Записки по „Езици, автомати, изчислимост”

Стефан Вътев[1]

11 юни 2024 г.

[1]Това е чернова. Възможни са неточности и грешки, а също и несъответсвия с терминологията въведена на лекции. За забележки и коментари: stefanv@fmi.uni-sofia.bg

Съдържание

Глава 1

Увод

1.1 Съждително смятане

На англ. Propositional calculus

Както при езиците за програмиране, всяка логика има свой синтаксис и семантика. Тук ще разгледаме класическата съждителна логика, при която те са сравнително прости.

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.

Това не е формална дефиниция, но за момента е достатъчно.

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	$\neg p$	$p \vee q$	$p \wedge q$	$p \rightarrow q$	$\neg p \vee q$	$p \Leftrightarrow q$	$(p \wedge q) \vee (\neg p \wedge \neg q)$
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 $p \rightarrow q \equiv \neg p \vee q$ и $p \Leftrightarrow q \equiv (\neg p \wedge q) \vee (p \wedge \neg q)$.

Съждителни закони

I) **Закон за идемпотентността**

$$p \wedge p \equiv p$$
$$p \vee p \equiv p$$

II) **Комутативен закон**

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

III) **Асоциативен закон**

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

IV) **Дистрибутивен закон**

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

V) **Закони на де Морган**

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

VI) **Закон за контрапозицията**

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

VII) **Обобщен закон за контрапозицията**

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

VIII) **Закон за изключеното трето**

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

IX) **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 Предикати и квантори

Квантори

Свойствата или отношенията на елементите в произволно множество се наричат **предикати**. Нека да разгледаме един едноместен предикат $P(\cdot)$.

твърдение	Кога е истина?	Кога е неистина?
$\forall x P(x)$	$P(x)$ е вярно за всяко x	съществува x, за което $P(x)$ **не** е вярно
$\exists x P(x)$	съществува x, за което $P(x)$ е вярно	$P(x)$ **не** е вярно за всяко x

(I) **Квантор за общност** $\forall x$. Записът $(\forall x \in A)P(x)$ означава, че за всеки елемент a в A, твърдението $P(a)$ има стойност истина. Например, $(\forall x \in \mathbb{R})[(x+1)^2 = x^2 + 2x + 1]$.

(II) **Квантор за съществуване** $\exists x$. Записът $(\exists x \in A)P(x)$ означава, че съществува елемент a в A, за който твърдението $P(a)$ има стойност истина. Например, $(\exists x \in \mathbb{C})[x^2 = -1]$, но $(\forall x \in \mathbb{R})[x^2 \neq -1]$.

Закони на предикатното смятане

(I) $\neg\forall x P(x) \Leftrightarrow \exists x \neg P(x)$

(II) $\neg\exists x P(x) \Leftrightarrow \forall x \neg P(x)$

(III) $\forall x P(x) \Leftrightarrow \neg\exists x \neg P(x)$

(IV) $\exists x P(x) \Leftrightarrow \neg\forall x \neg P(x)$

(V) $\forall x \forall y P(x,y) \Leftrightarrow \forall y \forall x P(x,y)$

(VI) $\exists x \exists y P(x,y) \Leftrightarrow \exists y \exists x P(x,y)$

(VII) $\exists x \forall y P(x,y) \rightarrow \forall y \exists x P(x,y)$

Закони на Де Морган за квантори			
Твърдение	Еквивалентно твърдение	Кога е истина?	Кога е неистина?
$\neg\exists x P(x)$	$\forall x \neg P(x)$	за всяко x $P(x)$ **не** е вярно	съществува x, за което $P(x)$ е вярно
$\neg\forall x P(x)$	$\exists x \neg P(x)$	съществува x, за което $P(x)$ **не** е вярно	$P(x)$ е вярно за всяко x

Задача 1.1. Да означим с $K(x,y)$ твърдението "x познава y". Изразете като предикатна формула следните твърдения.

$\forall x \exists y K(x,y)$ 1) Всеки познава някого.

$\exists x \forall y K(x,y)$ 2) Някой познава всеки.

$\exists x \forall y K(y,x)$ 3) Някой е познаван от всички.

$\forall x \exists y (K(x,y) \wedge \neg K(y,x))$ 4) Всеки знае някой, който не го познава.

$\exists x \forall y (K(y,x) \rightarrow K(x,y))$ 5) Има такъв, който знае всеки, който го познава.

$(\forall x,y)(K(x,y)\ \&\ K(y,x) \rightarrow \exists z (K(x,z)\ \&\ K(y,z))$ 6) Всеки двама познати имат общ познат.

Пример 1.1. Нека $D \subseteq \mathbb{R}$. Казваме, че $f : D \rightarrow \mathbb{R}$ е *непрекъсната* в точката $x_0 \in D$, ако

$$(\forall \varepsilon > 0)(\exists \delta > 0)(\forall x \in D)(\ |x_0 - x| < \delta \ \rightarrow\ |f(x_0) - f(x)| < \varepsilon\ ).$$

f е прекъсната в x_0 точно тогава, когато f не е непрекъсната в x_0

Да видим какво означава f да бъде *прекъсната* в точката $x_0 \in D$:

$$\begin{aligned}
&\neg(\forall \varepsilon > 0)(\exists \delta > 0)(\forall x \in D)(\ |x_0 - x| < \delta \ \rightarrow\ |f(x_0) - f(x)| < \varepsilon\ ) \equiv \\
&(\exists \varepsilon > 0)\neg(\exists \delta > 0)(\forall x \in D)(\ |x_0 - x| < \delta \ \rightarrow\ |f(x_0) - f(x)| < \varepsilon\ ) \equiv \\
&(\exists \varepsilon > 0)(\forall \delta > 0)\neg(\forall x \in D)(\ |x_0 - x| < \delta \ \rightarrow\ |f(x_0) - f(x)| < \varepsilon\ ) \equiv \\
&(\exists \varepsilon > 0)(\forall \delta > 0)(\exists x \in D)\neg(\ |x_0 - x| < \delta \ \rightarrow\ |f(x_0) - f(x)| < \varepsilon\ ) \equiv \\
&(\exists \varepsilon > 0)(\forall \delta > 0)(\exists x \in D)\neg(\ \neg(|x_0 - x| < \delta) \vee |f(x_0) - f(x)| < \varepsilon\ ) \equiv \\
&(\exists \varepsilon > 0)(\forall \delta > 0)(\exists x \in D)(\ \neg\neg(|x_0 - x| < \delta) \wedge \neg(|f(x_0) - f(x)| < \varepsilon)\ ) \equiv \\
&(\exists \varepsilon > 0)(\forall \delta > 0)(\exists x \in D)(\ |x_0 - x| < \delta \ \wedge\ |f(x_0) - f(x)| \geq \varepsilon\ ).
\end{aligned}$$

1.3 Множества, релации, функции

Основни отношения между множества

За произволни множества A и B, ще казваме, че:

- A е подмножество на B, което ще означаваме като $A \subseteq B$, ако:

$$(\forall x)[x \in A \implies x \in B].$$

- A е равно на B, което ще означаваме като $A = B$, ако:

$$(\forall x)[x \in A \Leftrightarrow x \in B],$$

или

$$A = B \Leftrightarrow A \subseteq B \,\&\, B \subseteq A.$$

Основни операции върху множества

Ще разгледаме няколко операции върху произволни множества A и B.

- **Сечение**

На англ. *intersection*

$$A \cap B = \{x \mid x \in A \wedge x \in B\}.$$

Казано по-формално, $A \cap B$ е множеството, за което е изпълнено, че:

$$(\forall x)[x \in A \cap B \Leftrightarrow (x \in A \wedge x \in B)].$$

Примери:

- $A \cap A = A$, за всяко множество A.
- $A \cap \emptyset = \emptyset$, за всяко множество A.
- $\{1, \emptyset, \{\emptyset\}\} \cap \{\emptyset\} = \{\emptyset\}$.
- $\{1, 2, \{1, 2\}\} \cap \{1, \{1\}\} = \{1\}$.

Макар и $\emptyset$, $\{\emptyset\}$ и $\{1, 2\}$ да са множества, те може да са елементи на други множества.

- **Обединение**

На англ. *union*

$$A \cup B = \{x \mid x \in A \vee x \in B\}.$$

$A \cup B$ е множеството, за което е изпълнено, че:

$$(\forall x)[x \in A \cup B \Leftrightarrow (x \in A \vee x \in B)].$$

Примери:

- $A \cup A = A$, за всяко множество A.
- $A \cup \emptyset = A$, за всяко множество A.
- $\{1, 2, \emptyset\} \cup \{1, 2, \{\emptyset\}\} = \{1, 2, \emptyset, \{\emptyset\}\}$.
- $\{1, 2, \{1, 2\}\} \cup \{1, \{1\}\} = \{1, 2, \{1\}, \{1, 2\}\}$.

- **Разлика**

$$A \setminus B = \{x \mid x \in A \wedge x \notin B\}.$$

$A \setminus B$ е множеството, за което е изпълнено, че:

$$(\forall x)[x \in A \setminus B \Leftrightarrow (x \in A \wedge x \notin B)].$$

Примери:

- $A \setminus A = \emptyset$, за всяко множество A.
- $A \setminus \emptyset = A$, за всяко множество A.
- $\emptyset \setminus A = \emptyset$, за всяко множество A.

- $\{1,2,\emptyset\} \setminus \{1,2,\{\emptyset\}\} = \{\emptyset\}$.
- $\{1,2,\{1,2\}\} \setminus \{1,\{1\}\} = \{2,\{1,2\}\}$.

- **Степенно множество**

$$\mathscr{P}(A) = \{x \mid x \subseteq A\}.$$

На англ. *power set*

$\mathscr{P}(A)$ е множеството, за което е изпълнено, че:

$$(\forall x)[x \in \mathscr{P}(A) \Leftrightarrow (\forall y)[y \in x \to y \in A]].$$

В литературата се среща също така и означението 2^A за степенното множество на A.

Примери:

- $\mathscr{P}(\emptyset) = \{\emptyset\}$.
- $\mathscr{P}(\{\emptyset\}) = \{\emptyset, \{\emptyset\}\}$.
- $\mathscr{P}(\{\emptyset, \{\emptyset\}\}) = \{\emptyset, \{\emptyset\}, \{\{\emptyset\}\}, \{\emptyset, \{\emptyset\}\}\}$.
- $\mathscr{P}(\{1,2\}) = \{\emptyset, \{1\}, \{2\}, \{1,2\}\}$.

Задача 1.2. 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) $(A \cup B) \setminus C = (A \setminus C) \cup (B \setminus C)$;

Закони на Де Морган

з) $C \setminus (A \cup B) = (C \backslash A) \cap (C \backslash B)$ и $C \backslash (A \cap B) = (C \backslash A) \cup (C \backslash B)$

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

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

$X \subseteq A \cup B \stackrel{?}{\Rightarrow} X \subseteq A \vee X \subseteq B$

л) $\mathscr{P}(A \cap B) = \mathscr{P}(A) \cap \mathscr{P}(B)$ и $\mathscr{P}(A \cup B) = \mathscr{P}(A) \cup \mathscr{P}(B)$;

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 опрецията **наредена двойка** $\langle a, b\rangle$. Наредената двойка $\langle a, b\rangle$ има следното характеристичното свойство:

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

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

Norbert Wiener (1914)

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

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

Kazimierz Kuratowski (1921)

2) Определението на Куратовски се приема за „стандартно” в наши дни:

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

Задача 1.3. 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 дадено понятието наредена n-орка, без да се интересуваме от нейната формална дефиниция.

Декартово произведение

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

На англ. cartesian product. 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

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

Задача 1.4. 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) $(A \setminus B) \times C = (A \times C) \setminus (B \times C)$.

Видове функции

Функцията $f : A \to B$ е:

- **инекция**, ако е изпълнено свойството: // 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 е изпълнено свойството: // или f е **върху** B

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

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

Задача 1.5. Докажете, че $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ е биекция, където

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

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

Мощност на множество

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, бинарната релация $\leq$ над $\mathbb{N}$ 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, бинарната релация $\leq$ над $\mathbb{N}$ е транзитивна, защото

$$(\forall x, y, z \in \mathbb{N})[\, x \leq y \,\&\, y \leq z \;\to\; x \leq z \,].$$

III) **симетрична**, ако

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

Например, бинарната релация $=$ над $\mathbb{N}$ 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 1.2. За всяко естествено число $n \geq 2$, дефинираме релацията R_n като

$$\langle x, y\rangle \in R_n \;\Leftrightarrow\; x \equiv y \,(\mathrm{mod}\, n).$$

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:

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

Това е малко объркващо

II) **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.

III) **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 $R^1 = R$

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

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

✍ Проверете, че R^+ е транзитивна релация!

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

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

1.4 Доказателства на твърдения

1.4.1 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 1.6. Докажете, че за всяко $a \in \mathbb{Z}$, ако a^2 е четно, то a е четно.

Доказателство. Ние искаме да докажем твърдението P, където:

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

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

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

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

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

Доказателство. 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 1.6, a е четно число. Нека $a = 2k$, за някое естествено число k. 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 естествено число n. 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. □

1.4.2 Индукция върху естествените числа

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

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

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

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 $P(x+1)$.

Да видим едно приложение на принципа на математическа индукция.

Задача 1.8. Докажете, че за всяко естествено число n,

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

Доказателство. Да разгледаме свойството

$$P(n) \overset{\text{деф}}{=} \sum_{i=0}^{n} 2^i = 2^{n+1} - 1.$$

Ще докажем с индукция по n, че $(\forall n)P(n)$, т.е. ще докажем следния извод:

$$\frac{P(0) \qquad (\forall n)[P(n) \implies P(n+1)]}{(\forall n)P(n)}$$

Това е базата на индукцията.

- Нека първо $n = 0$. Очевидно е, че $P(0)$ е изпълнено, защото

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

$P(n)$ се нарича индукционно предположение, а $P(n+1)$ се нарича индукционна стъпка.

- Да разгледаме сега произволно естествено число n, като приемем, че свойството $P(n)$ е изпълнено. Ще докажем, че $P(n+1)$ също е изпълнено. Но това е лесно защото имаме следната верига от равенства:

$$\begin{aligned}
\sum_{i=0}^{n+1} 2^i &= \sum_{i=0}^{n} 2^i + 2^{n+1} \\
&= 2^{n+1} - 1 + 2^{n+1} && \text{// защото } P(n) \text{ е изпълнено} \\
&= 2.2^{n+1} - 1 \\
&= 2^{1+(n+1)} - 1 \\
&= 2^{n+2} - 1.
\end{aligned}$$

□

1.4.3 Пълна индукция върху $\mathbb{N}$

Доказателство с пълна индукция по $\mathbb{N}$ за свойството P представлява следната схема:

$$\frac{(\forall x \in \mathbb{N})[(\forall y \in \mathbb{N})[y < x \ \rightarrow P(y)] \rightarrow P(x)]}{(\forall x \in \mathbb{N})P(x)}$$

Нека да проверим принципа за пълна индукция. Да допуснем, че принципът не е верен, т.е. за някое свойство P е изпълнено, че

$$(\forall x \in \mathbb{N})[(\forall y \in \mathbb{N})[y < x \ \rightarrow P(y)] \rightarrow P(x)] \ \wedge \ (\exists x \in \mathbb{N})\neg P(x).$$

Да вземем най-малкия елемент n_0, за който $\neg P(n_0)$. От нашето допускане знаем, че такова n_0 съществува. Тогава

$$(\forall y \in \mathbb{N})[y < n_0 \ \rightarrow P(y)]$$

и следователно:

$$\dfrac{(\forall y \in \mathbb{N})[y < n_0 \ \rightarrow P(y)] \qquad \dfrac{(\forall x \in \mathbb{N})[(\forall y \in \mathbb{N})[y < x \ \rightarrow P(y)] \rightarrow P(x)]}{(\forall y \in \mathbb{N})[y < n_0 \ \rightarrow P(y)] \rightarrow P(n_0)} \ {\scriptstyle (x = n_0)}}{P(n_0)}$$

Така достигаме до противоречие, защото получаваме, че $P(n_0) \wedge \neg P(n_0)$.

Сега да видим, че често е полезно да използваме пълната индукция.

Задача 1.9. Докажете, че всяко естествено число $n \geq 2$ може да се запише като произведение на прости числа.

Доказателство. Искаме да докажем, че $(\forall n \geq 2)P(n)$, където $P(n)$ казва, че n може да се запише като произведение на прости числа, т.е.

$$n = p_1^{m_1} p_2^{m_2} \cdots p_k^{m_k},$$

за някои прости числа $p_1, p_2, \ldots, p_k$ и естествени числа $m_1, m_2, \ldots, m_k$.

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

а) За $n = 2$ е ясно, защото 2 е просто число. В този случай $n = p_1^{m_1}$ и $p_1 = 2$ и $m_1 = 1$.

б) Да приемем, че $P(n)$ е изпълнено за някое естествено число $n > 2$.

в) Да разгледаме следващото естествено число $n + 1$. Ако $n + 1$ е просто число, то всичко е ясно. Ако $n + 1$ е съставно, то съществуват естествени числа $n_1, n_2 \geq 2$, за които

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

Тогава, понеже $n_1, n_2 \leq n$, от от **(И.П.)** следва, че $P(n_1)$ и $P(n_2)$, т.е.

$$n_1 = p_1^{\ell_1} \cdots p_k^{\ell_k} \text{ и } n_2 = q_1^{m_1} \cdots q_r^{m_r},$$

където $p_1, \ldots, p_k$ и $q_1, \ldots, q_r$ са прости числа, а $\ell_1, \ldots, \ell_k$ и $m_1, \ldots, m_r$ са естествени числа. Тогава е ясно, че $n + 1$ също е произведение на прости числа.

□

Глава 2

Регулярни езици

2.1 Азбуки, думи, езици

> Language is an app for converting a web of thoughts into a string of words. [Steven Pinker]

Основни понятия

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

  Обикновено ще използваме малки латински букви като 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 малки гръцки букви като $\alpha, \beta, \gamma, \omega$.

- 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 $\emptyset^\star = \{\varepsilon\}$.

- **Език** над азбуката Σ ще наричаме всяко подмножество на $\Sigma^\star$. Например, за азбуката $\Sigma = \{a, b\}$, множоеството от думи $L = \{\alpha \in \{a, b\}^\star \mid \alpha \text{ започва с } a\}$ е пример за език над Σ.

Операции и релации 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 $\alpha^{\mathtt{rev}}$ като обръщането на α по следния начин.

  Например, $reverse^{\mathtt{rev}} = esrever$

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

$$\alpha^{\mathtt{rev}} \stackrel{\text{деф}}{=} (\beta^{\mathtt{rev}})a.$$

- Дефинираме конкатенацията на езиците A и B като

  Обърнете внимание, че:

  $\emptyset \cdot A = A \cdot \emptyset = \emptyset$
  $\{\varepsilon\} \cdot A = A \cdot \{\varepsilon\} = A.$

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

- Сега за един език 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 $A^0 = \{\varepsilon\}$, $A^1 = A$, $A^2 = \{abab, abba, baba, baab\}$.

- За един език A, дефинираме:

  Операцията $\star$ е известна като звезда на Клини.

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

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 $L^0 = \{\varepsilon\}$, $L^1 = \emptyset$ и $L^2 = L^1 \cdot L^1 = \emptyset$. Получаваме, че $L^\star = \{\varepsilon\}$, т.е. *краен* език
- Нека $L = \{0^i \mid i \in \mathbb{N}\} = \{\varepsilon, 0, 00, 000, \ldots\}$. Лесно се вижда, че $L = L^\star$.

Задача 2.1. 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 произволни езици A, B и C,

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

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

г) за произволен език A,
$$A^\star = A^\star \cdot A^\star \text{ и } (A^\star)^\star = A^\star;$$

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

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

Задача 2.2. Докажете, че за всеки две думи α и β е изпълнено:

а) $(\alpha \cdot \beta)^{\mathtt{rev}} = \beta^{\mathtt{rev}} \cdot \alpha^{\mathtt{rev}}$;

б) α е префикс на β точно тогава, когато $\alpha^{\mathtt{rev}}$ е суфикс на $\beta^{\mathtt{rev}}$;

в) $(\alpha^{\mathtt{rev}})^{\mathtt{rev}} = \alpha$;

г) $(\alpha^n)^{\mathtt{rev}} = (\alpha^{\mathtt{rev}})^n$, за всяко $n \geq 0$.

Отрези на дума

Понякога може да е удобно да вземем назаем от `python` нотацията за `slices` на масив и ако думата $\alpha = a_0 a_1 \cdots a_{n-1}$, то нека

$$\begin{aligned}
\alpha\mathtt{[}i\mathtt{]} &\stackrel{\text{деф}}{=} a_i \\
\alpha\mathtt{[}i : j\mathtt{]} &\stackrel{\text{деф}}{=} \begin{cases} a_i \cdots a_{m-1}, & \text{ако } i < j \ \&\ m = \min\{j, |\alpha|\} \\ \varepsilon, & \text{иначе} \end{cases} \\
\alpha\mathtt{[}i :\mathtt{]} &\stackrel{\text{деф}}{=} \alpha\mathtt{[}i : |\alpha|\mathtt{]} \\
\alpha\mathtt{[}: i\mathtt{]} &\stackrel{\text{деф}}{=} \alpha\mathtt{[}0 : i\mathtt{]}
\end{aligned}$$

Например, за $\alpha = abc$, то $\alpha\mathtt{[}1\mathtt{]} = b = \alpha\mathtt{[}1 : 2\mathtt{]}$ и $\alpha\mathtt{[}1 : 3\mathtt{]} = bc = \alpha\mathtt{[}1 : 4\mathtt{]}$, $\alpha\mathtt{[}1 : 1\mathtt{]} = \varepsilon$.

Релации между думи

- Казваме, че думата α е **префикс** на думата β, което ще записваме като $\alpha \preceq \beta$, ако съществува дума γ, такава че $\beta = \alpha \cdot \gamma$. Да обърнем внимание, че позволяваме $\gamma = \varepsilon$. Тогава β е префикс на самата себе си. Ако се ограничим до думи $\gamma \neq \varepsilon$, то ще казваме, че α е *същински* префикс на β, което ще записваме като $\alpha \prec \beta$.

- За един език L, можем да дефинираме езика от префиксите на думите от L, т.е.
$$\mathtt{Pref}(L) = \{\alpha \in \Sigma^\star \mid (\exists \gamma \in \Sigma^\star)[\alpha \cdot \gamma \in L]\}.$$

- α е **суфикс** на β, ако $\beta = \gamma \cdot \alpha$, за някоя дума γ.

- За един език L, можем да дефинираме езика от суфиксите на думите от L, т.е.
$$\mathtt{Suff}(L) = \{\alpha \in \Sigma^\star \mid (\exists \gamma \in \Sigma^\star)[\gamma \cdot \alpha \in L]\}.$$

- Нека имаме азбука $\Sigma = \{0, 1, 2, \ldots, n\}$. Казваме, че една дума α е **лексикографски** по-малка от β, което ще означаваме като $\alpha <_{\mathtt{lex}} \beta$, ако:
$$\alpha \prec \beta \ \vee\ (\exists i < \min\{|\alpha|, |\beta|\})[\ \alpha\mathtt{[}: i\mathtt{]} = \beta\mathtt{[}: i\mathtt{]} \ \&\ \alpha\mathtt{[}i\mathtt{]} < \beta\mathtt{[}i\mathtt{]}\ ].$$

Тук не е съществено, че буквите в азбуката Σ са числа. Можем да дефинираме наредба между елементите на произволна крайна азбука.

2.2 Уравнения от езици

В този раздел ще видим, че операцията звезда на Клини се оказва доста важна в нашите разглеждания. Това ще направим като разгледаме уравнения от вида $X = L \cdot X \cup M$, където X е неизвестна променлива, а L и M са произволни езици. Решение на такова уравнение е език S, за който $S = L \cdot S \cup M$.

Удобно е понякога за език L и думи α и β да пишем $\alpha L \beta$ вместо $\{\alpha\} \cdot L \cdot \{\beta\}$. Така например, можем да запишем

$$X = aX \cup \{\varepsilon\}$$

вместо

$$X = \{a\} \cdot X \cup \{\varepsilon\}.$$

Също така, за два езика L и M можем просто да пишем LM вместо $L \cdot M$.

Пример 2.2. Уравнението $X = \{a\} \cdot X \cup \{\varepsilon\}$ има решение $S = \{a\}^\star$. Съобразете, че това е единственото решение.

Пример 2.3. Уравнението $X = \{a\} \cdot X$ има решение $S = \emptyset$. Съобразете, че това е единственото решение.

Пример 2.4. Уравнението $X = \{a, \varepsilon\} \cdot X \cup \{b\}$ има решение $S = \{a\}^\star \cdot \{b\}$. Друго решение на това уравнение е $S = \{a, b\}^\star$, а също и $S = \{a\}^\star \cdot \{b\}^+$.

Лема 2.1 (Ардън [2]). Нека L и M са произволни езици над азбука Σ. Тогава езикът $L^\star \cdot M$ е *най-малкото решение* на уравнението

$$X = L \cdot X \cup M. \tag{2.1}$$

Ако $\varepsilon \notin L$, то това решение е и *единствено*.

В повечето учебници този резултат отсъства. Ние го смятаме за основен. Все пак може да се намери в някои учебници [19, стр. 100], [17, стр. 53], [14, стр. 60].

Доказателство. Първо, да видим защо $L^\star M$ е решение на уравнението (2.1). Това е лесно. Просто заместваме променливата X с езика $L^\star M$ и получаваме равенствата:

$$\begin{aligned} L^\star M &= L \cdot (L^\star M) \cup M \\ &= L^+ \cdot M \cup \{\varepsilon\} \cdot M \\ &= (L^+ \cup \{\varepsilon\}) \cdot M \\ &= L^\star \cdot M. \end{aligned}$$

Второ, да видим защо $L^\star \cdot M$ е най-малкото решение на уравнението (2.1). За целта, нека приемем, че K е произволно решение, т.е.

$$K = L \cdot K \cup M.$$

Трябва да проверим, че $L^\star \cdot M \subseteq K$. Тук ще използваме представянето

$$L^\star \cdot M = \bigcup_n L^n \cdot M.$$

Ще докажем с индукция по n, че за всяко n е изпълнено включването:

$$L^n \cdot M \subseteq K.$$

- Нека $n = 0$. Понеже $K = L \cdot K \cup M$ е ясно, че $M \subseteq K$ или с други думи, $L^0 \cdot M \subseteq K$.
- Нека сега $n > 0$. От **(И.П.)** имаме, че $L^{n-1} \cdot M \subseteq K$. Тогава, като конкатенираме с L от двете страни, получаваме:

$$\begin{aligned} L \cdot L^{n-1} \cdot M &\subseteq L \cdot K \\ &\subseteq L \cdot K \cup M \\ &= K. \end{aligned}$$

Така заключаваме, че за всяко n, е изпълнено включването $L^n \cdot M \subseteq K$ и следователно $L^\star \cdot M \subseteq K$.

Остана да докажем, че ако $\varepsilon \notin L$, то $L^\star \cdot M$ е *единственото* решение на уравнението (2.1). Ще направим това като докажем, че всяко решение K на уравнението (2.1) е такова, че $K = L^\star \cdot M$. Достатъчно да докажем, че за всяко решение K на (2.1) е изпълнено, че $K \subseteq L^\star \cdot M$, защото обратното включване $L^\star \cdot M \subseteq K$ следва от факта, че $L^\star \cdot M$ е най-малкото решение на (2.1).

Да отбележим, че ако $\varepsilon \in L$, то $\Sigma^\star$ е решение на уравнението (2.1).

Щом трябва да докажем, че $K \subseteq L^\star \cdot M$, то това означава да докажем импликацията

$$(\forall \alpha \in \Sigma^\star)[\, \alpha \in K \implies \alpha \in L^\star \cdot M \,].$$

И така, ще докажем с индукция по n, че

$$(\forall n)(\forall \alpha \in \Sigma^{\leq n})[\, \alpha \in K \implies \alpha \in L^\star \cdot M \,].$$

- Нека $n = 0$ и да вземем дума $\alpha \in \Sigma^{\leq 0}$, което означава, че $\alpha = \varepsilon$. Да приемем, че $\varepsilon \in K$, защото в противен случай импликацията би била автоматично изпълнена за $n = 0$. Щом K е решение, то $\varepsilon \in L \cdot K \cup M$, но понеже $\varepsilon \notin L$, то със сигурност $\varepsilon \in M$. Така получаваме, че

  $$\varepsilon \in K \implies \varepsilon \in L^\star \cdot M,$$

  откъдето веднага следва, че

  $$(\forall \alpha \in \Sigma^{\leq 0})[\, \alpha \in K \implies \alpha \in L^\star \cdot M \,].$$

- Нека сега $n > 0$, като приемем, че следното имаме индукционно предположение:

  $$(\forall \alpha \in \Sigma^{\leq n-1})[\, \alpha \in K \implies \alpha \in L^\star \cdot M \,].$$

  Ще докажем, че

  $$(\forall \alpha \in \Sigma^{\leq n})[\, \alpha \in K \implies \alpha \in L^\star \cdot M \,].$$

  За целта, да вземем произволна дума $\alpha \in \Sigma^{\leq n}$. Ако $|\alpha| < n$, то всъщност $\alpha \in \Sigma^{\leq n-1}$ и от **(И.П.)** веднага следва, че импликацията е изпълнена за α.

  Нека $|\alpha| = n$. Ако $\alpha \notin K$, то отново импликацията е изпълнена по тривиални причини. Интересният случай е когато $|\alpha| = n$ и $\alpha \in K$. Понеже K е решение на (2.1), то $\alpha \in L \cdot K \cup M$. Сега имаме два случая.

  – Ако $\alpha \in M$, то всичко е ясно, защото тогава $\alpha \in L^\star \cdot M$.
  – Ако $\alpha \in L \cdot K$, то $\alpha = \alpha_1 \cdot \alpha_2$, за някои думи α_1 и α_2, за които $\alpha_1 \in L$ и $\alpha_2 \in K$. Понеже $\varepsilon \notin L$, то $|\alpha_1| \geq 1$ и следователно $|\alpha_2| < |\alpha| = n$. Това означава, че от **(И.П.)** получаваме, че $\alpha_2 \in L^\star \cdot M$, откъдето заключаваме, че следното:

    $$\alpha = \alpha_1 \cdot \alpha_2 \in L \cdot (L^\star \cdot M) \subseteq L^\star \cdot M.$$

Така доказахме включването $K \subseteq L^\star \cdot M$. □

Задача 2.3. Нека L и M са произволни езици. Тогава езикът $M \cdot L^\star$ е *най-малкото* решение на уравнението

$$X = X \cdot L \cup M.$$

Ако $\varepsilon \notin L$, то това решение е и *единствено*.

Задача 2.4. Нека L_1, L_2 и M са прозволни езици. Тогава езикът $L_1^\star \cdot M \cdot L_2^\star$ е *най-малкото* решение на уравнението [17, стр. 54].

$$X = L_1 \cdot X \cup X \cdot L_2 \cup M.$$

Ако $\varepsilon \notin L_1$ и $\varepsilon \notin L_2$, то това решение е и *единствено*.

Задача 2.5. Нека L и M са произволни езици над азбуката Σ и $\varepsilon \in L$. Докажете, че за всеки език $N \supseteq M$ над Σ, то $L^\star \cdot N$ е решение на уравнението

$$X = L \cdot X \cup M.$$

2.3 Системи от уравнения

В този раздел ще опишем един алгебричен поглед към регулярните езици, с чиято помощ можем да построим регулярен израз по даден краен автомат.

Изглежда естествено да се запитаме дали можем да обобщим лемата на Ардън до системи от уравнения.

Теорема 2.1. Да разгледаме системата:

$$\begin{array}{|rcl} X_1 &=& L_{1,1} \cdot X_1 \cup \cdots \cup L_{1,n} \cdot X_n \cup M_1 \\ &\vdots& \\ X_n &=& L_{n,1} \cdot X_1 \cup \cdots \cup L_{n,n} \cdot X_n \cup M_n, \end{array}$$

където $L_{i,j}$ и M_i са произволни езици, за $i, j = 1, \ldots, n$, като $\varepsilon \notin L_{i,j}$, за всяко $i, j = 1, \ldots, n$. Тогава тази система има *единствено* решение.

Системата може да се запише по-компактно така:

$$\begin{array}{|l} X_1 = \bigcup\limits_{i=1}^{n} L_{1,i} \cdot X_i \cup M_1 \\ \quad\vdots \\ X_n = \bigcup\limits_{i=1}^{n} L_{n,i} \cdot X_i \cup M_n. \end{array}$$

Кратко описание на този подход може да се намери и в [19, стр. 134], [14, стр. 66].

Доказателство. Индукция по броят на променливите n в системата.

Нека $n = 1$. Тогава системата представлява просто това:

$$X_1 = L_{1,1} \cdot X_1 \cup M_1.$$

Щом $\varepsilon \notin L_{1,1}$, от лемата на Ардън веднага следва, че тази система има *единствено* решение

$$S_1 = L_{1,1}^{\star} \cdot M_1.$$

Нека сега $n > 1$. Да разгледаме само последния ред на системата във вида, който ни трябва за да приложим лемата на Ардън.

$$X_n = L_{n,n} \cdot X_n \cup \bigcup_{i=1}^{n-1} L_{n,i} \cdot X_i \cup M_n.$$

Понеже $\varepsilon \notin L_{n,n}$, то за всеки избор на езици, с които да заменим променливите $X_1, \ldots, X_{n-1}$, горното уравнение ще има *единствено* решение и то е следното:

$$X_n = L_{n,n}^{\star} \cdot (\bigcup_{i=1}^{n-1} L_{n,i} \cdot X_i \cup M_n), \tag{2.2}$$

което, като разкрием скобите, придобива следния вид:

$$X_n = \bigcup_{i=1}^{n-1} L_{n,n}^{\star} \cdot L_{n,i} \cdot X_i \cup L_{n,n}^{\star} \cdot M_n.$$

Сега заместваме в първите $n-1$ реда на системата всяко срещане на X_n с неговото единствено решение, зависещо от $X_1, \ldots, X_{n-1}$. Получаваме следната система:

$$\begin{array}{|rcl} X_1 &=& \bigcup_{i=1}^{n-1}(L_{1,i} \cup L_{1,n} \cdot L_{n,n}^{\star} \cdot L_{n,i}) \cdot X_i \cup L_{1,n} \cdot L_{n,n}^{\star} \cdot M_n \cup M_1 \\ &\vdots& \\ X_{n-1} &=& \bigcup_{i=1}^{n-1}(L_{n-1,i} \cup L_{n-1,n} \cdot L_{n,n}^{\star} \cdot L_{n,i}) \cdot X_i \cup L_{n-1,n} \cdot L_{n,n}^{\star} \cdot M_n \cup M_{n-1}. \end{array}$$

Това е вече система с $n-1$ уравнения и $n-1$ неизвестни. От условието знаем, че за всяко $k = 1, \ldots, n$ и всяко $i = 1, \ldots, n$, празната дума $\varepsilon \notin L_{k,i}$. Тогава е ясно, че за всяко $k = 1, \ldots, n-1$ и всяко $i = 1, \ldots, n-1$,

$$\varepsilon \notin L_{k,i} \cup L_{k,n} \cdot L_{n,n}^{\star} \cdot L_{n,i}.$$

От **(И.П.)** тази система има *единствено* решение съставено от езиците

$$S_1, \ldots, S_{n-1}.$$

Тогава уравнението (2.2) за X_n също има единствено решение

$$S_n = \bigcup_{i=1}^{n-1} L_{n,n}^{\star} \cdot L_{n,i} \cdot S_i \cup L_{n,n}^{\star} \cdot M_n.$$

Така получаваме, че $S_1, \ldots, S_n$ е единственото решение на първоначалната система. □

От доказателството на *Теорема 2.1* веднага се вижда, че ако се ограничим само до регулярни езици, то единственото решение на системата ще е също съставено от регулярни езици.

Следствие 2.1. Да разгледаме системата:

$$\left|\begin{array}{rcl} X_1 & = & L_{1,1} \cdot X_1 \cup L_{1,2} \cdot X_2 \cup \cdots \cup L_{1,n} \cdot X_n \cup M_1 \\ & \vdots & \\ X_n & = & L_{n,1} \cdot X_1 \cup L_{n,2} \cdot X_2 \cup \cdots \cup L_{n,n} \cdot X_n \cup M_n, \end{array}\right.$$

където $L_{i,j}$ и M_i са *регулярни* езици, за $i, j = 1, \ldots, n$, като $\varepsilon \notin L_{i,j}$, за всяко $i, j = 1, \ldots, n$. Тогава тази система има *единствено* решение, което е съставено от *регулярни* езици.

2.4 Регулярни изрази

Да фиксираме една непразна азбука Σ. **Регулярните изрази** $\mathbf{r}$ могат да се опишат със следната граматика:

$$\mathbf{r} ::= \boldsymbol{\emptyset} \mid \boldsymbol{\varepsilon} \mid \mathbf{a} \mid (\boldsymbol{r} \cdot \boldsymbol{r}) \mid (\boldsymbol{r} + \boldsymbol{r}) \mid \boldsymbol{r}^\star.$$

Това е пример за индуктивна дефиниция

Регулярните изрази могат да се опишат и по следния начин:

- Символите $\boldsymbol{\emptyset}$, $\boldsymbol{\varepsilon}$ са регулярни изрази;
- за всяка буква $a \in \Sigma$, символът $\boldsymbol{a}$ е регулярен израз;

В литературата също се среща записът $(\boldsymbol{r}_1 \mid \boldsymbol{r}_2)$ вместо $(\boldsymbol{r}_1 + \boldsymbol{r}_2)$

- ако $\boldsymbol{r_1}$ и $\boldsymbol{r_2}$ са регулярни изрази, то думите $(\boldsymbol{r}_1 \cdot \boldsymbol{r}_2)$, $(\boldsymbol{r}_1 + \boldsymbol{r}_2)$ и $\boldsymbol{r}_1^\star$ също са регулярни изрази;
- Всеки регулярен израз е получен чрез крайно прилагане на горните правила.

Това е друг пример за индуктивна (рекурсивна) дефиниция.

Сега ще дефинираме езиците, които се описват с регулярни изрази. Тези езици се наричат **регулярни**. Това ще направим следвайки индуктивната дефиниция на регулярните изрази, т.е. за всеки регулярен израз $\boldsymbol{r}$ ще определим език $\mathcal{L}[\![\boldsymbol{r}]\!]$.

- $\emptyset$ е регулярен език, който се описва от регулярния израз $\boldsymbol{\emptyset}$. Означаваме

$$\mathcal{L}[\![\boldsymbol{\emptyset}]\!] \stackrel{\text{деф}}{=} \emptyset;$$

- $\{\varepsilon\}$ е регулярен език, който се описва от регулярния израз $\boldsymbol{\varepsilon}$. Означаваме

$$\mathcal{L}[\![\boldsymbol{\varepsilon}]\!] \stackrel{\text{деф}}{=} \{\varepsilon\};$$

- за всяка буква $a \in \Sigma$, $\{a\}$ е регулярен език, който се описва от регулярния израз $\boldsymbol{a}$. Означаваме

$$\mathcal{L}[\![\boldsymbol{a}]\!] \stackrel{\text{деф}}{=} \{a\};$$

Понякога, когато приоритетът на операциите е ясен, ще изпускаме да пишем скоби.

