	0	$1^\star$	$2^\star$	3	4	$5^\star$
$\equiv_1$	B_0	B_1	B_1	B_0	B_0	B_2
a	B_1	B_0	B_0	B_2	B_2	B_2
b	B_1	B_0	B_0	B_2	B_2	B_2

  Имаме, че $0 \equiv_1 3$, но $0 \not\equiv_2 3$. Следователно $\equiv_1 \neq \equiv_2$. Класовете на еквивалентност на $\equiv_2$ са $C_0 = \{0\}$, $C_1 = \{1, 2\}$, $C_2 = \{3, 4\}$, $C_3 = \{5\}$.
- Отново опитваме да разбием класовете на $\equiv_2$.

Q	0	$1^\star$	$2^\star$	3	4	$5^\star$
$\equiv_2$	C_0	C_1	C_1	C_2	C_2	C_3
a	C_1	C_2	C_2	C_3	C_3	C_3
b	C_1	C_2	C_2	C_3	C_3	C_3

  Виждаме, че не можем да разбием C_1 или C_2. Следователно, $\equiv_2 = \equiv_3$ и минималният автомат разпознаващ езика L има четири състояния. Вижте Фигура 2.11б за преходите на минималния автомат. Минималният автомат може да се представи и таблично:

Получаваме, че $\equiv_{\mathcal{A}} = \equiv_2$

δ	C_0	C_1	C_2	C_3
a	C_1	C_2	C_3	C_3
b	C_1	C_2	C_3	C_3

2.6 Допълнителни задачи

Задача 2.11. Докажете, че следните езици са регулярни:

Озн. $N_a(\omega)$ - броят на срещанията на буквата a в думата ω

а) $L = \{\alpha \in \{a,b\}^\star \mid |N_a(\omega) - N_b(\omega)| \leq 2$ за всяка представка ω на $\alpha\}$;

б) $L = \{\alpha \in \{a,b\}^\star \mid |N_a(\omega) - N_b(\omega)| > 2$ за някоя представка ω на $\alpha\}$;

в) $L = \{\alpha \in \{a,b\}^\star \mid |N_a(\omega) - N_b(\omega)| > 2$ за някоя наставка ω на $\alpha\}$.

Задача 2.12. Нека $\Sigma = \{a,b\}$. Проверете дали L е регулярен, където

а) $L = \{\alpha^R \mid \alpha \in L_0\}$, където L_0 е регулярен;

б) $L = \{a^i b^i \mid i \in \mathbb{N}\}$; $\alpha = a^p b^p$

в) $L = \{a^i b^i \mid i,j \in \mathbb{N}\ \&\ i \neq j\}$;

г) $L = \{a^i b^j \mid i > j\}$; $\alpha = a^{p+1} b^p$.

д) $L = \{a^n b^m \mid n$ дели $m\}$.

е) $L = \{a^{2^n} \mid n \geq 1\}$;

ж) $L = \{a^m b^n a^{m+n} \mid m \geq 1\ \&\ n \geq 1\}$;

з) $L = \{a^{n.m} \mid n,m$ са прости числа$\}$;

и) $L = \{\omega \in \{a,b\}^\star \mid N_a(\omega) = N_b(\omega)\}$;

к) $L = \{\omega\omega \mid \omega \in \{a,b\}^\star\}$; $\alpha = a^p b a^p b$

л) $L = \{\omega\omega^R \mid \omega \in \{a,b\}^\star\}$;

м) $L = \{\alpha\beta\beta \in \{a,b\}^\star \mid \beta \neq \varepsilon\}$;

н) $L = \{a^n b^n c^n \mid n \geq 0\}$;

о) $L = \{\omega\omega\omega \mid \omega \in \Sigma^\star\}$;

п) $L = \{a^{2^n} \mid n \geq 0\}$;

р) $L = \{a^m b^n \mid n \neq m\}$;

с) $L = \{a^{n!} b^{n!} \mid n \neq 1\}$;

т) $L = \{a^{f_n} \mid f_0 = f_1 = 1\ \&\ f_{n+2} = f_{n+1} + f_n\}$;

у) $L = \{\alpha \in \{a,b\}^\star \mid |N_a(\alpha) - N_b(\alpha)| \leq 2\}$;

ф) $L = \{\alpha \in \{a,b\}^\star \mid \alpha = \alpha\beta\alpha\ \&\ |\beta| \leq |\alpha|\}$;

х) $L = \{\alpha \in \{a,b\}^\star \mid \alpha = \beta\gamma\gamma^R\ \&\ |\beta| \leq |\gamma|\}$;

ц) $L = \{c^k a^n b^m \mid k,m,n > 0\ \&\ n \neq m\}$;

ч) $L = \{c^k a^n b^n \mid k > 0\ \&\ n \geq 0\} \cup \{a,b\}^\star$;

ш) $L = \{\omega \in \{a,b\}^\star \mid N_a(\omega)$ не дели $N_b(\omega)\}$;

щ) $L = \{\omega \in \{a,b\}^\star \mid N_a(\omega) < N_b(\omega)\}$;

ю) $L = \{\omega \in \{a,b\}^\star \mid N_a(\omega) = 2N_b(\omega)\}$;

я) $L = \{\omega \in \{a,b\}^\star \mid |N_a(\omega) - N_b(\omega)| \leq 3\}$.

Задача 2.13. Нека L е регулярен език. Докажете, че

$$\text{Infix}(L) = \{\alpha \mid (\exists \beta, \gamma)[\beta\alpha\gamma \in L]\}$$

също е регулярен език.

Упътване. Най-лесно е да се построи автомат за $\text{Infix}(L)$ като се използва автомата за L. □

Задача 2.14. Нека $\Sigma = \{a,b,c,d\}$. Да се докаже, че езика

$$L = \{a_1a_2\cdots a_{2n} \in \Sigma^\star \mid (\forall j \in [1,n])[a_{2j-1} = a_{2j}] \ \&\ d \text{ се среща} \leq 3 \text{ пъти}\}$$

е регулярен.

Задача 2.15. Нека L_1 и L_2 са регулярни езици. Докажете, че L също е регулярен език, където

$$L = \{\alpha \mid (\exists \beta, \gamma)[\beta\alpha\gamma \in L_1] \ \&\ (\alpha \in L_2 \vee \alpha^R \in L_2)\}.$$

Определение 2.6. Да фиксираме две азбуки Σ_1 и Σ_2. Хомоморфизъм е изображение $h : \Sigma_1^\star \to \Sigma_2^\star$ със свойството, че за всеки две думи $\alpha, \beta \in \Sigma_1^\star$,

$$h(\alpha\beta) = h(\alpha) \cdot h(\beta).$$

Лесно се съобразява, че за всеки хомоморфизъм h, $h(\varepsilon) = \varepsilon$.

Задача 2.16. Нека $L \subseteq \Sigma_1^\star$ е регулярен език и $h : \Sigma_1^\star \to \Sigma_2^\star$ е хомоморфизъм. Тогава $h(L) = \{h(\alpha) \in \Sigma_2^\star \mid \alpha \in L\}$ е регулярен.

Упътване. Индукция по построението на регулярни езици. □

Задача 2.17. Нека $L \subseteq \Sigma_2^\star$ е регулярен език и $h : \Sigma_1^\star \to \Sigma_2^\star$ е хомоморфизъм. Тогава езикът $h^{-1}(L) = \{\alpha \in \Sigma_1^\star \mid h(\alpha) \in L\}$ е регулярен.

Упътване. Конструкция на автомат за $h^{-1}(L)$ при даден автомат за L. □

Задача 2.18. При дадени езици L, L' над азбуката Σ, да разгледаме: [PL98] стр. 84

а) $\text{Pref}(L) = \{\alpha \in \Sigma^\star \mid (\exists \beta \in \Sigma^\star)[\alpha\beta \in L]\}$;

б) $\text{Suf}(L) = \{\beta \in \Sigma^\star \mid (\exists \alpha \in \Sigma^\star)[\alpha\beta \in L]\}$;

в) $\text{Infix}(L) = \{\alpha \mid (\exists \beta, \gamma)[\beta\alpha\gamma \in L]\}$;

г) $\frac{1}{2}(L) = \{\omega \in \Sigma^\star \mid (\exists \alpha \in \Sigma^\star)[\omega\alpha \in L \ \&\ |\omega| = |\alpha|]\}$;

д) $L/L' = \{\alpha \in \Sigma^\star \mid (\exists \beta \in L')[\alpha\beta \in L]\}$;

е) $\text{Max}(L) = \{\alpha \in \Sigma^\star \mid (\forall \beta \in \Sigma^\star)[\beta \neq \varepsilon \implies \alpha\beta \notin L]\}$.

За всички тези езици, докажете, че са регулярни при условие, че L и L' са регулярни. Освен това, докажете, че L/L' е регулярен и при условието, че L е регулярен, но L' е произволен език.

Тази конструкция няма да бъде ефективна

Упътване.

а) Индукция по дефиницията на регулярен израз.

в) Най-лесно е да се построи автомат за $\mathrm{Infix}(L)$ като се използва автомата за L.

г) Конструкция с автомат за L и автомат за L^R.

□

Задача 2.19. За даден език L над азбуката Σ, да разгледаме езиците:

а) $L' = \{\alpha \mid (\exists \beta \in \Sigma^\star)[|\alpha| = 2|\beta| \ \&\ \alpha\beta \in L]\}$;

б) $L'' = \{\alpha \mid (\exists \beta \in \Sigma^\star)[2|\alpha| = |\beta| \ \&\ \alpha\beta \in L]\}$;

в) $\frac{1}{3}(L) = \{\alpha \mid (\exists \beta, \gamma)[|\alpha| = |\beta| = |\gamma| \ \&\ \alpha\beta\gamma \in L]\}$;

г) $\frac{2}{3}(L) = \{\beta \mid (\exists \beta, \gamma)[|\alpha| = |\beta| = |\gamma| \ \&\ \alpha\beta\gamma \in L]\}$;

д) $\frac{3}{3}(L) = \{\gamma \mid (\exists \beta, \gamma)[|\alpha| = |\beta| = |\gamma| \ \&\ \alpha\beta\gamma \in L]\}$;

е) $\sqrt{L} = \{\alpha \mid (\exists \beta)[|\beta| = |\alpha|^2 \ \&\ \alpha\beta \in L]\}$.