- Нека L_1 и L_2 са регулярни езици, т.е. съществуват регулярни изрази $\boldsymbol{r}_1$ и $\boldsymbol{r}_2$, за които $\mathcal{L}[\![\boldsymbol{r_1}]\!] = L_1$ и $\mathcal{L}[\![\boldsymbol{r_2}]\!] = L_2$. Тогава:
  - $L_1 \cup L_2$ е регулярен език, който се описва с регулярния израз $(\boldsymbol{r}_1 + \boldsymbol{r}_2)$. Тогава

$$\mathcal{L}[\![(\boldsymbol{r_1} + \boldsymbol{r_2})]\!] \stackrel{\text{деф}}{=} \mathcal{L}[\![\boldsymbol{r_1}]\!] \cup \mathcal{L}[\![\boldsymbol{r_2}]\!].$$

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

  - $L_1 \cdot L_2$ е регулярен език, който се описва с регулярния израз $(\boldsymbol{r}_1 \cdot \boldsymbol{r}_2)$. Тогава

$$\mathcal{L}[\![(\boldsymbol{r_1} \cdot \boldsymbol{r_2})]\!] \stackrel{\text{деф}}{=} \mathcal{L}[\![\boldsymbol{r_1}]\!] \cdot \mathcal{L}[\![\boldsymbol{r_2}]\!].$$

Звезда на Клини

  - $L_1^\star$ е регулярен език, който се описва с регулярния израз $\boldsymbol{r_1^\star}$. Тогава

$$\mathcal{L}[\![\boldsymbol{r_1^\star}]\!] \stackrel{\text{деф}}{=} \mathcal{L}[\![\boldsymbol{r_1}]\!]^\star.$$

Забележка. Ние знаем, че:

- Всеки регулярен израз представлява крайна дума над крайна азбука. Това означава, че множеството от всички регулярни изрази е изброимо безкрайно. Оттук следва, че всички регулярни езици образуват изброимо безкрайно множество.
- Понеже Σ е крайна азбука, то $\Sigma^\star$ е изброимо безкрайно множество;
- Един език над азбуката Σ представлява елемент на $\mathscr{P}(\Sigma^\star)$. Това означава, че всички езици над азбуката Σ представляват неизброимо безкрайно множество.

От всичко това следва, че има езици, които не са регулярни. По-нататък ще видим примери за такива езици.

Пример 2.5. Нека да построим регулярни изрази за всеки от езиците от *Пример 3.1*.

В [21, стр. 70] е показан алгоритъм, за който по един автомат може да се получи регулярен израз описващ езика на автомата. Ние няма да разглеждаме този алгоритъм.

а) Нека $\boldsymbol{r} \stackrel{\text{деф}}{=} (\boldsymbol{a}+\boldsymbol{b})^\star\boldsymbol{bab}(\boldsymbol{a}+\boldsymbol{b})^\star$. Тогава

$$\mathcal{L}[\![\boldsymbol{r}]\!] = \{\omega \in \{a,b\}^\star \mid \omega \text{ съдържа } bab\}.$$

б) Нека $\boldsymbol{r} \stackrel{\text{деф}}{=} \boldsymbol{b}^\star\boldsymbol{ab}^\star\boldsymbol{a}(\boldsymbol{a}+\boldsymbol{b})^\star$. Тогава

$$\mathcal{L}[\![\boldsymbol{r}]\!] = \{\omega \in \{a,b\}^\star \mid |\omega|_a \geq 2\}.$$

в) Нека $\boldsymbol{r} \stackrel{\text{деф}}{=} (\boldsymbol{b}+\boldsymbol{ab})^\star$. Тогава

За момента не е ясно как можем да верифицираме дали регулярният израз $\boldsymbol{r}$ наистина описва посочения език.

$$\mathcal{L}[\![\boldsymbol{r}]\!] = \{\omega \in \{a,b\}^\star \mid \text{ всяко } a \text{ в } \omega \text{ се следва от поне едно } b\}.$$

г) Нека $\boldsymbol{r} \stackrel{\text{деф}}{=} (\boldsymbol{b}^\star\boldsymbol{ab}^\star\boldsymbol{ab}^\star\boldsymbol{ab}^\star)^\star$. Тогава

$$\mathcal{L}[\![\boldsymbol{r}]\!] = \{\omega \in \{a,b\}^\star \mid |\omega|_a \equiv 0 \bmod 3\}.$$

Нека да въведем следните означения, които ще олеснят записа на регулярни изрази.

Определение 2.1. За произволни регулярни изрази $\boldsymbol{r}$ и $\boldsymbol{s}$, ще използваме записа $\boldsymbol{r} = \boldsymbol{s}$, ако $\mathcal{L}[\![\boldsymbol{r}]\!] = \mathcal{L}[\![\boldsymbol{s}]\!]$.

Задача 2.6. Докажете, че са изпълнени следните равенства:

а) $(\boldsymbol{rs})^\star\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{sr})^\star$;

б) $(\boldsymbol{r}+\boldsymbol{s})^\star = \boldsymbol{r}^\star(\boldsymbol{sr}^\star)^\star$

Задача 2.7. Проверете дали са изпълнени следните равенства:

а) $\boldsymbol{r}+\boldsymbol{s} = \boldsymbol{s}+\boldsymbol{r}$;
б) $(\boldsymbol{\varepsilon}+\boldsymbol{r})^\star = \boldsymbol{r}^\star$;
в) $(\boldsymbol{r}^\star\boldsymbol{s}^\star)^\star = (\boldsymbol{r}+\boldsymbol{s})^\star$;
г) $(\boldsymbol{r}^\star)^\star = \boldsymbol{r}^\star$;
д) $\boldsymbol{r}^\star\boldsymbol{r}^\star = \boldsymbol{r}^\star$;
е) $(\boldsymbol{rs}+\boldsymbol{r})^\star\boldsymbol{r} = \boldsymbol{r}(\boldsymbol{sr}+\boldsymbol{r})^\star$;
ж) $\boldsymbol{s}(\boldsymbol{rs}+\boldsymbol{s})^\star\boldsymbol{r} = \boldsymbol{rr}^\star\boldsymbol{s}(\boldsymbol{rr}^\star\boldsymbol{s})^\star$;
з) $(\boldsymbol{r}+\boldsymbol{s})^\star = \boldsymbol{r}^\star+\boldsymbol{s}^\star$;
и) $(\boldsymbol{r}+\boldsymbol{s})^\star\boldsymbol{s} = (\boldsymbol{r}^\star\boldsymbol{s})^\star$;
к) $(\boldsymbol{rs}+\boldsymbol{r})^\star\boldsymbol{rs} = (\boldsymbol{rr}^\star\boldsymbol{s})^\star$;
л) $\boldsymbol{\emptyset}^\star = \boldsymbol{\varepsilon}$;
м) $\boldsymbol{rs} = \boldsymbol{st} \implies \boldsymbol{r}^\star\boldsymbol{s} = \boldsymbol{st}^\star$;
н) $\boldsymbol{r}^\star = (\boldsymbol{\varepsilon}+\boldsymbol{r})^n(\boldsymbol{r}^n)^\star$, за всяко $n \in \mathbb{N}$.

Сега вече като имаме понятието регулярен израз, можем да преформулираме *Следствие 2.1* то следния начин.

Следствие 2.2. Да разгледаме системата:

$$\left|\begin{array}{rcl} X_1 & = & \boldsymbol{r_{1,1}}\cdot X_1 + \boldsymbol{r_{1,2}}\cdot X_2 + \cdots + \boldsymbol{r_{1,n}}\cdot X_n + \boldsymbol{p_1} \\ & \vdots & \\ X_n & = & \boldsymbol{r_{n,1}}\cdot X_1 + \boldsymbol{r_{n,2}}\cdot X_2 + \cdots + \boldsymbol{r_{n,n}}\cdot X_n + \boldsymbol{p_n}, \end{array}\right.$$

където $\boldsymbol{r_{i,j}}$ и $\boldsymbol{p_i}$ са *регулярни изрази* , за $i,j = 1,\ldots,n$, като $\varepsilon \notin \mathcal{L}[\![\boldsymbol{r_{i,j}}]\!]$, за всяко $i,j = 1,\ldots,n$. Тогава тази система има *единствено* решение, което е се описва с *регулярни изрази*.

2.5 Нерегулярни езици

Лема 2.2 (Слаба форма на лемата за покачването). Ако L е безкраен регулярен език, то съществуват думи α, β и γ, за които:

- $\beta \neq \varepsilon$;
- $(\forall n \in \mathbb{N})[\alpha\beta^n\gamma \in L]$.

Доказателство. Индукция по построението на регулярните езици.

- Ако $L = \emptyset$, $L = \{\varepsilon\}$, $L = \{a\}$, то твърдението е изпълнено по тривиални съображения, защото тогава L не е безкраен.
- Нека $L = L_1 \cup L_2$. Нека без ограничение на общността, L_1 е безкраен език. Тогава от **(И.П.)** , съществуват думи α, $\beta \neq \varepsilon$, γ, за които $\alpha\beta^n\gamma \in L_1$ за всяко n. Тогава е ясно, че $\alpha\beta^n\gamma \in L_1 \cup L_2$ за всяко n.
- Нека $L = L_1 \cdot L_2$. Нека без ограничение на общността, L_1 е безкраен език. Тогава от **(И.П.)** , съществуват думи α, $\beta \neq \varepsilon$, γ_1, за които $\alpha\beta^n\gamma_1 \in L_1$ за всяко n. Да вземем някоя дума $\gamma_2 \in L_2$. Да положим $\gamma = \gamma_1 \cdot \gamma_2$. Тогава е ясно, че $\alpha\beta^n\gamma \in L_1 \cdot L_2$ за всяко n.
- Нека $L = L_1^\star$. Нека просто вземем непразна дума $\beta \in L_1$ и $\alpha = \gamma = \varepsilon$. Ясно е, че $\alpha\beta^n\gamma \in L$ за всяко n.

□

Пример 2.6. Да видим защо езикът $L = \{a^n b^n \mid n \in \mathbb{N}\}$ *не* е регулярен. Да допуснем, че L е регулярен. Тогава какви са вариантите за избора на α, $\beta \neq \varepsilon$ и γ ?

- $\beta = a^i$, за някое $i \geq 1$, или
- $\beta = b^j$, за някое $j \geq 1$, или
- $\beta = a^i b^j$, за някои $i \geq 1$ и $j \geq 1$.

И в трите случая получаваме, че $\alpha\beta^2\gamma \notin L$. В първите два случая нарушаваме равния брой срещания на a и b. В третия случай нарушаваме последователността на буквите - след срещане на b се среща отново a.

Задача 2.8. Докажете, че езикът $L = \{a^n b^n \mid n \in \mathbb{N}\}$ е *единственото* решение на уравнението

$$X = \{a\} \cdot X \cdot \{b\} \cup \{\varepsilon\}.$$

2.6 Производна на език

Може би по-добро означение е L_a вместо $a^{-1}(L)$.

Нека въведем следната операция за произволна буква a,

$$a^{-1}(L) \stackrel{\text{деф}}{=} \{\omega \in \Sigma^\star \mid a \cdot \omega \in L\}.$$

Бжозовски [5] описва алгоритъм за строене на автомат по регулярен израз. Вижте [19, стр. 112].

Нека още тук да въведем следното означение, което ще ни бъде полезно по-нататък:

$$\varepsilon(L) \stackrel{\text{деф}}{=} \begin{cases} \{\varepsilon\}, & \text{ако } \varepsilon \in L \\ \emptyset, & \text{иначе.} \end{cases}$$

Да видим как се държи тази операция върху регулярните езици. За целта следваме дефиницията на регулярните езици.

Задача 2.9. Докажете, че за произволна буква a и език L са изпълнени равенствата:

(1) $a^{-1}(\emptyset) = \emptyset$;

(2) $a^{-1}(\{\varepsilon\}) = \emptyset$;

(3) $a^{-1}(\{b\}) = \begin{cases} \{\varepsilon\}, & \text{ако } a = b \\ \emptyset, & \text{ако } a \neq b \end{cases}$

(4) $a^{-1}(L_1 \cup L_2) = a^{-1}(L_1) \cup a^{-1}(L_2)$;

(5) $a^{-1}(L_1 \cdot L_2) = a^{-1}(L_1) \cdot L_2 \cup \varepsilon(L_1) \cdot a^{-1}(L_2)$;

(6) $a^{-1}(L \cdot L) = a^{-1}(L) \cdot L$;

(7) $a^{-1}(L^\star) = a^{-1}(L) \cdot L^\star$.

Лема 2.3. За всяка буква $a \in \Sigma$ и регулярен език L, езикът $a^{-1}(L)$ е регулярен.

Упътване. Индукция по построението на регулярниете езици. □

Определение 2.2. За произволна дума ω, нека положим

$$\omega^{-1}(L) \stackrel{\text{деф}}{=} \{\alpha \in \Sigma^\star \mid \omega \cdot \alpha \in L\}.$$

Задача 2.10. Докажете, че за всеки две думи α и β е изпълнено, че:

$$(\alpha \cdot \beta)^{-1}(L) = \beta^{-1}(\alpha^{-1}(L)).$$

Задача 2.11. Докажете, че за произволна дума ω и произволни езици L и M са изпълнени равенствата:

(а) $\omega^{-1}(L \cup M) = \omega^{-1}(L) \cup \omega^{-1}(M)$;

(б) $\omega^{-1}(L \cdot M) = \omega^{-1}(L) \cdot M \cup \bigcup_{\omega=\omega_1\omega_2} \varepsilon(\omega_1^{-1}(L)) \cdot \omega_2^{-1}(M)$;

(в) $\omega^{-1}(L^{n+1}) = \bigcup_{\ell+r=n} \bigcup_{\omega=\omega_1\cdot\omega_2} \varepsilon(\omega_1^{-1}(L^\ell)) \cdot \omega_2^{-1}(L) \cdot L^r$;

(г) $\omega^{-1}(L^\star) = \varepsilon(\omega^{-1}(L^\star)) \cdot L^\star \cup \bigcup_{\omega=\omega_1\omega_2} \varepsilon(\omega_1^{-1}(L^\star)) \cdot \omega_2^{-1}(L) \cdot L^\star$.

Лема 2.4. За всяка дума $\omega \in \Sigma^\star$ и регулярен език L, езикът $\omega^{-1}(L)$ е регулярен.

Упътване. Индукция по построението на регулярните езици. □

Задача 2.12. Докажете, че за всеки език L е изпълнено равенството:

$$L = \{\omega \in \Sigma^\star \mid \varepsilon \in \omega^{-1}(L)\}. \tag{2.3}$$

За произволен език L, нека положим

$$Q_L \stackrel{\text{деф}}{=} \{\omega^{-1}(L) \mid \omega \in \Sigma^\star\}.$$

Оказва се, че е интересно да се изследва мощността на множеството Q_L.

Да напомним, че ако X е език, то $X \in \mathcal{P}(\Sigma^\star)$. Ако $\mathcal{X}$ е фамилия от езици, то $\mathcal{X} \subseteq \mathcal{P}(\Sigma^\star)$. За следващата теорема ще се наложи да работим с множества от езици. Нека $\mathcal{X}$ е множество от езици и L е език. Да обобщим операцията конкатенация така:

$L \cdot \mathcal{X} \stackrel{\text{деф}}{=} \{L \cdot M \mid M \in \mathcal{X}\}$

$\mathcal{X} \cdot L \stackrel{\text{деф}}{=} \{M \cdot L \mid M \in \mathcal{X}\}.$

Вижте[19, стр. 142].

Теорема 2.2. Ако L е регулярен, то Q_L е крайно множество.

Доказателство. Индукция по построението на регулярните езици.

- Нека $L = \emptyset$. Ясно е, че $Q_L = \{\emptyset\}$.
- Нека $L = \{\varepsilon\}$. Ясно е, че $Q_L = \{\{\varepsilon\}, \emptyset\}$.
- Нека $L = \{a\}$. Ясно е, че $Q_L = \{\{a\}, \{\varepsilon\}, \emptyset\}$.
- Нека $L = L_1 \cup L_2$. Ясно е, че за произволна дума ω,

  $$\omega^{-1}(L_1 \cup L_2) = \omega^{-1}(L_1) \cup \omega^{-1}(L_2).$$

  Това означава, че за всяка дума ω, съществуват езици $M_1 \in Q_{L_1}$ и $M_2 \in Q_{L_2}$, за които $\omega^{-1}(L_1 \cup L_2) = M_1 \cup M_2$. Да положим

  $$\mathcal{X} \stackrel{\text{деф}}{=} \{M_1 \cup M_2 \mid M_1 \in Q_{L_1} \,\&\, M_2 \in Q_{L_2}\}.$$

  Тогава:

  $$Q_{L_1 \cup L_2} \subseteq \mathcal{X}.$$

  Да разгледаме функцията $f : Q_{L_1} \times Q_{L_2} \to \mathcal{X}$, където

  $$f(M_1, M_2) \stackrel{\text{деф}}{=} M_1 \cup M_2.$$

  Ясно е, че f е сюрекция, откъдето следва, че $|\mathcal{X}| \leq |Q_{L_1} \times Q_{L_2}|$. Оттук заключаваме, че:

  $$|Q_{L_1 \cup L_2}| \leq |\mathcal{X}| \leq |Q_{L_1} \times Q_{L_2}| = |Q_{L_1}| \cdot |Q_{L_2}|.$$

  От **(И.П.)** имаме, че Q_{L_1} и Q_{L_2} са крайни множества. Следователно $Q_{L_1 \cup L_2}$ също е крайно множество.

- Нека $L = L_1 \cdot L_2$. Тогава, за произволна дума ω,

  $$\omega^{-1}(L_1 \cdot L_2) = \omega^{-1}(L_1) \cdot L_2 \cup \bigcup_{\omega = \omega_1 \cdot \omega_2} \varepsilon(\omega_1^{-1}(L_1)) \cdot \omega_2^{-1}(L_2).$$

  Това означава, че за всяка дума ω, съществува език $M \in Q_{L_1} \cdot L_2$ и множество от езици $\mathcal{X} \subseteq Q_{L_2}$, за които

  $$\omega^{-1}(L_1 \cdot L_2) = M \cup \bigcup \mathcal{X}.$$

  Така получаваме, че

  $$Q_{L_1 \cdot L_2} \subseteq \left\{M \cup \bigcup \mathcal{X} \mid M \in Q_{L_1} \cdot L_2 \,\&\, \mathcal{X} \in \mathcal{P}(Q_{L_2})\right\}.$$

  Понеже $|Q_{L_1} \cdot L_2| \leq |Q_{L_1}|$, то можем да заключим, че:

  $$|Q_{L_1 \cdot L_2}| \leq |Q_{L_1} \times \mathcal{P}(Q_{L_2})| = |Q_{L_1}| \cdot 2^{|Q_{L_2}|}.$$

  От **(И.П.)** имаме, че Q_{L_1} и Q_{L_2} са крайни множества. Следователно $Q_{L_1 \cdot L_2}$ също е крайно множество.

- Остана да разгледаме езика $L^\star$. Имаме, че

$$\omega^{-1}(L^\star) = \varepsilon(\omega^{-1}(L^\star)) \cdot L^\star \cup \bigcup_{\omega_1 \cdot \omega_2 = \omega} \varepsilon(\omega_1^{-1}(L^\star)) \cdot \omega_2^{-1}(L) \cdot L^\star.$$

Така получаваме, че за всяко ω, съществува крайно множество от езици

$$\mathcal{X} \subseteq Q_L \cdot L^\star,$$

за което $\omega^{-1}(L^\star) = \bigcup \mathcal{X}$. Така получаваме, че

$$Q_{L^\star} \subseteq \left\{ \bigcup \mathcal{X} \mid \mathcal{X} \in \mathcal{P}(Q_L \cdot L^\star) \right\}.$$

Понеже $|Q_L \cdot L^\star| \leq |Q_L|$, то можем да заключим, че:

$$|Q_{L^\star}| \leq |\mathcal{P}(Q_L \cdot L^\star)| \leq 2^{|Q_L|}.$$

От **(И.П.)** имаме, че Q_L е крайно множество. Следователно $Q_{L^\star}$ също е крайно множество.

□

По-късно ще видим, че това свойство точно характеризира регулярните езици. За момента ще се задоволим със следното наблюдение.

Задача 2.13. Нека $L = \{a^n b^n \mid n \in \mathbb{N}\}$. Тогава Q_L е безкрайно множество.

Упътване. Съобразете, че всички езици $(a^k)^{-1}(L) = \{a^n b^{n+k} \mid n, k \in \mathbb{N}\}$ са различни. □ Вижте също *Пример 3.2*.

Джон Конуей [6, стр. 43] предлага да обобщим операцията $\omega^{-1}(L)$ по следния начин. Нека M и L са произволни езици. Да положим

$$M^{-1}(L) \stackrel{\text{деф}}{=} \{\omega^{-1}(L) \mid \omega \in M\}.$$

Задача 2.14. Докажете, че:

- $M^{-1}(\emptyset) = \emptyset$;
- $M^{-1}(\{a\}) = \begin{cases} \emptyset, & \text{ако } a \notin M \\ \{\varepsilon\}, & \text{ако } a \in M \\ \{a\}, & \text{ако } \varepsilon \in M \ \& \ a \notin M \\ \{\varepsilon, a\}, & \text{ако } \varepsilon \in M \ \& \ a \in M. \end{cases}$
- $M^{-1}(L_1 \cup L_2) = M^{-1}(L_1) \cup M^{-1}(L_2)$.
- $M^{-1}(L_1 \cdot L_2) = M^{-1}(L_1) \cdot L_2 \cup (L_1^{-1}(M))^{-1}(L_2)$;
- $M^{-1}(L^\star) = \varepsilon(M) \cup ((L^\star)^{-1}(M))^{-1}(L) \cdot L^\star$;

Лема 2.5. За всеки език M и регулярен език L, езикът $M^{-1}(L)$ е регулярен.

Упътване. Индукция по построението на регулярните езици. □

2.7 Хомоморфизми

Определение 2.3. За 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 е изпълнено, че $h(\varepsilon) = \varepsilon$. За да дефинирате един хомоморфизъм h е достатъчно да кажете как h преобразува всяка буква над азбуката Σ_1 в дума над азбуката Σ_2.

За произволна функция $h : \Sigma_1^\star \to \Sigma_2^\star$, казваме, че **образът** на $L \subseteq \Sigma_1^\star$ относно h е множеството:

$$h(L) \stackrel{\text{деф}}{=} \{h(\omega) \in \Sigma_2^\star \mid \omega \in L\}.$$

Твърдение 2.1. Нека $h : \Sigma_1^\star \to \Sigma_2^\star$ е функция. Тогава за всеки два езика $L_1 \subseteq \Sigma_1^\star$ и $L_2 \subseteq \Sigma_2^\star$ е изпълнено, че:

$$h(L_1 \cup L_2) = h(L_1) \cup h(L_2).$$

Тук няма нужда h да бъде хомоморфизъм.

Доказателство. Достатъчно е да проследим следната верига от равенства:

$$\begin{aligned} h(L_1 \cup L_2) &= \{h(\omega) \mid \omega \in L_1 \cup L_2\} \\ &= \{h(\omega) \mid \omega \in L_1 \vee \omega \in L_2\} \\ &= \{h(\omega) \mid \omega \in L_1\} \cup \{h(\omega) \mid \omega \in L_2\} \\ &= h(L_1) \cup h(L_2). \end{aligned}$$

□

Твърдение 2.2. Нека $h : \Sigma_1^\star \to \Sigma_2^\star$ е функция и нека за всяко n, L_n е език над азбуката Σ_1. Тогава

$$h(\bigcup_n L_n) = \bigcup_n h(L_n).$$

Доказателство. Просто проследяваме равенствата:

$$\begin{aligned} h(\bigcup_n L_n) &= \{h(\alpha) \mid (\exists n)[\alpha \in L_n]\} \\ &= \bigcup_n \{h(\alpha) \mid \alpha \in L_n\} \\ &= \bigcup_n h(L_n). \end{aligned}$$

□

Твърдение 2.3. Нека $h : \Sigma_1^\star \to \Sigma_2^\star$ е хомоморфизъм. Тогава за всеки два езика $L_1 \subseteq \Sigma_1^\star$ и $L_2 \subseteq \Sigma_2^\star$ е изпълнено, че:

$$h(L_1 \cdot L_2) = h(L_1) \cdot h(L_2).$$

Доказателство. Достатъчно е да проследим веригата от равенства:

$$\begin{aligned} h(L_1 \cdot L_2) &= \{h(\alpha) \mid \alpha \in L_1 \cdot L_2\} \\ &= \{h(\alpha_1\alpha_2) \mid \alpha_1 \in L_1 \,\&\, \alpha_2 \in L_2\} \\ &= \{h(\alpha_1) \cdot h(\alpha_2) \mid \alpha_1 \in L_1 \,\&\, \alpha_2 \in L_2\} && \text{// } h \text{ е хомоморфизъм} \\ &= \{\omega_1\omega_2 \mid \omega_1 \in h(L_1) \,\&\, \omega_2 \in h(L_2)\} \\ &= h(L_1) \cdot h(L_2). \end{aligned}$$

□

Твърдение 2.4. Нека $h : \Sigma_1^\star \to \Sigma_2^\star$ е хомоморфизъм и $L \subseteq \Sigma_1^\star$. Тогава за всяко естествено число n е изпълнено, че:

$$h(L^n) = h(L)^n.$$

Доказателство. Индукция по n.

- Нека $n = 0$. Тогава всичко е ясно, защото:

$$L^0 = \{\varepsilon\} = h(L)^0.$$

- Да приемем като индукционно предположение, че $h(L^n) = h(L)^n$.
- За да докажем индукционната стъпка е достатъчно да проследим равенствата:

$$\begin{aligned} h(L^{n+1}) &= h(L^n \cdot L) \\ &= h(L^n) \cdot h(L) && \text{// от } \textit{Твърдение 2.3} \\ &= h(L)^n \cdot h(L) && \text{// от } \textbf{(И.П.)} \\ &= h(L)^{n+1}. \end{aligned}$$

□

Лема 2.6. Нека $h : \Sigma_1^\star \to \Sigma_2^\star$ е хомоморфизъм. Тогава, ако $L \subseteq \Sigma_1^\star$ е регулярен език, то $h(L)$ е регулярен език.

Лема 2.6 ни казва, че регулярните езици са затворени относно хомоморфизми.

Доказателство. Индукция по построението на регулярни езици.

- Нека $L = \emptyset$. Тогава е ясно, че

$$h(L) = \emptyset.$$

- Нека $L = \{a\}$, за някоя буква $a \in \Sigma$, то тогава

$$h(L) = \{h(a)\},$$

  който е ясно, че е регулярен.

- Нека L_1 и L_2 са регулярни езици, за които като индукционно предположение приемаме, че $h(L_1)$ и $h(L_2)$ са регулярни езици.
  - Според *Твърдение 2.1* имаме, че

$$h(L_1 \cup L_2) = h(L_1) \cup h(L_2).$$

    Щом от **(И.П.)** имаме, че $h(L_1)$ и $h(L_2)$ са регулярни езици, то заключаваме, че $h(L_1 \cup L_2)$ е регулярен език.

– Според *Твърдение 2.3* имаме, че

$$h(L_1 \cdot L_2) = h(L_1) \cdot h(L_2).$$

Щом от **(И.П.)** имаме, че $h(L_1)$ и $h(L_2)$ са регулярни езици, то заключаваме, че $h(L_1 \cdot L_2)$ е регулярен език.

- Нека сега L е регулярен език, за който да приемем като индукционно предположение, че $h(L)$ е регулярен език. Имаме равенствата:

$$\begin{aligned} h(L^\star) &= h(\bigcup_n L^n) && \text{// деф. на звезда на Клини} \\ &= \bigcup_n h(L^n) && \text{// от } \textit{Твърдение 2.2} \\ &= \bigcup_n h(L)^n && \text{// от } \textit{Твърдение 2.4} \\ &= h(L)^\star. && \text{// деф. на звезда на Клини} \end{aligned}$$

Щом от **(И.П.)** имаме, че $h(L)$ е регулярен език, то $h(L^\star) = h(L)^\star$ също е регулярен език.

□

Пример 2.7. Да разгледаме езика $L = \{ca^nb^n \mid n \in \mathbb{N}\}$. Да видим как лесно можем да докажем, че L е нерегулярен. Да разгледаме хомоморфизма h, който изтрива буквата c, т.е. $h(a) \stackrel{\text{деф}}{=} a$, $h(b) \stackrel{\text{деф}}{=} b$ и $h(c) \stackrel{\text{деф}}{=} \varepsilon$. Тогава, ако допуснем, че L е регулярен, то би следвало, че и

$$h(L) = \{a^nb^n \mid n \in \mathbb{N}\}$$

е регулярен. Достигхнаме до противоречие.

Забележка. И така, вече знаем, че ако L е регулярен и h е хомоморфизъм, то $h(L)$ е регулярен. В общия случай, не можем да твърдим, че имаме обратната посока, т.е. съществува хомоморфизъм h, нерегулярен език L, за които $h(L)$ е регулярен.

Нека за простота да фиксираме $\Sigma_1 = \Sigma_2 = \{a, b\}$. Да вземем възможно най-простия хомоморфизъм h - този, който трие всичко, т.е. $h(a) \stackrel{\text{деф}}{=} \varepsilon$ и $h(b) \stackrel{\text{деф}}{=} \varepsilon$. Тогава за всеки език L, $h(L) = \{\varepsilon\}$.

Като друг пример, нека да вземем константен хомоморфизъм h - такъв, който винаги връща едно и също. Например, нека $h(a) \stackrel{\text{деф}}{=} a$ и $h(b) \stackrel{\text{деф}}{=} a$. За нерегулярния език $L = \{a^nb^n \mid n \in \mathbb{N}\}$ имаме, че $h(L) = \{a^k \mid k \text{ е четно}\}$, който е регулярен.

Глава 3

Автоматни езици

3.1 Детерминирани крайни автомати

Определение 3.1. Детерминиран краен автомат (ДКА) е петорка

$$\mathcal{A} = \langle \Sigma, Q, q_{\texttt{start}}, \delta, F \rangle,$$

където

- Σ е азбука;
- Q е крайно множество от състояния;
- $\delta : Q \times \Sigma \to Q$ е тотална функция, която ще наричаме *функция на преходите*;
- $q_{\texttt{start}} \in Q$ е начално състояние;
- $F \subseteq Q$ е множеството от финални състояния

Нека имаме една дума $\alpha \in \Sigma^\star$, където $\alpha = a_0 a_1 \cdots a_{n-1}$. 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:

- $q_0 = q_{\texttt{start}}$, началното състояние на автомата;
- $\delta(q_i, a_i) = q_{i+1}$, 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. $L = \{\alpha \in \Sigma^\star \mid \mathcal{A} \text{ разпознава } \alpha\}$. В такъв случай ще казваме, че езикът L е **автоматен**.

Algorithm 1 Проблемът за принадлежност за автоматни езици

```
1: procedure BELONG(A, ω)
2:     q := q_start
3:     for all i < |ω| do
4:         q := δ(q, ω[i])
5:     if q ∈ F then
6:         return True
7:     else
8:         return False
```

При дадена функция на преходите δ, често е удобно да разглеждаме функция $\delta^\star : Q \times \Sigma^\star \to Q$, която е дефинирана за всяко $q \in Q$ и $\alpha \in \Sigma^\star$ по следния начин:

Възможно е да се дефинира $\delta^\star$ и така:

$$\delta^\star(q, \varepsilon) = q$$
$$\delta^\star(q, a\beta) = \delta^\star(\delta(q, a), \beta).$$

- Ако $\alpha = \varepsilon$, то $\delta^\star(q, \varepsilon) \stackrel{\text{деф}}{=} q$;
- Ако $\alpha = \beta a$, то $\delta^\star(q, \beta a) \stackrel{\text{деф}}{=} \delta(\delta^\star(q, \beta), a)$.

✍ Съобразете го сами!

Лесно се съобразява, че една дума α се *разпознава* от автомата $\mathcal{A}$ точно тогава, когато $\delta^\star(q_{\texttt{start}}, \alpha) \in F$.

Определение 3.2. За произволен ДКА $\mathcal{A}$, дефинираме

$$\mathcal{L}(\mathcal{A}) \stackrel{\text{деф}}{=} \{\, \omega \in \Sigma^\star \mid \delta^\star(q_{\texttt{start}}, \omega) \in F \,\}.$$

Понякога е удобно да разгледаме произволно състояние на автомата $\mathcal{A}$ като начално и да разгледаме съответния език, който получаваме. Поради тази причина полагаме за произволно $q \in Q$,

$$\mathcal{L}_\mathcal{A}(q) \stackrel{\text{деф}}{=} \{\omega \in \Sigma^\star \mid \delta^\star(q, \omega) \in F\}.$$

В частност, $\mathcal{L}(\mathcal{A}) = \mathcal{L}_\mathcal{A}(q_{\texttt{start}})$.

Твърдение 3.1. Нека $\mathcal{A}$ е ДКА. Тогава за всяко състояние q и произволни думи α и β е изпълнено равенството:

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

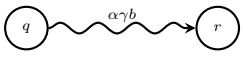
Обърнете внимание, че $\delta(q, a) = \delta^\star(q, a)$ за $a \in \Sigma$

Упътване. Индукция по дължината на думата β.

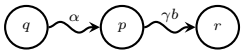
- $|\beta| = 0$, т.е. $\beta = \varepsilon$. Тогава за всяко състояние q и произволна дума α имаме:

$$\delta^\star(q, \alpha\varepsilon) = \delta^\star(\underbrace{\delta^\star(q, \alpha)}_{p}, \varepsilon).$$

- Да приемем, че твърдението е изпълнено за думи β с дължина n.
- За индукционната стъпка имаме изчислението:



То може да се разбие така:



Нека $|\beta| = n + 1$, т.е. $\beta = \gamma b$, където $|\gamma| = n$. Тогава за всяко състояние q и произволна дума α имаме:

$$\begin{aligned}
\delta^\star(q, \alpha\beta) &= \delta^\star(q, \alpha\gamma b) \\
&= \delta(\delta^\star(q, \alpha\gamma), b) && \text{// от деф. на } \delta^\star \\
&= \delta(\delta^\star(\underbrace{\delta^\star(q, \alpha)}_{p}, \gamma), b) && \text{// от } \textbf{(И.П.)} \text{ приложено за } \gamma \\
&= \delta(\delta^\star(p, \gamma), b) \\
&= \delta^\star(p, \gamma b) && \text{// от деф. на } \delta^\star \\
&= \delta^\star(\delta^\star(q, \alpha), \beta). && \text{// } p = \delta^\star(q, \alpha)
\end{aligned}$$

□

Забележка. Формално погледнато, доказателството на *Твърдение 3.1* протича по следната схема:

$$\frac{P(0) \qquad (\forall n \in \mathbb{N})[P(n) \implies P(n+1)]}{(\forall n \in \mathbb{N})[P(n)]},$$

където свойството P е дефинирано така:

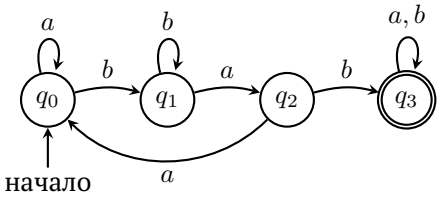
$$P(n) \stackrel{\text{деф}}{=} (\forall \beta \in \Sigma^n)(\forall \alpha \in \Sigma^\star)(\forall q \in Q)[\delta^\star(q, \alpha\beta) = \delta^\star(\delta^\star(q, \alpha), \beta)].$$

Примерни задачи

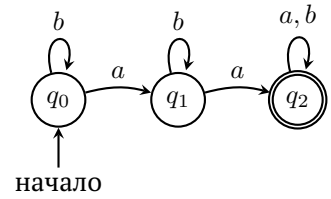
В този раздел ще разгледаме няколко примера за автоматни езици и ще видим как можем да докажем, че конкретен автомат разпознава даден език.

Винаги с удвоени окръжности ще отбелязваме финалните състояния. За момента нямаме общ метод, с който да докажем, че даденият автомат разпознава точно съответния език.

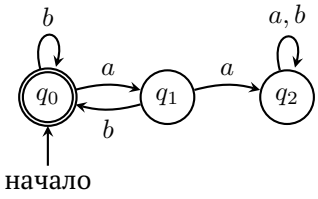
Пример 3.1. Да разгледаме няколко примера за автомати и езиците, които разпознават. Дефинирайте функцията на преходите δ за всеки автомат.



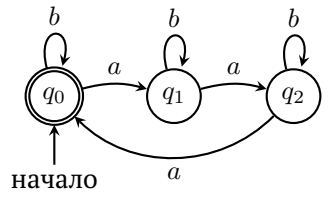
(а) $\{\omega \in \{a,b\}^\star \mid \omega \text{ съдържа } bab\}$



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



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



(г) $\{\omega \in \{a,b\}^\star \mid |\omega|_a \equiv 0 \mod 3\}$

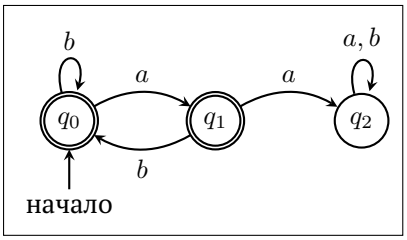
В повечето от горните примери може сравнително лесно да се съобрази, че построеният автомат разпознава желания език. При по-сложни задачи, обаче, ще се наложи да дадем доказателство, като обикновено се прилага *метода на математическата индукция* върху дължината на думите. Ще разгледаме няколко такива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

Задача 3.1. Докажете, че езикът L е автоматен, където:

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

Доказателство. Да разгледаме $\mathcal{A} = \langle \Sigma, Q, q_{\texttt{start}}, \delta, F \rangle$ с функция на преходите описана на *Фигура 3.2*.

Най-лесния начин да се сетим как да построим $\mathcal{A}$ е като първо построим автомат за езика, който разпознава тези думи, в които се съдържат две поредни срещания на a и вземем неговото допълнение съгласно Твърдение 3.3. По-късно ще разгледаме общ метод за строене на автомат по език.



Фигура 3.2: Автомат $\mathcal{A}$ разпознаващ думите, които не съдържат две поредни срещания на a

Ще докажем, че $L = \mathcal{L}(\mathcal{A})$. Първо ще се концентрираме върху доказателството на включването $\mathcal{L}(\mathcal{A}) \subseteq L$. Ще докажем, че за всяка дума $\alpha \in \Sigma^\star$ са изпълнени свойствата:

Да напомним, че $|\alpha| \stackrel{\text{деф}}{=}$ дължината на α.

(1) ако $\delta^\star(q_0, \alpha) = q_0$, то α не съдържа две поредни срещания на a и ако $|\alpha| > 0$, то α завършва на b;

(2) ако $\delta^\star(q_0, \alpha) = q_1$, то α не съдържа две поредни срещания на a и завършва на a.

Ще докажем (1) и (2) едновременно с индукция по дължината на думата α.

- За $|\alpha| = 0$, т.е. $\alpha = \varepsilon$, 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 $\delta^\star(q_0, \beta x) = q_0 = \delta(\delta^\star(q_0, \beta), x)$. Според дефиницията на функцията δ, $x = b$ и $\delta^\star(q_0, \beta) \in \{q_0, q_1\}$. 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 $\delta^\star(q_0, \beta x) = q_1 = \delta(\delta^\star(q_0, \beta), x)$. Според дефиницията на δ, $x = a$ и $\delta^\star(q_0, \beta) = q_0$. Тогава по **(И.П.)** за (1), β не съдържа две поредни срещания на a и ако $|\beta| > 0$, то β 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 $\delta^\star(q_0, \alpha) \in F = \{q_0, q_1\}$ 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 включването

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

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

$$(\forall \alpha \in \Sigma^\star)[\alpha \in L \;\Rightarrow\; \delta^\star(q_0, \alpha) \in F],$$

Да напомним, че от съждителното смятане знаем, че $p \to q \equiv \neg q \to \neg p$.

което, разгледана в контрапозиция, е еквивалентна на

$$(\forall \alpha \in \Sigma^\star)[\delta^\star(q_0, \alpha) \notin F \;\Rightarrow\; \alpha \notin L]. \tag{3.1}$$

Това е лесно да се съобрази. Щом $\delta^\star(q_0, \alpha) \notin F$, то $\delta^\star(q_0, \alpha) = q_2$ и думата α може да се представи по следния начин:

$$\alpha = \beta a \gamma \;\&\; \delta^\star(q_0, \beta) = q_1.$$

Използвайки свойство (2) от по-горе, понеже $\delta^\star(q_0, \beta) = q_1$, 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 3.1, а следователно и посоката $L \subseteq \mathcal{L}(\mathcal{A})$. □

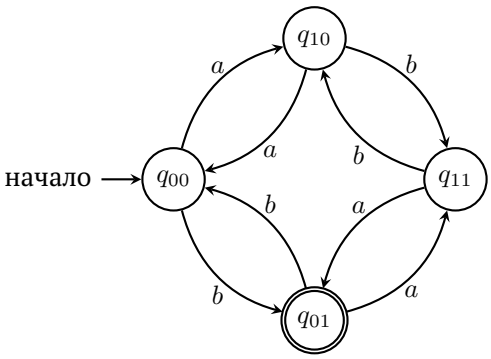
Да напомним, че $|\omega|_a \stackrel{\text{деф}}{=}$ броят на срещанията на буквата a в думата ω.

Задача 3.2. Докажете, че езикът L е автоматен, където:

$$L \stackrel{\text{деф}}{=} \{\omega \in \{a, b\}^\star \mid |\omega|_a \equiv 0 \bmod 2 \;\&\; |\omega|_b \equiv 1 \bmod 2\}.$$

Тази задача е важна, защото в явен вид показва как кодираме информация в състоянията на автомата, а именно остатъците при деление на 2 на $|\omega|_a$ и $|\omega|_b$.

Упътване. Разгледайте детерминирания автомат $\mathcal{A}$:



Фигура 3.3: $\mathcal{L}(\mathcal{A}) \stackrel{?}{=} L$

Докажете с индукция по дължината на думата ω, че:

а) $\delta^\star(q_{00}, \omega) = q_{00} \implies |\omega|_a \equiv 0 \bmod 2 \;\&\; |\omega|_b \equiv 0 \bmod 2$;

б) $\delta^\star(q_{00}, \omega) = q_{01} \implies |\omega|_a \equiv 0 \bmod 2 \;\&\; |\omega|_b \equiv 1 \bmod 2$;

в) $\delta^\star(q_{00}, \omega) = q_{10} \implies |\omega|_a \equiv 1 \bmod 2 \;\&\; |\omega|_b \equiv 0 \bmod 2$;

г) $\delta^\star(q_{00}, \omega) = q_{11} \implies |\omega|_a \equiv 1 \bmod 2 \;\&\; |\omega|_b \equiv 1 \bmod 2$;

Оттук направете извода, че за произволна дума ω,

$$(\forall i < 2)(\forall j < 2)[\, \delta^\star(q_{00}, \omega) = q_{ij} \;\Leftrightarrow\; |\omega|_a \equiv i \bmod 2 \;\&\; |\omega|_b \equiv j \bmod 2\,].$$

□

За една дума $\alpha \in \{0,1\}^\star$, нека с $\overline{\alpha}_{(k)}$ да означим числото, което се представя в k-ична бройна система като α. Например,

$$\overline{1101}_{(2)} = 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0 = \overline{13}_{(10)} = 1.10^1 + 3.10^0.$$

За $k = 2$, можем да изразим $\overline{\alpha}_{(2)}$ рекурсивно така:

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

Да отбележим, че за всяко число n има безкрайно много думи α, за които $\overline{\alpha}_{(2)} = n$. Например,

$$\begin{aligned}\overline{10}_{(2)} &= \overline{010}_{(2)} \\ &= \overline{0010}_{(2)} \\ &= \cdots\end{aligned}$$

Задача 3.3. Докажете, че езикът L е автоматен, където:

$$L \stackrel{\text{деф}}{=} \{\alpha \in \{0,1\}^\star \mid \overline{\alpha}_{(2)} \equiv 2 \bmod 3\}.$$

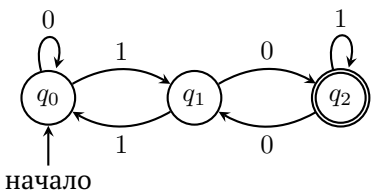
Доказателство. Нашият автомат ще има три състояния $\{q_0, q_1, q_2\}$, като началното състояние ще бъде q_0. Целта ни е да дефинираме така автомата, че да имаме следното свойство:

$$(\forall \alpha \in \Sigma^\star)(\forall i < 3)[\, \overline{\alpha}_{(2)} \equiv i \bmod 3 \;\Leftrightarrow\; \delta^\star(q_0, \alpha) = q_i\,], \tag{3.2}$$

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 по следния начин:

$$\begin{aligned}
&\delta(q_0, 0) = q_0 && // \;\overline{\alpha}_{(2)} \equiv 0 \bmod 3 \implies \overline{\alpha 0}_{(2)} \equiv 0 \bmod 3 \\
&\delta(q_0, 1) = q_1 && // \;\overline{\alpha}_{(2)} \equiv 0 \bmod 3 \implies \overline{\alpha 1}_{(2)} \equiv 1 \bmod 3 \\
&\delta(q_1, 0) = q_2 && // \;\overline{\alpha}_{(2)} \equiv 1 \bmod 3 \implies \overline{\alpha 0}_{(2)} \equiv 2 \bmod 3 \\
&\delta(q_1, 1) = q_0 && // \;\overline{\alpha}_{(2)} \equiv 1 \bmod 3 \implies \overline{\alpha 1}_{(2)} \equiv 0 \bmod 3 \\
&\delta(q_2, 0) = q_1 && // \;\overline{\alpha}_{(2)} \equiv 2 \bmod 3 \implies \overline{\alpha 0}_{(2)} \equiv 1 \bmod 3 \\
&\delta(q_2, 1) = q_2 && // \;\overline{\alpha}_{(2)} \equiv 2 \bmod 3 \implies \overline{\alpha 1}_{(2)} \equiv 2 \bmod 3
\end{aligned}$$

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



Фигура 3.4: $\mathcal{L}(\mathcal{A}) \stackrel{?}{=} \{\alpha \in \{0,1\}^\star \mid \overline{\alpha}_{(2)} \equiv 2 \,(\mathrm{mod}\, 3)\}$

Да разгледаме твърденията:

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

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

(3) $\delta^\star(q_0, \alpha) = q_2 \implies \overline{\alpha}_{(2)} \equiv 2 \mod 3$.

Обърнете внимание, че в доказателството на (3) използваме И.П. не само за (3), но и за (2). Поради тази причина трябва да докажем трите свойства едновременно

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. $\alpha = \beta x$, $|\beta| = n$.

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:

- $x = 0$. 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 имаме импликацията:

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

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

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

- $x = 1$. 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 имаме импликацията:

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

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

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

✍ Довършете доказателствата на (1) и (2)

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).

✍ Защо?

От (1), (2) и (3) следва директно, че е изпълнено свойството:

$$(\forall \alpha \in \Sigma^\star)(\forall i < 3)[\, \overline{\alpha}_{(2)} \equiv i \,(\mathrm{mod}\ 3) \Leftrightarrow \delta^\star(q_0, \alpha) = q_i \,],$$

откъдето получаваме, че $\mathcal{L}(\mathcal{A}) = L$. □

Затвореност относно булеви операции

В този раздел ще видим, че автоматните езици са затворени относно основните булеви операции над езици - обединение, сечение и разлика.

Твърдение 3.2. 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.

Доказателство. Да разгледаме два автомата

$$\mathcal{A}_1 = \langle \Sigma, Q_1, q'_{\mathtt{start}}, \delta_1, F_2 \rangle \text{ и } \mathcal{A}_2 = \langle \Sigma, Q_2, q''_{\mathtt{start}}, \delta_2, F_2 \rangle.$$

Определяме автомата $\mathcal{A} = \langle \Sigma, Q, q_{\mathtt{start}}, \delta, F \rangle$, по следния начин:

Изчислението на автомата $\mathcal{A}$ върху думата α едновременно симулира изчислението на $\mathcal{A}_1$ и $\mathcal{A}_2$ върху α.

Съобразете, че δ е тотална функция.

- $Q \stackrel{\text{деф}}{=} Q_1 \times Q_2$;
- За всяко $\langle r_1, r_2 \rangle \in Q$ и всяко $a \in \Sigma$,

$$\delta(\langle r_1, r_2 \rangle, a) \stackrel{\text{деф}}{=} \langle \delta_1(r_1, a), \delta_2(r_2, a) \rangle;$$

- $q_{\mathtt{start}} \stackrel{\text{деф}}{=} \langle q'_{\mathtt{start}}, q''_{\mathtt{start}} \rangle$;
- $F \stackrel{\text{деф}}{=} \{\langle r_1, r_2 \rangle \mid r_1 \in F_1 \;\&\; r_2 \in F_2\} = F_1 \times F_2$

Трябва да докажем, че за всяка дума $\alpha \in \Sigma^\star$ е изпълнено, че:

$$(\forall p \in Q_1)(\forall q \in Q_2)[\; \delta^\star(\langle p, q \rangle, \alpha) = \langle \delta_1^\star(p, \alpha), \delta_2^\star(q, \alpha) \rangle \;]. \tag{3.3}$$

Ще докажем *Свойство 3.3* с индукция по дължината на думата α.

- Нека $|\alpha| = 0$, т.е. $\alpha = \varepsilon$. Тогава всичко е ясно, защото

$$\begin{aligned} \delta^\star(\langle p, q \rangle, \varepsilon) &= \langle p, q \rangle && \text{// деф. на } \delta^\star \\ &= \langle \delta_1^\star(p, \varepsilon), \delta_2^\star(q, \varepsilon) \rangle. && \text{// деф. на } \delta_1^\star \text{ и } \delta_2^\star \end{aligned}$$

- Да приемем, че *Свойство 3.3* е изпълнено за думи α с дължина n.
- Нека $|\alpha| = n + 1$, т.е. $\alpha = \beta a$ и $|\beta| = n$. Тогава:

$$\begin{aligned} \delta^\star(\langle p, q \rangle, \beta a) &= \delta(\delta^\star(\langle p, q \rangle, \beta), a) && \text{// деф. на } \delta^\star \\ &= \delta(\langle \delta_1^\star(p, \beta), \delta_2^\star(q, \beta) \rangle, a) && \text{// от } \textbf{(И.П.)} \\ &= \langle \delta_1(\delta_1^\star(p, \beta), a), \delta_2(\delta_2^\star(q, \beta), a) \rangle && \text{// деф. на } \delta \\ &= \langle \delta_1^\star(p, \beta a), \delta_2^\star(q, \beta a) \rangle && \text{// деф. на } \delta_1^\star \text{ и } \delta_2^\star. \end{aligned}$$

Използвайки *Свойство 3.3* лесно можем да докажем, че

$$\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2).$$

Имаме следните еквивалентности:

$$\begin{aligned} \omega \in \mathcal{L}(\mathcal{A}) &\Leftrightarrow \delta^\star(q_{\mathtt{start}}, \omega) \in F && \text{// деф. на } \mathcal{L}(\mathcal{A}) \\ &\Leftrightarrow \delta^\star(\langle q'_{\mathtt{start}}, q''_{\mathtt{start}} \rangle, \omega) \in F_1 \times F_2 && \text{// деф. на } \mathcal{A} \\ &\Leftrightarrow \langle \delta_1^\star(q'_{\mathtt{start}}, \omega), \delta_2^\star(q''_{\mathtt{start}}, \omega) \rangle \in F_1 \times F_2 && \text{// от (3.3)} \\ &\Leftrightarrow \delta_1^\star(q'_{\mathtt{start}}, \omega) \in F_1 \;\&\; \delta_2^\star(q''_{\mathtt{start}}, \omega) \in F_2 \\ &\Leftrightarrow \omega \in \mathcal{L}(\mathcal{A}_1) \;\&\; \omega \in \mathcal{L}(\mathcal{A}_2) \\ &\Leftrightarrow \omega \in \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2). \end{aligned}$$

□

Твърдение 3.3. Класът на автоматните езици са затворени относно операцията *допълнение*, т.е. ако L е автоматен език, то $\Sigma^\star \setminus L$ също е автоматен език.

✍ Проверете, че $\Sigma^\star \setminus L = \mathcal{L}(\mathcal{A}')$. Съобразете, че тук е важно, че δ е тотална функция на преходите, а не просто частична функция.

Упътване. Нека $L = L(\mathcal{A})$, където $\mathcal{A} = \langle \Sigma, Q, q_{\texttt{start}}, \delta, F \rangle$. Да вземем автомата

$$\mathcal{A}' = \langle Q, \Sigma, q_{\texttt{start}}, \delta, Q \setminus F \rangle,$$

т.е. $\mathcal{A}'$ е същия като $\mathcal{A}$, с единствената разлика, че финалните състояния на $\mathcal{A}'$ са тези състояния, които **не** са финални в $\mathcal{A}$. □

Твърдение 3.4. Класът на автоматните езици е затворен относно операцията *обединение*. Това означава, че ако L_1 и L_2 са два произволни автоматни езика, то $L_1 \cup L_2$ също е автоматен език.

✍ Докажете, че така построения автомат $\mathcal{A}$ разпознава $L_1 \cup L_2$. Тук отново е важно, че δ_1 и δ_2 са тотални функции на преходите.

Упътване. Първият подход е да използваме конструкцията на автомата $\mathcal{A}$ от *Твърдение 3.2*, с единствената разлика, че тук избираме финалните състояния да бъдат елементите на множеството

$$\begin{aligned} F &\stackrel{\text{деф}}{=} \{\langle q_1, q_2 \rangle \in Q_1 \times Q_2 \mid q_1 \in F_1 \ \vee \ q_2 \in F_2\} \\ &= F_1 \times Q_2 \cup Q_1 \times F_2. \end{aligned}$$

Друг подход е да се използва правилото на Де Морган, а именно:

$$L_1 \cup L_2 = \overline{\overline{L_1} \cap \overline{L_2}}.$$

□

Като приложение на наученото тук, нека отново да разгледаме *Задача 3.2*.

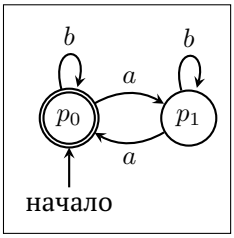
Задача 3.4. Докажете, че езикът L е автоматен, където:

$$L = \{\omega \in \{a, b\}^\star \mid |\omega|_a \equiv 0 \bmod 2 \ \& \ |\omega|_b \equiv 1 \bmod 2\}.$$

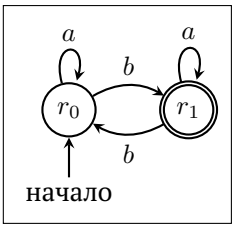
Упътване. Нека да представим езика L като $L = L_1 \cap L_2$, където

$$\begin{aligned} L_1 &= \{\omega \in \{a, b\}^\star \mid |\omega|_a \equiv 0 \bmod 2\} \\ L_2 &= \{\omega \in \{a, b\}^\star \mid |\omega|_b \equiv 1 \bmod 2\}. \end{aligned}$$

Да разгледаме следните детерминирани крайни автомати $\mathcal{A}_1$ и A_2:



(а) $\mathcal{L}(\mathcal{A}_1) = L_1$

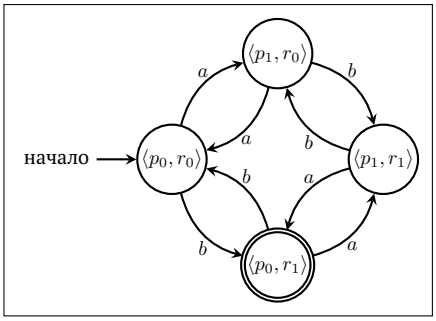


(б) $\mathcal{L}(\mathcal{A}_2) = L_2$

Ясно е, че $L = \mathcal{L}(\mathcal{A}_1) \cap \mathcal{L}(\mathcal{A}_2)$. Ще построим ДКА $\mathcal{A}$ като използваме конструкцията от *Твърдение 3.2*. Така финалното състояние на $\mathcal{A}$ ще бъде $\langle p_0, r_1 \rangle$, защото p_0 и r_1 са единствените финални състояния съответно на $\mathcal{A}_1$ и $\mathcal{A}_2$. Началното състояние на $\mathcal{A}$ ще бъде $\langle p_0, r_0 \rangle$, защото p_0 е началното състояние на A_1 и r_0 е началното състояние на $\mathcal{A}_2$. Сега, използвайки, че $\delta(\langle p_i, r_j \rangle, x) = \langle \delta_1(p_i, x), \delta_2(r_j, x) \rangle$, получаваме следния автомат $\mathcal{A}$.

✍ Съобразете сами, че $\mathcal{L}(\mathcal{A}_1) = L_1$ и $\mathcal{L}(\mathcal{A}_2) = L_2$!

Ако означим състоянията $\langle p_i, r_j \rangle$ като q_{ij}, то ще получим точно автомата от *Задача 3.2*.



Фигура 3.6: $\mathcal{L}(\mathcal{A}) = \{\omega \in \{a,b\}^\star \mid |\omega|_a \equiv 0 \bmod 2 \;\&\; |\omega|_b \equiv 1 \bmod 2\}$

□

3.2 Каноничен автомат

Тук виждаме, че езиците играят ролята на състояния на автомата на Бжозовски [19, стр. 112]. На автоматът на Бжозовски може да се гледа като на *каноничния* автомат за дадения език [8, стр. 53].

Нека е даден произволен език L. Ще покажем конструкция на детерминиран автомат

$$\mathcal{B}_L = \langle \Sigma, Q_L, q^L_{\texttt{start}}, \delta_L, F_L \rangle,$$

който разпознава L. Този автомат ще наречем **автомат на Бжозовски**. Ще дивим, че ако L е регулярен, то $\mathcal{B}_L$ ще бъде детерминиран краен автомат, но ако L не е регулярен, то $\mathcal{B}_L$ ще бъде детерминиран *безкраен* автомат. Конструкцията на автомата $\mathcal{B}_L$ е следната:

Q_L е същото множество, което е дефинирано в Раздел 2.6.

- Състоянията Q_L на автомата на Бжозовски ще бъдат езици над азбуката Σ, където:

$$Q_L \overset{\text{деф}}{=} \{\alpha^{-1}(L) \mid \alpha \in \Sigma^\star\}.$$

- Началното състояние $q_{\texttt{start}}$ на автомата на Бжозовски е дефинирано като:

$$q^L_{\texttt{start}} \overset{\text{деф}}{=} L \qquad // \; L = \varepsilon^{-1}(L).$$

- За произволно състояние (език) $M \in Q_L$ и буква a, дефинираме функцията на преходите ето така:

$$\delta_L(M, a) \overset{\text{деф}}{=} a^{-1}(M).$$

Да напомним, че Свойство (2.3) ни казва, че $L = \{\omega \in \Sigma^\star \mid \varepsilon \in \omega^{-1}(L)\}$.

- Финалните състояния на автомата на Бжозовски са следните:

$$F_L \overset{\text{деф}}{=} \{M \in Q_L \mid \varepsilon \in M\}.$$

Твърдение 3.5. За всяка дума α и всяко състояние $M \in Q_L$ е изпълнено равеството:

$$\delta^\star_L(M, \alpha) = \alpha^{-1}(M). \tag{3.4}$$

Доказателство.

- Твърдението очевидно е изпълнено за $\alpha = \varepsilon$, защото

$$\delta^\star_L(M, \varepsilon) \overset{\text{деф}}{=} M = \varepsilon^{-1}(M).$$

- Да примем, че за думи α с дължина n е изпълнено Свойство (3.4).
- Сега да разгледаме дума α с дължина $n+1$, т.е. $\alpha = \beta a$ и $|\beta| = n$. Тогава

$$\begin{aligned}
\delta^\star_L(M, \beta a) &= \delta_L(\delta^\star_L(M, \beta), a) && // \text{ от деф. на } \delta^\star_L \\
&= \delta_L(\beta^{-1}(M), a) && // \text{ от } \textbf{(И.П.)} \\
&= a^{-1}(\beta^{-1}(M)) && // \text{ от деф. на } \delta_L \\
&= (\beta a)^{-1}(M). && // \text{ от } \textit{Задача 2.10}
\end{aligned}$$

□

Твърдение 3.6. За произволен език L е изпълнено равенството $L = \mathcal{L}(\mathcal{B}_L)$.

Упътване. Съобразете, че имаме следните еквивалентности:

$$
\begin{aligned}
\alpha \in L &\Leftrightarrow \varepsilon \in \alpha^{-1}(L) && \text{// от Свойство (2.3)}\\
&\Leftrightarrow \varepsilon \in \delta_L^\star(L, \alpha) && \text{// от } \textit{Твърдение 3.5}\\
&\Leftrightarrow \delta_L^\star(q^L_{\mathtt{start}}, \alpha) \in F_L && \text{// } \varepsilon \in \alpha^{-1}(L)\\
&\Leftrightarrow \alpha \in \mathcal{L}(\mathcal{B}_L). && \text{// деф. на език на автомат}
\end{aligned}
$$

□

Тук е важно да отбележим, че все още не знаем дали $\mathcal{B}_L$ има крайно много състояния. В *Пример 3.2* ще видим един детерминиран безкраен автомат за език, който не е регулярен.

Примерни задачи

Задача 3.5. Постройте автомат $\mathcal{B}$ по метода на Бжозовски за регулярния език

$$L = \mathcal{L}[\![((\boldsymbol{a}+\boldsymbol{b})^+ \cdot \boldsymbol{a})^\star]\!].$$

Решение.

Да напомним, че $\boldsymbol{r}^+ \stackrel{\text{деф}}{=} \boldsymbol{r} \cdot \boldsymbol{r}^\star$. Очевидно е, че $\boldsymbol{r}^\star = \boldsymbol{\varepsilon} + \boldsymbol{r}^+$. Веднага се вижда, че L ще бъде и финално състояние, защото $\varepsilon \in L$. Често ще използваме съкратения запис αL вместо $\{\alpha\} \cdot L$.

- Ясно е, че ще започнем от началното състояние L. Удобно е да имаме предвид следното представяне, при което явно да се вижда кои думи на L започват с буквата a и кои с буквата b:

$$\begin{aligned} L &= (\{a,b\}^+ \cdot \{a\})^\star \\ &= \{\varepsilon\} \cup (\{a,b\}^+ \cdot \{a\})^+ \\ &= \{\varepsilon\} \cup \{a,b\}^+ \cdot \{a\} \cdot L \\ &= \{\varepsilon\} \cup \{a,b\} \cdot \{a,b\}^\star aL \\ &= \{\varepsilon\} \cup a\{a,b\}^\star aL \cup b\{a,b\}^\star aL \end{aligned}$$

- Сега като имаме това представяне на L, лесно се съобразява, че

$$a^{-1}(L) = b^{-1}(L) = \{a,b\}^\star aL.$$

Нека положим $M \stackrel{\text{деф}}{=} \{a,b\}^\star aL$. Лесно се съобразява, че $M \neq L$, защото $\varepsilon \in L$, но $\varepsilon \notin M$. Това означава, че имаме ново състояние M и

$$\delta(L,a) \stackrel{\text{деф}}{=} M \qquad \text{// защото } a^{-1}(L) = M$$
$$\delta(L,b) \stackrel{\text{деф}}{=} M. \qquad \text{// защото } b^{-1}(L) = M$$

Веднага се вижда, че M няма да бъде финално състояние.

- За следващата стъпка е удобно да представим езика M по следния начин:

$$\begin{aligned} M &= \{a,b\}^\star aL \\ &= aL \cup \{a,b\}^+ aL \\ &= aL \cup \{a,b\} \cdot \{a,b\}^\star aL \\ &= aL \cup \{a,b\} \cdot M \\ &= aL \cup aM \cup bM \end{aligned}$$

От това представяне на M веднага се съобразява, че

$$a^{-1}(M) = L \cup M$$
$$b^{-1}(M) = M.$$

Обърнете внимание, че $L \subset N$, но L и N са различни състояния.

Нека да положим $N \stackrel{\text{деф}}{=} L \cup M$. Имаме, че $N \neq L$, защото $a \in N$, но $a \notin L$. Освен това, $N \neq M$, защото $\varepsilon \in N$, но $\varepsilon \notin M$. Това означава, че имаме ново състояние N и тогава

$$\delta(M,a) \stackrel{\text{деф}}{=} N \qquad \text{// защото } a^{-1}(M) = N$$
$$\delta(M,b) \stackrel{\text{деф}}{=} M \qquad \text{// защото } b^{-1}(M) = M.$$

- Да разгледаме следното представяне:

$$\begin{aligned} N &= L \cup M \\ &= \{\varepsilon\} \cup aM \cup bM \cup M \\ &= \{\varepsilon\} \cup aM \cup bM \cup aL \\ &= \{\varepsilon\} \cup aN \cup bM. \end{aligned}$$

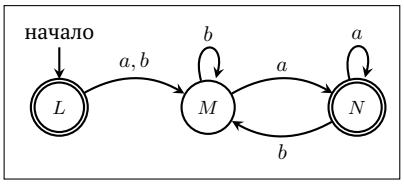
Веднага можем да съобразим, че

$$a^{-1}(N) = N$$
$$b^{-1}(N) = M.$$

Сега полагаме:

$$\delta(N,a) \stackrel{\text{деф}}{=} N \qquad \text{// защото } a^{-1}(N) = N$$
$$\delta(N,b) \stackrel{\text{деф}}{=} M \qquad \text{// защото } b^{-1}(N) = M.$$

- Нямаме повече нови състояния. Следователно, $Q \stackrel{\text{деф}}{=} \{L, M, N\}$.
- Понеже $\varepsilon \in L$ и $\varepsilon \in N$ е ясно, че $F = \{L, N\}$. Сега вече сме готови да нарисуваме картинка на автомата.



Фигура 3.7: Автомат на Бжозовски за езика L.

□

Задача 3.6. Постройте автомат $\mathcal{B}$ по метода на Бжозовски, който рапознава регулярния език

$$L = \mathcal{L}[\![\mathbf{a} \cdot (\mathbf{a} + \mathbf{b})^\star \cdot \mathbf{b}]\!].$$

Решение.

- Започваме с началното състояние L, за което е ясно, че няма да бъде финално, защото $\varepsilon \notin L$. Първата стъпка е лесна, защото е явно, че всички думи на L започват с буквата a.
- $a^{-1}(L) = \{a,b\}^\star b \stackrel{\text{деф}}{=} M$.
  Имаме, че $M \neq L$, защото $b \in M$, но $b \notin L$. Тогава

  $$\delta(L,a) \stackrel{\text{деф}}{=} M \qquad \text{// защото } a^{-1}(L) = M$$
  $$\delta(L,b) \stackrel{\text{деф}}{=} \emptyset \qquad \text{// защото } b^{-1}(L) = \emptyset$$

  $\emptyset$ е съвсем нормален език и напълно нормално да имаме състояние $\emptyset$.
- За по-нататък ще е удобно да представим M по следния начин, така че да се вижда кои думи на M започват с буквата a и кои с буквата b:

  $$\begin{aligned} M &= \{a,b\}^\star b \\ &= \{a,b\}^+ b \cup \{b\} \\ &= a \cdot \{a,b\}^\star \cdot b \cup b \cdot \{a,b\}^\star \cdot b \cup \{b\} \\ &= aM \cup bM \cup \{b\}. \end{aligned}$$

  Сега е ясно, че

  $$a^{-1}(M) = M$$
  $$b^{-1}(M) = \{\varepsilon\} \cup M.$$

  Нека да положим $N \stackrel{\text{деф}}{=} \{\varepsilon\} \cup M$. Имаме, че $N \neq L$ и $N \neq M$, защото $\varepsilon \in N$, но $\varepsilon \notin L$ и $\varepsilon \notin M$. Тогава

  $$\delta(M,a) \stackrel{\text{деф}}{=} M \qquad \text{// защото } a^{-1}(M) = M$$
  $$\delta(M,b) \stackrel{\text{деф}}{=} N \qquad \text{// защото } b^{-1}(M) = N$$

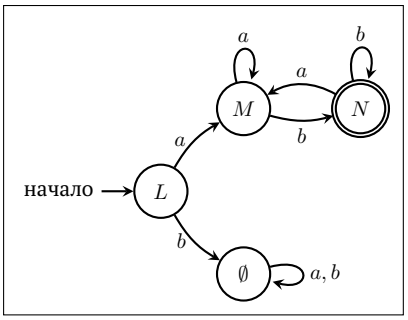
- Понеже $N = \{\varepsilon\} \cup aM \cup bM \cup \{b\}$, то

  $$\delta(N,a) \stackrel{\text{деф}}{=} M \qquad \text{// защото } a^{-1}(N) = M$$
  $$\delta(N,b) \stackrel{\text{деф}}{=} N \qquad \text{// защото } b^{-1}(N) = M.$$

- Имаме примки за състоянието $\emptyset$:

  $$\delta(\emptyset,a) \stackrel{\text{деф}}{=} \emptyset \qquad \text{// защото } a^{-1}(\emptyset) = \emptyset$$
  $$\delta(\emptyset,b) \stackrel{\text{деф}}{=} \emptyset \qquad \text{// защото } b^{-1}(\emptyset) = \emptyset.$$

- Съобразете сами, че $F = \{N\}$.



Фигура 3.8: Автомат за $\mathcal{L}[\![\boldsymbol{a} \cdot (\boldsymbol{a} + \boldsymbol{b})^\star \cdot \boldsymbol{b}]\!]$ по метода на Бжозовски.

□

Задача 3.7. Да фиксираме азбуката $\Sigma = \{a,b\}$. Постройте автомат по метода на Бжозовски за регулярния език

$$L = \{\omega \in \Sigma^\star \mid |\omega|_a \text{ е четно и } |\omega|_b = 1\}.$$

Решение.

Да започнем с преходите от началното състояние L:

$$\begin{aligned} a^{-1}(L) &= \{\omega \in \Sigma^\star \mid |\omega|_a \text{ е нечет. и } |\omega|_b = 1\} \\ &\stackrel{\text{деф}}{=} M \\ b^{-1}(L) &= \{\omega \in \Sigma^\star \mid |\omega|_a \text{ е четно и } |\omega|_b = 0\} \\ &\stackrel{\text{деф}}{=} N. \end{aligned}$$

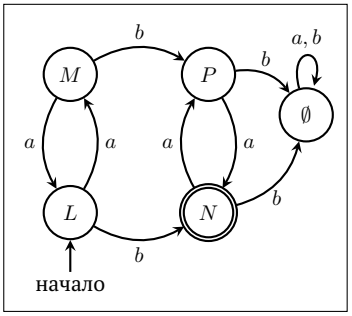
Лесно се вижда, че M и N са различни езици от L. Например, $aab \in L$, но $aab \notin M$; $aa \in N$, но $aa \notin L$ и $aa \notin M$. Сега да видим какви преходи имаме от състоянието M:

$$\begin{aligned} a^{-1}(M) &= L; \\ b^{-1}(M) &= \{\omega \in \Sigma^\star \mid |\omega|_a \text{ е нечет. и } |\omega|_b = 0\} \\ &\stackrel{\text{деф}}{=} P. \end{aligned}$$

Ясно е, че $P \neq M, L, N$, защото $a \in P$, но $a \notin M, L, N$. Завършваме с преходите от състоянията N и P:

$$a^{-1}(N) = P \quad \text{и} \quad b^{-1}(N) = \emptyset$$
$$a^{-1}(P) = N \quad \text{и} \quad b^{-1}(P) = \emptyset.$$

Вече сме достатъчно опитни за да ни е очевидно, че L е регулярен. Да напомним, че $|\omega|_a$ означава броя на срещанията на a в думата ω.



Фигура 3.9: Автомат, който приема думи с четен брой a и точно едно b, получен чрез метода на Бжозовски.

□

Задача 3.8. Постройте автомат $\mathcal{B}$ по метода на Бжозовски, който разпознава езика

$$L = \{\alpha \in \{0,1\}^\star \mid \overline{\alpha}_{(2)} \equiv 2 \bmod 3\}.$$

Да припомним, че в *Задача 3.3* се искаше да се докаже, че същият език е регулярен. Ние направихме това като построихме автомат за L и доказахме, че той разпознава L.

Решение.

За целта ще ни трябва алтернативна дефиниция на $\overline{\alpha}_{(2)}$. За една дума $\alpha \in \{0,1\}^\star$, можем да дадем следната дефиниция на $\overline{\alpha}_{(2)}$:

- $\overline{\varepsilon}_{(2)} = 0$,
- $\overline{0\alpha}_{(2)} = \overline{\alpha}_{(2)}$,
- $\overline{1\alpha}_{(2)} = 2^{|\alpha|} + \overline{\alpha}_{(2)}$.

Лесно се съобразява, че началното състояние L на автомата на Бжозовски не е финално, защото $\varepsilon \notin L$. Да започнем с преходите от началното състояние:

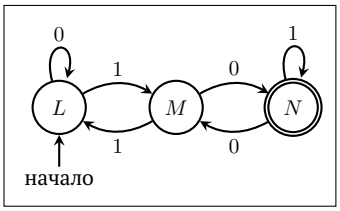
$$\begin{aligned}
0^{-1}(L) &= \{\alpha \mid 0\alpha \in L\} \\
&= \{\alpha \mid \overline{0\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= L. \\
1^{-1}(L) &= \{\alpha \mid 1\alpha \in L\} \\
&= \{\alpha \mid \overline{1\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&\overset{\text{деф}}{=} M.
\end{aligned}$$

Лесно се съобразява, че езикът M е различен от L, защото например думата $10 \in L$, но $10 \notin M$. Също така, понеже $2^{|\varepsilon|} + \overline{\varepsilon}_{(2)} = 1$, то $\varepsilon \notin M$ и следователно M няма да бъде финално състояние в автомата на Бжозовски. Продължаваме нататък:

$$\begin{aligned}
0^{-1}(M) &= \{\alpha \mid 0\alpha \in M\} \\
&= \{\alpha \mid 2^{|0\alpha|} + \overline{0\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 2 \cdot 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&\overset{\text{деф}}{=} N. \\
1^{-1}(M) &= \{\alpha \mid 1\alpha \in M\} \\
&= \{\alpha \mid 2^{|1\alpha|} + \overline{1\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 2 \cdot 2^{|\alpha|} + 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 3 \cdot 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= L.
\end{aligned}$$

Сега трябва да се уверим, че езикът N е различен от L и M. Това отново е лесно. Непосредствено се проверява, че думата $11 \in N$, но $11 \notin L$ и $11 \notin M$. Също така да отбележим, че понеже $2.2^{|\varepsilon|} + \overline{\varepsilon}_{(2)} = 2$, то N ще бъде финално състояние в автомата на Бжозовски, защото $\varepsilon \in N$. Продължаваме нататък с преходите от N:

$$\begin{aligned}
0^{-1}(N) &= \{\alpha \mid 0\alpha \in N\} \\
&= \{\alpha \mid 2 \cdot 2^{|0\alpha|} + \overline{0\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 4 \cdot 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 3 \cdot 2^{|\alpha|} + 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= M \\
1^{-1}(N) &= \{\alpha \mid 1\alpha \in N\} \\
&= \{\alpha \mid 2 \cdot 2^{|1\alpha|} + \overline{1\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 4 \cdot 2^{|\alpha|} + 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 3 \cdot 2^{|\alpha|} + 2 \cdot 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= \{\alpha \mid 2 \cdot 2^{|\alpha|} + \overline{\alpha}_{(2)} \equiv 2 \bmod 3\} \\
&= N.
\end{aligned}$$



Фигура 3.10: Автомат на Бжозовски за езика L.

□

Ние формално не сме дефинирали понятието *безкраен* детерминиран автомат. Този пример се разглежда и в [19, стр. 113].

Пример 3.2. Да разгледаме езика $L = \{a^n b^n \mid n \in \mathbb{N}\}$. Ние знаем от *Пример 3.8*, че L не е регулярен език. Да се опитаме да построим автомат, който го разпознава.

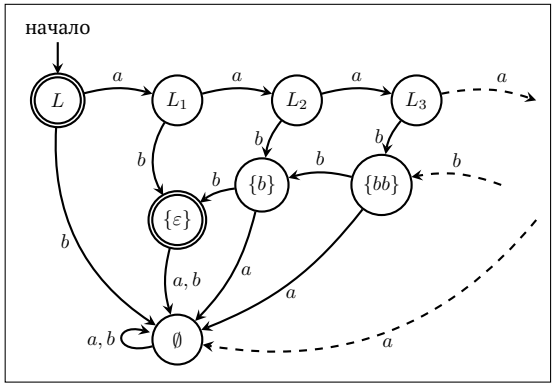
Нека за всяко k да разгледаме езика

$$L_k \overset{\text{деф}}{=} \{a^n b^{n+k} \mid n \in \mathbb{N}\}.$$

Да видим какво се получава като приложим процедурата за строене на минимален автомат.

- $a^{-1}(L) = L_1$ и $b^{-1}(L) = \emptyset$;
- $a^{-1}(L_1) = L_2$ и $b^{-1}(L_1) = \{\varepsilon\}$;
- $a^{-1}(\{\varepsilon\}) = b^{-1}(\{\varepsilon\}) = \emptyset$;
- Лесно можем да докажем, че за всяко k е изпълнено, че $a^{-1}(L_k) = L_{k+1}$.
- Лесно се вижда, че $b^{-1}(L_{k+1}) = \{b^k\}$, за всяко k.
- Ясно е, че $b^{-1}(\{b^k\}) = \{b^{k-1}\}$, за $k \geq 1$.

Получаваме, че езикът L се разпознава от автомат с *безкрайно много състояния*.



Фигура 3.11: *Безкраен* автомат за езика $\{a^n b^n \mid n \in \mathbb{N}\}$ построен по метода на Бжозовски.

3.3 Регулярните езици са автоматни (алгебричен подход)

Нашата цел в този раздел ще бъде да покажем как като използваме директно конструкцията на Бжозовски можем да покажем, че автоматните езици са затворени относно операциите обединение, конкатенация и звезда на Клини. За щастие, ние вече сме свършили тежката работа в *Теорема 2.2*. Нека да напомним какво ни казва тя.

Теорема 3.1 (*Теорема 2.2*)**.** Ако L е регулярен език, то Q_L е крайно множество.

Теорема 3.2 (Клини[12])**.** Всеки регулярен език е автоматен.

Алтернативно доказателство на тази теорема е дадено в *Раздел 3.7*.

Доказателство. Знаем от *Твърдение 3.6*, че $\mathcal{L}(\mathcal{B}_L) = L$. Щом L е регулярен език, то от *Теорема 2.2* имаме, че Q_L е крайно множество и следователно $\mathcal{B}_L$ е ДКА. □

Следствие 3.1 (Критерий за регулярност на език)**.** Един език L е регулярен точно тогава, когато автоматът $\mathcal{B}_L$ има крайно много състояния.

Доказателство. Първо, ако L е регулярен, то от *Теорема 2.2* веднага имаме, че $\mathcal{B}_L$ е ДКА.

Второ, ако $\mathcal{B}_L$ има крайно много състояния, то $\mathcal{B}_L$ е ДКА. Понеже $\mathcal{L}(\mathcal{B}_L) = L$ според *Твърдение 3.6*, то L е автоматен и следователно регулярен по теоремата на Клини, за която имаме доказателство с алгоритмичен подход и алгебричен подход. □

Използвайки подобни разсъждения, можем директно да съобразим, че регулярните езици са затворени и относно операциите сечение и допълнение.

Следствие 3.2. Ако L_1 и L_2 са регулярни езици, то $L_1 \cap L_2$ и $L_1 \setminus L_2$ са регулярни езици.

Ние знаем, че автоматните езици са затврени относно сечение и разлика чрез декартови конструкции на автомати.
Сравнете с *Твърдение 3.2*.

Доказателство. Първо, съобразете, че $\omega^{-1}(L_1 \cap L_2) = \omega^{-1}(L_1) \cap \omega^{-1}(L_2)$. Това означава, че

$$Q_{L_1 \cap L_2} \subseteq \{M_1 \cap M_2 \mid M_1 \in Q_{L_1} \ \& \ M_2 \in Q_{L_2}\}.$$

Следователно, $|Q_{L_1 \cap L_2}| \leq |Q_{L_1} \times Q_{L_2}| = |Q_{L_1}| \cdot |Q_{L_2}|$.

Второ, съобразете, че $\omega^{-1}(L_1 \setminus L_2) = \omega^{-1}(L_1) \setminus \omega^{-1}(L_2)$. Оттук, отново получаваме, че

$$Q_{L_1 \setminus L_2} \subseteq \{M_1 \setminus M_2 \mid M_1 \in Q_{L_1} \ \& \ M_2 \in Q_{L_2}\}.$$

Следователно, $|Q_{L_1 \setminus L_2}| \leq |Q_{L_1} \times Q_{L_2}| = |Q_{L_1}| \cdot |Q_{L_2}|$. □

3.4 Недетерминирани крайни автомати

В този раздел ще разгледаме едно обобщение на детерминираните крайни автомати, при които ще позволяваме от дадено състояние и дадена буква да отиваме в много различни състояния, а не само едно. Това е пример **недетерминизъм** - важно понятие в информатиката. Най-общо казано, недетерминизмът настъпва, когато попаднем в ситуация, при която следващата стъпка в изчислението не е еднозначно определено от текущото състояние. В нашия конкретен случай - текущото състояние е просто един елемент на множеството Q. При по-сложни модели, текущото състоянието ще бъде по-сложен обект. Това няма да бъде единствения ни сблъсък с този феномен.