Проверете, ако L е регулярен, то кои от горните езици също са регулярни.

Задача 2.20. Да разгледаме езика ([Sip97], стр. 90)

$$L = \{\omega \in \{0,1\}^\star \mid \omega \text{ съдържа равен брой поднизове 01 и 10}\}.$$

Например, $101 \in L$, защото съдържа по веднъж 10 и 01. $1010 \notin L$, защото съдържа два пъти 10 и само веднъж 01. Докажете, че L е регулярен.

Задача 2.21. Да фиксираме азбука само с един символ $\Sigma = \{a\}$. Да положим за всяко $p, q \in \mathbb{N}$, ([Koz97], стр. 75; [PL98], стр. 89)

$$\mathcal{L}(p,q) = \{a^k \mid (\exists n \in \mathbb{N})[k = p + q \cdot n]\}.$$

Ако за един език L съществуват константи $p_1, \ldots, p_k$ и $q_1, \ldots, q_k$, такива че

$$L = \bigcup_{1 \le i \le k} \mathcal{L}(p_i, q_i),$$

то казваме, че L е *породен от аритметични прогресии*.

а) Докажете, че $L \subseteq \{a\}^\star$ е регулярен език точно тогава, когато L е породен от аритметична прогресия.

б) За *произволна* азбука Σ, докажете, че ако $L \subseteq \Sigma^\star$ е регулярен език, то езикът $\{a^{|\omega|} \mid \omega \in L\}$ е породен от аритметични прогресии.

Упътване.

а) За едната посока, разгледайте КДА за L.

б) За втората част, разгледайте $h : \Sigma \to \{a\}$ деф. като $(\forall b \in \Sigma)[h(b) = a]$. Докажете, че h поражда хомоморфизъм между $\Sigma^\star$ и $\{a\}^\star$. Тогава $h(L) = \{a^{|\omega|} \mid \omega \in L\}$, а ние знаем, че регулярните езици са затворени относно хомоморфни образи.

□

Задача 2.22. Да разгледаме азбуката:

$$\Sigma_3 = \left\{ \begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\end{bmatrix}, \begin{bmatrix}0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\1\\1\end{bmatrix}, \ldots, \begin{bmatrix}1\\1\\1\end{bmatrix} \right\}.$$

Докажете, че $L = \left\{ \begin{bmatrix}\alpha\\\beta\\\gamma\end{bmatrix} \in \Sigma_3^\star \mid \alpha_{(2)} + \beta_{(2)} = \gamma_{(2)} \right\}$ е автоматен език.

Задача 2.23. Да разгледаме азбуката:

$$\Sigma_2 = \left\{ \begin{bmatrix}0\\0\end{bmatrix}, \begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}, \begin{bmatrix}1\\1\end{bmatrix} \right\}.$$

Да разгледаме езиците:

а) $L_1 = \left\{ \begin{bmatrix}\alpha\\\beta\end{bmatrix} \in \Sigma_2^\star \mid \alpha_{(2)} < \beta_{(2)} \right\}$;

б) $L_2 = \left\{ \begin{bmatrix}\alpha\\\beta\end{bmatrix} \in \Sigma_2^\star \mid 3\alpha_{(2)} = \beta_{(2)} \right\}$;

в) $L_2 = \left\{ \begin{bmatrix}\alpha\\\beta\end{bmatrix} \in \Sigma_2^\star \mid \alpha = \beta^R \right\}$;

Докажете, че L_1 и L_2 са автоматни, а L_3 не е автоматен.

Глава 3

Безконтекстни езици и стекови автомати

3.1 Безконтекстни граматики

Определение 3.1. Безконтекстна граматика е четворка от вида

$$G = (V, \Sigma, R, S),$$

На англ. **context-free grammar**

Други срещани наименования на български са **контекстно-свободна**, **контекстно-независима**

където

- V е крайно множество от *променливи*;
- Σ е крайно множество от *букви*, $\Sigma \cap V = \emptyset$;
- $R \subseteq V \times (V \cup \Sigma)^\star$, крайно множество от *правила*;
- $S \in V$ е началната променлива.

Променливите се наричат също нетерминали

Буквите се наричат също терминали.

При дадена граматика G, за правилата на граматиката обикновено ще пишем $A \to \alpha$ вместо $(A, \alpha) \in R$. Ще въведем и релация между думи $\alpha, \beta \in (V \cup \Sigma)^\star$, която ще казва, че думата β се получава от α като приложим правило от граматиката. За две думи $u, v \in (V \cup \Sigma)^\star$ ще пишем $u \to_G v$, ако съществуват думи $x, y \in (\Sigma \cup V)^\star$, $A \in V$, правило $A \to \alpha$ и $u = xAy$, $v = x\alpha y$. С $\to_G^\star$ ще означаваме рефлексивното и транзитивно затваряне на релацията $\to_G$.

Езикът породен от граматиката G е множеството от думи

$$\mathcal{L}(G) = \{\alpha \in \Sigma^\star \mid S \to_G^\star \alpha\}.$$

Определение 3.2. Казваме, че езикът L е **безконтекстен**, ако съществува безконтекстна граматика G, за която $L = \mathcal{L}(G)$.

Задача 3.1. Докажете, че езикът $L = \{a^m b^n c^k \mid m+n \geq k\}$ е безконтекстен.

Док. Да разгледаме граматиката G с правила

$$S \to aSc|aS|B$$
$$B \to bBc|bB|\varepsilon.$$

Лесно се вижда с индукция по n, че за всяко n имаме свойствата:

✍ Докажете!

- $S \to^\star a^n S c^n$,
- $S \to^\star a^n S$,
- $B \to^\star a^n B c^n$,
- $B \to^\star b^n B$.

Комбинирайки горните свойства, можем да видим, че за всяко $n \geq k$,

- $S \to^\star a^n S c^k$,
- $B \to^\star b^n B c^k$.

За да докажем, че $L \subseteq L(G)$, да разгледаме една дума $\omega \in L$, т.е. $\omega = a^m b^n c^k$, където $m + n \geq k$. Имаме два случая:

- $k \leq m$, т.е. $m = k + l$ и $m + n = k + l + n$. Тогава имаме изводите:

  $$S \to^\star a^k S c^k,\ S \to^\star a^l S,\ S \to B,\ B \to^\star b^n B,\ B \to \varepsilon.$$

  Обединявайки всичко това, получаваме:

  $$S \to^\star a^m b^n c^k.$$

- $k > m$, т.е. $k = m + l$, за някое $l > 0$, и $m + n = k + r = m + l + r$, за някое r. Тогава имаме изводите:

  $$S \to^\star a^m S c^m,\ S \to B,\ B \to^\star b^l B c^l,\ B \to b^r B,\ B \to \varepsilon,$$

  и отново получаваме $S \to^\star a^m b^n c^k$.

Така доказахме, че $\omega \in \mathcal{L}(G)$.

Сега ще докажем, че $\mathcal{L}(G) \subseteq L$. С индукция по дължината на извода l, ще докажем, че ако $S \overset{l}{\to} \omega$, то $\omega \in M$, където

$$M = \{a^n S c^k \mid n \geq k\} \cup \{a^n b^m B c^k \mid n + m \geq k\} \cup \{a^n b^m c^k \mid n + m \geq k\}.$$

Ако $l = 0$, то е ясно, че $S \overset{0}{\to} S$ и $S \in M$.

Нека $l > 0$ и $S \overset{l-1}{\to} \alpha \to \omega$. От **И.П.** имаме, че $\alpha \in M$. Нека ω се получава от α с прилагане на правилото $C \to \gamma$. Разглеждаме всички варианти за думата $\alpha \in M$ и за правилото $C \to \gamma$ в граматиката G за да докажем, че $\omega \in M$. Удобно е да представим всички случаи в таблица.

$\alpha \in M$	$C \to \gamma$	$\omega \in M$?
$a^n S c^k$	$S \to aSc$	$a^{n+1} S c^{k+1}$
$a^n S c^k$	$S \to aS$	$a^{n+1} S c^k$
$a^n S c^k$	$S \to B$	$a^n B c^k$
$a^n b^m B c^k$	$B \to bBc$	$a^n b^{m+1} B c^{k+1}$
$a^n b^m B c^k$	$B \to bB$	$a^n b^{m+1} B c^k$
$a^n b^m B c^k$	$B \to \varepsilon$	$a^n b^m c^k$

Във всички случаи се установява, че $\omega \in M$. Сега, за всяка дума $\omega \in L(G)$ следва, че

$$\omega \in \Sigma^\star \cap M = \{a^m b^n c^k \mid m + n \geq k\}.$$

□

Задача 3.2. Докажете, че езикът $L = \{a^m b^n c^k \mid m + n \geq k + 1\}$ е безконтекстен.

$S \to aS \mid aSc \mid aB \mid bB$
$B \to bB \mid bBc \mid \varepsilon$

Задача 3.3. Докажете, че за произволна дума ω над азбуката $\{a, b\}$ са изпълнени свойствата:

а) ако $n_a(\omega) = n_b(\omega) + 1$, то съществуват думи ω_1, ω_2, за които $\omega = \omega_1 a \omega_2$, $n_a(\omega_1) = n_b(\omega_1)$ и $n_a(\omega_2) = n_b(\omega_2)$.

б) ако $n_b(\omega) = n_a(\omega) + 1$, то съществуват думи ω_1, ω_2, за които $\omega = \omega_1 b \omega_2$, $n_a(\omega_1) = n_b(\omega_1)$ и $n_a(\omega_2) = n_b(\omega_2)$.

Док. Пълна индукция по дължината на думата ω, за които $n_a(\omega) = n_b(\omega) + 1$.