Определение 3.3. Недетерминиран краен автомат представлява петорка от вида

$$\mathcal{N} = \langle \Sigma, Q, Q_{\texttt{start}}, \Delta, F \rangle,$$

Въведени от Рабин и Скот [18]. За по-голяма яснота, често ще означаваме с $\mathcal{N}$ недетерминирани автомати, като ще запазим $\mathcal{A}$ за детерминираните. Мотивация - [13, стр. 25].

където

- Q е крайно множество от състояния;
- Σ е крайна азбука;
- $\Delta : Q \times \Sigma \to \mathscr{P}(Q)$ е функцията на преходите. Да обърнем внимание, че е възможно за някоя двойка (q, a) да няма нито един преход в автомата. Това е възможно, когато $\Delta(q, a) = \emptyset$;
- $Q_{\texttt{start}} \subseteq Q$ е множество от начални състояния;
- $F \subseteq Q$ е множеството от финални състояния.

Да напомним, че $\mathscr{P}(Q) \stackrel{\text{деф}}{=} \{R \mid R \subseteq Q\}$, $|\mathscr{P}(Q)| = 2^{|Q|}$ Съобразете, че в този случай можем да си мислим за Δ като релация.

Удобно е да разширим функцията на преходите $\Delta : Q \times \Sigma \to \mathscr{P}(Q)$ до функцията $\Delta^\star : \mathscr{P}(Q) \times \Sigma^\star \to \mathscr{P}(Q)$, която дефинираме за произволно множество от състояния $R \subseteq Q$ и дума $\alpha \in \Sigma^\star$ по следния начин:

В [16] Δ е релация и се позволяват ε-преходи. В [21] пък е функция, но пак се позволяват ε-преходи. В [10] е функция без ε-преходи. Навсякъде има само едно начално.

- Ако $\alpha = \varepsilon$, то
$$\Delta^\star(R, \varepsilon) \stackrel{\text{деф}}{=} R.$$
- Ако $\alpha = \beta a$, то
$$\Delta^\star(R, \beta a) \stackrel{\text{деф}}{=} \bigcup \{\Delta(p, a) \mid p \in \Delta^\star(R, \beta)\}.$$

Дефиницията на език разпознаван от недерерминиран краен автомат $\mathcal{N}$ е малко по-сложна от тази за ДКА.

Понеже операцията $\bigcup$ е сложна, да напомним, че $\bigcup\{\{0,1\},\{1,2,3\}\} = \{0,1\} \cup \{1,2,3\}$.

$$\mathcal{L}(\mathcal{N}) \stackrel{\text{деф}}{=} \{\omega \in \Sigma^\star \mid \Delta^\star(Q_{\texttt{start}}, \omega) \cap F \neq \emptyset\}.$$

Удачно е да дефинираме език на автомата $\mathcal{N}$, ако приемем, че множеството от състояния R са начални. Това можем да направим така:

$$\mathcal{L}_{\mathcal{N}}(R) = \{\omega \in \Sigma^\star \mid \Delta^\star(R, \omega) \cap F \neq \emptyset\}.$$

Ако $R = \{q\}$, удобно е да пишем просто $\mathcal{L}_{\mathcal{N}}(q)$ вместо тромавия запис $\mathcal{L}_{\mathcal{N}}(\{q\})$.

Твърдение 3.7. За всеки две думи $\alpha, \beta \in \Sigma^\star$ и всяко $R \subseteq Q$,

$$\Delta^\star(R, \alpha\beta) = \Delta^\star(\Delta^\star(R, \alpha), \beta).$$

Сравенете с *Твърдение 3.1*.

Упътване. Индукция по дължината на β. □

✍ Трябва да може да докажете това твърдение сами!

Доказателство. Ще докажем, че $(\forall n)P(n)$ с индукция по n, където

$$P(n) \stackrel{\text{деф}}{=} (\forall \beta \in \Sigma^n)(\forall \alpha \in \Sigma^\star)[\Delta^\star(R, \alpha\beta) = \Delta^\star(\Delta^\star(R, \alpha), \beta)].$$

- Нека $|\beta| = 0$, т.е. $\beta = \varepsilon$. Тогава:

$$\begin{aligned} \Delta^\star(R, \alpha\varepsilon) &= \Delta^\star(R, \alpha) && \text{// } \alpha\varepsilon = \alpha \\ &= \Delta^\star(\Delta^\star(R, \alpha), \varepsilon). && \text{// деф. на } \Delta^\star \end{aligned}$$

- Да приемем, че твърдението е вярно за думи β с дължина n.
- Нека $|\beta| = n + 1$, т.е. $\beta = \gamma b$, където $|\gamma| = n$.

$$\begin{aligned} \Delta^\star(R, \alpha\gamma b) &= \bigcup\{\Delta(p, b) \mid p \in \Delta^\star(R, \alpha\gamma)\} && \text{// от деф. на } \Delta^\star \\ &= \bigcup\{\Delta(p, b) \mid p \in \Delta^\star(\underbrace{\Delta^\star(R, \alpha)}_{U}, \gamma))\} && \text{// от И.П. за } \gamma \\ &= \bigcup\{\Delta(p, b) \mid p \in \Delta^\star(U, \gamma)\} && \text{// нека } U \stackrel{\text{деф}}{=} \Delta^\star(R, \alpha) \\ &= \Delta^\star(U, \gamma b) && \text{// от деф. на } \Delta^\star \\ &= \Delta^\star(\Delta^\star(R, \alpha), \gamma b) && \text{// } U = \Delta^\star(R, \alpha) \end{aligned}$$

□

(На англ. *instantaneous description*). В случая въвеждането на това понятие не е напълно необходимо, но по-късно, когато въведем стекови автомати и машини на Тюринг, ще работим с такива конфигурации на изчисления и затова е добре още отначало да свикнем с това понятие.

Конфигурация (или моментното описание) представлява описание на текущото състояние на едно изчисление с краен автомат. То 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 $\vdash_{\mathcal{N}}$ над $Q \times \Sigma^\star$, която ще ни казва как моментното описание (конфигурацията) на автомата $\mathcal{N}$ се променя след изпълнение на една стъпка.

$$\frac{p \in \Delta(q, a)}{(q, a\beta) \vdash_{\mathcal{N}} (p, \beta)}$$

Фигура 3.12: Едностъпков преход в недетерминиран краен автомат $\mathcal{N}$

Рефл. и транз. затваряне на една релация е разгледано в Глава 1. Тук $\vdash^\star_{\mathcal{N}}$ е рефлексивното и транзитивно затваряне на релацията $\vdash_{\mathcal{N}}$.

Ще дефинираме релацията $\vdash^\ell_{\mathcal{N}}$, която определя работата на автомата $\mathcal{N}$ за ℓ на брой стъпки.

$$\frac{}{\kappa \vdash^0_{\mathcal{N}} \kappa} \qquad \frac{\kappa \vdash_{\mathcal{N}} \kappa'' \qquad \kappa'' \vdash^\ell_{\mathcal{N}} \kappa'}{\kappa \vdash^{\ell+1}_{\mathcal{N}} \kappa'}$$

Сега можем да дефинираме $\vdash^\star_{\mathcal{N}}$ като:

$$(q, \alpha) \vdash^\star_{\mathcal{N}} (p, \beta) \stackrel{\text{деф}}{\Leftrightarrow} (\exists \ell \in \mathbb{N})[(q, \alpha) \vdash^\ell_{\mathcal{N}} (p, \beta)].$$

Друг начин да дефинираме релацията $\vdash^\star_{\mathcal{N}}$ е следния: $(q, \alpha\beta) \vdash^\star_{\mathcal{N}} (p, \beta)$ точно тогава, когато $p \in \Delta^\star(\{q\}, \alpha)$.

Получаваме, че

$$\mathcal{L}(\mathcal{N}) = \{\alpha \in \Sigma^\star \mid (\exists q \in Q_{\texttt{start}})(\exists f \in F)[(q, \alpha) \vdash^\star_{\mathcal{N}} (f, \varepsilon)]\}.$$

Теорема 3.3 (Рабин-Скот [18]). За всеки НКА $\mathcal{N}$ съществува еквивалентен на него ДКА $\mathcal{D}$, т.е.

$$\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{D}).$$

Упътване. Нека $\mathcal{N} = \langle \Sigma, Q, Q_{\texttt{start}}, \Delta, F \rangle$. Ще построим детерминиран автомат

$$\mathcal{D} = (\Sigma, Q', q_{\texttt{start}}, \delta, F'),$$

за който $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{D})$. Конструкцията е следната:

- $Q' \stackrel{\text{деф}}{=} \{R \mid R \subseteq Q\}$;
- За произволна буква $a \in \Sigma$ и произволно $R \subseteq Q$,

$$\delta(\underbrace{R}_{\text{състояние}}, a) \stackrel{\text{деф}}{=} \Delta^\star(\underbrace{R}_{\text{множество}}, a).$$

- $q_{\texttt{start}} \stackrel{\text{деф}}{=} Q_{\texttt{start}}$;
- $F' \stackrel{\text{деф}}{=} \{R \in Q' \mid R \cap F \neq \emptyset\}$.

Да отбележим, че детерминираният автомат $\mathcal{D}$ има не повече от $2^{|Q|}$ на брой състояния. Реално на нас ни трябват само тези множества $R \subseteq Q$, за които съществува дума α и $\Delta^\star(Q_{\texttt{start}}, \alpha) = R$. Тук използваме, че

$$\Delta^\star(R, a) = \bigcup_{q \in R} \Delta(q, a).$$

Ще докажем, че за произволна дума α и произволно множество $R \subseteq Q$ е изпълнено следното равенство:

$$\Delta^\star(\underbrace{R}_{\text{множество}}, \alpha) = \delta^\star(\underbrace{R}_{\text{състояние}}, \alpha). \tag{3.5}$$

Това ще направим с индукция по дължината на думата α.

- Ако $|\alpha| = 0$, т.е. $\alpha = \varepsilon$, то е ясно от дефиницията на $\Delta^\star$ и $\delta^\star$, че за всяко $R \subseteq Q$ е изпълнено:

$$\Delta^\star(R, \varepsilon) = R = \delta^\star(R, \varepsilon).$$

- Да приемем, че (3.5) е изпълнено за думи α с дължина n, т.е.

$$(\forall \alpha \in \Sigma^n)(\forall R \subseteq Q)[\, \Delta^\star(R, \alpha) = \delta^\star(R, \alpha)\,].$$

- Нека сега α има дължина $n+1$, т.е. $\alpha = \beta a$, където $|\beta| = n$ и $a \in \Sigma$.

$$\begin{aligned}
\delta^\star(R, \beta a) &= \delta(\delta^\star(R, \beta a)) && \text{// деф. на } \delta^\star \\
&= \delta(\Delta^\star(R, \beta), a) && \text{// от } \textbf{(И.П.)} \text{ за } \beta \\
&= \Delta^\star(\Delta^\star(R, \beta), a) && \text{// от деф. на } \delta \\
&= \Delta^\star(R, \beta a). && \text{// от } \textit{Твърдение 3.7}
\end{aligned}$$

Така доказахме, че

$$(\forall \alpha \in \Sigma^{n+1})(\forall R \subseteq Q)[\, \Delta^\star(R, \alpha) = \delta^\star(R, \alpha)\,].$$

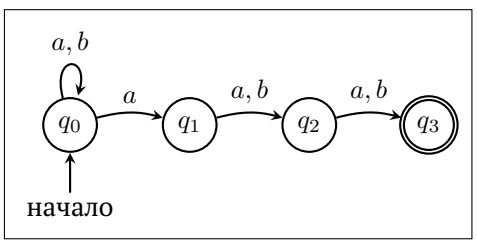
Това означава, че според принципа на математическата индукция имаме Свойство (3.5).

Сега вече е лесно да съобразим, че

$$\begin{aligned}
\omega \in \mathcal{L}(\mathcal{D}) &\Leftrightarrow \delta^\star(q_{\texttt{start}}, \omega) \in F' && \text{// деф. на } \mathcal{L}(\mathcal{D}) \\
&\Leftrightarrow \Delta^\star(Q_{\texttt{start}}, \omega) \cap F \neq \emptyset && \text{// от (3.5)} \\
&\Leftrightarrow \omega \in \mathcal{L}(\mathcal{N}) && \text{// деф. на } \mathcal{L}(\mathcal{N}).
\end{aligned}$$

□

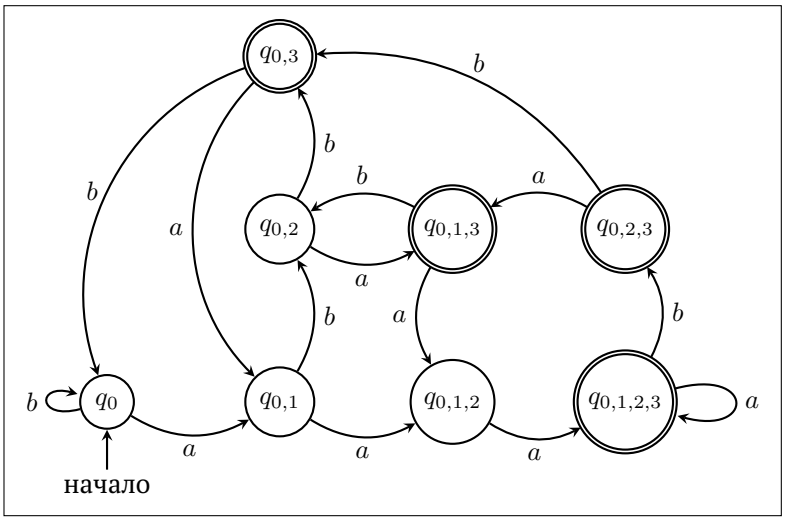
Пример 3.3. Да разгледаме следния недетерминиран краен автомат $\mathcal{N}$ зададен на *Фигура 3.14*.



Фигура 3.14: Недетерминиран автомат $\mathcal{N}$ за езика съставен думите, в които има a на трета позиция от дясно на ляво.

Нека да приложим конструкцията на Рабин-Скот за получаване на ДКА $\mathcal{D}$ разпознаващ езика на $\mathcal{N}$.

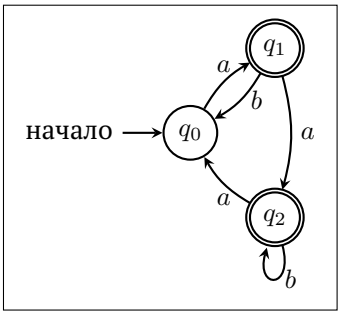
Тук индексите на състоянията на новия автомат кодират множеството от състояния на първоначалния автомат. Например, $q_{0,1}$ кодира състоянието $\{q_0, q_1\}$. В следващия раздел ще видим, че не можем да получим детерминиран автомат с по-малко състояния за този език.



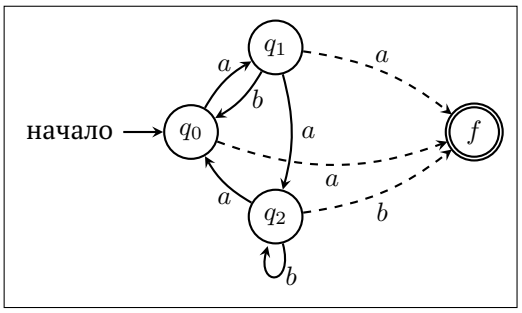
Фигура 3.15: Детерминиран автомат $\mathcal{D}$ за езика съставен думите, в които има a на трета позиция от дясно на ляво.

Задача 3.9. Докажете, че за всеки недетерминиран краен автомат $\mathcal{N}$ съществува недетерминиран краен автомат $\mathcal{N}'$ 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 $\mathcal{N}'$, за който $\mathcal{L}(\mathcal{N}') = \mathcal{L}(\mathcal{N})$

Обърнете внимание, че примерът показва, че е възможно $\mathcal{N}$ да е детерминиран автомат, но полученият $\mathcal{N}'$ да бъде недетерминиран. □

Задача 3.10. Докажете, че за всеки недетерминиран краен автомат $\mathcal{N}$ съществува недетерминиран краен автомат $\mathcal{N}'$ с *едно начално* състояние, за който $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{N}')$.

Задача 3.11. Докажете, че автоматните езици са затворени относно операцията `rev`. С други думи, докажете, че ако L е автоматен език, то $L^{\texttt{rev}} \stackrel{\text{деф}}{=} \{\omega^{\texttt{rev}} \mid \omega \in L\}$ също е автоматен.

Вече знаем, че ако L е регулярен, то $L^{\texttt{rev}}$ е регулярен.

Упътване. Идеята е съвсем проста - просто обръщаме стрелките и правим финалните състояния да са начални, а началното става финално. Нека $L = \mathcal{L}(\mathcal{A})$. Ще построим НКА $\mathcal{N}$, за който $\mathcal{L}(\mathcal{N}) = L^{\texttt{rev}}$.

- $Q_{\mathcal{N}} \stackrel{\text{деф}}{=} Q_{\mathcal{A}}$;

- $Q^{\mathcal{N}}_{\mathtt{start}} \stackrel{\text{деф}}{=} F_{\mathcal{A}}$;
- $\Delta_{\mathcal{N}}(q,a) \stackrel{\text{деф}}{=} \{p \in Q_{\mathcal{A}} \mid \delta_{\mathcal{A}}(p,a) = q\}$;
- $F_{\mathcal{N}} \stackrel{\text{деф}}{=} \{q_{\mathtt{start}}\}$.

Достатъчно е да се докаже, че $\Delta^{\star}_{\mathcal{N}}(Q^{\mathcal{N}}_{\mathtt{start}}, \alpha) = \{q \in Q_{\mathcal{A}} \mid \delta^{\star}_{\mathcal{A}}(q, \alpha^{\mathtt{rev}}) \in F_{\mathcal{A}}\}$. □

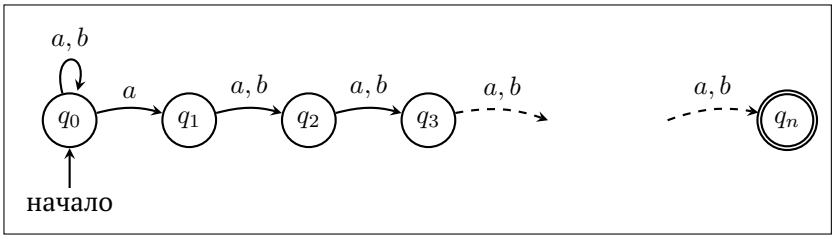
3.4.1 Експоненциална експлозия

Тук следваме [11, стр. 66] и [1, стр. 164]. В [20, стр. 80] има друг пример за НКА с n състояния вместо $n+1$, но доказателството изглежда по-сложно.

Сега ще видим, че всеки алгоритъм за детерминизация е експоненциален по време и памет.

Твърдение 3.8. Съществува НКА $\mathcal{N}$ с $n+1$ състояния, за който не съществува ДКА $\mathcal{A}$ с по-малко от 2^n състояния.

Упътване. Да разгледаме следния недетерминиран краен автомат $\mathcal{N}$ зададен на *Фигура 3.17*.



Фигура 3.17: Недетерминиран автомат $\mathcal{N}$ за езика съставен думите, в които има a на n-та позиция от дясно на ляво.

Лесно се съобразява, че за $n > 0$, недетерминирният краен автомат $\mathcal{N}$ на *Фигура 3.17* с $n+1$ на брой състояния разпознава езика

$$L = \{\lambda \cdot a \cdot \rho \in \{a,b\}^\star \mid |\rho| = n-1\}.$$

Езикът L може да се представи и така:

$$L = \{a,b\}^\star \cdot \{a\} \cdot \{a,b\}^{n-1}.$$

Нека да съобразим, че не е възможно да съществува краен детерминиран автомат $\mathcal{A} = \langle \Sigma, Q, q_{\texttt{start}}, \delta, F\rangle$ разпознаващ същия език L с по-малко от 2^n състояния.

Да допуснем, че $|Q| < 2^n$. От принципа на Дирихле имаме, че съществуват две различни думи α и β с дължина n, за които съществува $q \in Q$ и

Да напомним, че броят на всички думи с дължина n над азбука с k букви е k^n. В нашия случай, $k = 2$.

$$\delta^\star(q_{\texttt{start}}, \alpha) = q = \delta^\star(q_{\texttt{start}}, \beta).$$

Нека *първата* разлика в тези две думи е на позиция $i < n$, т.е. думите α и β могат да се представят така:

$$\alpha = \lambda \cdot \alpha[i] \cdot \rho \text{ и } \beta = \lambda \cdot \beta[i] \cdot \rho'.$$

Без ограничение на общността, нека $\alpha[i] = a$ и $\beta[i] = b$.

Тук $\omega = \varepsilon$.

- Ако $i = 0$. Това означава, че $\alpha \in L$, но $\beta \notin L$. Следователно състоянието q е едновременно финално и нефинално. Това е противоречие.

Съобразете, че тук $|\lambda| = i$ и $|\rho| = n - i - 1$. Не е важно как разширяваме α и β за да получим α_0 и β_0. Тук, за да бъдем конкретни, сме избрали разширението да е просто a^i. Важното е това разширение да има дължина i.

- Ако $i > 0$. Да разгледаме следните думи:

$$\alpha_0 = \alpha \cdot a^i$$
$$\beta_0 = \beta \cdot a^i.$$

Тогава $\alpha_0 = \lambda \cdot a \cdot \rho_0$, където $|\rho_0| = n-1$. Аналогично, $\beta_0 = \lambda \cdot b \cdot \rho_1$, където $|\rho_1| = n-1$. Така отново получаваме, че $\alpha_0 \in L$, но $\beta_0 \notin L$. И в този случай получаваме противоречие, защото

$$\delta^\star(q_{\texttt{start}}, \alpha_0) = p = \delta^\star(q_{\texttt{start}}, \beta_0)$$

и състоянието p трябва да е едновременно финално и нефинално.

□

3.5 Фундаментална теорема за автоматните езици

Сега ще видим една естествена характеризация на езиците $\mathcal{L}_{\mathcal{A}}(q)$ като решение на система от уравнения. Този подход ще ни помогне в Раздел 3.6 да направим връзка между автоматните и регулярните езици.

[19, стр. 133], [14, стр. 63].

Лема 3.1. Да разгледаме произволен краен автомат $\mathcal{A}$, за който $Q_{\mathcal{A}} = \{q_1, q_2, \ldots, q_n\}$. За произволни индекси $i, j = 1, 2, \ldots, n$, да положим:

$$R_{i,j} \stackrel{\text{деф}}{=} \{a \in \Sigma \mid q_j \in \Delta_{\mathcal{A}}(q_i, a)\}$$

$$P_i \stackrel{\text{деф}}{=} \begin{cases} \{\varepsilon\}, & \text{ако } q_i \in F_{\mathcal{A}} \\ \emptyset, & \text{ако } q_i \notin F_{\mathcal{A}}. \end{cases}$$

Тогава езиците $L_{\mathcal{A}}(q_1), \ldots, \mathcal{L}_{\mathcal{A}}(q_n)$ задават *единственото* решение на системата:

$$\left| \begin{array}{rcl} X_1 & = & R_{1,1} \cdot X_1 \cup \cdots R_{1,n} \cdot X_n \cup P_1 \\ & \vdots & \\ X_n & = & R_{n,1} \cdot X_1 \cup \cdots R_{n,n} \cdot X_n \cup P_n. \end{array} \right.$$

Упътване. Можем да разгледаме всеки ред поотделно. Достатъчно е да докажем, че за произволно i, където $1 \leq i \leq n$, е изпълнено равенството

$$\mathcal{L}_{\mathcal{A}}(q_i) = \bigcup_{j=1}^{n} R_{i,j} \cdot \mathcal{L}_{\mathcal{A}}(q_j) \cup P_i.$$

Първо, за включването ($\subseteq$), нека разгледаме дума ω, за която

$$\omega \in \mathcal{L}_{\mathcal{A}}(q_i).$$

- Нека $\omega = \varepsilon$. Ясно е, че $q_i \in F$. Тогава, по дефиниция, $\omega \in P_i$.
- Нека $|\omega| > 0$. Тогава думата ω може да се запише във вида $\omega = b\alpha$. Щом $\omega \in \mathcal{L}_{\mathcal{A}}(q_i)$, то $\Delta^{\star}_{\mathcal{A}}(\{q_i\}, b\alpha) \in F_{\mathcal{A}}$. Това означава, че $\Delta^{\star}_{\mathcal{A}}(\Delta_{\mathcal{A}}(q_i, b), \alpha) \in F_{\mathcal{A}}$. Нека $q_j \in \Delta_{\mathcal{A}}(q_i, b)$ е такова, за което $\Delta^{\star}_{\mathcal{A}}(\{q_j\}, \alpha) \in F_{\mathcal{A}}$. Тогава $\alpha \in \mathcal{L}_{\mathcal{A}}(q_j)$ и $b \in R_{i,j}$. Заключаваме, че

$$\underbrace{b\alpha}_{\omega} \in R_{i,j} \cdot \mathcal{L}_{\mathcal{A}}(q_j).$$

Второ, за включването ($\supseteq$), нека разгледаме дума ω, за която

$$\omega \in \bigcup_{j=1}^{n} R_{i,j} \cdot \mathcal{L}_{\mathcal{A}}(q_j) \cup P_i.$$

- Нека $\omega \in P_i$. Тогава $\omega = \varepsilon$. Ясно е, по дефиниция на P_i, че $q_i \in F$. Ясно е, че $\omega \in \mathcal{L}_{\mathcal{A}}(q_i)$.
- Нека $\omega \in R_{i,j} \cdot \mathcal{L}_{\mathcal{A}}(q_j)$, за някое j, където $1 \leq j \leq n$. Тогава ω може да се представи като $\omega = b\alpha$, където $b \in R_{i,j}$ и $\alpha \in \mathcal{L}_{\mathcal{A}}(q_j)$. Заключаваме, че $q_j \in \Delta_{\mathcal{A}}(q_i, b)$ и $\Delta^{\star}_{\mathcal{A}}(\{q_j\}, \alpha) \in F_{\mathcal{A}}$. Накрая, $\Delta^{\star}_{\mathcal{A}}(\{q_i\}, \omega) \in F_{\mathcal{A}}$ и тогава можем да заключим, че

$$\omega \in \mathcal{L}_{\mathcal{A}}(q_i).$$

□

Сега ще видим, че имаме и обратната посока.

Лема 3.2. Да разгледаме следната произволна система от вида

$$\left|\begin{array}{rcl} X_1 & = & \hat{R}_{1,1}\cdot X_1 \cup \cdots \hat{R}_{1,n}\cdot X_n \cup \hat{P}_1 \\ & \vdots & \\ X_n & = & \hat{R}_{n,1}\cdot X_1 \cup \cdots \hat{R}_{n,n}\cdot X_n \cup \hat{P}_n, \end{array}\right. \tag{3.6}$$

при която $\hat{P}_i \subseteq \{\varepsilon\}$ и $\hat{R}_{i,j} \subseteq \Sigma$, за всяко $i,j = 1,\ldots,n$. Да дефинираме НКА $\mathcal{N}$ по следния начин:

- $Q_{\mathcal{N}} = \{q_1,\ldots,q_n\}$,
- $F_{\mathcal{N}} = \{q_i \in Q_{\mathcal{N}} \mid \varepsilon \in \hat{P}_i\}$,
- $\Delta_{\mathcal{N}}(q_i,a) = \{q_j \in Q_{\mathcal{N}} \mid a \in \hat{R}_{i,j}\}$.

Тогава $\mathcal{L}_{\mathcal{N}}(q_1),\ldots,\mathcal{L}_{\mathcal{N}}(q_n)$ е *единственото* решение на системата (3.6).

Доказателство. Нека имаме автомата $\mathcal{N}$ от условието на лемата, построен по системата (3.6). За него по *Лема 3.1* имаме, че $\mathcal{L}_{\mathcal{N}}(q_1),\ldots,\mathcal{L}_{\mathcal{N}}(q_n)$ е единственото решение на системата

$$\left|\begin{array}{rcl} X_1 & = & R_{1,1}\cdot X_1 \cup \cdots \cup R_{1,n}\cdot X_n \cup P_1 \\ & \vdots & \\ X_n & = & R_{n,1}\cdot X_1 \cup \cdots \cup R_{n,n}\cdot X_n \cup P_n, \end{array}\right. \tag{3.7}$$

където $R_{i,j} \stackrel{\text{деф}}{=} \{a \in \Sigma \mid q_j \in \Delta_{\mathcal{N}}(q_i,a)\}$, $P_i \stackrel{\text{деф}}{=} \{\varepsilon\}$, ако $q_i \in F_{\mathcal{N}}$ и $P_i \stackrel{\text{деф}}{=} \emptyset$ в противен случай.

Но $\varepsilon \in P_i \Leftrightarrow q_i \in F_{\mathcal{N}} \Leftrightarrow \varepsilon \in \hat{P}_i$. Ясно е тогава, че $P_i = \hat{P}_i$. Освен това,

$$a \in R_{i,j} \Leftrightarrow q_j \in \Delta_{\mathcal{N}}(q_i,a) \Leftrightarrow a \in \hat{R}_{i,j}.$$

Заключаваме, че $R_{i,j} = \hat{R}_{i,j}$. Оттук следва, че системата (3.6) съвпада със системата (3.7) и тогава $\mathcal{L}_{\mathcal{N}}(q_1),\ldots,\mathcal{L}_{\mathcal{N}}(q_n)$ е решението на системата (3.6). □

Теорема 3.4 (Фундаментална теорема за автоматните езици)**.** Един език L е *автоматен* точно тогава, когато L е измежду езиците $L_1,\ldots,L_n$, които представляват *единственото* решение на система от вида

$$\left|\begin{array}{rcl} X_1 & = & R_{1,1}\cdot X_1 \cup \cdots \cup R_{1,n}\cdot X_n \cup P_1 \\ & \vdots & \\ X_n & = & R_{n,1}\cdot X_1 \cup \cdots \cup R_{n,n}\cdot X_n \cup P_n, \end{array}\right.$$

където $P_i \subseteq \{\varepsilon\}$ и $R_{i,j} \subseteq \Sigma$, за всяко $i,j = 1,\ldots,n$.

3.6 Автоматните езици са регулярни (алгебричен подход)

Следващата стъпка е да видим как системите от уравнения от регулярни изрази се „връзват" с автоматните езици. Да разгледаме произволен ДКА $\mathcal{A}$. Оказва се, че ние вече знаем как. Достатъчно е да си припомним *Лема 3.1*. Според него, езиците $\mathcal{L}_\mathcal{A}(q_1), \ldots, \mathcal{L}_\mathcal{A}(q_n)$ са решение на системата

$$\left|\begin{array}{rcl} X_1 & = & R_{1,1} \cdot X_1 \cup \cdots R_{1,n} \cdot X_n \cup P_1 \\ & \vdots & \\ X_n & = & R_{n,1} \cdot X_1 \cup \cdots R_{n,n} \cdot X_n \cup P_n. \end{array}\right.$$

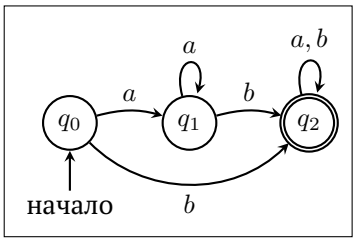
Знаем, че $R_{i,j} \stackrel{\text{деф}}{=} \{a \in \Sigma \mid \delta(q_i, a) = q_j\}$ е краен език и следователно е регулярен. Ясно е също така, че $\varepsilon \notin R_{i,j}$. Знаем също, че

$$P_i \stackrel{\text{деф}}{=} \begin{cases} \{\varepsilon\}, & \text{ако } q_i \in F \\ \emptyset, & \text{ако } q_i \notin F \end{cases},$$

които отново са крайни езици и следователно регулярни. Според *Следствие 2.1* системата има *единствено* решение, което е съставено от регулярни езици. Щом $\mathcal{L}_\mathcal{A}(q_1), \ldots, \mathcal{L}_\mathcal{A}(q_n)$ е решение на системата, то $\mathcal{L}_\mathcal{A}(q_1), \ldots, \mathcal{L}_\mathcal{A}(q_n)$ са регулярни езици. В частност, $\mathcal{L}(\mathcal{A})$ е регулярен език. Така доказахме следната теорема.

Теорема 3.5 (Клини [12]). Всеки автоматен език е регулярен.

Пример 3.4. Да разгледаме отново автомата от *Фигура 3.18*.



Фигура 3.18: $\mathcal{L}(\mathcal{A}) \stackrel{?}{=} \mathcal{L}[\![\boldsymbol{a^\star b(a+b)^\star}]\!]$.

На него съответства следната система от уравнения на езици:

$$\left|\begin{array}{rcl} \mathcal{L}_\mathcal{A}(q_0) & = & \{a\} \cdot \mathcal{L}_\mathcal{A}(q_1) \cup \{b\} \cdot \mathcal{L}_\mathcal{A}(q_2) \cup \emptyset \\ \mathcal{L}_\mathcal{A}(q_1) & = & \{a\} \cdot \mathcal{L}_\mathcal{A}(q_1) \cup \{b\} \cdot \mathcal{L}_\mathcal{A}(q_2) \cup \emptyset \\ \mathcal{L}_\mathcal{A}(q_2) & = & \{a\} \cdot \mathcal{L}_\mathcal{A}(q_2) \cup \{b\} \cdot \mathcal{L}_\mathcal{A}(q_2) \cup \{\varepsilon\}. \end{array}\right.$$

За простота, да преобразуваме системата в система от уравнения на регулярни изрази:

$$\left|\begin{array}{rcl} \boldsymbol{r_0} & = & \boldsymbol{a \cdot r_1 + b \cdot r_2 + \emptyset} \\ \boldsymbol{r_1} & = & \boldsymbol{a \cdot r_1 + b \cdot r_2 + \emptyset} \\ \boldsymbol{r_2} & = & \boldsymbol{a \cdot r_2 + b \cdot r_2 + \varepsilon}. \end{array}\right.$$

Въпреки, че $\emptyset$ нищо не добавя към езика, пишем $\emptyset$ за да видим общия вид на уравненията, които приличат на полиноми. Ясно е, че $\mathcal{L}[\![\boldsymbol{r_i}]\!] = \mathcal{L}_\mathcal{A}(q_i)$.

Да разгледаме само последния ред на системата:

$$\boldsymbol{r_2 = (a+b) \cdot r_2 + \varepsilon}.$$

Чрез прилагане на лемата на Ардън получаваме, че единственото решение на това уравнение е $(a+b)^\star$. В първите два реда на системата заместваме r_2 с $(a+b)^\star$ и получаваме системата:

$$\left|\begin{array}{lcl} r_0 & = & a\cdot r_1 + b\cdot (a+b)^\star \\ r_1 & = & a\cdot r_1 + b\cdot (a+b)^\star \\ r_2 & = & (a+b)^\star. \end{array}\right.$$

Сега вече би трябвало да е ясно как продължаваме. Разглеждаме само втория ред на системата и прилагаме лемата на Ардън за него и получаваме, че единственото решение на втория ред от системата е $a^\star\cdot b\cdot (a+b)^\star$. Така получаваме, след заместване, следните уравнения:

$$\left|\begin{array}{lcl} r_0 & = & a\cdot a^\star\cdot b\cdot (a+b)^\star + b\cdot (a+b)^\star \\ r_1 & = & a^\star\cdot b\cdot (a+b)^\star \\ r_2 & = & (a+b)^\star. \end{array}\right.$$

Тук е сравнително лесно да опростим регулярния израз, но в общия случай това може да е много трудно. Оказва се, че $r_0 = r_1$. По-късно ще видим, че това означава, че можем да „слеем" двете състояния и да получим по-компактен автомат за същия език.

Заключаваме, че езикът на автомата $\mathcal{A}$ може да се опише с регулярния израз:

$$\begin{aligned} r_0 &= a\cdot a^\star\cdot b\cdot (a+b)^\star + b\cdot (a+b)^\star \\ &= (a\cdot a^\star + \varepsilon)\cdot b\cdot (a+b)^\star && \text{// } r\cdot p + p = (r+\varepsilon)\cdot p \\ &= a^\star\cdot b\cdot (a+b)^\star. && \text{// } a\cdot a^\star + \varepsilon = a^\star \end{aligned}$$

Макар и нашият подход тук да беше алгебричен, от горния пример лесно се вижда как може да се построи алгоритъм за намиране на регулярен израз описващ езика на даден ДКА.

Твърдение 3.9. Съществува алгоритъм, за който при вход краен детерминиран автомат $\mathcal{A}$, извежда като изход регулярен израз r, такъв че $\mathcal{L}(\mathcal{A}) = \mathcal{L}[\![r]\!]$.

За подробно изложение на този въпрос, вижте [21, стр. 69]. Алгоритъмът на практика следва идеята изложена в Пример 3.4.

3.7 Регулярните езици са автоматни (алгебричен подход)

Ще докажем, че регулярните езици са автоматни с индукция по построението на регулярните езици. Да започнем с базата на индукцията.

Лема 3.3. Езиците $\emptyset$, $\{\varepsilon\}$, $\{a\}$ за всяко $a \in \Sigma$ са автоматни.

Упътване. Достатъчно е да покажем, че съществуват недетерминирани крайни автомати $\mathcal{N}$.

Според нашата дефиниция, тук описваме недетерминирани автомати, защото за да бъдат детерминирани, трябва функцията на преходите да бъде тотална.

(а) $L(\mathcal{N}) = \emptyset$ **(б)** $\mathcal{L}(\mathcal{N}) = \{\varepsilon\}$ **(в)** $\mathcal{L}(\mathcal{N}) = \{a\}$

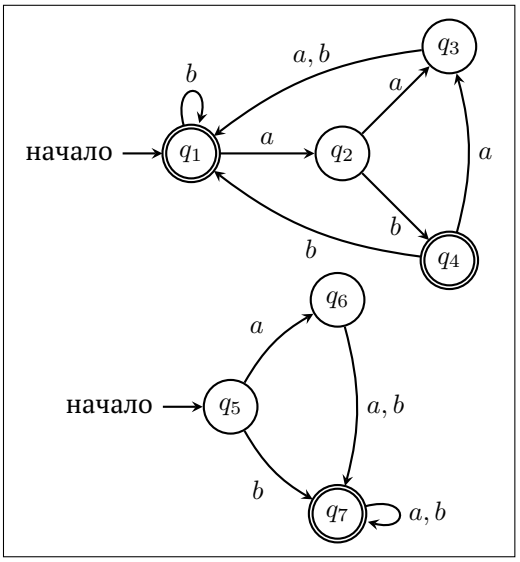
□

Сега преминаваме към индукционната стъпка.

Лема 3.4. Автоматните езици са затворени относно операцията обединение.

Това означава, че ако два езика L_1 и L_2 се разпознават с крайни автомати, то трябва да докажем, че $L_1 \cup L_2$ се разпознава от краен автомат. Понеже приемаме, че недетерминираните крайни автомати могат да имат множество от начални състояния, то автомат за $L_1 \cup L_2$ може да се построи като просто обединим състоянията на автоматите за L_1 и L_2 както и функциите на преходите и множествата от финални състояния.

Пример 3.5. За да построим автомат, който разпознава обединението на $\mathcal{L}(\mathcal{A}_1)$ и $\mathcal{L}(\mathcal{A}_2)$, трябва да добавим ново начално състояние, което да свържем с наследниците на началните състояния на $\mathcal{A}_1$ и $\mathcal{A}_2$. Обърнете внимание, че $\mathcal{A}_1$ и $\mathcal{A}_2$ са детерминирани автомати, но $\mathcal{N}$ е недетерминиран.



Фигура 3.20: $\mathcal{L}(\mathcal{N}) = \mathcal{L}(\mathcal{A}_1) \cup \mathcal{L}(\mathcal{A}_2)$.

Лема 3.5. Автоматните езици са затворени относно операцията конкатенация.

Да започнем с един автомат $\mathcal{A}_1$ разпознаващ езика L_1. Знаем, че на него съответства системата

$$\begin{vmatrix} X_1 &=& R_{1,1}\cdot X_1 \cup \cdots \cup R_{1,n}\cdot X_n \cup P_1 \\ & \vdots & \\ X_n &=& R_{n,1}\cdot X_1 \cup \cdots \cup R_{n,n}\cdot X_n \cup P_n, \end{vmatrix}$$

чието *единствено* решение е n-орка от езици $L_1, \ldots, L_n$.

$L_1, \ldots, L_n$ е решението на системата за $\mathcal{A}_1$ означава, че:

$$\begin{vmatrix} L_1 = \bigcup\limits_{i=1}^{n} R_{1,i}\cdot L_i \cup P_1 \\ \vdots \\ L_n = \bigcup\limits_{i=1}^{n} R_{n,i}\cdot L_i \cup P_n. \end{vmatrix}$$

Да вземем и друг автомат $\mathcal{A}_2$ разпознаващ езика M_1. Знаем, че на него съответства системата

$$\begin{vmatrix} Y_1 &=& T_{1,1}\cdot Y_1 \cup \cdots \cup T_{1,n}\cdot Y_k \cup E_1 \\ & \vdots & \\ Y_k &=& T_{n,1}\cdot Y_1 \cup \cdots \cup T_{k,k}\cdot Y_k \cup E_k, \end{vmatrix}$$

чието *единствено* решение е k-орка от езици $M_1, \ldots, M_k$.

$M_1, \ldots, M_k$ е решението на системата за $\mathcal{A}_2$ означава, че:

$$\begin{vmatrix} M_1 = \bigcup\limits_{i=1}^{k} T_{1,i}\cdot M_i \cup E_1 \\ \vdots \\ M_k = \bigcup\limits_{i=1}^{k} T_{n,i}\cdot M_i \cup E_k. \end{vmatrix}$$

Ще дефинираме нова система от $(n+k)$ уравнения, чието единствено решение ще бъдат езиците $L_1M_1, L_2M_1, \ldots, L_nM_1, M_1, \ldots, M_k$.

Понеже имаме равенствата

$$\begin{vmatrix} L_1 &=& R_{1,1}\cdot L_1 \cup \cdots R_{1,n}\cdot L_n \cup P_1 \\ & \vdots & \\ L_n &=& R_{n,1}\cdot L_1 \cup \cdots R_{n,n}\cdot L_n \cup P_n, \end{vmatrix}$$

то е ясно, че можем да конкатенираме навсякъде отдясно с M_1 и да получим следното:

$$\begin{vmatrix} L_1M_1 &=& R_{1,1}\cdot L_1M_1 \cup \cdots R_{1,n}\cdot L_nM_1 \cup P_1M_1 \\ & \vdots & \\ L_nM_1 &=& R_{n,1}\cdot L_1M_1 \cup \cdots R_{n,n}\cdot L_nM_1 \cup P_nM_1. \end{vmatrix}$$

Да разгледаме произволно $i = 1, \ldots, n$.

- Ако $P_i = \emptyset$, то е ясно, че $P_iM_1 = P_i$. Тогава имаме, че

$$L_iM_1 = R_{i,1}\cdot L_1M_1 \cup \cdots R_{i,n}\cdot L_nM_1 \cup P_i.$$

- Ако $P_i = \{\varepsilon\}$, то е ясно, че $P_iM_1 = M_1$. Тогава използваме дясната страна на равенството за M_1 за да получим равенството:

$$L_iM_1 = R_{i,1}\cdot L_1M_1 \cup \cdots R_{i,n}\cdot L_nM_1 \cup \underbrace{T_{1,1}\cdot M_1 \cup \cdots T_{1,k}\cdot M_k \cup E_1}_{M_1}.$$

От тези две наблюдения следва, че можем да запишем следната система от равенства:

$$\begin{vmatrix} L_1M_1 &=& R_{1,1}\cdot L_1M_1 \cup \cdots R_{1,n}\cdot L_nM_1 \cup \hat{T}_{1,1}M_1 \cup \cdots \hat{T}_{1,k}M_k \cup \hat{E}_1 \\ & \vdots & \\ L_nM_1 &=& R_{n,1}\cdot L_1M_1 \cup \cdots R_{n,n}\cdot L_nM_1 \cup \hat{T}_{n,1}M_1 \cup \cdots \hat{T}_{n,k}M_k \cup \hat{E}_n, \end{vmatrix}$$

където за $i = 1, \ldots, n$ и $j = 1, \ldots, k$ сме положили:

$$\hat{T}_{i,j} \stackrel{\text{деф}}{=} \begin{cases} T_{1,j}, & \text{ако } P_i = \{\varepsilon\} \\ \emptyset, & \text{ако } P_i = \emptyset, \end{cases}$$

а също така за $i = 1, \ldots, n$,

$$\hat{E}_i \stackrel{\text{деф}}{=} \begin{cases} E_1, & \text{ако } P_i = \{\varepsilon\} \\ \emptyset, & \text{ако } P_i = \emptyset. \end{cases}$$

Оттук заключаваме, че системата

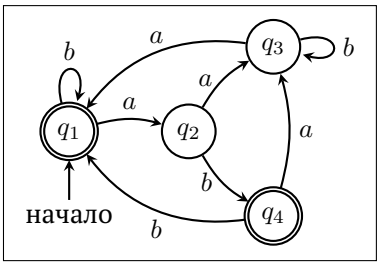
$$\left|\begin{array}{lcl} X_1 & = & R_{1,1} \cdot X_1 \cup \cdots R_{1,n} \cdot X_n \cup \hat{T}_{1,1} Y_1 \cup \cdots \hat{T}_{1,k} Y_k \cup \hat{E}_1 \\ & \vdots & \\ X_n & = & R_{n,1} \cdot X_1 \cup \cdots R_{n,n} \cdot X_n \cup \hat{T}_{n,1} Y_1 \cup \cdots \hat{T}_{n,k} Y_k \cup \hat{E}_n \\ & & \\ Y_1 & = & T_{1,1} \cdot Y_1 \cup \cdots T_{1,n} \cdot Y_k \cup E_1 \\ & \vdots & \\ Y_k & = & T_{n,1} \cdot Y_1 \cup \cdots T_{k,k} \cdot Y_k \cup E_k. \end{array}\right.$$

има единствено решение, което е $L_1 M_1, \ldots, L_n M_1, M_1, \ldots, M_k$.

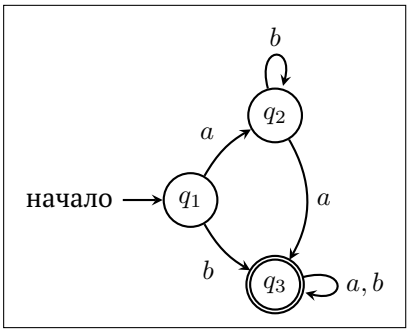
Директно от тази конструкция получаваме алгоритъм за намиране на автомат за конкатенацията на двата дадени първоначално автомата. Ползата от този подход е, че понеже на всяка стъпка правим еквивалентни преобразования, то безплатно получаваме коректността на конструкцията на автомат за конкатенацията.

Пример 3.6. Нека да видим как можем да приложим горната конструкция върху два конкретни автомата $\mathcal{A}_1$ и $\mathcal{A}_2$.

От автоматна гледна точка, ние трябва да свържем финалните състояния на $\mathcal{A}_1$ с наследниците на началното състояние на $\mathcal{A}_2$, като внимаваме да запазим етикетите по новите преходи. От алгебрична гледна точка, просто заместваме $\{\varepsilon\}$ в системата за $\mathcal{A}_1$ с дефиницията на Y_1.



(а) автомат $\mathcal{A}_1$ за езика L_1.



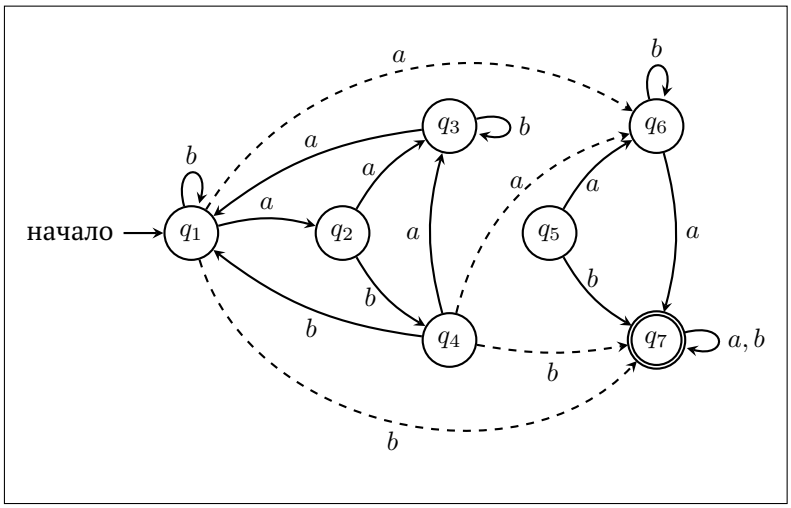
(б) автомат $\mathcal{A}_2$ за езика L_2.

$$\left|\begin{array}{lcl} X_1 & = & \{a\}X_2 \cup \{b\}X_1 \cup \{\varepsilon\} \\ X_2 & = & \{a\}X_3 \cup \{b\}X_4 \\ X_3 & = & \{a\}X_1 \cup \{b\}X_3 \\ X_4 & = & \{a\}X_3 \cup \{b\}X_1 \cup \{\varepsilon\}. \end{array}\right.$$

(в) Система за автомата $\mathcal{A}_1$.

$$\left|\begin{array}{lcl} Y_1 & = & \{a\}Y_2 \cup \{b\}Y_3 \\ Y_2 & = & \{a\}Y_3 \cup \{b\}Y_2 \\ Y_3 & = & \{a\}Y_3 \cup \{b\}Y_3 \cup \{\varepsilon\}. \end{array}\right.$$

(г) Система за автомата $\mathcal{A}_2$.



(д) Автомат $\mathcal{N}$ за езика $L_1 \cdot L_2$.

$$\left|\begin{array}{lcl} X_1 & = & \{a\}X_2 \cup \{b\}X_1 \cup \boldsymbol{\{a\}Y_2 \cup \{b\}Y_3} \\ X_2 & = & \{a\}X_3 \cup \{b\}X_4 \\ X_3 & = & \{a\}X_1 \cup \{b\}X_3 \\ X_4 & = & \{a\}X_3 \cup \{b\}X_1 \cup \boldsymbol{\{a\}Y_2 \cup \{b\}Y_3} \\ Y_1 & = & \{a\}Y_2 \cup \{b\}Y_3 \\ Y_2 & = & \{a\}Y_3 \cup \{b\}Y_2 \\ Y_3 & = & \{a\}Y_3 \cup \{b\}Y_3 \cup \{\varepsilon\}. \end{array}\right.$$

(е) Система за автомата $\mathcal{N}$.

Лема 3.6. Автоматните езици са затворени относно операцията звезда на Клини.

Да започнем от един автомат $\mathcal{A}$ разпознаващ езика $L_1 = \mathcal{L}_{\mathcal{A}}(q_{\texttt{start}})$. Знаем, че на него съответства системата

$$\left|\begin{array}{rcl} X_1 &=& R_{1,1}\cdot X_1 \cup \cdots \cup R_{1,n}\cdot X_n \cup P_1 \\ &\vdots& \\ X_n &=& R_{n,1}\cdot X_1 \cup \cdots \cup R_{n,n}\cdot X_n \cup P_n, \end{array}\right.$$

чието *единствено* решение е n-орка от езици $L_1, \ldots, L_n$.

Да напомним, че $R_{i,j} \subseteq \Sigma$ и $P_j \subseteq \{\varepsilon\}$. Освен това, щом $L_1, \ldots, L_n$ е решението на системата, то имаме равенствата:

$$\left|\begin{array}{l} L_1 = \bigcup\limits_{j=1}^{n} R_{1,j}\cdot L_j \cup P_1 \\ \quad\vdots \\ L_n = \bigcup\limits_{j=1}^{n} R_{n,j}\cdot L_j \cup P_n. \end{array}\right.$$

Ще видим, че като направим редица еквивалентни преобразувания върху тази система от уравнения, ще получим нова система, чието *единствено* решение ще бъде n-орката от езици $L_1L_1^\star, L_2L_1^\star, \cdots, L_nL_1^\star$. Тогава от фундаменталната теорема за автомати ще следва, че имаме автомат $\hat{\mathcal{A}}$ разпознаващ езика L_1^+. Първата стъпка е да конкатенираме навсякъде с $L_1^\star$ за да получим равенствата:

$$\left|\begin{array}{rcl} L_1L_1^\star &=& R_{1,1}\cdot L_1L_1^\star \cup \cdots R_{1,n}\cdot L_nL_1^\star \cup P_1\cdot L_1^\star \\ &\vdots& \\ L_nL_1^\star &=& R_{n,1}\cdot L_1L_1^\star \cup \cdots R_{n,n}\cdot L_nL_1^\star \cup P_n\cdot L_1^\star. \end{array}\right.$$

Ясно е, че членовете $P_i \cdot L_1^\star$ са проблематични. За да преведем новата система от уравнения в подходящия за нас вид, трябва да разгледаме няколко случая в зависимост от това дали $P_i = \emptyset$ или $P_i = \{\varepsilon\}$.

Нека първо да разгледаме случая, когато $P_i = \emptyset$, за някое i. Тогава $P_i \cdot L_1^\star = P_i$. Получаваме, че

$$L_iL_1^\star = R_{i,1}\cdot L_1L_1^\star \cup \cdots R_{i,n}\cdot L_nL_1^\star \cup \underbrace{P_i}_{\emptyset}.$$

Нека сега да разгледаме случая, когато $P_i = \{\varepsilon\}$. Първо да направим наблюдението, че за произволен език L е изпълнено равенството

$$L^\star = (L \setminus \{\varepsilon\})\cdot L^\star \cup \{\varepsilon\}. \tag{3.8}$$

По този начин „разгръщаме" рекурсията. Обръщането към L_1 се замества с неговата дефиниция.

Оттук получаваме следните равенства:

$$\begin{aligned} P_iL_1^\star &= L_1^\star \\ &= (L_1 \setminus \{\varepsilon\})L_1^\star \cup \underbrace{P_i}_{\{\varepsilon\}} \\ &= \underbrace{(R_{1,1}\cdot L_1 \cup \cdots R_{1,n}\cdot L_n)}_{L_1\setminus\{\varepsilon\}}\cdot L_1^\star \cup P_i \\ &= R_{1,1}\cdot L_1L_1^\star \cup \cdots R_{1,n}\cdot L_nL_1^\star \cup P_i. \end{aligned}$$

Накрая заключаваме, че i-тия ред на системата може да се запише така:

Тук единственото, което правим е да приведем системата във вид, за който можем да приложим фундаменталната теорема за автомати езици.

$$\begin{aligned} L_iL_1^\star &= R_{i,1}\cdot L_1L_1^\star \cup \cdots R_{i,n}\cdot L_nL_1^\star \cup P_iL_1^\star \\ &= R_{i,1}\cdot L_1L_1^\star \cup \cdots R_{i,n}\cdot L_nL_1^\star \cup \underbrace{R_{1,1}\cdot L_1L_1^\star \cup \cdots R_{1,n}\cdot L_nL_1^\star \cup P_i}_{P_iL_1^\star} \\ &= \underbrace{(R_{i,1}\cup R_{1,1})}_{\hat{R}_{i,1}}\cdot L_1L_1^\star \cup \cdots \underbrace{(R_{i,n}\cup R_{1,n})}_{\hat{R}_{i,n}}\cdot L_nL_1^\star \cup P_i. \end{aligned}$$

Можем да обобщим всичко така. Нека положим

$$\hat{R_{i,j}} \stackrel{\text{деф}}{=} \begin{cases} R_{i,j}, & \text{ако } P_i = \emptyset \\ R_{i,j} \cup R_{1,j}, & \text{ако } P_i = \{\varepsilon\}. \end{cases}$$

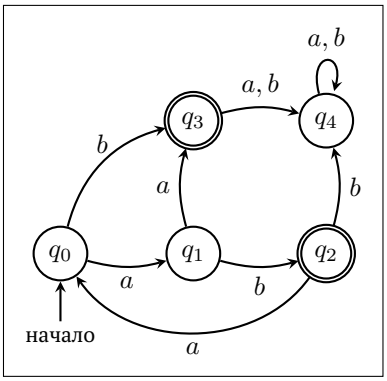
Тогава заключаваме, че *единственото* решение на системата

$$\left|\begin{array}{rcl} X_1 &=& \hat{R}_{1,1} \cdot X_1 \cup \cdots \hat{R}_{1,n} \cdot X_n \cup P_1 \\ & \vdots & \\ X_n &=& \hat{R}_{n,1} \cdot X_1 \cup \cdots \hat{R}_{n,n} \cdot X_n \cup P_n \end{array}\right.$$

е n-орката от езици $L_1 L_1^\star, L_2 L_1^\star, \ldots, L_n L_1^\star$.

Тук отново директно получаваме алгоритъм за намиране на автомат разпознаващ звездата на Клини на езика на входния автомат. Ползвата от нашият подход е, че няма нужда да доказваме коректност на този алгоритъм.

Пример 3.7. Нека да видим как можем да приложим горната конструкция върху конкретен автомат.



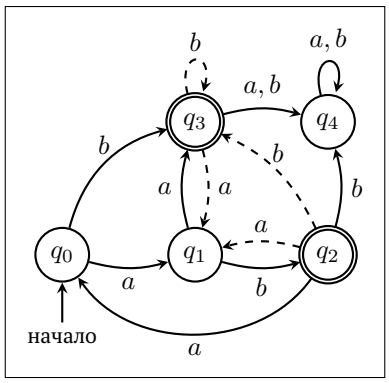
(а) автомат $\mathcal{A}$ за езика L.

$$\left|\begin{array}{rcl} X_0 &=& \{a\}X_1 \cup \{b\}X_3 \\ X_1 &=& \{a\}X_3 \cup \{b\}X_2 \\ X_2 &=& \{a\}X_0 \cup \{b\}X_4 \cup \{\varepsilon\} \\ X_3 &=& \{a,b\}X_4 \cup \{\varepsilon\} \\ X_4 &=& \{a,b\}X_4 \end{array}\right.$$

(б) Система за автомата $\mathcal{A}$.

От автоматна гледна точка, свързваме финалните състояния на $\mathcal{A}$ с наследниците на неговото начално състояние, като внимаваме да запазим етикетите по новите преходи. От алгебрична гледна точка, просто копираме дефиницията на X_1 на тези редове, където имаме $\{\varepsilon\}$.

Да видим как можем да приложим горната конструкция за да намерим автомат за L^+. Така получаваме системата:



(а) Автомат $\mathcal{N}$ за езика L^+.

$$\left|\begin{array}{l} X_0 = \{a\}X_1 \cup \{b\}X_3 \\ X_1 = \{a\}X_3 \cup \{b\}X_2 \\ X_2 = \{a\}X_0 \cup \{b\}X_4 \cup \boldsymbol{\{a\}X_1 \cup \{b\}X_3} \cup \{\varepsilon\} \\ X_3 = \{a,b\}X_4 \cup \boldsymbol{\{a\}X_1 \cup \{b\}X_3} \cup \{\varepsilon\} \\ X_4 = \{a,b\}X_4 \end{array}\right.$$

(б) Системата за автомата $\mathcal{N}$.

Оттук вече е ясно как можем да получим автомат за $L^\star$. Просто добавяме ново начално състояние, което едновременно с това е и финално и от което не излизат преходи.

3.8 Критерий за регулярност (автоматен подход)

Лема 3.7 (Лема за покачването). Нека L да бъде *безкраен* регулярен език. Съществува число $p \geq 1$, зависещо само от L, за което за всяка дума $\alpha \in L$, за която $|\alpha| \geq p$, може да бъде записана във вида $\alpha = xyz$ и

1) $|y| \geq 1$;

2) $|xy| \leq p$;

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

На англ. се нарича *Pumping Lemma*, макар според някои студенти да се нарича *Pumpkin Lemma*. Има подобна лема и за безконтекстни езици, която ще разгледаме по-нататък. Обърнете внимание, че $0 \in \mathbb{N}$ и $xy^0z = xz$. Тази лема я има на практика във всеки добър учебник по този предмет. Например, [16, стр. 88], [21, стр. 77], [10, стр. 55]. Оригиналното доказателство е на Бар-Хилел, Перлес и Шамир [3].

Упътване. Понеже L е регулярен, то L е и автоматен език. Нека

$$\mathcal{A} = \langle \Sigma, Q, q_{\texttt{start}}, \delta, F \rangle$$

е краен детерминиран автомат, за който $L = \mathcal{L}(\mathcal{A})$. Да положим $p = |Q|$ и нека ω е дума, за която $|\omega| = k \geq p$. Да разгледаме изпълнението на ω върху $\mathcal{A}$:

$$q_{\texttt{start}} \overset{\omega[:p]}{\rightsquigarrow} q_p \overset{\omega[p:]}{\rightsquigarrow} q_k$$

и по специално първите p стъпки. Понеже в това доказателство ще работим с индексите на състоянията, нека за улеснение да положим $q_0 = q_{\texttt{start}}$. Тъй като $|Q| = p$, а по пътя

$$q_0 \overset{a_0}{\to} q_1 \overset{a_1}{\to} q_2 \cdots \overset{a_{p-1}}{\to} q_p$$

участват $p+1$ състояния $q_0, q_1, \ldots, q_p$, то съществуват индекси i, j, за които $0 \leq i < j \leq p$ и $q_i = q_j$. Нека разделим думата ω на три части по следния начин:

$$q_0 \overset{\omega[:i]}{\rightsquigarrow} q_i \overset{\omega[i:j]}{\rightsquigarrow} q_j \overset{\omega[j:p]}{\rightsquigarrow} q_p$$

Нека положим $x \overset{\text{деф}}{=} \omega[:i]$, $y \overset{\text{деф}}{=} \omega[i:j]$, $z \overset{\text{деф}}{=} \omega[j:]$. Ясно е, че $|y| \geq 1$ и $|xy| = j \leq p$.

Можем да опишем изчислението по следния начин:

$$(q_0, xyz) \vdash^\star_{\mathcal{A}} (q_i, yz) \vdash^\star_{\mathcal{A}} (q_j, z) \vdash^\star_{\mathcal{A}} (q_k, \varepsilon).$$

Да разгледаме случая за $n = 0$. Думата $xy^0z = xz \in L$, защото имаме следното изчисление:

$$q_0 \overset{\omega[:i]}{\rightsquigarrow} \underbrace{q_i}_{q_j} \overset{\omega[j:]}{\rightsquigarrow} q_k \in F.$$

защото $q_i = q_j$, или с други думи,

$$(q_0, xz) \vdash^\star_{\mathcal{A}} (q_i, z) \vdash^\star_{\mathcal{A}} (q_k, \varepsilon).$$

Сега да разгледаме случая $n = 2$. Тогава думата $xy^2z = xyyz \in L$, защото имаме следното изчисление:

$$q_0 \overset{\omega[:i]}{\rightsquigarrow} q_i \overset{\omega[i:j]}{\rightsquigarrow} \underbrace{q_i}_{q_j} \overset{\omega[i:j]}{\rightsquigarrow} q_j \overset{\omega[j:]}{\rightsquigarrow} q_k \in F,$$

или с други думи,

$$(q_0, xyyz) \vdash^\star_{\mathcal{A}} (q_i, yyz) \vdash^\star_{\mathcal{A}} (q_i, yz) \vdash^\star_{\mathcal{A}} (q_i, z) \vdash^\star_{\mathcal{A}} (q_k, \varepsilon).$$

Аналогично, за произволно $n \geq 2$, имаме изчислението:

$$q_0 \overset{\omega[:i]}{\rightsquigarrow} \underbrace{q_i \overset{\omega[i:j]}{\rightsquigarrow} q_i \overset{\omega[i:j]}{\rightsquigarrow} q_i \cdots \overset{\omega[i:j]}{\rightsquigarrow} \cdots q_i}_{n \text{ пъти}} \overset{\omega[j:]}{\rightsquigarrow} q_k \in F,$$

или с други думи,

$$(q_0, xy^n z) \vdash^\star_{\mathcal{A}} \underbrace{(q_i, y^n z) \vdash^\star_{\mathcal{A}} (q_i, y^{n-1} z) \vdash^\star_{\mathcal{A}} \cdots \vdash^\star_{\mathcal{A}} (q_i, z)}_{n \text{ пъти}} \vdash^\star_{\mathcal{A}} (q_k, \varepsilon).$$

От направените по-горе разсъждения, можем да заключим, че за всяко естествено число n е изпълнено, че $xy^n z \in L$. □

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

Следствие 3.3 (Контрапозиция на лемата за покачването)**.** Нека L е произволен 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 $|xy| \leq p$ и $|y| \geq 1$,

(∃) можем да посочим $i \in \mathbb{N}$, за което е изпълнено, че $xy^i z \notin L$.

Тогава L **не** е регулярен език.

Подобно е изложението и в [13, стр. 70].

Доказателство. Да означим с $P_{\text{reg}}(L)$ следната формула:

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

Понеже имаме, че $p \to q \equiv \neg p \vee q$, то горната формула може да се запише и така:

$$\begin{aligned}(\exists p \geq 1)(\forall \alpha \in L)[|\alpha| \not\geq p \vee (\exists x, y, z \in \Sigma^\star)[\ & \alpha = xyz \wedge \\ & |y| \geq 1 \wedge \\ & |xy| \leq p \wedge \\ & (\forall i \in \mathbb{N})[xy^i z \in L]]].\end{aligned}$$

Така условието на лемата за покачването представлява твърдението:

„Ако L е регулярен език, то е изпълнено $P_{reg}(L)$."

Контрапозиция на твърдението $P \to Q$ е твърдението $\neg Q \to \neg P$

Лемата може да се запише по следния еквивалентен начин:

„Ако $P_{reg}(L)$ не е изпълнено, то L не е регулярен език."

Или еквивалентно,

„Ако $\neg P_{reg}(L)$ е изпълнено, то L не е регулярен език."

Използваме, че $\neg\exists\forall\exists\forall(\dots) \equiv \forall\exists\forall\exists\neg(\dots)$

Сега $\neg P_{\text{reg}}(L)$ преставлява формулата

$$\begin{aligned}(\forall p \geq 1)(\exists \alpha \in L)[\ |\alpha| \geq p\ \wedge\ (\forall x, y, z \in \Sigma^\star)[\ & \alpha \neq xyz \vee \\ & |y| \not\geq 1 \vee \\ & |xy| \not\leq p \vee \\ & (\exists i \in \mathbb{N})[xy^i z \notin L]\ ]\ ].\end{aligned}$$

Горната формула е еквивалентна на:

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

Използваме, че

$$\frac{\neg P \vee \neg Q \vee \neg S \vee R}{\neg(P\ \wedge\ Q \wedge S) \vee R}$$
$$\frac{}{(P \wedge Q \wedge S) \rightarrow R}$$

Това означава, че ако

(∀) вземем произволна константа $p \geq 1$,

(∃) за нея намерим конкретна дума $\alpha \in L$, такава че $|\alpha| \geq p$ и

(∀) докажем, че за всяко нейно разбиване на три части x, y, z, със свойствата $|y| \geq 1$ и $|xy| \leq p$,

(∃) можем да посочим естествено число i, за което $xy^i z \notin L$,

то можем да заключим, че езикът L не е регулярен. □

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

Пример 3.8. Да видим защо езикът $L = \{a^n b^n \mid n \in \mathbb{N}\}$ не е регулярен. За да направим това като използваме контрапозицията на лемата за покачването, то трябва да докажем, че $\neg P_{\text{reg}}(L)$ е изпълнено. Както обяснихме по-горе, доказателството следва стъпките:

(∀) Нямаме власт над избора на числото p.

(∀) Разглеждаме произволно число $p \geq 1$.

(∃) Няма общо правило, което да ни казва как избираме думата α. Трябва сами да се досетим. Обърнете внимание, че думата α зависи от константата 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 $\alpha = a^p b^p \in L$. Очевидно е, че $|\alpha| \geq p$.

(∀) Не знаем нищо друго за x, y и z освен тези две свойства.

(∀) Разглеждаме произволно разбиване на α на три части, $\alpha = xyz$, за които изискваме свойствата $|xy| \leq p$ и $|y| \geq 1$.

(∃) Изборът на i може да зависи разбиването x, y, z. В конкретния пример не зависи.

(∃) 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 $i = 0$, получаваме $xy^0 z = a^{p-k} b^p$. Ясно е, че $xz \notin L$, защото $p - k < p$.

Тогава от *Следствие 3.3* следва, че L не е регулярен език.

✍ Съобразете сами!

Забележка. Много често студентите правят следното разсъждение:

$$L \text{ е регулярен } \& \; L' \subseteq L \implies L' \text{ е регулярен.}$$

Съобразете, че в общия случай това твърдение е *невярно*. За да видим това, достатъчно е да посочим регулярен език L, който има като подмножество нерегулярен език L'. Също лесно се вижда, че твърдението

$$L \text{ е регулярен } \& \; L \subseteq L' \implies L' \text{ е регулярен}$$

е невярно.

Поради подобни съображения, следните твърдения също са *неверни*:

$$L \text{ не е регулярен } \& \; L' \subseteq L \implies L' \text{ не е регулярен}$$
$$L' \text{ не е регулярен } \& \; L' \subseteq L \implies L \text{ не е регулярен.}$$

Примерни задачи

Задача 3.12. Докажете, че езикът $L = \{a^m b^n \mid m, n \in \mathbb{N} \,\&\, m < n\}$ не е регулярен.

Доказателство. 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 $|\alpha| \geq p$.

На стъпка от вида (∀) нямаме власт над това как избираме съответния елемент. На стъпка от вида (∃) имаме тази власт. Тогава трябва да посочим конкретен елемент.

(∀) Разглеждаме произволно разбиване на α на три части, $\alpha = xyz$, за които изискваме свойствата $|xy| \leq p$ и $|y| \geq 1$.

(∃) Ще намерим конкретно $i \in \mathbb{N}$, 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

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

Тук изборът на i не зависи от изборите, които сме направили на предишните стъпки.

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

Тогава от контрапозицията на лемата за покачването следва, че L не е регулярен език. □

Задача 3.13. Докажете, че езикът $L = \{a^n \mid n$ е просто число$\}$ не е регулярен.

Доказателство. Доказателството следва стъпките:

(∀) Разглеждаме произволно число $p \geq 1$.

(∃) Избираме дума $\omega \in L$, за която $|\omega| \geq p$. Можем да изберем каквото ω си харесаме, стига то да принадлежи на L и да има дължина поне p. Нека да изберем една конкретна дума $\omega \in L$, такава че $|\omega| > p + 1$. 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 ω на три части, $\omega = xyz$, за които изискваме свойствата $|xy| \leq p$ и $|y| \geq 1$.

Обърнете внимание, че тук е по-интересно. Изборът на i зависи от предишната стъпка, на която сме разбили думата ω на три части.

(∃) Ще намерим конкретно 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 $xy^i z \notin L$.

Изискваме $|\omega > p + 1$, защото искаме да гарантираме, че $|xz| > 1$.

Тогава от контрапозицията на лемата за покачването следва, че L не е регулярен език. □

Задача 3.14. Докажете, че езикът $L = \{a^{n^2} \mid n \in \mathbb{N}\}$ не е регулярен.

Доказателство. В тази задача ще използваме следното свойство:

$$n \text{ не е точен квадрат } \Leftrightarrow (\exists p \in \mathbb{N})[p^2 < n < (p+1)^2].$$

Доказателството следва стъпките:

(∀) Разглеждаме произволно число $p \geq 1$.

(∃) Избираме достатъчно дълга дума, която принадлежи на езика L. За да бъдем конкретни, нека $\omega = a^{p^2}$.

(∀) Разглеждаме произволно разбиване на ω на три части, $\omega = xyz$, като $|xy| \leq p$ и $|y| \geq 1$.

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

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

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 контрапозицията на лемата за покачването следва, че L не е регулярен език. □

Задача 3.15. Докажете, че езикът $L = \{a^{n!} \mid n \in \mathbb{N}\}$ не е регулярен.

Доказателство. Доказателството следва стъпките:

(∀) Разглеждаме произволно число $p \geq 1$.

(∃) Избираме достатъчно дълга дума, която принадлежи на езика L. В този случай, добра работа ще ни свърши думата $\omega = a^{(p+1)!}$.

(∀) 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 конкретно i, за което $xy^iz \notin L$. Това означава да съществува n, за което

$$n! < |xy^iz| < (n+1)!$$

Да разгледаме $i = 2$. Тогава:

$$\begin{aligned}(p+1)! &< |xy^2z| \\ &= (p+1)! + |y| \\ &\leq (p+1)! + p \\ &< (p+1)! + (p+1)!(p+1) \\ &= (p+2)!\end{aligned}$$

Възможно е да вземем $\omega = a^{(p+2)!}$. Тогава възможно ли е $xy^0z \notin L$? Понеже $|xyz| = (p+2)!$, това означава, че би трябвало $|xz| = k!$, за някое $k \leq p+1$. Тогава

$$\begin{aligned}|y| &= |xyz| - |xz| \\ &= (p+2)! - k! \\ &\geq (p+2)! - (p+1)! \\ &= (p+1).(p+1)! \\ &> p.\end{aligned}$$

Достигнахме до противоречие с условието, че $|y| \leq p$.

Тогава от контрапозицията на лемата за покачването следва, че L не е регулярен език. □

Задача 3.16. Докажете, че езикът $L = \{\alpha\beta \in \{a,b\}^\star \mid |\alpha| = |\beta| \ \& \ \alpha \neq \beta\}$ не е регулярен.

Упътване. Да допуснем, че L е регулярен. Тогава езикът $\overline{L} = \{a,b\}^\star \setminus L$ също е регулярен. Ясно е, че

$$\overline{L} = \{\alpha\beta \in \{a,b\}^\star \mid \alpha = \beta\} \cup \{\omega \in \{a,b\}^\star \mid |\omega| \text{ е нечетно число}\}.$$

Тогава езикът $L_1 = \overline{L} \cap \{\omega \in \{a,b\}^\star \mid |\omega| \text{ е четно число}\}$ също е регулярен. Ясно е, че $L_1 = \{\alpha\beta \in \{a,b\}^\star \mid \alpha = \beta\}$. Сега можем да разгледаме регулярния език

$$L_2 = L_1 \cap \mathcal{L}[\![\boldsymbol{a^\star ba^\star b}]\!] = \{a^nba^nb \mid n \in \mathbb{N}\}.$$

За него вече лесно можем да приложим лемата за покачването и да получим, че L_2 не е регулярен. Така достигаме до противоречие с допускането, че L е регулярен. □

Задача 3.17. Докажете, че езикът $L = \{a^nb^k \mid n \neq k\}$ не е регулярен.

Упътване. Доказателството следва стъпките:

(∀) Разглеждаме произволно число $p \geq 1$.

(∃) Избираме достатъчно дълга дума, която принадлежи на езика L. В този случай, добра работа ще ни свърши думата $\omega = a^p \cdot b^{p+p!}$.

Тук е важно да отбележим, че изборът на i зависи от избора на разбиването xyz.

(∀) Разглеждаме произволно разбиване на ω на три части, $\omega = xyz$, като $|xy| \leq p$ и $|y| \geq 1$. Да обърнем внимание, че $1 \leq |y| \leq p$. Ясно е, че $y = a^\ell$, за някое ℓ.

(∃) Да изберем числото $i = \frac{p!}{\ell}$. Тогава е ясно, че $y^i = a^{p!}$. Заключаваме, че

$$xy^{i+1}z = a^{p+p!} \cdot b^{p+p!} \notin L.$$

□

Забележка. Тази задача е добър пример защо е добре първо да се опитаме да преобразуваме дадения език с операции, които запазват регулярността. Тогава можем далеч по-лесно да решим тази задача. Знаем, че регулярните езици са затворени относно допълнение. Следователно, ако допуснем, че L е регулярен, то езикът

$$L' = \{a\}^\star \cdot \{b\}^\star \setminus L$$

също трябва да е регулярен. Но $L' = \{a^nb^n \mid n \in \mathbb{N}\}$, за който вече знаем, че не е регулярен.

Пример, за който критерият не е приложим

Да напомним, че условието на лемата за покачването представлява твърдението:

„Ако L е регулярен език, то е изпълнено $P_{reg}(L)$".

Естествено е да се запитаме дали имаме обратната посока на горната импликация, т.е. вярно ли е, че:

„Ако $P_{reg}(L)$ е изпълнено, то L е регулярен"?

Сега ще видим, че можем да посочим език L, за който условието $P_{\text{reg}}(L)$ е изпълнено, но въпреки това L не е регулярен. Това означава, че нямаме обратната посока на горната импликация и може да срещнем примери за езици, които макар и нерегулярни, не можем да докажем тяхната нерегулярност с помощта на контрапозицията на лемата за покачването. По-късно ще видим един пълен критерий за проверка за регулярност на език.

Пример 3.9. Езикът

$$L = \{c\}^+ \cdot \{a^n b^n \mid n \in \mathbb{N}\} \cup \{a\}^\star \cdot \{b\}^\star$$

не е регулярен, но условието $P_{\text{reg}}(L)$ е изпълнено.

Упътване. Ако допуснем, че L е регулярен, то тогава ще следва, че

$$L_1 = L \cap \mathcal{L}[\![\boldsymbol{ca^\star b^\star}]\!] = \{ca^n b^n \mid n \in \mathbb{N}\}$$

е регулярен, но с контрапозицията на лемата за покачването лесно се вижда, че L_1 не е. Сега да проверим, че $P_{\text{reg}}(L)$ е изпълнено.

(∃) Нека изберем $p = 1$.

(∀) Сега трябва да разгледаме всички думи $\alpha \in L$, за които $|\alpha| \geq 1$.

(∃) Нека разбием думата α на три части по следния начин:

$$x = \varepsilon,\ y = \alpha[0],\ z = \alpha[1:].$$

(∀) Съобразете, че за всяко $i \in \mathbb{N}$ имаме, че $xy^i z \in L$.

□

За да покажем, че $P_{\text{reg}}(L)$ е изпълнено, трябва да следваме стъпките:

(∃) Избираме конкретно число $p \geq 1$.

(∀) Разглеждаме произволна дума $\alpha \in L$ и $|\alpha| \geq p$.

(∃) Посочваме конкретно разбиване на думата α като $\alpha = xyz$ със свойството $|xy| \leq p$ и $|y| \geq 1$.

(∀) За всяко i трябва да покажем, че $xy^i z \in L$.

Приложения

[11, стр. 93].

В този раздел ще видим, че с помощта на лемата за покачването някои основни проблеми са алгоритмично разрешими за автоматни езици. По-нататък ще разгледаме тези проблеми и за други видове езици и ще видим, че не винаги те са алгоритмично разрешими.

Проблемът за празнота: Съществува ли алгоритъм, който по вход краен автомат $\mathcal{A}$, определя дали $\mathcal{L}(\mathcal{A}) = \emptyset$?

Проблемът за включване: Съществува ли алгоритъм, който по вход два крайни автомата $\mathcal{A}$ и $\mathcal{B}$, определя дали $\mathcal{L}(\mathcal{A}) \subseteq \mathcal{L}(\mathcal{B})$?

Проблемът за равенство: Съществува ли алгоритъм, който по вход два крайни автомата $\mathcal{A}$ и $\mathcal{B}$, определя дали $\mathcal{L}(\mathcal{A}) = \mathcal{L}(\mathcal{B})$?

Проблемът за безкрайност: Съществува ли алгоритъм, който по вход краен автомат $\mathcal{A}$, определя дали $|\mathcal{L}(\mathcal{A})| = \infty$?

Проблемът за универсалност: Съществува ли алгоритъм, който по вход краен автомат $\mathcal{A}$, определя дали $\mathcal{L}(\mathcal{A}) = \Sigma^\star$?