- $|\omega| = 1$. Тогава $\omega_1 = \omega_2 = \varepsilon$ и $\omega = a$.
- $|\omega| = n+1$. Ще разгледаме два случая, в зависимост от първия символ на ω.
  - Случаят $\omega = a\omega'$ е лесен. (Защо?)
  - Интересният случай е $\omega = b\omega'$. Тогава $\omega = b^{i+1}a\omega'$. Да разгледаме думата ω'', която се получава от ω като премахнем първото срещане на думата ba, т.е. $\omega'' = b^i\omega'$ и $|\omega''| = n - 1$. Понеже от ω сме премахнали равен брой a-та и b-та, $n_a(\omega'') = n_b(\omega'') + 1$. Според **И.П.** за ω'', можем да запишем думата като $\omega'' = \omega_1''a\omega_2''$ и $n_a(\omega_1'') = n_b(\omega_1'')$, $n_a(\omega_2'') = n_b(\omega_2'')$. Понеже b^i е префикс на ω_1'', за да получим обратно ω, трябва да прибавим премахнатата част ba веднага след b^i в ω_1''.

□

Задача 3.4. За произволна дума $\omega \in \{a, b\}^\star$, докажете, че ако $n_a(\omega) > n_b(\omega)$, то съществуват думи ω_1 и ω_2, за които $\omega = \omega_1 a \omega_2$ и $n_a(\omega_1) \geq n_b(\omega_1)$, $n_a(\omega_2) \geq n_b(\omega_2)$.

Задача 3.5. Да се докаже, че езикът $L = \{\alpha \in \{a, b\}^\star \mid n_a(\alpha) = n_b(\alpha)\}$ е безконтекстен.

Док. Една възможна граматика G е следната:

Алтернативна граматика за езика L е

$S \to aB|bA$
$A \to a|aS|bAA$
$B \to b|bS|aBB$

$$S \to aSbS|bSaS|\varepsilon.$$

Например, да разгледаме извода на думата $aabbba$ в тази граматика:

$$S \to aSbS \to aaSbSbS \to aa\varepsilon bSbS \to aab\varepsilon bS \to aabbbSaS$$
$$\to aabbb\varepsilon aS \to aabbba.$$

Като следствие от *Задача 3.3* може лесно да се изведе, че за думи ω, за които $n_a(\omega) = n_b(\omega)$, е изпълнено следното:

а) ако $\omega = a\omega'$, то $\omega = a\omega_1 b\omega_2$ и $n_a(\omega_1) = n_b(\omega_1)$, $n_a(\omega_2) = n_b(\omega_2)$;

б) ако $\omega = b\omega'$, то $\omega = b\omega_1 a\omega_2$ и $n_a(\omega_1) = n_b(\omega_1)$, $n_a(\omega_2) = n_b(\omega_2)$.

Сега първо ще проверим, че $L \subseteq L(G)$. За целта ще докажем с *пълна индукция* по дължината на думата ω, че за всяка дума ω със свойството $n_a(\omega) = n_b(\omega)$ е изпълнено $S \to^\star \omega$.

- Нека $|\omega| = 0$. Тогава $S \to \varepsilon$.
- Нека $|\omega| = k + 1$. Имаме два случая.
  - $\omega = a\omega'$, т.е. от свойство а), $\omega = a\omega_1 b\omega_2$ и $n_a(\omega_1) = n_b(\omega_1)$, $n_a(\omega_2) = n_b(\omega_2)$. Тогава $|\omega_1| \leq k$ и по И.П. $S \to^\star \omega_1$. Аналогично, $S \to^\star \omega_2$. Понеже имаме правило $S \to aSbS$, заключаваме че $S \to^\star a\omega_1 b\omega_2$.
  - $\omega = b\omega'$, т.е. свойство б), $\omega = b\omega_1 a\omega_2$ и $n_a(\omega_1) = n_b(\omega_1)$, $n_a(\omega_2) = n_b(\omega_2)$. Този случай се разглежда аналогично.

Преминаваме към доказателството на другата посока, т.е. $L(G) \subseteq L$. Тук с индукция по дължината на извода l ще докажем, че $S \overset{l}{\to} \omega$, то $\omega \in M$, където

$$M = \{\omega \in \{a, b, S\}^\star \mid n_a(\omega) = n_b(\omega)\}.$$

За $l = 0$ е ясно, че $S \overset{0}{\to}^\star S$. За $l = k + 1$, то $S \overset{k}{\to}^\star \alpha \to \omega$. От **И.П.** имаме, че $\alpha \in M$. Нека ω се получава от α с прилагане на правилото $C \to \gamma$. Разглеждаме всички варианти за думата $\alpha \in M$ и за правилото $C \to \gamma$ в граматиката G за да докажем, че $\omega \in M$. Удобно е да представим всички случаи в таблица.

α	$C \to \gamma$	ω
$\in M$	$S \to aSbS$	$\in M$
$\in M$	$S \to bSaS$	$\in M$
$\in M$	$S \to \varepsilon$	$\in M$

Във всички случаи лесно се установява, че $\omega \in M$. Така за всяка дума $\omega \in L(G)$ следва, че

$$\omega \in \Sigma^\star \cap M = L.$$

□

Задача 3.6. Докажете, че следните езици са безконтекстни.

а) $L = \{ww^R \mid w \in \{a, b\}^\star\}$; $S \to aSa \mid bSb \mid \varepsilon$

б) $L = \{w \in \{a, b\}^\star \mid w = w^R\}$; $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$

в) $L = \{a^n b^{2m} c^n \mid m, n \in \mathbb{N}\}$;

г) $L = \{a^n b^m c^m d^n \mid m, n \in \mathbb{N}\}$;

д) $L = \{a^n b^{2k} \mid n, k \in \mathbb{N} \ \&\ n \neq k\}$;

е) $L = \{a^n b^k \mid n > k\}$; $S \to aSb|aS|a$

ж) $L = \{a^n b^k \mid n \geq 2k\}$;

з) $L = \{a^n b^m c^{n+m} \mid n, m \in \mathbb{N}\}$; $S \to aSc|B,\ B \to bBc|\varepsilon$

и) $L = \{a^n b^k c^m \mid n + k \geq m\}$; $S \to aSc|aS|B,\ B \to bBc|bB|\varepsilon$

к) $L = \{a^n b^k c^m \mid n + k \geq m + 1\}$; $S \to aSc|aS|aB|bB,$ $B \to bBc|bB|\varepsilon$

л) $L = \{a^n b^k c^m \mid n + k \geq m + 2\}$;

м) $L = \{a^n b^k c^m \mid n + k + 1 \geq m\}$; $S \to aSc|aS|B|Bc,$ $B \to bBc|bB|\varepsilon$

н) $L = \{a^n b^k c^m \mid n + k + 2 \geq m\}$;

о) $L = \{a^n b^k c^m \mid n + k \leq m\}$;

п) $L = \{a^n b^k c^m \mid n + k \leq m + 1\}$;

р) $L = \{a^n b^m c^k \mid n, m, k$ не са страни на триъгълник$\}$. Обединение на три езика

с) $L = \{a, b\}^\star \setminus \{a^{2n} b^n \mid n \in \mathbb{N}\}$;

т) $L = \{\alpha \in \{a, b\}^\star \mid n_a(\alpha) = n_b(\alpha) + 1\}$; $S \to EaE,\ E \to aEbE|bEaE|\varepsilon$

у) $L = \{\alpha \in \{a, b\}^\star \mid n_a(\alpha) \geq n_b(\alpha)\}$; $S \to E|SaS,$ $E \to aEbE|bEaE|\varepsilon$

ф) $L = \{\alpha \in \{a, b\}^\star \mid n_a(\alpha) > n_b(\alpha)\}$;

х) $L = \{\omega_1 a \omega_2 b \mid \omega_1, \omega_2 \in \{a, b\}^\star \ \&\ |\omega_1| = |\omega_2|\}$;

ц) $L = \{\alpha c \beta \mid \alpha, \beta \in \{a, b\}^\star \ \&\ \alpha^R$ е поддума на $\beta\}$.

Задача 3.7. Да разгледаме граматиката $G = \langle V, \Sigma, R, S\rangle$, където $V = \{S, A, B\}$, $\Sigma = \{a, b\}$, а правилата R са

$$S \to AA|B, A \to B|bb, B \to aa|aB.$$

Да се намери езика на тази граматика и да се докаже, че граматиката разпознава точно този език.

3.2 Езици, които не са безконтекстни

Лема 3.1 (за покачването (безконтекстни езици))**.** За всеки безконтекстен език L съществува $p > 0$, такова че ако $\alpha \in L, |\alpha| \geq p$, то съществува разбиване на думата на пет части, $\alpha = xyuvw$, за което е изпълнено: (стр. 123 от [Sip97]; стр. 125 от [HU79])

1) $|yv| \geq 1$,

2) $|yuv| \leq p$, и

3) $(\forall i \geq 0)[xy^i uv^i w \in L]$.

Док. Нека G е граматиката за езика L. Нека За простота, можем да си мислим, че G е в НФЧ. Тогава $b = 2$.

$$b = \max\{|\beta| \mid A \to_G \beta\}.$$

Можем да приемем, че $b \geq 2$. Това означава, че във всяко дърво на извод, всеки възел има не повече от b наследника. Нека $p = b^{|V|} + 1$. Ще покажем, че p е константа на покачването за граматиката G. Това означава, че всяка дума с дължина поне p в езика L има дърво на извод с височина поне $|V|+1$. Възлите във вътрешността на дървото са променливи, а листата са букви или ε

Нека $|\alpha| \geq p$ и T е дърво на извода за думата α. Понеже думата α може да има много дървета на извод, нека T да бъде **с минимален брой възли**. От направените по-горе разсъждения е ясно, че височината на T е поне $|V|+1$, Следователно, по най-дългия път π в T имаме поне $|V|+2$ възела, от които поне $|V|+1$ са променливи, защото само листата могат да не са променливи. Да разгледаме последните $|V|+1$ променливи по пътя π. От принципа на Дирихле следва, че измежду тези $|V|+1$ променливи има поне една повтаряща се. Нека R да бъде една такава променлива. Последните две повтаряния на R разделят думата α на пет части. Нека $\alpha = xyuvw$.