Вече имаме всичко необходимо за да отговорим сравнително лесно на тези въпроси. Първо ще разгледаме две помощни твърдения, които на практика следват директно от лемата за покачването.

Твърдение 3.10. Нека $\mathcal{A} = \langle \Sigma, Q, q_{\texttt{start}}, \delta, F \rangle$ е произволен ДКА. Тогава следните условия са еквивалентни:

(1) $\mathcal{L}(\mathcal{A})$ е *непразен*;

(2) $\mathcal{A}$ разпознава дума α, за която $|\alpha| < |Q|$.

Доказателство. Да положим $L = \mathcal{L}(\mathcal{A})$. Ще разгледаме двете посоки на твърдението.

$(1) \Rightarrow (2)$**.** Нека L 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 доказателството на лемата за покачването, съществува разбиване $xyz = \alpha$, такова че $xz \in L$. При положение, че $|y| \geq 1$, то $|xz| < m$, което е противоречие с минималността на думата α. Заключаваме, че нашето допускане е грешно. Тогава $m < |Q|$, откъдето следва, че съществува дума $\alpha \in L$ с $|\alpha| < |Q|$.

$(2) \Rightarrow (1)$**.** Тази посока е тривиална. Ако L съдържа дума α, за която $|\alpha| < |Q|$, то е очевидно, че L е непразен език.

□

Твърдение 3.11. Нека $\mathcal{A} = \langle \Sigma, Q, q_{\texttt{start}}, \delta, F \rangle$ е произволен ДКА. Тогава следните условия са еквивалентни:

(1) $\mathcal{L}(\mathcal{A})$ е *безкраен*;

(2) $\mathcal{A}$ разпознава дума α, за която $|Q| \leq |\alpha| < 2|Q|$.

Доказателство. Да положим $L = \mathcal{L}(\mathcal{A})$. Да разгледаме двете посоки на твърдението.

$(1) \Rightarrow (2)$**.** 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 лемата за покачването, 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 $xz \in L$.

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

□

Сега вече много лесно можем да видим, че следните проблеми са алгоритмично разрешими за автоматните езици.

(Проблемът за празнота) Според *Твърдение 3.10*, *Алгоритъм (2)* разрешава проблемът за празнота за автоматни езици.

Algorithm 2 Проблемът за празнота за автоматни езици

```
1: procedure isEmpty(A)
2:     for all n < |Q| do
3:         for all ω ∈ Σ^n do
4:             if belong(A, ω) then
5:                 return True
6:     return False
```

(Проблемът за включване) Проблемът за включването се свежда към проблема за празнота, защото за всеки два езика L_1 и L_2 е изпълнена веригата от еквивалентности:

$$\begin{aligned} L_1 \subseteq L_2 &\Leftrightarrow (L_1 \setminus L_2) = \emptyset \\ &\Leftrightarrow L_1 \cap \overline{L}_2 = \emptyset. \end{aligned}$$

Algorithm 3 Проблемът за включване за автоматни езици

```
1: procedure isIncluded(A, B)
2:     B_1 := complement(B)
3:     A_1 := intersect(A, B_1)
4:     return isEmpty(A_1)
```

(Проблемът за равенство) Понеже е изпълнена еквивалентността

$$L_1 = L_2 \Leftrightarrow L_1 \subseteq L_2 \ \&\ L_2 \subseteq L_1,$$

то е ясно, че проблемът за равенство на два автоматни езика е алгоритмично разрешим, защото просто трябва да приложим два пъти *Алгоритъм (3)*.

(Проблемът за безкрайност) От *Твърдение 3.11* директно получаваме следния прост алгоритъм, който разрешава проблемът за безкрайност на автоматен език.

Algorithm 4 Проблемът за безкрайност за автоматни езици

```
1: procedure isInfinite(𝒜)
2:     for all n := |Q|, ..., 2|Q| − 1 do
3:         for all ω ∈ Σ^n do
4:             if belong(𝒜, ω) then
5:                 return True
6:     return False
```

(Проблемът за универсалност) Този проблем отново е алгоритмично разрешим, защото имаме еквивалентността:

$$L = \Sigma^\star \;\Leftrightarrow\; \overline{L} = \emptyset.$$

3.9 Изоморфни автомати

Да разгледаме два произволни ДКА

$$\mathcal{A}_1 = (\Sigma, Q_1, q'_{\texttt{start}}, \delta_1, F_1),$$
$$\mathcal{A}_2 = (\Sigma, Q_2, q''_{\texttt{start}}, \delta_2, F_2).$$

Казваме, че $\mathcal{A}_1$ и $\mathcal{A}_2$ са **изоморфни**, което означаваме с $\mathcal{A}_1 \cong \mathcal{A}_2$, ако съществува *биективна* функция $f : Q_1 \to Q_2$, за която:

(1) $f(q'_{\texttt{start}}) = q''_{\texttt{start}}$;

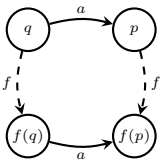
(2) $q \in F_1 \Leftrightarrow f(q) \in F_2$;

(3) $f(\delta_1(q, a)) = \delta_2(f(q), a)$.

С други думи, два автомата са изоморфни точно тогава, когато те са идентични с точност до преименуване на състоянията. Ще казваме, че f задава изоморфизъм на $\mathcal{A}_1$ върху $\mathcal{A}_2$ и ще означаваме $f : \mathcal{A}_1 \cong \mathcal{A}_2$.

[13, стр. 89]. Естествено, ние можем да дефинираме и изоморфни НКА, но за нашите цели е достатъчно да разгледаме само изоморфни ДКА.

Условие (3) може да се представи графично така:



Твърдение 3.12. Нека $f : \mathcal{A}_1 \cong \mathcal{A}_2$. Тогава за всяка дума α и състояние $q \in Q_1$ е изпълнена еквивалентността:

$$f(\delta_1^\star(q, \alpha)) = \delta_2^\star(f(q), \alpha). \tag{3.9}$$

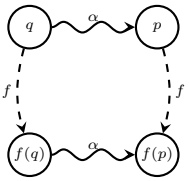
Доказателство. Ще докажем *Свойство 3.9* с индукция по дължината на думата α.

- Нека $|\alpha| = 0$, т.е. $\alpha = \varepsilon$. Ясно е, че *Свойство 3.9* е изпълнено за ε защото $\delta_1^\star(q, \varepsilon) = q$ и съответно $\delta_2^\star(f(q), \varepsilon) = f(q)$ за произволно състояние $q \in Q_1$.
- Да приемем, че *Свойство 3.9* е изпълнено за думи с дължина n.
- Да разгледаме произволна дума α с дължина $n+1$, т.е. $\alpha = \beta c$ и $|\beta| = n$. Тогава имаме следната верига от равенства:

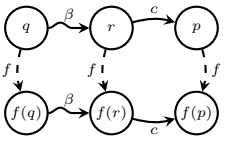
$$\begin{aligned}
f(\delta_1^\star(q, \beta c)) &= f(\delta_1(\overbrace{\delta_1^\star(q, \beta)}^{p}, c)) \\
&= f(\delta_1(p, c)) && \text{// } \textit{Свойство (3)} \\
&= \delta_2(f(p), c) \\
&= \delta_2(f(\delta_1^\star(q, \beta), c)) \\
&= \delta_2(\delta_2^\star(f(q), \beta), c) && \text{// } \textbf{(И.П.)} \text{ за } \beta \\
&= \delta_2^\star(f(q), \beta c).
\end{aligned}$$

$\square$

Свойство 3.9 може да се представи графично така:



Индукционната стъпка може да се представи така:



Твърдение 3.13. Ако $\mathcal{A}_1 \cong \mathcal{A}_2$, то $\mathcal{L}(\mathcal{A}_1) = \mathcal{L}(\mathcal{A}_2)$.

Упътване. Нека $f : \mathcal{A}_1 \cong \mathcal{A}_2$. Тогава имаме следните еквивалентности:

$$\begin{aligned}
\alpha \in \mathcal{L}(\mathcal{A}_1) &\Leftrightarrow \delta_1^\star(q'_{\texttt{start}}, \alpha) \in F_1 && \text{// деф. на } \mathcal{L}(\mathcal{A}_1) \\
&\Leftrightarrow f(\delta_1^\star(q'_{\texttt{start}}, \alpha)) \in F_2 && \text{// } \textit{Свойство (2)} \\
&\Leftrightarrow \delta_2^\star(f(q'_{\texttt{start}}), \alpha)) \in F_2 && \text{// } \textit{Твърдение 3.12} \\
&\Leftrightarrow \delta_2^\star(q''_{\texttt{start}}, \alpha) \in F_2 && \text{// } f(q'_{\texttt{start}}) \overset{\text{деф}}{=} q''_{\texttt{start}} \\
&\Leftrightarrow \alpha \in \mathcal{L}(\mathcal{A}_2). && \text{// деф. на } \mathcal{L}(\mathcal{A}_2)
\end{aligned}$$

$\square$

Лесно можем да съобразим, че в общия случай нямаме обратната посока на това твърдение.

3.10 Минимален автомат

Сега ще видим, че автоматът на Бжозовски $\mathcal{B}_L$ за даден регулярен език L е в известен смисъл най-добрият възможен. Накратко, $\mathcal{B}_L$ има най-малкия възможен брой състояния измежду всички детерминирани крайни автомати, които разпознават L. За да успеем да видим това, първо трябва да се подготвим.

Без ограничение на общността, нека приемем, че винаги разглеждаме само свързани ДКА $\mathcal{A}$, т.е. всяко състояние е достижимо от началното. Нека за всяка дума α да положим $q_\alpha \stackrel{\text{деф}}{=} \delta^\star(q_{\mathtt{start}}, \alpha)$. Понеже $\mathcal{A}$ е свързан, то всяко състояние на $\mathcal{A}$ може да се разглежда като q_α за някоя дума α.

Тук използваме, че δ е тотална функция. За някои състояния p може да съществуват безкрайно много думи α, за които $q_\alpha = p$.

Твърдение 3.14. Нека $L = \mathcal{L}(\mathcal{A})$. Тогава за всяка дума α е изпълнено:

$$\mathcal{L}_\mathcal{A}(q_\alpha) = \alpha^{-1}(L).$$

Доказателство. За произволна дума ω имаме следната верига от еквивалентности:

$$\begin{aligned}
\omega \in \mathcal{L}_\mathcal{A}(q_\alpha) &\Leftrightarrow \delta^\star(q_\alpha, \omega) \in F && \text{// от деф. на } \mathcal{L}_\mathcal{A}(q_\alpha)\\
&\Leftrightarrow \delta^\star(\delta^\star(q_{\mathtt{start}}, \alpha), \omega) \in F && \text{// } q_\alpha \stackrel{\text{деф}}{=} \delta^\star(q_{\mathtt{start}}, \alpha)\\
&\Leftrightarrow \delta^\star(q_{\mathtt{start}}, \alpha\omega) \in F && \text{// } \textit{Твърдение 3.1}\\
&\Leftrightarrow \alpha\omega \in \mathcal{L}(\mathcal{A}) && \text{// деф. на } \mathcal{L}(\mathcal{A})\\
&\Leftrightarrow \alpha\omega \in L && \text{// } L = \mathcal{L}(\mathcal{A})\\
&\Leftrightarrow \omega \in \alpha^{-1}(L). && \text{// деф. на } \alpha^{-1}(L)
\end{aligned}$$

□

Твърдение 3.15. Нека L е произволен език. Тогава за произволно състояние M на $\mathcal{B}_L$ имаме равенството:

$$\mathcal{L}_{\mathcal{B}_L}(\underbrace{M}_{\text{състояние}}) = \underbrace{M}_{\text{език}}.$$

Доказателство. Понеже за произволна дума α имаме еквивалентностите

$$\begin{aligned}
\alpha \in \mathcal{L}_{\mathcal{B}_L}(M) &\Leftrightarrow \delta_L^\star(M, \alpha) \in F_L\\
&\Leftrightarrow \varepsilon \in \alpha^{-1}(M) && \text{// деф. на } F_L\\
&\Leftrightarrow \alpha \in M, && \text{// от (2.3)}
\end{aligned}$$

заключаваме, че $\mathcal{L}_{\mathcal{B}_L}(M) = M$. □

Твърдение 3.16. Нека A и B са множества, като A е крайно, за които съществува *сюрективна* функция $f : A \to B$. Тогава $|B| \leq |A|$.

Упътване. Понеже A е крайно множество, можем да изброим елементите му в редица. Нека $A = \{a_0, a_1, \ldots, a_{n-1}\}$. Разгледайте $g : B \to A$, където

Вярно ли е това твърдение, ако A е безкрайно множество ?

$$g(b) \stackrel{\text{деф}}{=} a_m \text{ за } m = \min\{i < n \mid f(a_i) = b\}.$$

Да отбележим, че дефиницията на g е коректна, защото множеството

$$\{i < n \mid f(a_i) = b\}$$

е непразно, понеже f е сюрективна. Докажете, че g е инективна. □

Лема 3.8. Нека L е регулярен език и $\mathcal{A}$ е произволен ДКА, за който $L = \mathcal{L}(\mathcal{A})$. Тогава $|Q_L| \leq |Q_\mathcal{A}|$.

Тази лема ни казва, че автоматът на Бжозовски има възможно най-малкия брой състояния измежду всички ДКА разпознаващи L [19, стр. 113].

Доказателство. Да разгледаме функцията $f : Q_\mathcal{A} \to Q_L$ зададена по следния начин:

$$f(q) \stackrel{\text{деф}}{=} \mathcal{L}_\mathcal{A}(q). \tag{3.10}$$

Първо, да видим защо f е добре дефинирана функция, т.е. да видим защо за всяко състояние $q \in Q_\mathcal{A}$, имаме $f(q) \in Q_L$. Да напомним, че приехме още в началото на раздела, че $\mathcal{A}$ е свързан автомат. Това означава, че за всяко състояние p, съществува дума α, за която $p = \delta^\star_\mathcal{A}(q^\mathcal{A}_{\texttt{start}}, \alpha)$, т.е. $p = q_\alpha$ според означението, което въведохме в началото на *Раздел 3.10*. Тогава от *Твърдение 3.14* следва, че $f(q_\alpha) = \alpha^{-1}(L) \in Q_L$, защото $\alpha^{-1}(L) = \mathcal{L}_\mathcal{A}(q_\alpha)$.

Второ, да видим, че f е сюрективна функция. За тази цел, да разгледаме произволно състояние $M \in Q_L$, което означава, че има дума α, за която $M = \alpha^{-1}(L)$. Отново според *Твърдение 3.14*,

$$f(q_\alpha) = \mathcal{L}_\mathcal{A}(q_\alpha) = \alpha^{-1}(M).$$

Сега от *Твърдение 3.16* можем да заключим, че

$$|Q_L| \leq |Q_\mathcal{A}|.$$

□

Следствие 3.4 (Критерий за регулярност на език)**.** Един език L е регулярен точно тогава, когато $\mathcal{B}_L$ има крайно много състояния.

Доказателство. Ако сме избрали алгебричния подход описан в *Раздел 3.3*, то ние отдавна знаем този критерий формулиран като *Следствие 3.4*. □

Доказателство. Ако сме избрали автоматния подход описан в *Раздел* **??**, то ние чак сега можем да докажем този критерий. Нека L е регулярен. Тогава от *Теорема* **??**, $L = \mathcal{L}(\mathcal{A})$ за някой ДКА $\mathcal{A}$, защото НКА разпознават същите езици като ДКА. Да напомним, че от *Твърдение 3.6* също имаме и $\mathcal{L}(\mathcal{B}_L) = L$. Понеже от *Лема 3.8* имаме, че $|Q_L| \leq |Q_\mathcal{A}|$, то $\mathcal{B}_L$ има краен брой състояния.

Обратно, ако $\mathcal{B}_L$ има крайно много състояния, то тогава $\mathcal{B}_L$ е ДКА. Понеже $\mathcal{L}(\mathcal{B}_L) = L$ според *Твърдение 3.6*, то L е автоматен и следователно регулярен по теоремата на Клини, за която имаме две доказателства - автоматен подход и алгебричен подход. □

Следствие 3.5. Нека L е регулярен език. Тогава автоматът $\mathcal{B}_L$, построен по метода на Бжозовски за L, има *минималния* възможен брой състояния измежду всички детерминирани крайни автомати разпознаващи L.

✍ Проверете, че за произволен регулярен език L, за всеки ДКА $\mathcal{A}$, за който $\mathcal{L}(\mathcal{A}) = L$ е изпълнено, че $|F_L| \leq |F_\mathcal{A}|$.

Следващото твърдение е до голяма степен очевидно, но за пълнота на изложението, ще го разгледаме подробно.

Твърдение 3.17. Нека A и B са крайни равномощни множества. Докажете, че ако $g : A \to B$ е сюрекция, то g е биекция.

Ясно е, че щом A и B са равномощни, то има биекция между тях. Тук доказваме, че всяка сюрекция между тях е също така и биекция. Това твърдение трябва да може да докажете сами! Да напомним, че от курса по Дискретна математика имаме формулата $|X \cup Y| = |X| + |Y| + |X \cap Y|$. Просто в нашия случай $|X \cap Y| = 0$.

Доказателство. Нека $B = \{b_0, \ldots, b_{n-1}\}$. За всеки индекс $i < n$ да положим

$$A_i \stackrel{\text{деф}}{=} \{a \in A \mid g(a) = b_i\}.$$

Щом g е сюрекция, то $A_i \neq \emptyset$ за всеки индекс $i < n$. Понеже g е функция, то $A_i \cap A_j = \emptyset$ за всеки два различни индекса i и j. Това означава, че

$$n = |A| = |\bigcup_{i<n} A_i| = \sum_{i<n} |A_i|.$$

Оттук следва, че щом за всяко i имаме, че $|A_i| \neq 0$, то $|A_i| = 1$. Заключаваме, че g е инекция, защото в противен случай щяхме да имаме някое i, за което $|A_i| > 1$. □

Теорема 3.6. За всеки регулярен език L съществува *единствен* минимален ДКА с точност до изоморфизъм.

С други думи, ако имаме два минимални ДКА за L, то можем да получим единия автомат от другия чрез внимателно преименуване на състоянията.

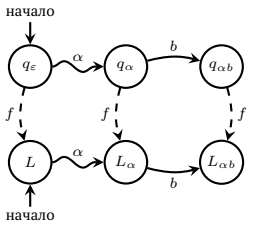
Доказателство. Вече знаем, че автоматът $\mathcal{B}_L$, построен по метода на Бжозовски за L, има минималния възможен брой състояния. Нека $\mathcal{A}$ е друг ДКА разпознаващ L и $|Q_\mathcal{A}| = |Q_L|$. Трябва да докажем, че $\mathcal{A} \cong \mathcal{B}_L$. Ясно е, че $\mathcal{A}$ е свързан автомат, т.е. всяко състояние p на $\mathcal{A}$ е от вида $p = q_\alpha$. В противен случай, $\mathcal{A}$ нямаше да бъде минимален. Да разгледаме функцията $f : Q_\mathcal{A} \to Q_L$, където

$$f(p) \stackrel{\text{деф}}{=} \mathcal{L}_\mathcal{A}(p). \tag{3.11}$$

От *Лема 3.8* знаем, че f е сюрективна. Понеже $|Q_\mathcal{A}| = |Q_L|$, то от *Твърдение 3.17* имаме, че f е всъщност биекция. Остава да видим защо $\mathcal{A} \cong_f \mathcal{B}$. Да напомним, че можем да разглеждаме състоянията на $Q_\mathcal{A}$ като q_α и тогава $\mathcal{L}_\mathcal{A}(q_\alpha) = \alpha^{-1}(L)$ от *Твърдение 3.14*.

- За началното състояние имаме, че:

$$\begin{aligned} f(q^\mathcal{A}_{\texttt{start}}) &= f(q_\varepsilon) && \text{// } q^\mathcal{A}_{\texttt{start}} = q_\varepsilon \\ &= \mathcal{L}_\mathcal{A}(q_\varepsilon) && \text{// от деф. на } f \\ &= \varepsilon^{-1}(L) && \text{// } \textit{Твърдение 3.14} \\ &= L. \end{aligned}$$



Тук сме означили

$L_\alpha = \alpha^{-1}(L)$

$L_{\alpha b} = (\alpha b)^{-1}(L)$.

- За финалните състояния имаме, че:

$$\begin{aligned} q_\alpha \in F_\mathcal{A} &\Leftrightarrow \delta^\star_\mathcal{A}(q_{\texttt{start}}, \alpha) \in F_\mathcal{A} && \text{// } q_\alpha = \delta^\star_\mathcal{A}(q_{\texttt{start}}, \alpha) \\ &\Leftrightarrow \alpha \in \mathcal{L}(\mathcal{A}) && \text{// деф. на } \mathcal{L}(\mathcal{A}) \\ &\Leftrightarrow \varepsilon \in \alpha^{-1}(L) && \text{// } L = \mathcal{L}(\mathcal{A}) \\ &\Leftrightarrow \alpha^{-1}(L) \in F_L. && \text{// деф. на } F_L \end{aligned}$$

- Остава да докажем, че за произволна буква b и произволни състояния $q, p \in Q_\mathcal{A}$ е изпълнено, че:

$$\delta_L(f(q), b) = f(\delta_\mathcal{A}(q, b)),$$

или с други думи,

$$\delta_L(\mathcal{L}_\mathcal{A}(q), b) = \mathcal{L}_\mathcal{A}(\delta_\mathcal{A}(q, b)). \tag{3.12}$$

Това се вижда директно от веригата от еквивалентности:

$$\begin{aligned}
\omega \in \delta_L(\mathcal{L}_\mathcal{A}(q), b) &\Leftrightarrow \omega \in b^{-1}(\mathcal{L}_\mathcal{A}(q)) \\
&\Leftrightarrow b\omega \in \mathcal{L}_\mathcal{A}(q) \\
&\Leftrightarrow \delta^\star_\mathcal{A}(q, b\omega) \in F_\mathcal{A} \\
&\Leftrightarrow \delta^\star_\mathcal{A}(\delta_\mathcal{A}(q, b), \omega) \in F_\mathcal{A} \\
&\Leftrightarrow \omega \in \mathcal{L}_\mathcal{A}(\delta^\star_\mathcal{A}(q, b)).
\end{aligned}$$

Така доказахме, че f задава изоморфизъм между $\mathcal{A}$ и $\mathcal{B}_L$. □

Следствие 3.6 (Критерий за минималност)**.** Нека $\mathcal{A}$ е ДКА, при който всяко състояние е достижимо от началното, разпознаващ регулярния език L. Тогава $\mathcal{A}$ е *минимален* автомат за езика L точно тогава, когато е изпълнена импликацията:

$$(\forall p \in Q_\mathcal{A})(\forall q \in Q_\mathcal{A})[\, p \neq q \implies \mathcal{L}_\mathcal{A}(p) \neq \mathcal{L}_\mathcal{A}(q)\,]. \tag{3.13}$$

Доказателство. Нека $\mathcal{B}_L$ е автоматът на Бжозовски за L и да разгледаме функцията $f : Q_\mathcal{A} \to Q_L$ дефинирана като $f(q) = \mathcal{L}_\mathcal{A}(q)$. Първо, нека $\mathcal{A}$ е минимален автомат за L. Тогава знаем от *Теорема 3.6*, че f е биекция. От инективността на f следва, че имаме импликцията (3.13).

Биекция = инекция + сюрекция.

Второто, нека импликацията (3.13) е изпълнена. Това означава, че f е инективна функция, откъдето имаме, че $|Q_\mathcal{A}| \leq |Q_L|$. Понеже $\mathcal{B}_L$ е минимален автомат, то $\mathcal{A}$ също е минимален автомат. □

3.11 Критерий за регулярност (алгебричен подход)

За да докажем, че един език L не е регулярен можем да приложим *Следствие 3.4* като докажем, че автоматът на Бжозовски за L има безкрайно много състояния. Обърнете внимание, че не е нужно да намерим всички състояния на автомата $\mathcal{B}_L$, а само това, че са безкрайно много.

Сравнете с *Пример 3.2*.

Пример 3.10. За езика $L = \{a^n b^n \mid n \in \mathbb{N}\}$ имаме, че за произволни естествени числа k и m е изпълнено следното свойство:

$$k < m \implies (a^k)^{-1}(L) \neq (a^m)^{-1}(L).$$

Проверете, че $(a^k)^{-1}(L) = \{a^n b^{n+k} \mid n \in \mathbb{N}\}$, за всяко $k \in \mathbb{N}$. Така получаваме, че автоматът на Бжозовски за L ще има безкрайно много състояния. Заключаваме, че този език **не** е регулярен.

Пример 3.11. За езика $L = \{a^{n^2} \mid n \in \mathbb{N}\}$ имаме, че за произволни естествени числа k и m е изпълнено следното свойство:

$$k < m \implies (a^{k^2})^{-1}(L) \neq (a^{m^2})^{-1}(L).$$

За да покажем, че $(a^{k^2})^{-1}(L) \neq (a^{m^2})^{-1}(L)$ е достатъчно да посочим дума γ, за която $\gamma \in (a^{k^2})^{-1}(L)$, но $\gamma \notin (a^{m^2})^{-1}(L)$. Да разгледаме думата $\gamma = a^{2k+1}$. Ясно е, че $\gamma \in (a^{k^2})^{-1}(L)$, защото $a^{k^2}\gamma = a^{(k+1)^2} \in L$, но понеже $k < m$, то

$$m^2 < m^2 + 2k + 1 < m^2 + 2m + 1 = (m+1)^2$$

и следователно $\gamma \notin (a^{m^2})(L)$, защото $a^{m^2}\gamma = a^{m^2+2k+1} \notin L$. Заключаваме, че този език **не** е регулярен.

Пример 3.12. За езика $L = \{a^{n!} \mid n \in \mathbb{N}\}$ имаме, че за произволни естествени числа k и m е изпълнено следното свойство:

$$k < m \implies (a^{k!})^{-1}(L) \neq (a^{m!})^{-1}(L).$$

Да разгледаме $k < m$ и думата $\gamma = a^{(k!)k}$. Тогава $\gamma \in (a^{k!})^{-1}(L)$, защото $a^{k!}\gamma = a^{k!+(k!)k} = a^{(k+1)!} \in L$, но

$$m! < m! + (k!)k < m! + (m!)m = (m+1)!$$

и следователно $\gamma \notin (a^{m!})^{-1}(L)$, защото $a^{m!}\gamma = a^{m!+(k!)k} \notin L$. Заключаваме, че този език **не** е регулярен.

Задача 3.18. Докажете, че езикът

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

не е регулярен.

3.12 Минимизация (алгебричен подход)

Този подход за минимизация е описан добре в [20, стр. 88].

Нека $\mathcal{A} = (\Sigma, Q, \delta, q_{\texttt{start}}, F)$ е детерминиран краен автомат. Тогава полагаме

$$\mathcal{A}_q \stackrel{\text{деф}}{=} (\Sigma, Q, \delta, q, F).$$

Нека $\mathcal{A} = (\Sigma, Q, \Delta, Q_{\texttt{start}}, F)$ е недетерминиран краен автомат. Тогава полагаме

$$\mathcal{A}_q \stackrel{\text{деф}}{=} (\Sigma, Q, \Delta, \{q\}, F).$$

За произволен краен автомат $\mathcal{A}$, за да не пишем много индекси, полагаме

$$\mathcal{L}_{\mathcal{A}}(q) \stackrel{\text{деф}}{=} \mathcal{L}(\mathcal{A}_q).$$

Нека да започнем с едно твърдение, което ни дава критерий, кога детерминизацията ни дава автомат с минимален брой състояния.

Твърдение 3.18. Нека $\mathcal{A}$ е НКА със следните свойства:

(а) $(\forall q \in Q_{\mathcal{A}})[\, \mathcal{L}_{\mathcal{A}}(q) \neq \emptyset \,]$;

(б) $(\forall p \in Q_{\mathcal{A}})(\forall q \in Q_{\mathcal{A}})[\, p \neq q \implies \mathcal{L}_{\mathcal{A}}(q) \cap \mathcal{L}_{\mathcal{A}}(p) = \emptyset \,]$.

Тогава $\mathcal{M} = \texttt{det}(\mathcal{A})$, получен по метода на Рабин-Скот, е минимален автомат за езика на $\mathcal{A}$.

Упътване. От контрукцията на $\mathcal{M}$ в теоремата на Рабин-Скот, знаем, че всяко състояние на $\mathcal{M}$ е достижимо от началното. Критерият за минималност ни казва, че за да докажем, че $\mathcal{M}$ е минимален е достатъчно да покажем, че за произволни състояния P и U на автомата $\mathcal{M}$ е изпълнена импликацията:

$$\mathcal{L}_{\mathcal{M}}(P) = \mathcal{L}_{\mathcal{M}}(U) \implies P = U.$$

Критерият за минималност на детерминиран автомат $\mathcal{A}$ гласи, че за всеки две различни състояния p и q, то $\mathcal{L}_{\mathcal{A}}(p) \neq \mathcal{L}_{\mathcal{A}}(q)$. Понеже тук $\mathcal{A}$ е недетерминиран, искаме по-силното свойство (б), а именно $\mathcal{L}_{\mathcal{A}}(p) \cap \mathcal{L}_{\mathcal{A}}(q) = \emptyset$. Съобразете, че това означава, че $\mathcal{A}$ има само едно финално състояние.

И така, да вземем две такива състояния P и U, за които $\mathcal{L}_{\mathcal{M}}(P) = \mathcal{L}_{\mathcal{M}}(U)$. Достатъчно е да докажем, че $P \subseteq U$, защото доказателството на другата посока е симетрично.

Да разгледаме произволен елемент $p \in P$. Щом от (а) имаме, че $\mathcal{L}_{\mathcal{A}}(p) \neq \emptyset$, то да разгледаме една дума $\omega \in \mathcal{L}_{\mathcal{A}}(p)$. Знаем, че $\Delta^{\star}_{\mathcal{A}}(\{p\}, \omega) \cap F_{\mathcal{A}} \neq \emptyset$ или с други думи, $\delta^{\star}_{\mathcal{M}}(P, \omega) \in F_{\mathcal{M}}$, откъдето имаме, че $\omega \in \mathcal{L}_{\mathcal{M}}(P)$. Сега, понеже приехме, че $\mathcal{L}_{\mathcal{M}}(P) = \mathcal{L}_{\mathcal{M}}(U)$, то $\delta^{\star}_{\mathcal{M}}(U, \omega) \in F_{\mathcal{M}}$, което означава, че за някое състояние $u \in U$, $\Delta^{\star}_{\mathcal{A}}(\{u\}, \omega) \cap F_{\mathcal{A}} \neq \emptyset$. Така получаваме, че $\omega \in \mathcal{L}_{\mathcal{A}}(u)$. От (б) следва, че $u = p$, защото $\mathcal{L}_{\mathcal{A}}(p) \cap \mathcal{L}_{\mathcal{A}}(u) \neq \emptyset$. Заключаваме, че $P \subseteq U$. □

Твърдение 3.19. Нека $\mathcal{D}$ е ДКА, за който всяко състояние е достижимо от началното. Да разгледаме $\mathcal{A} \stackrel{\text{деф}}{=} \texttt{rev}(\mathcal{D})$. Тогава са изпълнени двете свойства от условието на *Твърдение 3.18*, а именно:

(а) $(\forall q \in Q_{\mathcal{A}})[\, \mathcal{L}_{\mathcal{A}}(q) \neq \emptyset \,]$;

(б) $(\forall p \in Q_{\mathcal{A}})(\forall q \in Q_{\mathcal{A}})[\, p \neq q \implies \mathcal{L}_{\mathcal{A}}(q) \cap \mathcal{L}_{\mathcal{A}}(p) = \emptyset \,]$.

Упътване. Да напомним, че според *Задача 3.11*, $\mathcal{A} = (\Sigma, Q, \Delta, F, \{q_{\texttt{start}}\})$, където

$$\Delta_{\mathcal{A}}(q, a) \stackrel{\text{деф}}{=} \{p \in Q \mid \delta_{\mathcal{D}}(p, a) = q\}.$$

Щом в $\mathcal{D}$ всяко състояние q е достижимо от началното, то е ясно, че $\mathcal{L}_{\mathcal{A}}(q) \neq \emptyset$, което ни дава първото свойство. За второто свойство, нека приемем, че има дума $\omega \in \mathcal{L}_{\mathcal{A}}(p) \cap \mathcal{L}_{\mathcal{A}}(q)$. Ще

докажем, че $p = q$. И така, щом $\omega \in \mathcal{L}_{\mathcal{A}}(p)$, то $\Delta^{\star}_{\mathcal{A}}(\{p\}, \omega) \ni q_{\texttt{start}}$, защото $q_{\texttt{start}}$ е единственото финално състояние на $\mathcal{A}$. Това означава, че $\delta^{\star}_{\mathcal{D}}(q_{\texttt{start}}, \omega) = p$. От друга страна, имаме също, че $\omega \in \mathcal{L}_{\mathcal{A}}(q)$, откъдето получаваме, че $\delta^{\star}_{\mathcal{D}}(q_{\texttt{start}}, \omega) = q$. Ясно е, че $p = q$, защото $\mathcal{D}$ е детерминиран автомат. □

Лема 3.9. Нека $\mathcal{D}$ е ДКА, за който всяко състояние е достижимо от началното. Тогава детерминираният краен автомат $\mathcal{M}$ получен като

$$\mathcal{M} \stackrel{\text{деф}}{=} \texttt{det}(\texttt{rev}(\mathcal{D}))$$

е минимален за езика $L^{\texttt{rev}}(\mathcal{D})$.

Упътване. Просто комбинираме горните две твърдения. Нека $\mathcal{A} \stackrel{\text{деф}}{=} \texttt{rev}(\mathcal{D})$. Знаем, че $\mathcal{L}(\mathcal{A}) = \mathcal{L}^{\texttt{rev}}(\mathcal{D})$. От *Твърдение 3.19* знаем, че $\mathcal{A}$ притежава свойствата необходими за приложението на *Твърдение 3.18*. Така получаваме, че $\mathcal{M} \stackrel{\text{деф}}{=} \texttt{det}(\mathcal{A})$ е минимален детерминиран автомат за езика $\mathcal{L}(\mathcal{A})$. □

По този начин получаваме алгоритъм за минимизация на автомат.

Теорема 3.7 (Бжозовски)**.** Нека $\mathcal{A}$ е ДКА, за който всяко състояние е достижимо от началното. Тогава детерминираният краен автомат

$$\mathcal{M} \stackrel{\text{деф}}{=} \texttt{det}(\texttt{rev}(\texttt{det}(\texttt{rev}(\mathcal{A}))))$$

е минимален за езика на $\mathcal{A}$.

3.13 Допълнителни задачи

3.13.1 Лесни задачи

Задача 3.19. За всеки от следните езици L, постройте минимален краен детерминиран автомат $\mathcal{A}$, който разпознава езика L, където:

$|\omega|_a \stackrel{\text{деф}}{=}$ броят на срещанията на буквата a в думата ω,
$|\omega| \stackrel{\text{деф}}{=}$ дължината на ω.

а) $L = \{a^n b \mid n \geq 0\}$;
б) $L = \{a, b\}^\star \setminus \{\varepsilon\}$;
в) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a$ и $|\omega|_b$ са четни$\}$;
г) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a$ е четно $\&$ $|\omega|_b$ е нечетно$\}$;
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) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a \geq 2 \ \vee \ |\omega|_b \leq 3\}$;
и) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a \geq 2 \ \& \ |\omega|_b \geq 1\}$;
к) $L = \{\omega \in \{a, b\}^\star \mid$ на всяка нечетна позиция на ω е буквата $a\}$;
л) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a$ е четно $\&$ $|\omega|_b \leq 1\}$;
м) $L = \{\omega \in \{a, b\}^\star \mid |\omega| \leq 3\}$;
н) $L = \{\omega \in \{a, b\}^\star \mid \omega$ не започва с $ab\}$;
о) $L = \{\omega \in \{a, b\}^\star \mid \omega$ завършва с ab или ba$\}$;
п) $L = \{\omega \in \{a, b\}^\star \mid \omega$ започва или завършва с $a\}$;
р) $L = \{\omega \in \{a, b\}^\star \mid \omega$ започва с $a \ \Leftrightarrow \ \omega$ завършва с $b\}$;
с) $L = \{\omega \in \{a, b\}^\star \mid |\omega| \equiv 0 \pmod 2 \ \& \ |\omega|_a = 1\}$;
т) $L = \{\omega \in \{a, b\}^\star \mid$ всяко a в ω се следва веднага от поне едно $b\}$;
у) $L = \{\omega \in \{a, b\}^\star \mid |\omega| \equiv 0 \bmod 3\}$;
ф) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a \equiv 1 \bmod 3\}$;
х) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a \equiv 0 \bmod 3 \ \& \ |\omega|_b \equiv 1 \bmod 2\}$;
ц) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a \equiv 0 \bmod 2 \ \vee \ |\omega|_b = 2\}$;
ч) $L = \{\omega \in \{a, b\}^\star \mid \omega$ съдържа равен брой срещания на ab и на $ba\}$;
ш) $L = \{\omega_1 \sharp \omega_2 \sharp \omega_3 \mid |\omega_1| \geq 2 \ \& \ |\omega_2| \geq 3 \ \& \ |\omega_3| \geq 4 \ \& \ \omega_i \in \{a, b\}^\star$ за $i = 1, 2, 3\}$.

Задача 3.20. Нека $\Sigma = \{a, b\}$. Проверете дали L е регулярен, където

а) $L = \{a^i b^i \mid i \in \mathbb{N}\}$;
б) $L = \{a^i b^i \mid i, j \in \mathbb{N} \ \& \ i \neq j\}$;
в) $L = \{a^i b^j \mid i > j\}$;
г) $L = \{a^n b^m \mid n$ дели $m\}$.
д) $L = \{a^{2n} \mid n \geq 1\}$;
е) $L = \{a^m b^n a^{m+n} \mid m \geq 1 \ \& \ n \geq 1\}$;
ж) $L = \{a^{n.m} \mid n, m$ са прости числа$\}$;
з) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a = |\omega|_b\}$;
и) $L = \{\omega\omega \mid \omega \in \{a, b\}^\star\}$;
к) $L = \{\omega\omega^{\mathtt{rev}} \mid \omega \in \{a, b\}^\star\}$;
л) $L = \{\alpha\beta\beta \in \{a, b\}^\star \mid \beta \neq \varepsilon\}$;
м) $L = \{a^n b^n c^n \mid n \geq 0\}$;
н) $L = \{\omega\omega\omega \mid \omega \in \Sigma^\star\}$;
о) $L = \{a^{2^n} \mid n \geq 0\}$;
п) $L = \{a^m b^n \mid n \neq m\}$;
р) $L = \{a^{n!} b^{n!} \mid n \neq 1\}$;
с) $L = \{a^{f_n} \mid f_0 = f_1 = 1 \ \& \ f_{n+2} = f_{n+1} + f_n\}$;
т) $L = \{\alpha \in \Sigma^\star \mid |\, |\alpha|_a - |\alpha|_b \,| \leq 2\}$;
у) $L = \{\alpha\beta\alpha \mid \alpha, \beta \in \Sigma^\star \ \& \ |\beta| \leq |\alpha|\}$;
ф) $L = \{\beta\gamma\gamma^{\mathtt{rev}} \mid \beta, \gamma \in \Sigma^\star \ \& \ |\beta| \leq |\gamma|\}$;
х) $L = \{c^k a^n b^m \mid k, m, n > 0 \ \& \ n \neq m\}$;
ц) $L = \{c^k a^n b^n \mid k > 0 \ \& \ n \geq 0\} \cup \{a, b\}^\star$;
ч) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a$ не дели $|\omega|_b\}$;
ш) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a < |\omega|_b\}$;
щ) $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a = 2|\omega|_b\}$;
ю) $L = \{\omega \in \{a, b\}^\star \mid |\, |\omega|_a - |\omega|_b \,| \leq 3\}$.

Задача 3.21. Докажете, че следните езици са регулярни:

а) $L = \{\alpha \in \{a, b\}^\star \mid$ за всяка представка ω на α имаме $|\, |\omega|_a - |\omega|_b \,| \leq 2\}$;

б) $L = \{\alpha \in \{a, b\}^\star \mid$ за някоя представка ω на α имаме $|\, |\omega|_a - |\omega|_b \,| > 2\}$;

в) $L = \{\alpha \in \{a, b\}^\star \mid$ за някоя наставка ω на α имаме $|\, |\omega|_a - |\omega|_b \,| > 2\}$.

Задача 3.22. Нека $\Sigma = \{a, b, c, d\}$. Да се докаже, че езикът

$$L = \{a_1 a_2 \cdots a_{2n} \in \Sigma^\star \mid (\forall j \in [1, n])[a_{2j-1} = a_{2j}] \,\&\, d \text{ се среща} \leq 3 \text{ пъти}\}$$

е регулярен.

Задача 3.23. Нека L_1 и L_2 са регулярни езици. Докажете, че L също е регулярен език, където

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

Задача 3.24. Нека Σ_1 и Σ_2 са непресичащи се азбуки, а L_1 и L_2 са езици съответно над Σ_1 и Σ_2. За една дума $\omega \in (\Sigma_1 \cup \Sigma_2)^\star$, нека с $\omega_i \in \Sigma_i^\star$ да означим редицата от букви от Σ_i в реда, в който се срещат в ω. Да разгледаме следния език

$$L_1 \oplus L_2 = \{\omega \in (\Sigma_1 \cup \Sigma_2)^\star \mid \omega_1 \in L_1 \,\&\, \omega_2 \in L_2 \,\&\, |\omega_1| = |\omega_2|\}.$$

Да а) Вярно ли е, че ако L_1 е краен, то $L_1 \oplus L_2$ е регулярен език?

Не б) Вярно ли е, че ако L_1 и L_2 са регулярни езици, то $L_1 \oplus L_2$ е регулярен език?

Задача 3.25. Нека Σ_1 и Σ_2 са непресичащи се азбуки, а L_1 и L_2 са езици съответно над Σ_1 и Σ_2. За една дума $\omega \in (\Sigma_1 \cup \Sigma_2)^\star$, нека с $\omega_i \in \Sigma_i^\star$ да означим редицата от букви от Σ_i в реда, в който се срещат в ω. Да разгледаме следния език

$$L_1 \oplus L_2 = \{\omega \in (\Sigma_1 \cup \Sigma_2)^\star \mid \omega_1 \in L_1 \,\&\, \omega_2 \in L_2\}.$$

Да Вярно ли е, че ако L_1 и L_2 са регулярни езици, то $L_1 \oplus L_2$ е регулярен език?

[7, стр. 88]. **Задача 3.26.** Нека $\Sigma = \{a, b, c\}$ и да разгледаме следната операция

$$\mathtt{Sort} : \mathcal{P}(\Sigma^\star) \to \mathcal{P}(\Sigma^\star),$$

където

$$\mathtt{Sort}(L) = \{a^{|\omega|_a} b^{|\omega|_b} c^{|\omega|_c} \mid \omega \in L\}.$$

Вярно ли е, че ако L е регулярен, то $\mathtt{Sort}(L)$ е регулярен ?

3.13.2 Не толкова лесни задачи

Задача 3.27. При дадени езици L и M над азбуката Σ, да разгледаме: [16, стр. 84]

а) $\texttt{NoExtend}(L) = \{\alpha \in L \mid \alpha$ не е префикс на никоя дума от $L\}$;

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

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

г) $L/_\omega = \{\alpha \in \Sigma^\star \mid \alpha\omega \in L\}$; right quotient of L by ω

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

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 M са регулярни. Освен това, докажете, че $L/_M$ е регулярен и при условието, че L е регулярен, но M е произволен език над азбуката Σ. Тази конструкция няма да бъде ефективна

Задача 3.28. Да фиксираме азбука само с един символ $\Sigma = \{a\}$. Да положим за всяко $p, q \in \mathbb{N}$, [13, стр. 75]; [16, стр. 89]

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

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

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

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 аритметична прогресия.

Упътване.

а) За едната посока, разгледайте ДКА за L.

б) За втората част, разгледайте $h : \Sigma \to \{a\}$ деф. като $(\forall b \in \Sigma)[h(b) = a]$. Докажете, че h е 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 3.29. Вярно ли е, че:

- $\{a^m \mid a^{m^2} \in L(p, q)\}$ е регулярен език ?
- $\{a^m \mid a^{2^m} \in L(p, q)\}$ е регулярен език ?

Задача 3.30. За даден език L над азбуката Σ, да разгледаме езиците:

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

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

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

Задача 3.31. Да разгледаме езика

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

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 3.32. Да фиксираме една азбука Σ. Да дефинираме функция $\texttt{diff} : \Sigma^\star \times \Sigma^\star \to \mathbb{N}$, която брои колко разлики има между α и β по следния начин:

$$\texttt{diff}(\varepsilon, \beta) \stackrel{\text{деф}}{=} |\beta|$$

$$\texttt{diff}(\alpha, \varepsilon) \stackrel{\text{деф}}{=} |\alpha|$$

$$\texttt{diff}(x\alpha, y\beta) \stackrel{\text{деф}}{=} \begin{cases} 1 + \texttt{diff}(\alpha, \beta), & \text{ако } x \neq y \\ \texttt{diff}(\alpha, \beta), & \text{иначе.} \end{cases}$$

За произволни езици L и M, да дефинираме езика

$$\texttt{Diff}_n(L, M) \stackrel{\text{деф}}{=} \{\alpha \in L \mid (\exists \beta \in M)[\texttt{diff}(\alpha, \beta) = n]\}.$$

Докажете, че ако L и M са регулярни, то $\texttt{Diff}_n(L, M)$ е регулярен.

Да обърнем внимание, че езикът $\{\alpha\sharp\beta\sharp\gamma \mid \overline{\alpha}_{(2)} + \overline{\beta}_{(2)} = \overline{\gamma}_{(2)}\}$ не е регулярен.

Задача 3.33. 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

$$L = \left\{ \begin{bmatrix}\alpha\\\beta\\\gamma\end{bmatrix} \in \Sigma_3^\star \mid \overline{\alpha}_{(2)} + \overline{\beta}_{(2)} = \overline{\gamma}_{(2)} \right\}$$

е автоматен език.

Упътване. Доста по-удобно е да построим автомат $\mathcal{A}$, такъв че $\mathcal{L}(\mathcal{A}) = L^{\texttt{rev}}$. Да започнем с състоянието $q_=$, за което искаме да имаме свойството, че за произволно състояние q,

$$\delta^\star(q, \begin{bmatrix}\alpha\\\beta\\\gamma\end{bmatrix}) = q_= \Leftrightarrow \overline{\alpha^{\texttt{rev}}}_{(2)} + \overline{\beta^{\texttt{rev}}}_{(2)} = \overline{\gamma^{\texttt{rev}}}_{(2)}.$$

Понеже за $\overline{\varepsilon}_{(2)} + \overline{\varepsilon}_{(2)} = \overline{\varepsilon}_{(2)}$, състоянието $q_=$ ще бъде начално и финално за $\mathcal{A}$.

Нека $\overline{\alpha}_{(2)} + \overline{\beta}_{(2)} = \overline{\gamma}_{(2)}$. Тогава:

$$\delta(q_=, \begin{bmatrix}0\\0\\0\end{bmatrix}) \stackrel{\text{деф}}{=} q_= \qquad // \ \overline{0\alpha}_{(2)} + \overline{0\beta}_{(2)} = \overline{0\gamma}_{(2)}$$

$$\delta(q_=, \begin{bmatrix}0\\1\\1\end{bmatrix}) \stackrel{\text{деф}}{=} q_= \qquad // \ \overline{0\alpha}_{(2)} + \overline{1\beta}_{(2)} = \overline{1\gamma}_{(2)}$$

$$\delta(q_=, \begin{bmatrix}1\\0\\1\end{bmatrix}) = q_= \qquad // \ \overline{1\alpha}_{(2)} + \overline{0\beta}_{(2)} = \overline{1\gamma}_{(2)}$$

Остана случая $\overline{1\alpha}_{(2)} + \overline{1\beta}_{(2)} = \overline{10\gamma}_{(2)}$. Този случай е по-специален и трябва да бъде разгледан отделно. Трябва да отидем в състояние q_1, в което ще помним, че третия ред трябва да започва с 1-ца. Затова имаме следния преход:

$$\delta(q_=, \begin{bmatrix}1\\1\\0\end{bmatrix}) \stackrel{\text{деф}}{=} q_1.$$

За останалите $\gamma \in \Sigma_3$ имаме, че

$$\delta(q_=, \gamma) \stackrel{\text{деф}}{=} q_{\texttt{err}},$$

където $q_{\texttt{err}}$ е състоянието, от което не можем да излезем.

Така трябва да дефинираме функцията на преходите, че за състоянието q_1 трябва да е изпълнено, че за произволно q,

$$\delta^\star(q, \begin{bmatrix}\alpha\\\beta\\\gamma\end{bmatrix}) = q_1 \Leftrightarrow \overline{\alpha^{\texttt{rev}}}_{(2)} + \overline{\beta^{\texttt{rev}}}_{(2)} = \overline{1\gamma^{\texttt{rev}}}_{(2)}.$$

Да разгледаме сега случая $\overline{\alpha}_{(2)} + \overline{\beta}_{(2)} = \overline{1\gamma}_{(2)}$. Тогава:

$$\delta(q_1, \begin{bmatrix}0\\0\\1\end{bmatrix}) \stackrel{\text{деф}}{=} q_= \qquad // \ \overline{0\alpha}_{(2)} + \overline{0\beta}_{(2)} = \overline{1\gamma}_{(2)}$$

$$\delta(q_1, \begin{bmatrix}1\\1\\1\end{bmatrix}) \stackrel{\text{деф}}{=} q_1 \qquad // \ \overline{1\alpha}_{(2)} + \overline{1\beta}_{(2)} = \overline{11\gamma}_{(2)}$$

$$\delta(q_1, \begin{bmatrix}1\\0\\0\end{bmatrix}) \stackrel{\text{деф}}{=} q_1 \qquad // \ \overline{1\alpha}_{(2)} + \overline{0\beta}_{(2)} = \overline{10\gamma}_{(2)}$$

$$\delta(q_1, \begin{bmatrix}0\\1\\0\end{bmatrix}) \stackrel{\text{деф}}{=} q_1 \qquad // \ \overline{0\alpha}_{(2)} + \overline{1\beta}_{(2)} = \overline{10\gamma}_{(2)}$$

$$\delta(q_1, \gamma) \stackrel{\text{деф}}{=} q_{\texttt{err}} \qquad // \text{ за останалите } \gamma \in \Sigma_3$$

□

Задача 3.34. Да разгледаме азбуката:

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

Една дума над азбуката Σ_2 ни дава два реда от 0-ли и 1-ци, които ще разглеждаме като числа в двоична бройна система. Да разгледаме езиците:

- $L_1 = \left\{ \begin{bmatrix}\alpha\\\beta\end{bmatrix} \in \Sigma_2^\star \mid \overline{\alpha}_{(2)} < \overline{\beta}_{(2)} \right\}$;
- $L_2 = \left\{ \begin{bmatrix}\alpha\\\beta\end{bmatrix} \in \Sigma_2^\star \mid 3(\overline{\alpha}_{(2)}) = \overline{\beta}_{(2)} \right\}$;
- $L_3 = \left\{ \begin{bmatrix}\alpha\\\beta\end{bmatrix} \in \Sigma_2^\star \mid \alpha = \beta^{\mathtt{rev}} \right\}$;

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

Упътване. Ще построим автомат $\mathcal{A} = \langle \Sigma, Q, q_{\mathtt{start}}, \delta, F \rangle$ за езика $L_1^{\mathtt{rev}}$. За улеснение, в рамките на тази задача ще пишем:

- $\alpha \equiv \beta$, ако $\overline{\alpha^{\mathtt{rev}}}_{(2)} = \overline{\beta^{\mathtt{rev}}}_{(2)}$,
- $\alpha \prec \beta$, ако $\overline{\alpha^{\mathtt{rev}}}_{(2)} < \overline{\beta^{\mathtt{rev}}}_{(2)}$,
- $\alpha \succ \beta$, ако $\overline{\alpha^{\mathtt{rev}}}_{(2)} > \overline{\beta^{\mathtt{rev}}}_{(2)}$.

Нека състоянията на автомата са $Q = \{q_=, q_<, q_>\}$. Искаме да е изпълнено свойствата:

- $\delta^\star(q_=, \begin{bmatrix}\alpha\\\beta\end{bmatrix}) = q_=$ точно тогава, когато $\alpha \equiv \beta$;
- $\delta^\star(q_=, \begin{bmatrix}\alpha\\\beta\end{bmatrix}) = q_<$ точно тогава, когато $\alpha \prec \beta$;
- $\delta^\star(q_=, \begin{bmatrix}\alpha\\\beta\end{bmatrix}) = q_>$ точно тогава, когато $\alpha \succ \beta$.

Множеството от финални състояния ще бъде $F = \{q_<\}$, а началното състояние $q_{\mathtt{start}} = q_=$. За да дефинираме функцията на преходите, трябва да разгледа няколко случая, в зависимост от това какво е отношението между α и β.

- Нека $\alpha \equiv \beta$. Тогава:
  - $\alpha 0 \equiv \beta 0$ и $\alpha 1 \equiv \beta 1$. Следователно,
    $$\delta(q_=, \begin{bmatrix}0\\0\end{bmatrix}) = \delta(q_=, \begin{bmatrix}1\\1\end{bmatrix}) = q_=.$$
  - $\alpha 0 \prec \beta 1$. Следователно,
    $$\delta(q_=, \begin{bmatrix}0\\1\end{bmatrix}) = q_>.$$
  - $\alpha 1 \succ \beta 0$. Следователно,
    $$\delta(q_=, \begin{bmatrix}1\\0\end{bmatrix}) = q_<.$$
- Нека $\alpha \prec \beta$. Тогава:
  - $\alpha 0 \prec \beta 0$, $\alpha 1 \prec \beta 1$, $\alpha 0 \prec \beta 1$. Следователно,
    $$\delta(q_<, \begin{bmatrix}0\\0\end{bmatrix}) = \delta(q_<, \begin{bmatrix}1\\1\end{bmatrix}) = \delta(q_<, \begin{bmatrix}0\\1\end{bmatrix}) = q_<.$$
  - $\alpha 1 \succ \beta 0$. Следователно,
    $$\delta(q_<, \begin{bmatrix}1\\0\end{bmatrix}) = q_>.$$
- Нека $\alpha \succ \beta$. Тогава:
  - $\alpha 0 \succ \beta 0$, $\alpha 1 \succ \beta 1$, $\alpha 1 \succ \beta 0$. Следователно,
    $$\delta(q_>, \begin{bmatrix}0\\0\end{bmatrix}) = \delta(q_>, \begin{bmatrix}1\\1\end{bmatrix}) = \delta(q_>, \begin{bmatrix}1\\0\end{bmatrix}) = q_>.$$
  - $\alpha 0 \prec \beta 1$. Следователно,
    $$\delta(q_>, \begin{bmatrix}0\\1\end{bmatrix}) = q_<.$$

Докажете, че за така дефинирания автомат $\mathcal{A}$ е изпълнено, че $\mathcal{L}(\mathcal{A}) = L_1^{\text{rev}}$. □

Нека $\mathcal{A} = \langle \Sigma, Q, q_{\texttt{start}}, \delta, F \rangle$. За някои по-сложни задачи е удобно да въведем означението

$$\mathcal{L}_{\mathcal{A}}(q, P) \stackrel{\text{деф}}{=} \{\omega \in \Sigma^\star \mid \delta^\star_{\mathcal{A}}(q, \omega) \in P\},$$

където $q \in Q$ и $P \subseteq Q$. В частност, $\mathcal{L}(\mathcal{A}) = \mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, F)$. Лесно се съобразява, че:

$$\mathcal{L}(\mathcal{A}) = \bigcup_{p \in Q} \mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}) \cdot \mathcal{L}_{\mathcal{A}}(p, F).$$

Аналогично, ако $\mathcal{A} = \langle \Sigma, Q, Q_{\texttt{start}}, \Delta, F \rangle$ е НКА, то да положим

$$\mathcal{L}_{\mathcal{A}}(q, P) \stackrel{\text{деф}}{=} \{\omega \in \Sigma^\star \mid \Delta^\star_{\mathcal{A}}(\{q\}, \omega) \cap P \neq \emptyset\}.$$

Циклични разместванията

Да разгледаме следната операция върху езици

$$\texttt{Cycle}(L) \stackrel{\text{деф}}{=} \{\omega_2 \cdot \omega_1 \mid \omega_1 \cdot \omega_2 \in L\}.$$

Задача 3.35. Докажете, че ако L е регулярен, то $\texttt{Cycle}(L)$ е регулярен.

В [20, стр. 60] е дадено конструктивно решение на тази задача, т.е. в явен вид се строи автомат. Нашият подход е алгебричен.

Упътване. Нека $L = \mathcal{L}(\mathcal{A})$. Достатъчно е да проследим веригата от еквивалентности:

$$\begin{aligned}
\omega \in \texttt{Cycle}(L) &\Leftrightarrow (\exists i)[\, \omega\texttt{[}i :\texttt{]} \cdot \omega\texttt{[:}\, i\texttt{]} \in L \,] \\
&\Leftrightarrow (\exists i)(\exists p \in Q)[\, \omega\texttt{[}i :\texttt{]} \in \mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}) \,\&\, \omega\texttt{[:}\, i\texttt{]} \in \mathcal{L}_{\mathcal{A}}(p, F) \,] \\
&\Leftrightarrow (\exists p \in Q)[\, \omega\texttt{[:}\, i\texttt{]} \cdot \omega\texttt{[}i :\texttt{]} \in \mathcal{L}_{\mathcal{A}}(p, F) \cdot \mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}) \,] \\
&\Leftrightarrow (\exists p \in Q)[\, \omega \in \mathcal{L}_{\mathcal{A}}(p, F) \cdot \mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}) \,] \\
&\Leftrightarrow \omega \in \bigcup_{p \in Q} \mathcal{L}_{\mathcal{A}}(p, F) \cdot \mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}).
\end{aligned}$$

□

Разполовяване и удвояване на думи

Нека да видим сега следните две операции върху езици:

$$\texttt{Half}(L) \stackrel{\text{деф}}{=} \{\omega \mid \omega \cdot \omega \in L\},$$
$$\texttt{Double}(L) \stackrel{\text{деф}}{=} \{\omega \cdot \omega \mid \omega \in L\}.$$

На пръв поглед може би изглежда, че тези две операции имат еднакви свойства относно регулярните езици. Оказва се, че това не е така.

Задача 3.36. Докажете, че ако L е регулярен, то $\texttt{Half}(L)$ е регулярен.

Упътване. Нека $L = \mathcal{L}(\mathcal{A})$. Достатъчно е да проследим веригата от еквивалентности:

$$\begin{aligned}
\omega \in \texttt{Half}(L) &\Leftrightarrow \omega \cdot \omega \in L \\
&\Leftrightarrow (\exists p \in Q)[\delta^\star_{\mathcal{A}}(q_{\texttt{start}}, \omega) = p \,\&\, \delta^\star_{\mathcal{A}}(p, \omega) \in F] \\
&\Leftrightarrow (\exists p \in Q)[\omega \in \mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}) \,\&\, \omega \in \mathcal{L}_{\mathcal{A}}(p, F)] \\
&\Leftrightarrow \omega \in \bigcup_{p \in Q} (\mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}) \cap \mathcal{L}_{\mathcal{A}}(p, F)).
\end{aligned}$$

Ясно е, че:

$$\texttt{Half}(L) = \bigcup_{p \in Q} (\mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}) \cap \mathcal{L}_{\mathcal{A}}(p, F)).$$

Така изразихме $\texttt{Half}(L)$ като крайна булева комбинация на регулярни езици. Следователно, $\texttt{Half}(L)$ е регулярен език. □

Задача 3.37. Съществува регулярен език L, за който $\texttt{Double}(L)$ не е регулярен.

Упътване. Достатъчно е да разгледаме $L = \mathcal{L}[\![\boldsymbol{a}^\star \boldsymbol{b}]\!]$. Тогава

$$\texttt{Double}(L) = \{a^n b a^n b \mid n \in \mathbb{N}\},$$

за който лесно се съобразява, че не е регулярен. □

Триене на половината дума

Да разгледаме следната операция върху езици:

$$\frac{1}{2}(L) \stackrel{\text{деф}}{=} \{\alpha \in \Sigma^\star \mid (\exists \beta)[\, |\alpha| = |\beta| \,\&\, \alpha\beta \in L\,]\}.$$

Задача 3.38. Докажете, че ако L е регулярен, то $\frac{1}{2}(L)$ е регулярен.

✎ Опитайте се да построите директно автомат за $\frac{1}{2}(L)$. Това е подходът в [20, стр. 58].

Упътване. Един подход към решаването на тази задача е да изразим езика $\frac{1}{2}(L)$ чрез крайно много регулярни операции приложени върху регулярни езици. Така ще видим, че $\frac{1}{2}(L)$ е регулярен без да строим автомат за него в явен вид. Нека $L = \mathcal{L}(\mathcal{A})$. Да вземем НКА $\mathcal{B}$, който е същия като $\mathcal{A}$, но всеки преход в $\mathcal{A}$ се замества с преход с произволна буква.

- $Q_{\mathcal{B}} = Q_{\mathcal{A}}$;
- $Q^{\mathcal{B}}_{\texttt{start}} = \{q^{\mathcal{A}}_{\texttt{start}}\}$;
- $F_{\mathcal{B}} = F_{\mathcal{A}}$;
- $\Delta_{\mathcal{B}}(q, x) = \{\delta_{\mathcal{A}}(q, y) \mid y \in \Sigma\}$.

Тогава $\beta \in \mathcal{L}(\mathcal{B}) \;\Leftrightarrow\; (\exists \alpha)[|\alpha| = |\beta| \,\&\, \alpha \in \mathcal{L}(\mathcal{A})]$.

$$\begin{aligned} \alpha \in \frac{1}{2}(L) &\;\Leftrightarrow\; (\exists p \in Q)[\alpha \in \mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}) \,\&\, \alpha \in \mathcal{L}_{\mathcal{B}}(\{p\}, F)] \\ &\;\Leftrightarrow\; \alpha \in \bigcup_{p \in Q} (\mathcal{L}_{\mathcal{A}}(q_{\texttt{start}}, \{p\}) \cap \mathcal{L}_{\mathcal{B}}(p, F)). \end{aligned}$$

□

Разбиване на дума на три части

Да разгледаме следните операции върху езици.

$$\texttt{Triples}_2(L) \stackrel{\text{деф}}{=} \{\omega_2 \mid (\exists \omega_1)(\exists \omega_3)[\, \omega_1\omega_2\omega_3 \in L \,\&\, |\omega_1| = |\omega_2| = |\omega_3|\,]\,\};$$

$$\texttt{Triples}_{1,3}(L) \stackrel{\text{деф}}{=} \{\omega_1\omega_3 \mid (\exists \omega_2)[\, \omega_1\omega_2\omega_3 \in L \,\&\, |\omega_1| = |\omega_2| = |\omega_3|\,]\,\}.$$

Отново се оказва, че тези две операции са принципно различни относно регулярните езици.

Задача 3.39. Докажете, че ако L е регулярен език, то $\texttt{Triples}_2(L)$ е регулярен.

Интересно е, че за регулярен език L, $\texttt{Triples}_{1,3}(L)$ е безконтекстен. Вижте *Задача 4.44*.

Задача 3.40. Докажете, че съществува регулярен език L, за който $\texttt{Triples}_{1,3}(L)$ *не* е регулярен.

Упътване. Разгледайте езика $L = \mathcal{L}[\![\boldsymbol{a}^+\sharp\boldsymbol{a}^+\sharp\boldsymbol{a}^+]\!]$. Ако допуснем, че езикът $\texttt{Triples}_{1,3}(L)$ е регулярен, то

$$\texttt{Triples}_{1,3}(L) \cap \mathcal{L}[\![\boldsymbol{a}^+\sharp\sharp\boldsymbol{a}^+]\!] = \{a^n\sharp\sharp a^n \mid n \geq 1\}$$

също би трябвало да е регулярен, но лесно се съобразява, че $\{a^n\sharp\sharp a^n \mid n \geq 1\}$ не е регулярен. □

Триене на квадрат от дължината

Да разгледаме следната операция върху езици:

$$\sqrt{L} \stackrel{\text{деф}}{=} \{\alpha \mid (\exists\beta)[\, |\beta| = |\alpha|^2 \;\&\; \alpha\cdot\beta \in L\,]\}.$$

Задача 3.41. Докажете, че ако L е регулярен, то $\sqrt{L}$ е регулярен.

Упътване.

□

Триене на степен от дължината

Да разгледаме следната операция върху езици:

$$\log(L) \stackrel{\text{деф}}{=} \{\alpha \mid (\exists\beta)[\, |\beta| = 2^{|\alpha|} \;\&\; \alpha\cdot\beta \in L\,]\}.$$

Задача 3.42. Докажете, че ако L е регулярен, то $\log(L)$ е регулярен.

Упътване. □

Смесване на думи

Да дефинираме рекурсивно функцията $\texttt{merge} : \Sigma^\star \times \Sigma^\star \to \mathcal{P}(\Sigma^\star)$, която смесва две думи по всички възможни начини така:

$$\texttt{merge}(\alpha, \varepsilon) \stackrel{\text{деф}}{=} \{\alpha\};$$

$$\texttt{merge}(\varepsilon, \beta) \stackrel{\text{деф}}{=} \{\beta\};$$

$$\texttt{merge}(a\cdot\alpha, b\cdot\beta) \stackrel{\text{деф}}{=} \{a\}\cdot\texttt{merge}(\alpha, b\cdot\beta) \cup \{b\}\cdot\texttt{merge}(a\cdot\alpha, \beta).$$

Например,

$$\texttt{merge}(ab, cd) = \{abcd, acbd, acdb, cabd, cadb, cdab\}.$$

Сега можем да дефинираме смесването на два езика

$$\texttt{Merge}(L_1, L_2) \stackrel{\text{деф}}{=} \bigcup_{\alpha\in L_1, \beta\in L_2} \texttt{merge}(\alpha, \beta).$$

Задача 3.43. Ако L_1 и L_2 са регулярни езици над азбуката Σ, то $\texttt{Merge}(L_1, L_2)$ е регулярен език.

Да дефинираме следната операция върху езици:

$$\texttt{Repeat}(L) \stackrel{\text{деф}}{=} \{\omega\sharp a^n \mid \omega^n \in L\}.$$

Задача 3.44. Докажете, че ако L е регулярен, то $\texttt{Repeat}(L)$ е регулярен.

Упътване.

□

Глава 4

Безконтекстни езици и стекови автомати

Видяхме, че има много прости езици, например $\{a^n b^n \mid n \in \mathbb{N}\}$, които не са регулярни. В тази глава ще разгледаме един по-широк клас от езици, така наречените безконтекстни езици, които се характеризират като решения на системи от малко по-общ тип в сравнение със системите за регулярните езици.

4.1 Дървета

Понятието дърво е едно от най-основните в информатиката и може да се дефинира по много различни начини, в зависимост от това за какви цели се използва. Тук на практика следваме [15], защото по-късно ще е важно да имаме наредба между възлите на дървото. При нас всички дървета са крайно разклонени и всеки възел в дървото еднозначно се определя от пътя от възела до корена. Обърнете внимание, че тук дърветата, макар и крайно разклонени, могат да бъдат безкрайни.

Нека фиксираме $b \in \mathbb{N}$. Да разгледаме азбуката $\Sigma = \{0, 1, \ldots, b-1\}$. Непразното множество $T \subseteq \Sigma^\star$ се нарича **дърво**, ако T изпълнява свойствата:

(1) $\texttt{Pref}(T) = T$, или с други думи,

$$(\forall \alpha \in \Sigma^\star)(\forall \beta \in \Sigma^\star)[\, \alpha \in T \;\&\; \beta \preceq \alpha \implies \beta \in T \,].$$

Това свойство казва, че T е затворено относно префикси.

(2) За всяка дума $\alpha \in \Sigma^\star$ и $x \in \Sigma$,

$$\alpha \cdot x \in T \;\&\; y < x \implies \alpha \cdot y \in T.$$

Едно от удобствата да се работи с тази дефиниция на дърво е, че ако имаме един връх от дървото, т.е. дума $\alpha \in T$, то тя пази информация за целия път от корена до този връх. Например, ако $010 \in T$, то от корена първо сме отишли по левия клон, после по десния, и най-накрая пак по левия. Фактът, че според тази дефиниция имаме подредба на върховете на дървото ще ни бъде полезен по-нататък. Например, 1011 е по-наляво от 110, защото лексикографски 1011 по-малка от 110.

Ето няколко примера:

- $T = \{\varepsilon, 0, 1, 00, 01, 11\}$ не е дърво, защото (2) не е удовлетворено.
- $T = \{\varepsilon, 0, 1, 00, 01, 10\}$ е дърво.
- $T = \{0\}^\star$ е безкрайно дърво, докато $T = \{1\}^\star$ не е.
- $T = \{0\}^\star \cdot \{\varepsilon, 1\}$ е безкрайно дърво.
- $T = \{0, 1\}^\star$ е пълното двоично дърво.

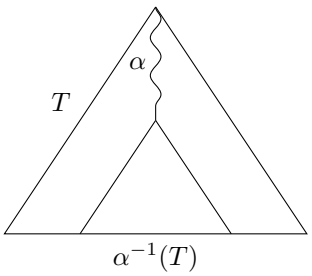
Нека да въведем следните означения:

$$\texttt{height}(T) \stackrel{\text{деф}}{=} \max\{\, |\alpha| \mid \alpha \in T \,\} \qquad \text{// височина}$$

$$\texttt{succ}_T(\alpha) \stackrel{\text{деф}}{=} \{\alpha c \mid \alpha c \in T \;\&\; c < b\} \qquad \text{// наследниците на } \alpha$$

$$T_\alpha \stackrel{\text{деф}}{=} \alpha^{-1}(T) \qquad \text{// поддървото на } \alpha$$

$$\texttt{leaves}(T) \stackrel{\text{деф}}{=} \{\alpha \in T \mid \texttt{succ}_T(\alpha) = \emptyset\}. \qquad \text{// листата на } T$$



Фигура 4.1: $\alpha^{-1}(T)$ е поддърво на T.

Задача 4.1. Докажете, че ако T е дърво, то $\texttt{Pref}(\texttt{leaves}(T)) = T$.

Задача 4.2. Нека $T \subseteq \{0, \ldots, b-1\}^\star$ е крайно дърво. Докажете, че

$$|\texttt{leaves}(T)| \leq b^{\texttt{height}(T)}.$$

Доказателство. Индукция по $\texttt{height}(T)$.

- Нека $\texttt{height}(T) = 0$. Тогава е ясно, че $|\texttt{leaves}(T)| = |\{\varepsilon\}| = 1 \leq b^0$.
- Нека $\texttt{height}(T) > 0$. За всяко $a \in T$ е ясно, че $\texttt{height}(T_a) < \texttt{height}(T)$. Тогава:

В този случай имаме, че $\texttt{leaves}(T) = \bigcup_{a \in T}(\{a\} \cdot \texttt{leaves}(T_a))$. Тук гледаме на a и b от една страна като букви в азбука, но и като числа.

$$\begin{aligned}
|\texttt{leaves}(T)| &= \sum_{a \in T} |\texttt{leaves}(T_a)| && \text{// } T_a \stackrel{\text{деф}}{=} a^{-1}(T) \\
&\leq \sum_{a \in T} b^{\texttt{height}(T_a)} && \text{// от } \textbf{(И.П.)} \\
&\leq \sum_{a \in T} b^{\texttt{height}(T)-1} && \text{// } \texttt{height}(T_a) \leq \texttt{height}(T) - 1 \\
&\leq \sum_{a < b} b^{\texttt{height}(T)-1} \\
&= b^{\texttt{height}(T)}.
\end{aligned}$$

□

$f \restriction n \stackrel{\text{деф}}{=} f(0)f(1)\cdots f(n-1)$.

Твърдение 4.1 (Лема на Кьониг)**.** Нека $T \subseteq \{0, 1, \ldots, k\}^\star$. Ако T е безкрайно дърво, то T съдържа безкраен път, т.е. съществува тотална функция $f : \mathbb{N} \to \{0, 1, \ldots, k\}$, такава че $f \restriction n \in T$ за всяко $n \in \mathbb{N}$.

Упътване. Дефинираме безкрайния път f на стъпки. На всяка стъпка избираме този наследник, който е корен на безкрайно дърво. Понеже T е безкрайно дърво с крайно разклонение, на всяка стъпка можем да изберем такъв наследник. □

Доказателство. По-формално, ще дефинираме редица от думи $\alpha_0 \prec \alpha_1 \prec \cdots \prec \alpha_n \prec \cdots$, които ще образуват безкрайния път, т.е. $f \restriction n = \alpha_n$, като искаме, за всяко n, $T_n \stackrel{\text{деф}}{=} (\alpha_n)^{-1}(T)$ да бъде безкрайно дърво. Нека $\alpha_0 = \varepsilon$, коренът на T. Ясно е, че по условие, $T_0 = T$ е безкрайно дърво. Да приемем, че сме дефинирали $\alpha_0 \prec \alpha_1 \prec \cdots \prec \alpha_n$ и T_n е безкрайно дърво. Ще дефинираме α_{n+1}. Понеже T_n е крайно разклонено дърво с максимално разклонение $k + 1$, възелът α_n в дървото има крайно много наследници. Понеже T_n е безкрайно дърво, то поне един от наследниците, да кажем $y \leq k$, на α_n ще има безкрайно много наследници. Нека $\alpha_{n+1} \stackrel{\text{деф}}{=} \alpha_n \cdot y$. Ясно е, че T_{n+1} е безкрайно дърво. □

4.2 Синтактични дървета

A parse tree is a graphical representation of a derivation that filters out the order in which productions are applied to replace nonterminals. [1]

Безконтекстна граматика е наредена четворка от вида

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

където:

- V е крайно множество от *променливи* (нетерминали);
- Σ е крайно множество от *букви* (терминали), като $\Sigma \cap V = \emptyset$;
- $R \subseteq V \times (V \cup \Sigma)^\star$ е крайно множество от *правила*. За по-добра яснота, обикновено правилата $(A, \beta) \in R$ ще означаваме като $A \to_G \beta$. Когато е ясно за коя граматика говорим, ще пишем просто $A \to \beta$.

В [10] правилата се наричат *productions* или *production rules*.

- $S \in V$ е началната променлива (нетерминал).

На английски се нарича parse tree, syntax tree, derivation tree. Обикновено, например в [16, стр. 123], дават рекурсивна дефиниция на синтактично дърво. Тук използваме дефиницията на понятието дърво разгледана в Раздел 4.1. Функцията λ облича голото дърво T с елементи на граматиката G.

- Нека фиксираме граматиката $G = (\Sigma, V, S, R)$ и

$$b \stackrel{\text{деф}}{=} \max\{|\gamma| \mid A \to_G \gamma \text{ е правило в } G\}.$$

- Двойката $P = (T, \lambda)$ се нарича **дърво на извод** съвместимо с G, ако са изпълнени свойствата:
  - T е непразно крайно дърво и $T \subseteq \{0, 1, \ldots, b-1\}^\star$.
  - Фукцията $\lambda : T \to V \cup \Sigma \cup \{\varepsilon\}$ асоциира етикет с всеки елемент на дървото. Нека означим тези етикети с $X_\alpha \stackrel{\text{деф}}{=} \lambda(\alpha)$.
  - Коренът на дървото винаги има етикет елемент на V или Σ, т.е. $\lambda(\varepsilon) \in V \cup \Sigma$.
  - Ако $\alpha \in T$ и $|\mathtt{succ}_T(\alpha)| = k + 1$, за някое $k \in \mathbb{N}$, то има правило в граматиката

    $$X_\alpha \to_G X_{\alpha\cdot 0} X_{\alpha\cdot 1} \cdots X_{\alpha\cdot k},$$

    където $X_{\alpha\cdot i} \in V \cup \Sigma$.

    С други думи, $\lambda(\alpha) \to_G \lambda(\alpha\cdot 0)\cdots\lambda(\alpha\cdot k)$.

    Това свойство е ключово, защото то ни казва, че дървото е съвместимо с правилата на граматиката G.
  - Имаме частен случай за $\alpha \in T$ и $|\mathtt{succ}_T(\alpha)| = 1$. Тогава позволяваме да имаме $X_{\alpha\cdot 0} = \varepsilon$, ако имаме правилото в граматиката

    $$X_\alpha \to_G \varepsilon.$$

- За дървото на извод P, нека $\mathtt{root}(P) \stackrel{\text{деф}}{=} X_\varepsilon$.
- Нека $\alpha_0, \alpha_1, \ldots, \alpha_k$ са всички думи от множеството $\mathtt{leaves}(T)$ подредени във възходящ ред относно лексикографската наредба. Тогава

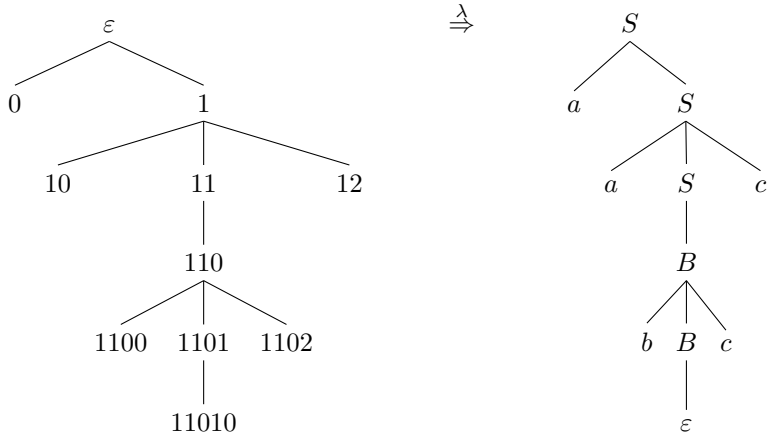
$$\mathtt{yield}(P) \stackrel{\text{деф}}{=} \lambda(\alpha_0)\cdot\lambda(\alpha_1)\cdots\lambda(\alpha_k).$$

Пример 4.1. Да разгледаме граматиката G зададена със следните правила:

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

По-нататък ще докажем, че езикът на граматиката G е $\{a^n b^k c^\ell \mid n + k \geq \ell\}$.

Да разгледаме дървото на извод $P = (T, \lambda)$, където:



Фигура 4.2: Дърво на извод за думата $aabcc$.

Имаме, че:

- $\texttt{height}(P) = 5$;
- $\texttt{leaves}(P) = \{0, 10, 1100, 11010, 1102, 12\}$;
- $\texttt{yield}(P) = aab\varepsilon cc = aabcc$.
- $\texttt{succ}_T(110) = \{1100, 1101, 1102\}$;
- $\lambda(\varepsilon) = S$;
- $\lambda(0) = a$, $\lambda(1) = S$;
- Ясно е, че $\underbrace{\lambda(\varepsilon)}_{S} \to_G \underbrace{\lambda(0)\lambda(1)}_{aS}$;
- $\lambda(10) = a$, $\lambda(11) = S$, $\lambda(12) = c$;
- Ясно е, че $\underbrace{\lambda(1)}_{S} \to_G \underbrace{\lambda(10)\lambda(11)\lambda(12)}_{aSc}$;
- $\lambda(110) = B$;
- Ясно е, че $\underbrace{\lambda(11)}_{S} \to_G \underbrace{\lambda(110)}_{B}$;
- $\lambda(1100) = b$, $\lambda(1101) = B$, $\lambda(1102) = c$;
- Ясно е, че

$$\underbrace{\lambda(110)}_{B} \to_G \underbrace{\lambda(1100)\lambda(1101)\lambda(1102)}_{bBc};$$

- $\lambda(11010) = \varepsilon$;
- Ясно е, че $\underbrace{\lambda(1101)}_{B} \to_G \underbrace{\lambda(11010)}_{\varepsilon}$;

От всичко по-горе следва, че $P = (T, \lambda)$ е дърво на извод за думата $aabcc$ в граматиката G.