1) $|yv| \geq 1$, защото ако допуснем, че $|yv| = 0$, то ще достигнем до противоречие с минималността на T.

2) $|yuv| \leq p$, защото сме избрали най-долното R.

3) $xy^iuv^iw \in L$, защото можем да заменим поддървото с корен последното R за поддървото с корен предпоследното R. В случая $i = 0$, правим обратното.

□

Следствие 3.1 (Контрапозиция на лемата за покачването)**.** Нека L е произволен **безкраен** език. Нека също така е изпълнено, че за всяко естествено число $p \geq 1$ можем да намерим дума $\alpha \in L$, $|\alpha| \geq p$, такава че за всяко разбиване на думата на пет части, $\alpha = xyuvw$, със свойствата $|yv| \geq 1$ и $|yuv| \leq p$, е изпълнено, че $(\exists i)[xy^iuv^iw \notin L]$. Тогава L **не** е безконтекстен език.

Ако L е краен език, то е ясно, че L е безконтекстен.

✍ Докажете! Аналогично е на *Следствие 2.2*

Следствие 3.2. Нека G е безконтекстна граматика и p е константата на покачването за G, $L = \mathcal{L}(G)$. Тогава $|L| = \infty$ точно тогава, когато съществува $\alpha \in L$, за която $p \leq |\alpha| < 2p$.

✍ Докажете!

Пример 3.1. Езикът $L = \{a^nb^nc^n \mid n \in \mathbb{N}\}$ не е безконтекстен.

Док. За да докажем, че L не е безконтекстен, прилагаме *Следствие 3.1* по следния начин:

- Разглеждаме произволна константа $p \geq 1$.
- Избираме дума $\alpha \in L$, $|\alpha| \geq p$. В случая, нека $\alpha = a^pb^pc^p$.
- Разглеждаме произволно разбиване $xyuvw = \alpha$, за което $|xyv| \leq p$ и $1 \leq |yv|$.
- Трябва да изберем i, за което $xy^iuv^iw \notin L$. Знаем, че поне едно от y и v не е празната дума. Имаме няколко случая за y и v.
  - y и v са думи съставени от една буква. В този случай получаваме, че xy^2uv^2w има различен брой букви a, b и c.
  - y или v е съставена от две букви. Тогава е възможно да се окаже, че xy^2uv^2w да има равен брой a, b и c, но тогава редът на буквите е нарушен.
  - понеже $|yuv| \leq p$, то не е възможно в y или v да се срещат и трите букви.

Оказа се, че във всички възможни случаи за y и v, $xy^2uv^2w \notin L$.

Така от *Следствие 3.1* следва, че езикът L не е безконтекстен. □

Пример 3.2. Приложете лемата за покачването за да докажете, че езикът L не е безконтекстен, където:

а) $L = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$;

б) $L = \{\beta\beta \mid \beta \in \{a, b\}^\star\}$;

в) $L = \{a^{n^2} \mid n \in \mathbb{N}\}$.

Док.

а) Да фиксираме думата $\alpha = a^p b^p c^p$ и да разгледаме едно произволно нейно разбиване, $\alpha = xyuvw$, за което $|yuv| \leq p$ и $1 \leq |yv|$. Знаем, че поне една от y и v не е празната дума.

- y и v са съставени от една буква. Имаме три случая.
  i) a не се среща в y и v. Тогава xy^0vu^0w съдържа повече a от b или c.
  ii) b не се среща в y и v. Ако a се среща в y или v, тогава xy^2uv^2w съдържа повече a от b Ако c се среща в y или v, тогава xy^0uv^0w съдържа по-малко c от b.
  iii) c не се среща в y и v. Тогава xy^2uv^2w съдържа повече a или b от c.
- y или v е съставена от две букви. Тук разглеждаме xy^2uv^2w и съобразяваме, че редът на буквите е нарушен.

б) Разгледайте $\alpha = a^p b^p a^p b^p$, т.е. $\beta = a^p b^p$ и $\alpha = \beta\beta$. Нека $xyuvw = \alpha$ е произволно разбиване на α, за което е изпълнено, че $|yuv| \leq p$ и $1 \leq |yv|$.

Защо $\alpha = a^p b a^p b$ не е добър кандидат?

- Ако yuv е в първата част на думата, то $xy^0uv^0w = a^i b^j a^p b^p \notin L$. Аналогично ако yuv е във втората част на думата.
- Ако yuv е в двете части на думата, то $xy^0uv^0w = a^p b^i a^j b^p \notin L$.

в) Решава се аналогично както за регулярни езици.

□

Теорема 3.1. Безконтекстните езици **не** са затворени относно сечение и допълнение.

Док. Да разгледаме езика

$$L_0 = \{a^n b^n c^n \mid n \in \mathbb{N}\},$$

за който вече знаем от Пример 3.1, че не е безконтекстен. Да вземем също така и безконтекстните езици

✍ Защо са безконтекстни?

$$L_1 = \{a^n b^n c^m \mid n, m \in \mathbb{N}\},\ L_2 = \{a^m b^n c^n \mid n, m \in \mathbb{N}\},$$

- Понеже $L_0 = L_1 \cap L_2$, то заключаваме, че безконтекстните езици не са затворени относно операцията сечение.

- Да допуснем, че безконтекстните езици са затворени относно операцията допълнение. Тогава $\overline{L}_1$ и $\overline{L}_2$ са безконтекстни. Знаем, че безконтекстните езици са затворени относно обединение. Следователно, езикът $L_3 = \overline{L}_1 \cup \overline{L}_2$ също е безконтекстен. Ние допуснахме, че безконтекстните са затворени относно допълнение, следователно $\overline{L}_3$ също е безконтекстен. Но тогава получаваме, че езикът

  Озн. $\overline{L} = \Sigma^\star \setminus L$

  $$L_0 = L_1 \cap L_2 = \overline{\overline{L}_1 \cup \overline{L}_2} = \overline{L}_3$$

  е безконтекстен, което е противоречие.

□

3.3 Алгоритми

3.3.1 Опростяване на безконтекстни граматики

Премахване на безполезните променливи

Нека е дадена безконтекстната граматика $G = \langle V, \Sigma, R, S\rangle$. Една променлива A се нарича **полезна**, ако съществува извод от следния вид:

[HU79] стр. 88

$$S \to^\star \alpha A \beta \to^\star \gamma,$$

където $\gamma \in \Sigma^\star$, а $\alpha, \beta \in (V \cup \Sigma)^\star$. Това означава, че една променлива е полезна, ако участва в извода на някоя дума в езика на граматиката. Една променлива се нарича **безполезна**, ако не е полезна. Целта ни е да получим еквивалентна граматика G' без безполезни променливи. Ще решим задачата като разгледаме две леми.

Лема 3.2. Нека е дадена безконтекстната граматика $G = \langle V, \Sigma, R, S\rangle$ и $\mathcal{L}(G) \neq \emptyset$. Съществува алгоритъм, който намира граматика $G' = \langle V', \Sigma, S, R'\rangle$, за която $\mathcal{L}(G) = \mathcal{L}(G')$, и за всяка променлива $A' \in V'$, съществува дума $\alpha \in \Sigma^\star$, за която $A' \to^\star \alpha$.

Док. Да разгледаме следната проста итеративна процедура.

Алгоритъм 1 Намираме $V' = \{A \in V \mid (\exists \alpha \in \Sigma^\star)[A \to^\star \alpha]\}$

```
1: V' := ∅
2: V'' := {A ∈ V | (∃α ∈ Σ*)[A → α]}
3: while V' ≠ V'' do
4:     V' := V''
5:     V'' := V' ∪ {A ∈ V | (∃α ∈ (Σ ∪ V')*)[A → α]}
6: return V'
```

Трябва да докажем, че във V' са точно полезните променливи за G. Очевидно е, че ако $A \in V'$, то A е полезна променлива. За другата посока, с индукция по дължината на извода се доказва, че ако $A \to^\star_G \omega$, то $A \in V'$.

✍ Докажете!

Правилата на G' са всички правила на G, в които участват променливи от V' и букви от Σ. □

Лема 3.3. Съществува алгоритъм, който по дадена безконтекстна граматика $G = \langle V, \Sigma, R, S\rangle$, намира $G' = \langle V', \Sigma', S, R'\rangle$, $\mathcal{L}(G') = \mathcal{L}(G)$, със свойството, че за всяко $x \in V' \cup \Sigma'$ съществуват $\alpha, \beta \in (V' \cup \Sigma')^\star$, за които $S \to^\star \alpha x \beta$, т.е. всяка променлива или буква в G' е достижима от началната променлива S.

Док. Намираме V' и Σ' итеративно, като в началото $V' = \{S\}$, $\Sigma' = \emptyset$. Ако $A \in V'$ и имаме правила $A \to \alpha_0|\alpha_1|\ldots|\alpha_n$ в G, то за всяко $i = 0, \ldots, n$ добавяме всички променливи на α_i към V' и всички нетерминали на α_i към Σ'. □

Теорема 3.2. Всеки непразен безконтекстен език L се поражда от безконтекстна граматика G без безполезни правила.

Док. Нека е дадена безконтекстна граматика G пораждаща L. Прилагаме върху G първо процедурата от *Лема 3.2* и след това върху резултата прилагаме процедурата от *Лема 3.3*. □

Важна ли е последователността на прилагане?

Премахване на ε-правила

За да премахнем правилата от вида $A \to \varepsilon$, следваме процедурата:

Броят на правилата може да се увеличи експоненциално, защото в най-лошия случай извеждаме всички подмножества на дадено множество от променливи

1) Намираме множеството $E = \{A \in V \mid A \to^\star \varepsilon\}$ по следния начин. Първо, $E := \{A \in V \mid A \to \varepsilon\}$. След това, за всяко правило от вида $B \to X_1 \cdots X_k$, ако всяко $X_i \in E$, то добавяме B към E.

2) Строим множеството от правила R', в което няма правила ε-правила по следния начин. За всяко правило $A \to X_1 \cdots X_k$ в R, добавяме към R' всички правила от вида $A \to \alpha_1 \cdots \alpha_k$, където:

   - ако $X_i \notin E$, то $\alpha_i = X_i$;
   - ако $X_i \in E$, то $\alpha_i = X_i$ или $\alpha_i = \varepsilon$;
   - не всички α_i-та са ε.

Пример 3.3. Нека е дадена граматиката G с правила

$$S \to D, D \to AD|b, A \to AB|BC|a, B \to AA|EC, C \to \varepsilon|CA|a, E \to \varepsilon|aEb.$$

Тогава $E = \{X \in V \mid X \to^\star_G \varepsilon\} = \{A, B, C, E\}$. Това означава, че $\varepsilon \notin \mathcal{L}(G)$. Граматиката G' без ε-правила, за която $\mathcal{L}(G') = \mathcal{L}(G)$ има следните правила $S \to D, D \to AD|D|b, A \to A|B|C|AB|BC|a, B \to A|E|C|AA|EC, C \to C|A|CA|a, E \to aEb|ab$.

Премахване на преименуващи правила

Преименуващите правила са от вида $A \to B$. Нека е дадена граматика $G = \langle V, \Sigma, R, S\rangle$, в която има преименуващи правила. Ще построим еквивалентна граматика G' без преименуващи правила. В началото нека в R' да добавим всички правила от R, които не са преименуващи. След това, за всяка променлива A, за която $A \to^\star_G B$, ако $B \to \alpha$ е правило в R, което не е преименуващо, то добавяме към R' правилото $A \to \alpha$.

Пример 3.4. Нека е дадена граматиката G с правила

$$A \to B|S, B \to C|BC, C \to AB|a|b, S \to B|CC|b.$$

Първо добавяме към R' правилата $B \to BC, C \to AB|a|b, S \to CC|b$.

- Лесно се съобразява, че $A \to^\star_G B, S, C$. Добавяме правилата $A \to BC|AB|a|b|CC$.
- Имаме $B \to^\star_G C$. Добавяме правилата $B \to AB|a|b$.
- Имаме $S \to^\star_G B, C$. Добавяме правилата $S \to BC|AB|a|b$.

Накрая получаваме, че граматиката G' има правила $A \to BC|AB|a|b|CC, B \to AB|a|b|BC, C \to AB|a|b, S \to BC|AB|CC|a|b$.

3.3.2 Нормална Форма на Чомски

Определение 3.3. Една безконтекстна граматика е в *нормална форма на Чомски*, ако всяко правило е от вида

$$A \to BC \text{ и } A \to a,$$

като B, C *не могат* да бъдат променливата за начало S. Освен това, позволяваме правилото $S \to \varepsilon$. [1]

Теорема 3.3. Всеки безконтекстен език L е генериран от контекстно-свободна граматика в нормална форма на Чомски.

Док. Нека имаме контекстно-свободна граматика G, за която $L = L(G)$. Ще построим контекстно-свободна граматика G' в нормална форма на Чомски, $L = L(G')$. Следваме следната процедура:

- Добавяме нов начален символ S_0 и правило $S_0 \to S$.
- Съкращаваме дължината на правилата. Заменяме правилата от вида $A \to u_1 \dots u_n$, $n \geq 3$, $u_i \in V \cup \Sigma$, с правилата — Време $O(n)$

$$A \to u_1 A_1,\ A_1 \to u_2 A_2,\ \dots,\ A_{n-2} \to u_{n-1} u_n.$$

  където A_i са нови променливи.

- За всяка променлива $A \neq S_0$ премахваме правилата от вида $A \to \varepsilon$. — Време $O(n^2)$
  Това правим по следния начин.

  Ако имаме правило от вида $R \to Au$ или $R \to uA$, $u \in V \cup \Sigma$, то добавяме правилото $R \to u$. Например,

  – ако имаме правило $R \to aA$, то добавяме правилото $R \to a$;
  – ако имаме правило $R \to AA$, то добавяме правилото $R \to A$.

  Ако имаме правило от вида $R \to A$, то добавяме правилото $R \to \varepsilon$ само ако променливата R още не е преминала през процедурата за премахване на ε.

- Премахваме преименуващите правила, т.е. правила от вида $A \to B$. Заменяме всяко правило от вида $B \to \beta$ с $A \to \beta$, освен ако $A \to \beta$ е вече премахнато преименуващо правило.

  Време $O(n^2)$
  Памет $O(n^2)$

- За правила от вида $A \to u_1 u_2$, където $u_1, u_2 \in V \cup \Sigma$, заменяме всяка буква u_i с новата променлива U_i и добавяме правилото $U_i \to u_i$. Например, правилото $A \to aB$ се заменя с правилото $A \to XB$ и добавяме правилото $X \to a$, където X е нова променлива.

  Време $O(n)$

□

Теорема 3.4. При дадена безконтекстна граматика G с дължина n, можем да намерим еквивалентна на нея граматика G' в нормална форма на Чомски за време $O(n^2)$, като получената граматика е с дължина $O(n^2)$.

Задача 3.8. Нека е дадена граматиката $G = \langle \{S, A, B, C\}, \{a, b\}, S, R\rangle$. Използвайте обща конструкция, за да премахнете „дългите" правила (т.е. правила с дължина поне 2, които не са в н.ф. на Чомски) от G като при това получите безконтекстна граматика G_1 с език $L(G) = L(G_1)$, където:

а) $R = \{S \to \varepsilon | ab | aAba, A \to aBCb, B \to bbb, C \to aC | aCaC\}\rangle$;

б) $R = \{S \to \varepsilon | ab | baAb, A \to BaBb, B \to b, C \to AbA | aCCa\}$;

в) $R = \{A \to BSB | a, B \to ba | BC, C \to BaSA | a | b, S \to CC | b\}$;

г) $R = \{A \to BAS, B \to CB, C \to ab | ABbS, S \to CC | b\}$;

3.3.3 Проблемът за принадлежност

Теорема 3.5. Съществува *полиномиален* алгоритъм , който проверява дали дадена дума принадлежи на граматиката G.

За дума α, алгоритъмът работи за време $O(|\alpha|^3)$

Можем да приемем, че $G = \langle V, \Sigma, R, S\rangle$ е граматика в нормална форма на Чомски. Нека $\alpha = a_1 a_2 \dots a_n$ е дума, за която искаме да проверим дали $\alpha \in L(G)$.

Това е алгоритъм на Cocke, Younger и Kasami (CYK), който е пример за динамично програмиране (стр. 195 от [Koz97])

[1] На стр. 151 в [PL98] дефиницията е малко по-различна. Там дефинират G да бъде в нормална форма на Чомски ако $R \subseteq V \times (V \cup \Sigma)^2$. В този случай губим езиците $\{\varepsilon\}$ и $\{a\}$, за $a \in \Sigma$.

Алгоритъм 2 Проверка за $\alpha \in L(G)$

```
1:  n := |α|                                          ▷ Вход дума α = a_1 ··· a_n
2:  for all i ∈ [1, n] do
3:      V[i, i] = {A ∈ V | A → a_i}
4:  for all i, j ∈ [1, n] & i ≠ j do
5:      V[i, j] = ∅
6:  for all s ∈ [1, n) do                             ▷ Дължина на интервала
7:      for all i ∈ [1, n − s] do                     ▷ Начало на интервала
8:          for all k ∈ [i, i + s) do                 ▷ Разделяне на интервала
9:              if ∃A → BC ∈ R & B ∈ V[i, k] & C ∈ V[k + 1, i + s] then
10:                 V[i, i + s] := V[i, i + s] ∪ {A}
11: if S ∈ V[1, n] then
12:     return True                                   ▷ Има извод на думата от S
13: else
14:     return False
```

Лема 3.4. За дадена граматика в нормална форма на Чомски и дума α, за всяко $0 \leq s < |\alpha|$, след s-тата итерация на алгоритъма (редове 6 - 10), за всяка позиция $i = 1, \ldots, n - s$,

$$V[i, i + s] = \{A \in V \mid A \to_G^\star a_i \ldots a_{i+s}\}.$$

Док. Пълна индукция по s. За $s = 0$ е ясно. (Защо?)

Нека твърдението е вярно за $s < n$. Ще докажем твърдението за $s + 1$, т.е. за всяко $i = 1, \ldots, n - s - 1$,

$$V[i, i + s + 1] = \{A \in V \mid A \to_G^\star a_i \ldots a_{i+s+1}\}.$$

За едната посока, да разгледаме първото правило в извода $A \to_G^\star a_i \cdots a_{i+s+1}$. Понеже граматиката G е в НФЧ, то е от вида $A \to BC$ и тогава съществува $t \in [0, s]$, за което $B \to^\star a_i \cdots a_{i+t}$ и $C \to^\star a_{i+t+1} \cdots a_{i+s+1}$. От **И.П.** получаваме, че $B \in V[i, i + t]$ и $C \in V[i + t + 1, i + s + 1]$. Тогава от ред 10 на алгоритъма е ясно, че $A \in V[i, i + s + 1]$.

За другата посока, нека $A \in V[i, i + s + 1]$. Единствената стъпка на алгоритъма, при която може да сме добавили A към множеството $V[i, i + s+1]$ е ред 10. Тогава имаме, че съществува $k \in [i, i+s]$, за което $B \in V[i, k]$, $C \in V[k + 1, i + s + 1]$, и $A \to BC$ е правило в граматиката G. От **И.П.** имаме, че $B \to_G^\star a_i \cdots a_k$ и $C \to_G^\star a_{k+1} \cdots a_{i+s+1}$. Заключаваме веднага, че $A \to_G^\star a_i \cdots a_{i+s+1}$. □

Пример 3.5. Нека е дадена граматиката G с правила $S \to a|AB|AC, C \to SB|AS, A \to a, B \to b$. Ще приложим CYK алгоритъма за да проверим дали думата $aaabb \in \mathcal{L}(G)$.

- $V[1, 1] = V[2, 2] = V[3, 3] = \{S, A\}$. $V[4, 4] = V[5, 5] = \{B\}$.
- $V[1, 2] = V[2, 3] = \{C\}$. $V[3, 4] = \{S, C\}$. $V[4, 5] = \emptyset$.
- $V[1, 3] = \{S\} \cup \emptyset$. $V[2, 4] = \{S, C\} \cup \emptyset$. $V[3, 5] = \emptyset \cup \{C\}$.
- $V[1, 4] = \{S, C\} \cup \emptyset \cup \emptyset = \{S, C\}$. $V[2, 5] = \{S\} \cup \emptyset \cup \{C\} = \{S, C\}$

- $V[1,5] = \{S, C\} \cup \emptyset \cup \emptyset \cup \{C\} = \{S, C\}$.

Понеже $S \in V[1,5]$, то $aaabb \in \mathcal{L}(G)$.

Теорема 3.6. Съществуват алгоритми, които определят по дадена безконтекстна граматика G дали: [HU79], стр. 137

а) $|\mathcal{L}(G)| = 0$;

б) $|\mathcal{L}(G)| < \infty$;

в) $|\mathcal{L}(G)| = \infty$.

Док. Нека е дадена една безконтекстна граматика G.

($\mathcal{L}(G) = \emptyset$?) Прилагаме алгоритъма за премахване на безполезните променливи. Ако открием, че S е безполезна променлива, то $\mathcal{L}(G) = \emptyset$.

($|\mathcal{L}(G)| < \infty$? **или** $|\mathcal{L}(G)| = \infty$?) Нека да разгледаме граматиката G' в НФЧ *без безполезни* променливи, за която $\mathcal{L}(G) = \mathcal{L}(G')$. От граматиката $G' = \langle V', \Sigma, S, R' \rangle$ строим граф с възли променливите от V' като за $A, B \in V'$ имаме ребро $A \to B$ точно тогава, когато съществува $C \in V'$, за което $A \to BC$ или $A \to CB$ е правило в R'.

Ако в получения граф имаме цикъл, то $|\mathcal{L}(G')| = \infty$. В противен случай, $|\mathcal{L}(G')| < \infty$.

□

3.4 Недетерминирани стекови автомати

Определение 3.4. Недетерминиран стеков автомат е 7-орка от вида На англ. **Push-down automaton**

$$P = \langle Q, \Sigma, \Gamma, \#, s, \Delta, F \rangle,$$

където:

- Q е крайно множество от състояния;
- Σ е крайна входна азбука;
- Γ е крайна стекова азбука;
- $\# \in \Gamma$ е символ за дъно на стека;
- $s \in Q$ е начално състояние;
- $\Delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to \mathscr{P}_{fin}(Q \times \Gamma^\star)$ е функция на преходите; Озн. $\mathscr{P}_{fin}(A)$ - крайните подмножества на A
- $F \subseteq Q$ е множество от заключителни състояния.

Моментно описание (или конфигурация) на изчислението със стеков автомат представлява тройка от вида $(q, \alpha, \gamma) \in Q \times \Sigma^\star \times \Gamma^\star$, т.е. автоматът се намира в състояние q, думата, която остава да се прочете е α, а съдържанието на стека е думата γ. Удобно е да въведем бинарната релация $\vdash_P$ Instanteneous description

над $Q \times \Sigma^\star \times \Gamma^\star$, която ще ни казва как моментното описание на автомата P се променя след изпълнение на една стъпка:

$$(q, x\alpha, Y\gamma) \vdash_P (p, \alpha, \beta\gamma), \text{ ако } \Delta(q, x, Y) \ni (p, \beta),$$

$$(q, \alpha, Y\gamma) \vdash_P (p, \alpha, \beta\gamma), \text{ ако } \Delta(q, \varepsilon, Y) \ni (p, \beta).$$

Рефлексивното и транзитивно затваряне на $\vdash_P$ ще означаваме с $\vdash_P^\star$. Сега вече можем да дадем дефиниция на език, разпознаван от стеков автомат P.

- $\mathcal{L}_F(P)$ е езика, който се разпознава от P **с финално състояние**,

$$\mathcal{L}_F(P) = \{\omega \mid (q_0, \omega, \#) \vdash_P^\star (q, \varepsilon, \alpha) \ \&\ q \in F\}.$$

- $\mathcal{L}_S(P)$ е езика, който се разпознава от P **с празен стек**,

$$\mathcal{L}_S(P) = \{w \mid (q_0, w, \#) \vdash_P^\star (q, \varepsilon, \varepsilon)\}.$$

Пример 3.6. За езика $L = \{a^n b^n \mid n \in \mathbb{N}\}$ съществува стеков автомат P, такъв че $L = \mathcal{L}_S(P)$. Да разгледаме $P = \langle Q, \Sigma, \Gamma, \#, s, \Delta, F\rangle$, където

- $Q = \{q\}$;
- $\Sigma = \{a, b\}$;
- $\Gamma = \{\#, A\}$, където символът $\#$ служи за дъно на стека, а броят на A-тата в стека ще показват колко букви a сме прочели от думата;
- $F = \emptyset$, защото разпознаваме с празен стек, а не с финално състояние;
- $\Delta(q, a, \#) = \{(q, A\#)\}$;
- $\Delta(q, \varepsilon, \#) = \{(q, \varepsilon)\}$;
- $\Delta(q, b, A) = \{(q, \varepsilon)\}$.

Вместо доказателство, да видим как думата a^2b^2 се разпознава от автомата с празен стек:

✍ Докажете, че $L = \mathcal{L}_S(P)$!

$$\begin{aligned}(q, a^2b^2, \#) &\vdash_P (q, ab^2, A\#)\\ &\vdash_P (q, b^2, AA\#)\\ &\vdash_P (q, b, A\#)\\ &\vdash_P (q, \varepsilon, \#)\\ &\vdash_P (q, \varepsilon, \varepsilon).\end{aligned}$$

Пример 3.7. За езика $L = \{\omega\omega^R \mid \omega \in \{a, b\}^\star\}$ съществува стеков автомат P, такъв че $L = \mathcal{L}_S(P)$. Нека $P = \langle Q, \Sigma, \Gamma, \#, s, \Delta, F\rangle$, където:

- $\Delta(q, a, \#) = \{(q, A\#)\}$;
- $\Delta(q, b, \#) = \{(q, B\#)\}$;
- $\Delta(q, a, A) = \{(q, AA), (p, \varepsilon)\}$;

- $\Delta(q, a, B) = \{(q, AB)\}$;
- $\Delta(q, b, B) = \{(q, BB), (p, \varepsilon)\}$;
- $\Delta(q, b, A) = \{(q, BA)\}$;
- $\Delta(p, a, A) = \{(p, \varepsilon)\}$;
- $\Delta(p, b, B) = \{(p, \varepsilon)\}$;
- $\Delta(q, \varepsilon, \#) = \{(q, \varepsilon)\}$;
- $\Delta(p, \varepsilon, \#) = \{(p, \varepsilon)\}$;

Основното наблюдение, което трябва да направим за да разберем конструкцията на автомата е, че всяка дума от вида $\omega\omega^R$ може да се запише като $\omega_1 aa\omega_1^R$ или $\omega_1 bb\omega_1^R$. Да видим защо P разпознава думата $abaaba$ с празен стек. Започваме по следния начин:

$$\begin{aligned}(q, abaaba, \#) &\vdash_P (q, baaba, A\#)\\ &\vdash_P (q, aaba, BA\#)\\ &\vdash_P (q, aba, ABA\#).\end{aligned}$$

Сега можем да направим два избора как да продължим. Състоянието p служи за маркер, което ни казва, че вече сме започнали да четем ω^R. Поради тази причина, продължаваме така:

$$\begin{aligned}(q, aba, ABA\#) &\vdash_P (p, ba, BA\#)\\ &\vdash_P (p, a, A\#)\\ &\vdash_P (p, \varepsilon, \#)\\ &\vdash_P (p, \varepsilon, \varepsilon).\end{aligned}$$

Да проиграем още един пример. Да видим защо думата aba не се извежда от автомата.

$$\begin{aligned}(q, aba, \#) &\vdash_P (q, ba, A\#)\\ &\vdash_P (q, a, BA\#)\\ &\vdash_P (q, \varepsilon, ABA\#).\end{aligned}$$

От последното моментно описание на автомата нямаме нито един преход, следователно думата aba не се разпознава от P с празен стек.

✍ Докажете, че $\mathcal{L}_S(P) = L$!

Теорема 3.7. Нека L е произволен език над азбука Σ. ([HU79], стр. 114)

1) Ако съществува НСА P, за който $L = \mathcal{L}_F(P)$, то съществува НСА P', за който $L = \mathcal{L}_S(P')$.

2) Ако съществува НСА P, за който $L = \mathcal{L}_S(P)$, то съществува НСА P', за който $L = \mathcal{L}_F(P')$.

С други думи, езиците разпознавани от НСА с празен стек са точно езиците разпознавани от НСА с финално състояние.

Док.

1) Нека $L = \mathcal{L}_F(P)$, където $P = \langle Q, \Sigma, \Gamma, \#, s, \Delta, F\rangle$. Ще построим P', така че да симулира P и като отидем във финално състояние ще изпразним стека. Нека

$$P' = \langle Q \cup \{q_e, s'\}, \Sigma, \Gamma \cup \{\$\}, \$, s', \Delta', \emptyset\rangle,$$

където $\$ \notin \Gamma$. Важно е P' да има собствен нов символ за дъно на стека. В противен случай е възможно за някоя дума $\alpha \notin \mathcal{L}_F(P)$ стековият автомат P' да си изчисти стека и така да разпознаем повече думи.

- $\Delta'(s', \varepsilon, \$) = \{(s, \#\$)\}$; - започваме симулацията
- $\Delta'(q, a, X)$ включва множеството $\Delta(q, a, X)$, за всяко $q \in Q$, $a \in \Sigma_\varepsilon$, $X \in \Gamma$; - симулираме P
- $\Delta'(q, \varepsilon, X)$ съдържа също и елемента (q_e, ε), за всяко $q \in F$, $X \in \Gamma \cup \{\$\}$; - ако сме във финално, започваме да чистим стека
- $\Delta'(q_e, \varepsilon, X) = \{(q_e, \varepsilon)\}$, за всяко $X \in \Gamma \cup \{\$\}$; - изчистваме стека
- Δ' няма други правила.

2) Сега имаме $L = \mathcal{L}_S(P)$, където $P = \langle Q, \Sigma, \Gamma, \#, s, \Delta, \emptyset\rangle$. Да положим

$$P' = \langle Q \cup \{s', q_f\}, \Sigma, \Gamma \cup \{\$\}, \Delta', \$, \{q_f\}\rangle.$$

P' ще симулира P като ще внимаваме кога P изчиства символа $\#$. Тогава ще искаме да отидем във финалното състояние q_f.

- $\Delta'(s', \varepsilon, \$) = \{(s, \#\$)\}$; - започваме симулацията
- $\Delta'(q, a, X) = \Delta(q, a, X)$, за всяко $q \in Q$, $a \in \Sigma_\varepsilon$, $X \in \Gamma$; - симулираме P
- $\Delta'(q, \varepsilon, \$) = \{(q_f, \varepsilon)\}$. - щом сме стигнали до \$, значи P е изчистил стека си

□

Задача 3.9. Като използвате стековия автомат от Пример 3.6, дефинирайте автомат P', за който $\mathcal{L}_F(P') = \{a^n b^n \mid n \in \mathbb{N}\}$.

Теорема 3.8. Класът на езиците, които се разпознават от краен стеков автомат съвпада с класа на безконтекстните езици.

Док. Ще разгледаме двете посоки на твърдението поотделно. ([HU79], стр. 117)

1) Нека е дадена безконтекстна граматика $G = \langle V, \Sigma, R, S\rangle$. Нашата цел е да построим стеков автомат P, така че $\mathcal{L}_S(P) = \mathcal{L}(G)$. Нека

$$P = \langle \{q\}, \Sigma, \Sigma \cup V, S, q, \Delta, \emptyset\rangle,$$

където функцията на преходите е:

$$\Delta(q, \varepsilon, A) = \{(q, \alpha) \mid A \to \alpha \text{ е правило в граматиката } G\}$$
$$\Delta(q, a, a) = \{(q, \varepsilon)\}$$

2) Нека имаме $P = \langle Q, \Sigma, \Gamma, \Delta, s, \#, \emptyset \rangle$. Ще дефинираме безконтекстна граматика G, за която $\mathcal{L}_S(P) = \mathcal{L}(G)$. Променливите на граматика са

$$V = \{[q, A, p] \mid q, p \in Q, A \in \Gamma\}.$$

Правилата на G са следните:

- $S \to [s, \#, q]$, за всяко $q \in Q$;
- $[q, A, q_{m+1}] \to a[q_1, B_1, q_2][q_2, B_2, q_3] \dots [q_m, B_m, q_{m+1}]$, където

$$(q_1, B_1 \dots B_m) \in \Delta(q, a, A)$$

и произволни $q, q_1, \dots, q_{m+1} \in Q$, $a \in \Sigma \cup \{\varepsilon\}$.
Да обърнем внимание, че е възможно $m = 0$. Това означава, че $(q_1, \varepsilon) \in \Delta(q, a, A)$ и тогава имаме правилото $[q, A, q_1] \to a$, където $a \in \Sigma \cup \{\varepsilon\}$.

Трябва да докажем, че:

$$[q, A, p] \to_G^\star \alpha \;\Leftrightarrow\; (q, \alpha, A) \vdash_P^\star (p, \varepsilon, \varepsilon).$$

($\Rightarrow$) С пълна индукция по i, ще докажем, че

$$(q, \alpha, A) \vdash_P^i (p, \varepsilon, \varepsilon) \implies [q, A, p] \Rightarrow_G^\star \alpha.$$

Ако $i = 1$, то е лесно, защото $\alpha \in \Sigma \cup \{\varepsilon\}$ и $m = 0$.
Ако $i > 1$, нека $\alpha = a\beta$. Тогава: Възможно е $a = \varepsilon$

$$(q, a\beta, A) \vdash_P (q_1, \beta, B_1 \dots B_n) \vdash_P^{i-1} (p, \varepsilon, \varepsilon).$$

Да разбием думата β на n части, $\beta = \beta_1 \cdots \beta_n$, със свойството, че след като прочетем β_i сме премахнали променливата B_i от върха на стека. Това означава, че :

$$(q_j, \beta_j, B_j) \vdash_P^{l_j} (q_{j+1}, \varepsilon, \varepsilon), \text{ за } j = 1, \dots, n-1,$$
$$(q_n, \beta_n, B_n) \vdash_P^{l_n} (p, \varepsilon, \varepsilon),$$

където $l_1 + l_2 + \dots + l_n = i - 1$. Сега по **И.П.** получаваме:

$$(q_j, \beta_j, B_j) \vdash_P^{l_j} (q_{j+1}, \varepsilon, \varepsilon) \implies [q_j, B_j, q_{j+1}] \to_G^\star \beta_j, \text{ за } j = 1, \dots, n-1,$$
$$(q_n, \beta_n, B_n) \vdash_P^{l_n} (p, \varepsilon, \varepsilon) \implies [q_n, B_n, p] \to_G^\star \beta_n.$$

Обединявайки тези изводи с правилото

$$[q, A, p] \to_G a[q_1, B_1, q_2] \dots [q_n, B_n, p],$$

получаваме извода

$$[q, A, p] \to_G^\star a\beta.$$

($\Leftarrow$) Отново с пълна индукция по i ще докажем, че

$$[q, A, p] \to_G^i \alpha \implies (q, \alpha, A) \vdash_P^\star (p, \varepsilon, \varepsilon).$$

Ако $i = 1$, то имаме $[q, A, p] \to \alpha$, където $\alpha = a$ или $\alpha = \varepsilon$. Ако $i > 1$, то имаме, че $\alpha = a\beta$ и за някое n,

$$[q, A, p] \to_G a[q_1, B_1, q_2][q_2, B_2, q_3] \dots [q_n, B_n, p] \to_G^{i-1} \beta.$$

Отново нека $\beta = \beta_1 \dots \beta_n$, където

$$[q_j, B_j, q_{j+1}] \to_G^{i_j} \beta_j, \text{ за } j = 1, \dots, n-1,$$
$$[q_n, B_n, p] \to_G^{i_n} \beta_n,$$

където $i_1 + i_2 + \dots + i_n = i - 1$. От **И.П.** получаваме, че

$$[q_j, B_j, q_{j+1}] \to_G^{i_j} \beta_j \implies (q_j, \beta_j, B_j) \vdash_P^\star (q_{j+1}, \varepsilon, \varepsilon),\ j = 1, \dots, n-1$$
$$[q_n, B_n, p] \to_G^{i_n} \beta_n \implies (q_n, \beta_n, B_n) \vdash_P^\star (p, \varepsilon, \varepsilon),$$

Обединявайки всичко, което знаем, получаваме:

$$\begin{aligned}(q, a\beta, A) &\vdash_P (q_1, \beta_1 \cdots \beta_n, B_1 \cdots B_n) \\ &\vdash_P^\star (q_2, \beta_2 \cdots \beta_n, B_2 \cdots B_n) \\ &\dots \\ &\vdash_P^\star (q_n, \beta_n, B_n) \\ &\vdash_P^\star (p, \varepsilon, \varepsilon)\end{aligned}$$

□

Пример 3.8. Нека е дадена граматиката G с правила $S \to ASB|\varepsilon, A \to aAa|a, B \to bBb|b$. Ще построим стеков автомат $P = \langle Q, \Sigma, \Gamma, \#, s, \Delta, F\rangle$, такъв че $\mathcal{L}_S(P) = \mathcal{L}(G)$.

- $\Sigma = \{a, b\}$;
- $\Gamma = \{A, S, B, a, b\}$;
- $\# = S$;
- $Q = \{q\}$;
- $F = \emptyset$;
- Дефинираме релацията на преходите, следвайки конструкцията от *Теорема 3.8*:
  - $\Delta(q, \varepsilon, S) = \{\langle q, ASB\rangle, \langle q, \varepsilon\rangle\}$;
  - $\Delta(q, \varepsilon, A) = \{\langle q, aAa\rangle, \langle q, a\rangle\}$;
  - $\Delta(q, \varepsilon, B) = \{\langle q, bBb\rangle, \langle q, b\rangle\}$;
  - $\Delta(q, a, a) = \{\langle q, \varepsilon\rangle\}$;
  - $\Delta(q, b, b) = \{\langle q, \varepsilon\rangle\}$.

Теорема 3.9. Нека L е безконтекстен език и R е регулярен език. Тогава тяхното сечение $L \cap R$ е безконтекстен език. (стр. 144 от [PL98])

Док. Нека имаме стеков автомат

$$\mathcal{M}_1 = \langle Q_1, \Sigma, \Gamma, \#, s_1, \Delta_1, F_1 \rangle, \text{ където } \mathcal{L}_F(\mathcal{M}_1) = L,$$

и краен детерминиран автомат

всъщност няма нужда да е детерминиран

$$\mathcal{M}_2 = \langle Q_2, \Sigma, s_2, \delta_2, F_2 \rangle, \text{ където } \mathcal{L}(\mathcal{M}_2) = R.$$

Ще определим нов стеков автомат $\mathcal{M} = \langle Q, \Sigma, \Gamma, \#, s, \Delta, F \rangle$, където

- $Q = Q_1 \times Q_2$;
- $s = \langle s_1, s_2 \rangle$;
- $F = F_1 \times F_2$;
- Функцията на преходите Δ е дефинирана както следва:
  - Ако $\Delta_1(q_1, a, b) \ni \langle r_1, c \rangle$ и $\delta_2(q_2, a) = r_2$, то

    симулираме едновременно изчислението и на двата автомата

    $$\Delta(\langle q_1, q_2 \rangle, a, b) \ni \langle \langle r_1, r_2 \rangle, c \rangle.$$

  - Ако $\Delta_1(q_1, \varepsilon, b) \ni \langle r_1, c \rangle$, то за всяко $q_2 \in Q_2$,

    празен ход на автомата M_2

    $$\Delta(\langle q_1, q_2 \rangle, \varepsilon, b) \ni \langle \langle r_1, q_2 \rangle, c \rangle.$$

  - Δ не съдържа други преходи;

✍ Докажете, че $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{M}_1) \cap \mathcal{L}(\mathcal{M}_2)$!

□

Пример 3.9. Езикът $L = \{w \in \{a, b, c\}^\star \mid n_a(w) = n_b(w) = n_c(w)\}$ не е безконтекстен.

Док. Да допуснем, че L е безконтекстен език. Тогава

$$L' = L \cap \mathcal{L}(a^\star b^\star c^\star)$$

също е безконтекстен език. Но $L' = \{a^n b^n c^n \mid n \in \mathbb{N}\}$, за който знаем от Пример 3.1, че *не* е безконтекстен. Достигнахме до противоречие. Следователно, L не е безконтекстни език. □

3.5 Допълнителни задачи

Задача 3.10. Докажете, че следните езици са безконтекстни:

а) $L = \{\omega_1 \sharp \omega_2 \mid \omega_1, \omega_2 \in \{a,b\}^\star \ \&\ |\omega_1| = |\omega_2|\}$;

б) $L = \{\omega_1 \sharp \omega_2 \sharp \cdots \sharp \omega_n \mid n \geq 2 \ \&\ \omega_1, \omega_2, \ldots, \omega_n \in \{a,b\}^\star \ \&\ |\omega_1| = |\omega_2|\}$;

в) $L = \{\omega_1 \sharp \omega_2 \sharp \cdots \sharp \omega_n \mid n \geq 2 \ \&\ \omega_1, \ldots, \omega_n \in \{a,b\}^\star \ \&\ (\exists i \neq j)[|\omega_i| = |\omega_j|]\}$;

г) $L = \{\omega_1 \sharp \omega_2 \sharp \cdots \sharp \omega_n \mid n \geq 2 \ \&\ (\forall i \in [1,n])[\omega_i \in \{a,b\}^\star \ \&\ |\omega_i| = |\omega_{n+1-i}|]\}$.

Задача 3.11. Проверете дали следните езици са безконтекстни:

а) $\{a^n b^{2n} c^{3n} \mid n \in \mathbb{N}\}$;

б) $\{a^n b^{2n} c^n \mid n \in \mathbb{N}\}$;

в) $\{a^m b^n \mid m \neq n\}$;

г) $\{a^n b^m c^k \mid n < m < k\}$;

д) $\{a^n b^m c^k \mid k = \min\{n,m\}\}$;

е) $\{a^n b^n c^m \mid m \leq n\}$;

ж) $\{a^n b^m c^k \mid k = n \cdot m\}$;

з) $L^\star$, където $L = \{\alpha\alpha^R \mid \alpha \in \{a,b\}^\star\}$;

и) $\{www \mid w \in \{a,b\}^\star\}$;

к) $\{ww^R \mid w \in \{a,b\}^\star\}$;

л) $\{a^{n^2} b^n \mid n \in \mathbb{N}\}$;

м) $\{a^p \mid p \text{ е просто }\}$;

н) $\{\omega \in \{a,b\}^\star \mid \omega = \omega^R\}$;

о) $\{\omega^n \mid \omega \in \{a,b\}^\star \ \&\ n \in \mathbb{N}\}$;

п) $\{a^{n^3+2n^2} \mid n \in \mathbb{N}\}$;

р) $L = \{wcx \mid w, x \in \{a,b\}^\star \ \&\ w \text{ е подниз на } x\}$;

с) $L = \{x_1 c x_2 c \ldots c x_k \mid k \geq 2 \ \&\ x_i \in \{a,b\}^\star \ \&\ (\exists i,j)[i \neq j \ \&\ x_i = x_j]\}$;

т) $L = \{a^i b^j c^k \mid i,j,k \geq 0 \ \&\ (i = j \vee j = k)\}$;

у) $L = \{\alpha \in \{a,b,c\}^\star \mid n_a(\alpha) > n_b(\alpha) > n_c(\alpha)\}$;

ф) $L = \{a,b\}^\star \setminus \{a^n b^n \mid n \in \mathbb{N}\}$;

х) $L = \{\omega \in \{a,b\}^\star \mid n_a(\omega) = 2n_b(\omega)\}$;

ц) $L = \{a^n b^m c^m a^n \mid m,n \in \mathbb{N} \ \&\ n = m + 42\}$;

ч) $L = \{babaabaaab \cdots ba^{n-1}ba^n b \mid n \geq 1\}$;

ш) $\{a^m b^n c^k \mid m = n \vee n = k \vee m = k\}$;

щ) $\{a^m b^n c^k \mid m \neq n \vee n \neq k \vee m \neq k\}$;

ю) $\{a^m b^n c^k \mid m = n \wedge n = k \wedge m = k\}$;

я) $\{w \in \{a,b,c\}^\star \mid n_a(w) \neq n_b(w) \vee n_a(w) \neq n_c(w) \vee n_b(w) \neq n_c(w)\}$.

Задача 3.12. Докажете, че ако L е безконтекстен език, то $L^R = \{\omega^R \mid \omega \in L\}$ също е безконтекстен.

Задача 3.13. Нека $\Sigma = \{a,b,c,d,f,e\}$. Докажете, че езикът L е безконтекстен, където за думите $\omega \in L$ са изпълнени свойствата:

- за всяко $n \in \mathbb{N}$, след всяко срещане на n последователни a-та следват n последователни b-та, и b-та не се срещат по друг повод в ω, и
- за всяко $m \in \mathbb{N}$, след всяко срещане на m последователни c-та следват m последователни d-та, и d-та не се срещат по друг повод в ω, и
- за всяко $k \in \mathbb{N}$, след всяко срещане на k последователни f-а следват k последователни e-та, и e-та не се срещат по друг повод в ω.

Задача 3.14. Да разгледаме езиците:

$$P = \{\alpha \in \{a,b,c\}^* \mid \alpha \text{ е палиндром с четна дължина}\}$$
$$L = \{\beta b^n \mid n \in \mathbb{N}, \beta \in P^n\}.$$

Да се докаже, че:

а) L не е регулярен;

б) L е безконтекстен.

Задача 3.15. Нека L_1 е произволен регулярен език над азбуката Σ, а L_2 е езика от всички думи палиндроми над Σ. Докажете, че L е безконтекстен език, където:

$$L = \{\alpha_1\alpha_2\cdots\alpha_{3n}\beta_1\cdots\beta_m\gamma_1\cdots\gamma_n \mid \alpha_i, \gamma_j \in L_1, \beta_k \in L_2, m, n \in \mathbb{N}\}.$$

Задача 3.16. Нека $L = \{\omega \in \{a,b\}^\star \mid N_a(\omega) = 2\}$. Да се докаже, че езикът $L' = \{\alpha^n \mid \alpha \in L, n \geq 0\}$ не е безконтекстен.

Задача 3.17. Нека $\Sigma = \{a,b,c\}$ и $L \subseteq \Sigma^\star$ е безконтестен език. Ако имаме дума $\alpha \in \Sigma^\star$, тогава *L-вариант* на α ще наричаме думата, която се получава като в α всяко едно срещане на символа a заменим с (евентуално различна) дума от L. Тогава, ако $M \subseteq \Sigma^*$ е произволен безконтестен език, да се докаже че езикът

$$M' = \{\beta \in \Sigma^\star \mid \beta \text{ е } L\text{-вариант на } \alpha \in M\}$$

също е безконтекстен.

Задача 3.18. Докажете, че всеки безконтекстен език над азбуката $\Sigma = \{a\}$ е регулярен.

Задача 3.19. Да фиксираме азбуката Σ. Нека L е безконтекстен език, а R е регулярен език. Докажете, че езикът $L/R = \{\omega \in \Sigma^\star \mid (\exists u \in R)[\omega u \in L]\}$ е безконтекстен.

Задача 3.20. Нека е дадена граматиката $G = \langle\{a, b\}, \{S, A, B, C\}, S, R\rangle$. Използвайте CYK-алгоритъма, за да проверите дали думата α принадлежи на $\mathcal{L}(G)$, където правилата на граматиката R и думата α са зададени като:

а) $R = \{S \to BA|CA|a, C \to BS|SA, A \to a, B \to b\}$, $\alpha = bbaaa$;

б) $R = \{S \to AB|BC, A \to BA|a, B \to CC|b, C \to AB|a\}$, $\alpha = baaba$;

в) $R = \{S \to AB, A \to AC|a|b, B \to CB|a, C \to a\}$, $\alpha = babaa$;

Задача 3.21. Постройте стеков автомат за езика L над азбуката $\{a, b, \sharp\}$, където

Интересно е също да се направи и безконтекстна граматика за L

$$L = \{\omega_1\sharp\omega_2\sharp\cdots\sharp\omega_{2n} \mid n \in \mathbb{N} \;\&\; \sum_{i=1}^{n}|\omega_{2i}| = \sum_{i=1}^{n}|\omega_{2i-1}|\}.$$

Библиография

[HU79] John E. Hopcroft and Jeffrey D. Ullman, *Introduction to automata theory, languages, and computation*, first ed., Addison-Wesley, 1979.

[Koz97] Dexter Kozen, *Automata and computability*, Springer, 1997.

[PL98] Christos Papadimitriu and Harry Lewis, *Elements of the theory of computation*, Prentice-Hall, 1998.

[RS59] M. O. Rabin and D. Scott, *Finite automata and their decision problems*, IBM Journal of Research and Development **3** (1959), pp. 114 – 125.

[Sip97] Michael Sipser, *Introduction to the theory of computation*, PWS Publishing Company, 1997.

Азбучен указател