4.3 Еднозначни граматики

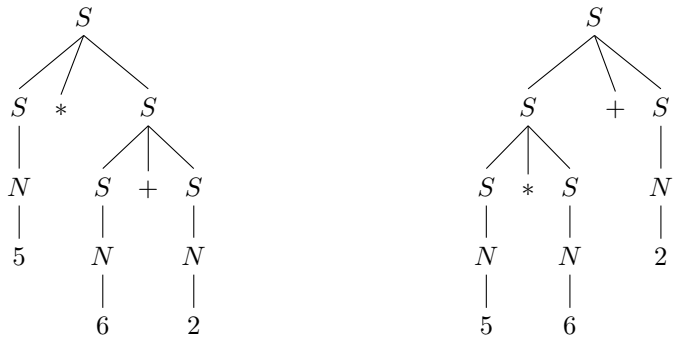
Практически, дърветата на извод са важни, когато искаме да зададем *смисъл* на една дума. Това е видно от *Пример 4.2*. Ако подходим наивно към задачата за построяване на граматика за аритметичните изрази, то смисълът на аритметичните изрази може да бъде неясен.

Пример 4.2. Да разгледаме граматика G_0 с правила

$$S \to S + S \mid S * S \mid (S) \mid N$$
$$N \to 0 \mid 1 \mid 2 \mid \ldots \mid 9$$

Подобен пример е разгледан и в [1]. Такъв вид граматики се наричат *неоднозначни* (на англ. *ambiguous*). Ако всяка дума от езика има единствено синтактично дърво, то тази граматика се нарича *еднозначна* (на англ. *unambiguous*). От практическа гледна точка е важно свойство дали една граматика е еднозначна или не.

В тази граматика нямаме приоритет между $+$ или $*$. Например, думата $5*6+2$ има две различни синтактични дървета P_1 и P_2, като и двете имат височина 4 и $\texttt{yield}(P_1) = \texttt{yield}(P_2) = 5 * 6 + 2$.



Фигура 4.3: Две синтактични дървета в граматиката G_0 за думата $5 * 6 + 2$.

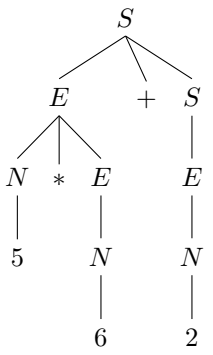
✍ Колко различни синтактични дървета за думата $1 + 5 * 6 + 2$ може да намерите?

Естествено е да искаме да модифицираме граматиката G_0, така че $*$ да има по-висок приоритет спрямо $+$. За целта ще разгледаме граматиката G_1, която ще поражда същия език като G_0, със следните правила:

$$S \to E + S \mid E$$
$$E \to N \mid (S) \mid N * E \mid (S) * E$$
$$N \to 0 \mid 1 \mid 2 \mid \ldots \mid 9$$

$*$ има по-висок приоритет от $+$, защото $*$ се среща по-близо до листата, или аналогично, $+$ се среща по-близо до корена.

Сега думата $5 * 6 + 2$ има само едно дърво на извод. Тук операцията $*$ е с по-висок приоритет в сравнение с операцията $+$.



Фигура 4.4: Единственото синтактично дърво в граматиката G_1 за думата $5 * 6 + 2$.

Пример 4.3. За безконтекстния език

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

да разгледаме две граматики G_1 и G_2.

- Правилата на G_1 са

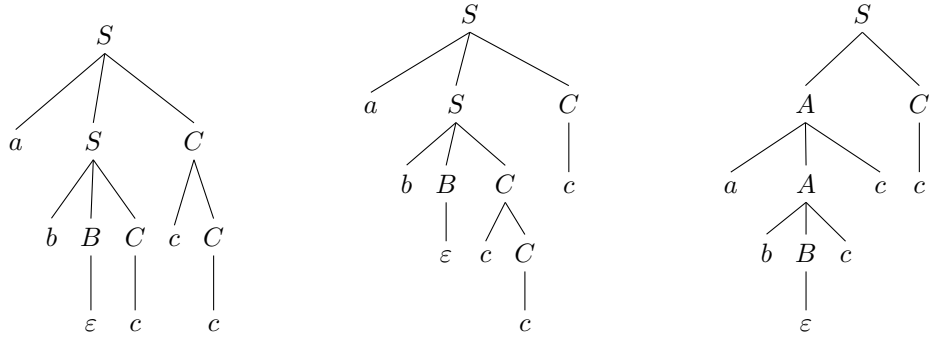
$$\begin{aligned} S &\to aSC \mid bBC \mid \varepsilon \\ B &\to bBC \mid \varepsilon \\ C &\to cC \mid c. \end{aligned}$$

- Правилата на G_2 са

$$\begin{aligned} S &\to AC \mid \varepsilon \\ A &\to aAc \mid bBc \mid \varepsilon \\ B &\to bBc \mid \varepsilon \\ C &\to cC \mid \varepsilon. \end{aligned}$$

Лесно се съобразява, че $L = \mathcal{L}(G_1) = \mathcal{L}(G_2)$.

Да разгледаме думата $abccc \in L$. Можем да посочим две синтактични дървета съвместими с G_1.



(а) Първо дърво за G_1 **(б)** Второ дърво за G_1 **(в)** Единственото дърво за G_2

Фигура 4.5: Синтактични дървета за думата $abccc$.

Съобразете, че всяка дума $\alpha \in L$ има единствено синтактично дърво съвместимо с G_2.

Ще завършим този раздел като формулираме един важен резултат, който все още нямаме сили да докажем.

Доказателството е в *Следствие* **??**.

Теорема 4.1. Не съществува алгоритъм, който да разпознава дали дадена безконтекстна граматика е еднозначна или не.

4.4 Извод върху синтактично дърво

Не знам да има учебник, в който формално да е въведена тази релация. Тя е удобна най-вече за решаване на задачи, както и за доказателството на лемата за покачването.

Определение 4.1. За произволна безконтекстна граматика G, дефинираме релацията $X \overset{\ell}{\lhd} \alpha$, където $X \in V \cup \Sigma$ и $\alpha \in (V \cup \Sigma)^\star$, по следния начин:

$$\frac{X \in V \cup \Sigma}{X \overset{0}{\lhd} X} \text{ правило (0)}$$

$$(\ell = \sup\{\ell_1, \ldots, \ell_n\}) \quad \frac{X \to_G X_1 \cdots X_n \qquad X_1 \overset{\ell_1}{\lhd} \gamma_1 \qquad \cdots \qquad X_n \overset{\ell_n}{\lhd} \gamma_n}{X \overset{\ell+1}{\lhd} \gamma_1 \cdots \gamma_n} \text{ правило (1)}$$

Според дефиницията $\sup(A) \overset{\text{деф}}{=} \min\{m \in \mathbb{N} \mid (\forall a \in A)[a \leq m]\}$ е ясно, че $\sup(\emptyset) = 0$.

Да напомним, че по дефиниция, ако $n = 0$, то $X_1 \cdots X_n = \varepsilon$. Освен това, понеже $\sup(A)$ означава най-малкото естествено число по-голямо от всички елементи на A, то $\sup(\emptyset) = 0$. Това означава, че ако в граматиката имаме правилото $A \to_G \varepsilon$, то според правило (1) получаваме, че $A \overset{1}{\lhd} \varepsilon$. Друг важен случай е когато $X \in \Sigma$. Тогава имаме, че $X \overset{0}{\lhd} X$, но *нямаме* $X \overset{\ell}{\lhd} X$ за всяко $\ell > 0$.

Определение 4.2. Нека G е произволна безконтекстна граматика. Тогава дефинираме езикът на G да бъде

$$\mathcal{L}(G) \overset{\text{деф}}{=} \{\alpha \in \Sigma^\star \mid S \overset{\star}{\lhd} \alpha\}.$$

Естествено е да разширим релацията $\overset{\ell}{\lhd}$ до бинарна релация над $(V \cup \Sigma)^\star$ следния начин:

$$(\ell = \sup\{\ell_1, \ldots, \ell_n\}) \quad \frac{X_1 \overset{\ell_1}{\lhd} \alpha_1 \qquad \cdots \qquad X_n \overset{\ell_n}{\lhd} \alpha_n}{X_1 \cdots X_n \overset{\ell}{\lhd} \alpha_1 \cdots \alpha_n}$$

Тук, отново, ако $n = 0$, то $\ell = 0$ и $X_1 \cdots X_n = \varepsilon$. Така получаваме извода $\varepsilon \overset{0}{\lhd} \varepsilon$.

Съобразете, че имаме:

$$\frac{X \to_G \gamma}{X \overset{1}{\lhd} \gamma.}$$

Да дефинираме релацията $\overset{\star}{\lhd}$ по следния начин:

$$\alpha \overset{\star}{\lhd} \beta \overset{\text{деф}}{\Leftrightarrow} (\exists \ell \in \mathbb{N})[\alpha \overset{\ell}{\lhd} \beta].$$

Задача 4.3. Докажете, че:

$$\frac{\alpha \overset{\ell_1}{\lhd} \beta \qquad \beta \overset{\ell_2}{\lhd} \gamma}{\alpha \overset{\leq \ell_1 + \ell_2}{\lhd} \gamma}$$

Релацията $\overset{\star}{\lhd}$ е удобна, когато не се интересуваме от реда, по който прилагаме правилата за извод в една безконтекстна граматика.

Следващото твърдение ни казва, че има пряка връзка между релацията $\overset{\star}{\lhd}$ и синтактичните дървета.

Твърдение 4.2. Нека G е безконтекстна граматика. Тогава $X \overset{\ell}{\lhd} \gamma$ точно тогава, когато съществува дърво на извод P съвместимо с G, за което $\texttt{root}(P) = X$, $\texttt{yield}(P) = \gamma$ и $\texttt{height}(P) = \ell$.

Упътване. Индукция по ℓ.

- Нека $\ell = 0$. Тогава твърдението е ясно.
- Нека $\ell > 0$. Първо, нека $X \overset{\ell}{\lhd} \gamma$. От правилата за извод върху синтактично дърво следва, че съществува правило $X \to_G X_0 \cdots X_{n-1}$ и $X_i \overset{\ell_i}{\lhd} \gamma_i$ за $i < n$, като $\ell = \sup\{\ell_0, \ldots, \ell_{n-1}\} + 1$ и $\gamma = \gamma_0 \cdots \gamma_{n-1}$. От **(И.П.)** , съществуват синтактични дървета P_i, за които $\texttt{root}(P_i) = X_i$, $\texttt{yield}(P_i) = \gamma_i$ и $\texttt{height}(P_i) = \ell_i$. Така можем да направим ново дърво $P = (T, \lambda)$, за което $T = \{\varepsilon\} \cup \bigcup_{i<n} \{i\} \cdot T_i$ и $\lambda(\varepsilon) = X$, $\lambda(i\alpha) = \lambda_i(\alpha)$.

  Второ, нека $P = (T, \lambda)$ е синтактично дърво, за което $\texttt{root}(P) = X$, $\texttt{yield}(P) = \gamma$ и $\texttt{height}(P) = \ell$. Нека $0, 1, \ldots, n-1$ са всички наследници на ε в T. Понеже дървото е съвместимо с G, то $\lambda(\varepsilon) \to_G \lambda(0)\lambda(1)\cdots\lambda(n-1)$. Да положим $T_i \overset{\text{деф}}{=} i^{-1}(T)$ и $\lambda_i(\alpha) \overset{\text{деф}}{=} \lambda(i\alpha)$. Ясно е, че $P_i = (T_i, \lambda_i)$ са синтактични дървета с височина $\ell_i < \ell$. От **(И.П.)** получаваме, че $X_i \overset{\ell_i}{\lhd} \gamma_i$ като $\gamma = \gamma_0 \cdot \gamma_1 \cdots \gamma_{n-1}$. Заключаваме, че $X \overset{\ell}{\lhd} \gamma$.

□

4.5 Апроксимации на безконтекстен език

Обикновено използваме a, b, c за елементи на Σ и A, B, C за елементи на V. Ще използваме X за елементи на $\Sigma \cup V$.

Определение 4.3. Да напомним, че когато говорихме за автомати, беше удобно да разсъждаваме за езика на всяка състояние. По подобен начин, сега ще дефинираме език $\mathcal{L}_G(X)$, за произволно $X \in \Sigma \cup V$ в граматиката G, по следния начин:

$$\mathcal{L}_G(X) \stackrel{\text{деф}}{=} \{\alpha \in \Sigma^\star \mid X \stackrel{\star}{\lhd} \alpha\}.$$

Ясно е, че $\mathcal{L}(G) = \mathcal{L}_G(S)$. Нека да дефинираме и *апроксимациите* $\mathcal{L}^\ell_G(X)$ на езика $\mathcal{L}_G(X)$ по следния начин:

$$\mathcal{L}^\ell_G(X) \stackrel{\text{деф}}{=} \{\alpha \in \Sigma^\star \mid X \stackrel{\leq \ell}{\lhd} \alpha\}.$$

След един бърз поглед върху дефиницията на релацията $\lhd$ веднага съобразяваме, че ако $X \in \Sigma$, то $\mathcal{L}^\ell_G(X) = \{X\}$ за всяко ℓ. Да видим сега какво става, когато $X \in V$.

Задача 4.4. Нека G е безконтекстна граматика и A е променлива в G. Докажете, че следните свойства са изпълнени:

- $\mathcal{L}^0_G(A) = \emptyset$;
- $\mathcal{L}^\ell_G(A)$ е краен език за всяко ℓ;
- $\mathcal{L}^\ell_G(A) \subseteq \mathcal{L}^{\ell+1}_G(A)$ за всяко ℓ;
- $\mathcal{L}_G(A) = \bigcup_{\ell \geq 0} \mathcal{L}^\ell_G(A)$.

Следната характеризация на езиците $\mathcal{L}^\ell_G(A)$ ще е удобна за нас, когато искаме да докажем, че една безконтекстна граматика разпознава даден език.

Нека да напомним, че

$\bigcup\{\{1,2\},\{3\}\} = \{1,2\} \cup \{3\} = \{1,2,3\}$.

Твърдение 4.3. Нека G е безконтекстна граматика и A е променлива в G. Тогава имаме следните зависимости:

$$\mathcal{L}^0_G(A) = \emptyset$$
$$\mathcal{L}^{\ell+1}_G(A) = \bigcup\{\mathcal{L}^\ell_G(X_1) \cdots \mathcal{L}^\ell_G(X_n) \mid A \to_G X_1 \cdots X_n \text{ е правило в } G\}.$$

Доказателство. Доказателството протича с индукция по ℓ. Твърдението очевидно е изпълнено за $\ell = 0$. За индукционната стъпка, първо ще докажем включването $\subseteq$. Нека сега да разгледаме произволна дума α, за която $\alpha \in \mathcal{L}^{\ell+1}_G(A)$, т.е. $A \stackrel{\leq \ell+1}{\lhd} \alpha$. Ясно е, че е невъзможно да имаме $A \stackrel{0}{\lhd} \alpha$. Тогава, според правилата на релацията $\lhd$, имаме следния извод:

$$\frac{A \to_G X_1 \cdots X_n \qquad X_1 \stackrel{\leq \ell}{\lhd} \alpha_1 \qquad \cdots \qquad X_n \stackrel{\leq \ell}{\lhd} \alpha_n}{A \stackrel{\leq \ell+1}{\lhd} \underbrace{\alpha_1 \cdots \alpha_n}_{\alpha}} \text{ правило (1)}$$

Понеже $X_i \stackrel{\leq \ell}{\lhd} \alpha_i$, то от **(И.П.)** имаме, че $\alpha_i \in \mathcal{L}^\ell_G(X_i)$ за всяко $i = 1, 2, \ldots, n$. Тогава

$$\alpha \in \bigcup\{\mathcal{L}^\ell_G(X_1) \cdots \mathcal{L}^\ell_G(X_n) \mid A \to_G X_1 \cdots X_n\}.$$

Така доказахме включването $\subseteq$. Сега преминаваме към доказателството на включването $\supseteq$. Нека вземем произволна дума α принадлежаща на дясното множество. Това означава, че можем да представим α като $\alpha = \alpha_1 \cdots \alpha_n$, където $A \to_G X_1 \cdots X_n$ и $\alpha_i \in \mathcal{L}^\ell_G(X_i)$. Получаваме следния извод:

$$\dfrac{A \to_G X_1 \cdots X_n \qquad \dfrac{\alpha_1 \in \mathcal{L}_G^\ell(X_1)}{X_1 \overset{\leq \ell}{\lhd} \alpha_1}\ \text{(деф.)} \qquad \cdots \qquad \dfrac{\alpha_n \in \mathcal{L}_G^\ell(X_n)}{X_n \overset{\leq \ell}{\lhd} \alpha_n}\ \text{(деф.)}}{A \overset{\leq \ell+1}{\lhd} \alpha_1 \cdots \alpha_n}\ \text{правило (1)}$$

Заключаваме, че думата α принадлежи на езика $\mathcal{L}^{\ell+1}(A)$. Така доказахме и включването $\supseteq$. $\square$

Понеже знаем, че $\mathcal{L}_G(A) = \bigcup_\ell \mathcal{L}_G^\ell(A)$, то можем да получим директно следствие от *Твърдение 4.3*.

Следствие 4.1. Нека G е безконтекстна граматика и A е променлива в G. Тогава

$$\mathcal{L}_G(A) = \bigcup \{\mathcal{L}_G(X_1) \cdots \mathcal{L}_G(X_n) \mid A \to_G X_1 \cdots X_n \text{ е правило в } G\}.$$

Доказателство. Нека $\alpha \in \mathcal{L}_G(A)$, т.е. $\alpha \in \mathcal{L}_G^{\ell+1}(A)$, за някое ℓ. Тогава, според *Твърдение 4.3*, $\alpha \in \mathcal{L}_G^\ell(X_1) \cdots \mathcal{L}_G^\ell(X_n)$, за някое правило $A \to_G X_1 \cdots X_n$. Оттук веднага получаваме, че $\alpha \in \mathcal{L}_G(X_1) \cdots \mathcal{L}_G(X_n)$ и следователно

$$\alpha \in \bigcup \{\mathcal{L}_G(X_1) \cdots \mathcal{L}_G(X_n) \mid A \to_G X_1 \cdots X_n \text{ е правило в } G\}.$$

За обратната посока, нека $\alpha \in \mathcal{L}_G(X_1) \cdots \mathcal{L}_G(X_n)$, където $A \to_G X_1 \cdots X_n$ е правило в G. Това означава, че $\alpha \in \mathcal{L}_G^\ell(X_1) \cdots \mathcal{L}_G^\ell(X_n)$, за някое ℓ, и следователно, според *Твърдение 4.3*, $\alpha \in \mathcal{L}_G^{\ell+1}(A)$. Заключаваме, че $\alpha \in \mathcal{L}_G(A)$. $\square$

Ще се окаже удобно да обобщим $\mathcal{L}_G^\ell(X)$ като дефинираме $\mathcal{L}_G^\ell(\alpha)$ за произволна дума $\alpha \in (V \cup \Sigma^\star)^\star$ по следния начин:

$$\mathcal{L}_G^\ell(\varepsilon) \overset{\text{деф}}{=} \varepsilon$$
$$\mathcal{L}_G^\ell(\beta \cdot X) \overset{\text{деф}}{=} \mathcal{L}_G^\ell(\beta) \cdot \mathcal{L}_G^\ell(X).$$

С други думи, ако $\alpha = X_1 \ldots X_n$, то

$$\mathcal{L}_G^\ell(X_1 \cdots X_n) = \mathcal{L}_G^\ell(X_1) \cdots \mathcal{L}_G^\ell(X_n).$$

Така можем да преформулираме *Твърдение 4.3* по следния начин.

Твърдение 4.4 (*Твърдение 4.3*)**.** Нека G е безконтекстна граматика и A е променлива в G. Тогава имаме следните зависимости:

$$\mathcal{L}_G^0(A) = \emptyset$$
$$\mathcal{L}_G^{\ell+1}(A) = \bigcup_{A \to \alpha} \mathcal{L}^\ell(\alpha).$$

Аналогично можем да обобщим $\mathcal{L}_G(X)$ като дефинираме $\mathcal{L}_G(\alpha)$ за произволна дума $\alpha \in (V \cup \Sigma^\star)^\star$ по следния начин:

$$\mathcal{L}_G(\varepsilon) \overset{\text{деф}}{=} \varepsilon$$
$$\mathcal{L}_G(\beta \cdot X) \overset{\text{деф}}{=} \mathcal{L}_G(\beta) \cdot \mathcal{L}_G(X).$$

С други думи, ако $\alpha = X_1 \ldots X_n$, то

$$\mathcal{L}_G(X_1 \cdots X_n) = \mathcal{L}_G(X_1) \cdots \mathcal{L}_G(X_n).$$

Така можем да преформулираме *Следствие 4.1* по следния начин:

Следствие 4.2 (*Следствие 4.1*)**.** Нека G е безконтекстна граматика и A е променлива в G. Тогава

$$\mathcal{L}_G(A) = \bigcup_{A \to \alpha} \mathcal{L}_G(\alpha).$$

4.6 Фундаментална теорема за безконтекстните езици

Нека имаме безконтекстна граматика G, където $V = \{A_1, \ldots, A_n\}$. Дефинираме система за G по следния начин:

$$\left|\begin{array}{lcl} A_1 & = & \bigcup\{\alpha \mid A_1 \to \alpha \text{ е правило в } G\} \\ & \vdots & \\ A_n & = & \bigcup\{\alpha \mid A_n \to \alpha \text{ е правило в } G\} \end{array}\right.$$

Можем да запишем съкратено така:

$$A_1 = \bigcup_{A_1 \to \alpha} \alpha$$
$$\vdots$$
$$A_n = \bigcup_{A_n \to \alpha} \alpha.$$

Пример 4.4. Да разгледаме безконтекстната граматика G зададена със следните правила:

$$S \to aSb \mid \varepsilon.$$

Системата, която съответства на тази граматика е съставена от едно уравнение с една променлива и е следната:

$$\begin{aligned} S &= \bigcup_{S \to \alpha} \alpha \\ &= \bigcup\{aSb, \varepsilon\} \\ &= aSb \cup \{\varepsilon\}. \end{aligned}$$

Тогава според *Следствие 4.1*, $\mathcal{L}_G(S)$ е решение на системата

$$S = aSb \cup \{\varepsilon\}.$$

Тук е добре по-внимателно да обясним какво означава решение на такава система. Нека n-орката $\vec{K} = K_1, \ldots, K_n$ са езици. За произволна дума $\alpha \in (V \cup \Sigma)^\star$, дефинираме

$$\begin{aligned} &\vec{K}(\varepsilon) = \varepsilon; \\ &\vec{K}(\alpha \cdot b) = \vec{K}(\alpha) \cdot \{b\} \\ &\vec{K}(\alpha \cdot A_i) = \vec{K}(\alpha) \cdot K_i \end{aligned}$$

Тогава $K_1, \ldots, K_n$ е *решение* на системата, ако за всяко $i = 1, \ldots, n$ е изпълнено равенството:

$$K_i = \bigcup_{A_i \to \alpha} \vec{K}(\alpha).$$

В този раздел ще видим каква е връзката между решенията на тази система и езиците $\mathcal{L}_G(A_1), \cdots \mathcal{L}_G(A_n)$.

Лема 4.1. Нека имаме безконтекстна граматика G, където $V = \{A_1, \ldots, A_n\}$. Дефинираме система за G по следния начин:

$$\left|\begin{array}{lcl} A_1 & = & \bigcup\{\alpha \mid A_1 \to \alpha \text{ е правило в } G\} \\ & \vdots & \\ A_n & = & \bigcup\{\alpha \mid A_n \to \alpha \text{ е правило в } G\} \end{array}\right. \tag{4.1}$$

Тогава $\mathcal{L}_G(A_1), \ldots, \mathcal{L}_G(A_n)$ е *най-малкото решение* на системата.

Доказателство. Според *Следствие 4.1* вече знаем, че езиците $\mathcal{L}_G(A_i)$ са решения на системата. За да видим, че тези езици са *най-малкото решение*, ще използваме *Твърдение 4.3*, според което езиците $\mathcal{L}_G(A_i)$ могат да се представят чрез техните апроксимации. По-конкретно, ще използваме, че:

Имаме, че за всяко $i = 1, \ldots, n$,

$$\mathcal{L}_G(A_i) = \bigcup_{A_i \to \alpha} \mathcal{L}_G(\alpha).$$

$$\begin{aligned} &\mathcal{L}^0_G(A_i) = \emptyset \\ &\mathcal{L}^{\ell+1}_G(A_i) = \bigcup_{A_i \to \alpha} \mathcal{L}^\ell_G(\alpha). \end{aligned}$$

Имаме, че за всяко $i = 1, \ldots, n$, $K_i = \bigcup_{A_i \to \alpha} \vec{K}(\alpha)$.

И така, нека езиците $K_1, \ldots, K_n$ са произволно решение на системата. Понеже $\mathcal{L}_G(A_i) = \bigcup_\ell \mathcal{L}_G^\ell(A_i)$, достатъчно е да докажем, че $\mathcal{L}_G^\ell(A_i) \subseteq K_i$ за всяко ℓ.

- За $\ell = 0$ това е очевидно, защото $\mathcal{L}_G^0(A_i) = \emptyset$.
- Нека $\ell > 0$. Да разгледаме една дума $\omega \in \mathcal{L}_G^\ell(A_i)$. Да видим защо $\omega \in K_i$. Щом $\omega \in \mathcal{L}_G^\ell(A_i)$, то $\omega \in \mathcal{L}_G^{\ell-1}(\alpha)$ за някое правило $A_i \to \alpha$ в граматиката G. Достатъчно е да видим, че $\mathcal{L}_G^{\ell-1}(\alpha) \subseteq \vec{K}(\alpha)$. Нека $\alpha = X_1 \ldots X_n$.
  - Ако $X_i \in \Sigma$, то $\mathcal{L}_G^{\ell-1}(X_i) = \{X_i\} = \vec{K}(X_i)$.
  - Ако $X_i \in V$, то от **(И.П.)** следва, че $\mathcal{L}_G^{\ell-1}(X_i) \subseteq \vec{K}(X_i)$.

  Оттук следва, че $\mathcal{L}_G^{\ell-1}(\alpha) \subseteq \vec{K}(\alpha)$. Така заключаваме, че $\omega \in \vec{K}(\alpha)$. Понеже $K_i = \bigcup_{A_i \to \alpha} \vec{K}(\alpha)$, то $\omega \in K_i$.

□

Нека отново да разгледаме безконтекстната граматика G зададена със следните правила:

$$S \to aSb \mid \varepsilon.$$

Ясно е, че $\mathcal{L}_G(S)$ е най-малкото решение на системата

$$S = aSb \cup \{\varepsilon\}.$$

Може да се види лесно, че това решение е и *единствено*.

Сега да разгледаме граматиката G, където

$$S \to S \mid aSb \mid \varepsilon.$$

Системата за тази граматика е

$$S = S \cup aSb \cup \{\varepsilon\}.$$

Тук вече има смисъл да се говори за най-малко решение, защото е ясно, че езикът $\{a, b\}^\star$ също е решение на системата.

Лема 4.2. Да разгледаме следната произволна система от вида

$$\left| \begin{array}{lcl} A_1 & = & \bigcup_{j=1}^{m_i} \alpha_{1,j} \\ & \vdots & \\ A_n & = & \bigcup_{j=1}^{m_n} \alpha_{n,j}, \end{array} \right. \tag{4.2}$$

където $\alpha_{i,j} \in (V \cup \Sigma)^\star$ за всяко $i = 1, \ldots, n$ и $j = 1, \ldots, m_i$. Да дефинираме безконтекстната граматика G с променливи $V = \{A_1, \ldots, A_n\}$ и правила

$$A_i \to \alpha_{i,1} \mid \cdots \mid \alpha_{i,m_i},$$

за всяко $i = 1, \ldots, n$. Тогава $\mathcal{L}_G(A_1), \ldots, \mathcal{L}_G(A_n)$ е *най-малкото решение* на системата.

Доказателство. Нека имаме граматиката G от условието на лемата. От *Лема 4.1*, системата (4.1) съвпада точно със системата (4.2). Тогава според *Лема 4.1*, наистина $\mathcal{L}_G(A_1), \ldots, \mathcal{L}_G(A_n)$ е *най-малкото решение* на системата (4.2). □

Сравнете с *Теорема 3.4*.

Теорема 4.2 (Фундаментална теорема за безконтекстните езици)**.** Един език L е *безконтекстен* точно тогава, когато L е измежду езиците $L_1, \ldots, L_n$, които представляват *най-малкото решение* на система от вида

$$\left|\begin{array}{rcl} A_1 & = & \bigcup_{j=1}^{m_i} \alpha_{1,j} \\ & \vdots & \\ A_n & = & \bigcup_{j=1}^{m_n} \alpha_{n,j}, \end{array}\right.$$

където $\alpha_{i,j} \in (V \cup \Sigma)^\star$ за всяко $i = 1, \ldots, n$ и $j = 1, \ldots, m_i$.

Понеже системите, които описват регулярните езици са частен случай на системите, които описват безконтекстните езици, то веднага получаваме следното следствие.

Следствие 4.3. Всеки регулярен език е безконтекстен.

Доказателство. Нека имаме един регулярен език L. Тогава съществува краен автомат $\mathcal{A}$, за който $\mathcal{L}(\mathcal{A}) = L$. Според *Теорема 3.4*, съществува система от вида

$$\left|\begin{array}{rcl} X_1 & = & R_{1,1} \cdot X_1 \cup \cdots \cup R_{1,n} \cdot X_n \cup P_1 \\ & \vdots & \\ X_n & = & R_{n,1} \cdot X_1 \cup \cdots \cup R_{n,n} \cdot X_n \cup P_n, \end{array}\right. \tag{4.3}$$

където $P_i \subseteq \{\varepsilon\}$ и $R_{i,j} \subseteq \Sigma$, за всяко $i, j = 1, \ldots, n$, чието *единствено* решение е $L_1, \ldots L_n$ и L е измежду тези езици.

Тази система представлява частен случай на системата в условието на *Теорема 4.2*. Това означава, че можем да приложим *Теорема 4.2* за системата (4.3). Така получаваме, че можем да построим безконтекстна граматика G с променливи $V = \{X_1, \ldots, X_n\}$ и $L_G(X_1), \ldots, \mathcal{L}_G(X_n)$ е *най-малкото решение* на системата (4.3). Но понеже тази система има *единствено* решение, то нашият регулярен език L е измежду езиците $\mathcal{L}_G(X_1), \ldots, \mathcal{L}_G(X_n)$. Така заключаваме, че L е безконтекстен език. □

Като първи пример, нека да видим, че съществува безконтекстен език, който не е регулярен.

Да напомним, че вече видяхме, че езикът $L = \{a^n b^n \mid n \in \mathbb{N}\}$ не е регулярен.

Пример 4.5. Да разгледаме безконтекстната граматика G зададена със следните правила:

$$S \to aSb \mid \varepsilon.$$

Да разгледаме първите няколко члена на редицата от езици $\mathcal{L}_G^\ell(S)$:

$$\begin{aligned} &\mathcal{L}_G^0(S) = \emptyset \\ &\mathcal{L}_G^1(S) = \{a\} \cdot \emptyset \cdot \{b\} \cup \{\varepsilon\} = \{\varepsilon\} \\ &\mathcal{L}_G^2(S) = \{a\} \cdot \{\varepsilon\} \cdot \{b\} \cup \{\varepsilon\} = \{\varepsilon, ab\} \\ &\mathcal{L}_G^3(S) = \{a\} \cdot \{\varepsilon, ab\} \cdot \{b\} \cup \{\varepsilon\} = \{\varepsilon, ab, aabb\} = \{a^n b^n \mid n < 3\} \\ &\vdots \end{aligned}$$

Лесно се съобразява, че $\mathcal{L}_G^\ell(S) = \{a^n b^n \mid n < \ell\}$. Заключаваме, че:

$$\mathcal{L}(G) = \mathcal{L}_G(S) = \bigcup_\ell \mathcal{L}_G^\ell(S) = \{a^n b^n \mid n \in \mathbb{N}\}.$$

Примерни задачи

Задача 4.5. Докажете, че езикът $L = \{a^n b^k c^{n+k} \mid n, k \in \mathbb{N}\}$ е безконтекстен.

Доказателство. Да разгледаме безконтекстната граматика G зададена със следните правила:

$$S \to aSc \mid B$$
$$B \to bBc \mid \varepsilon.$$

Да видим защо $\mathcal{L}(G) = \{a^n b^k c^{n+k} \mid n, k \in \mathbb{N}\}$. Първо ще докажем *коректността* на граматиката. Това означава, че G не генерира думи извън желания език, т.е. $\mathcal{L}_G(S) \subseteq L$. За да направим това обаче, трябва да знаем също така и $\mathcal{L}_G(B)$. Формално казано, трябва да докажем, че за всяко ℓ е изпълнено следното:

$$\mathcal{L}^\ell_G(S) \subseteq \{a^n b^k c^{n+k} \mid n, k \in \mathbb{N}\} \tag{4.4}$$
$$\mathcal{L}^\ell_G(B) \subseteq \{b^k c^k \mid k \in \mathbb{N}\}. \tag{4.5}$$

Това ще направим с индукция по ℓ. Да напомним, че според *Твърдение 4.3* имаме следните връзки:

$$\mathcal{L}^0_G(S) = \emptyset$$
$$\mathcal{L}^{\ell+1}_G(S) = \{a\} \cdot \mathcal{L}^\ell_G(S) \cdot \{c\} \cup \mathcal{L}^\ell_G(B)$$
$$\mathcal{L}^0_G(B) = \emptyset$$
$$\mathcal{L}^{\ell+1}_G(B) = \{b\} \cdot \mathcal{L}^\ell_G(B) \cdot \{c\} \cup \{\varepsilon\}.$$

Обърнете внимание, че не можем да докажем *Свойство 4.4* независимо от *Свойство 4.5*.

Очевидно е, че *Свойство 4.4* и *Свойство 4.5* са изпълнени за $\ell = 0$. Да примем, че имаме *Свойство 4.4* и *Свойство 4.5* за някое ℓ. Ще докажем свойствата и за $\ell + 1$.

- Първо, нека $\alpha \in \mathcal{L}^{\ell+1}_G(S)$. Имаме два случая.
  - Нека $\alpha \in \{a\} \cdot \mathcal{L}^\ell_G(S) \cdot \{c\}$. От **(И.П.)** следва, че $\alpha = a^{n+1} b^k c^{n+k+1}$ за някои естествени числа n и k. Тогава е ясно, че $\alpha \in \{a^n b^k c^{n+k} \mid n, k \in \mathbb{N}\}$.
  - Нека $\alpha \in \mathcal{L}^\ell_G(B)$. От **(И.П.)** следва, че $\alpha \in \{b^k c^k \mid k \in \mathbb{N}\} \subseteq \{a^n b^k c^{n+k} \mid n, k \in \mathbb{N}\}$.
- Второ, нека $\alpha \in \mathcal{L}^{\ell+1}_G(B)$. Имаме два случая.
  - Нека $\alpha \in \{b\} \cdot \mathcal{L}^\ell_G(B) \cdot \{c\}$. От **(И.П.)** следва, че $\alpha = b^{k+1} c^{k+1}$ за някое естествено число k. Тогава е ясно, че $\alpha \in \{b^k c^k \mid k \in \mathbb{N}\}$.
  - Нека $\alpha = \varepsilon$. В този случай също е ясно, че $\alpha \in \{b^k c^k \mid k \in \mathbb{N}\}$.

Оттук заключаваме, че $\mathcal{L}_G(S) \subseteq \{a^n b^k c^{n+k} \mid n, k \in \mathbb{N}\}$.

Сега да разгледаме *пълнотата* на граматиката, което означава, че всяка дума от езика се генерира от G. С други думи, $\{a^n b^k c^{n+k} \mid n, k \in \mathbb{N}\} \subseteq \mathcal{L}_G(S)$. Първо ще докажем, че

$$\{b^k c^k \mid k \in \mathbb{N}\} \subseteq \mathcal{L}_G(B). \tag{4.6}$$

Това ще направим с *пълна* индукция по дължината на думата. Да разгледаме произволна дума $\alpha \in \{b^k c^k \mid k \in \mathbb{N}\}$.

- Нека $|\alpha| = 0$, т.е. $\alpha = \varepsilon$. Ясно е, че $\alpha \in \mathcal{L}_G(B)$.
- Нека $|\alpha| > 0$, т.е. $\alpha = b^{k+1} c^{k+1}$. От **(И.П.)** за *Свойство 4.6* имаме $\alpha \in \{b\} \cdot \mathcal{L}_G(B) \cdot \{c\} \subseteq \mathcal{L}_G(B)$.

Така доказахме *Свойство 4.6*. Вече сме готови да докажем, че:

$$\{a^n b^k c^{n+k} \mid n, k \in \mathbb{N}\} \subseteq \mathcal{L}_G(S). \tag{4.7}$$

Това включване пак ще докажем с *пълна* индукция по дължината на думата. Да разгледаме произволна дума $\alpha \in L$.

- Нека $|\alpha| = 0$, т.е. $\alpha = \varepsilon$. Ясно е, че $\alpha \in \mathcal{L}_G(B) \subseteq \mathcal{L}_G(S)$.
- Нека сега $|\alpha| > 0$, т.е. $\alpha = a^n b^k c^{n+k}$, където $n > 0$ или $k > 0$. Да разгледаме два случая.
  - Нека $n = 0$. Тогава $\alpha = b^k c^k$ и $k > 0$. Тогава от *Свойство 4.6* следва, че $\alpha \in \mathcal{L}_G(B) \subseteq \mathcal{L}_G(S)$.
  - Нека $n > 0$. Тогава от **(И.П.)** за *Свойство 4.7* имаме $\alpha \in \{a\} \cdot \mathcal{L}_G(S) \cdot \{c\} \subseteq \mathcal{L}_G(S)$.

Доказахме коректност и пълнота на граматиката и следователно можем да заключим, че $\mathcal{L}(G) = \{a^n b^k c^{n+k} \mid n, k \in \mathbb{N}\}$. □

Задача 4.6. Докажете, че езикът $L = \{a^m b^n c^k \mid m + n \geq k\}$ е безконтекстен.

Доказателство. Да разгледаме граматиката G с правила

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

От *Твърдение 4.3* имаме, че:

$$\mathcal{L}_G^0(S) = \emptyset$$
$$\mathcal{L}_G^{\ell+1}(S) = \{a\} \cdot \mathcal{L}_G^\ell(S) \cdot \{c\} \cup \{a\} \cdot \mathcal{L}_G^\ell(S) \cup \mathcal{L}_G^\ell(B)$$
$$\mathcal{L}_G^0(B) = \emptyset$$
$$\mathcal{L}_G^{\ell+1}(B) = \{b\} \cdot \mathcal{L}_G^\ell(B) \cdot \{c\} \cup \{b\} \cdot \mathcal{L}_G^\ell(B) \cup \{\varepsilon\}.$$

Да предположим, че за произволно естествено число ℓ е изпълнено следното:

Тези две свойства ще бъдат нашето **(И.П.)** . Очевидно е, че те са изпълнени за $\ell = 0$.

$$\mathcal{L}_G^\ell(S) \subseteq \{a^n b^m c^k \mid n + m \geq k\} \tag{4.8}$$
$$\mathcal{L}_G^\ell(B) \subseteq \{b^m c^k \mid m \geq k\}. \tag{4.9}$$

Ще докажем, че:

$$\mathcal{L}_G^{\ell+1}(S) \subseteq \{a^n b^m c^k \mid n + m \geq k\}$$
$$\mathcal{L}_G^{\ell+1}(B) \subseteq \{b^m c^k \mid m \geq k\}.$$

За първото включване, да разгледаме произволна дума $\alpha \in \mathcal{L}_G^{\ell+1}(S)$. Имаме три случая:

- Ако $\alpha \in \mathcal{L}_G^\ell(B)$, то от **(И.П.)** имаме, че
$$\alpha \in \{b^m c^k \mid m \geq k\} \subseteq \{a^n b^m c^k \mid n + m \geq k\}.$$
- Ако $\alpha \in \{a\} \cdot \mathcal{L}_G^\ell(S)$, то от **(И.П.)** имаме, че
$$\alpha \in \{a^{n+1} b^m c^k \mid n + m \geq k\} \subseteq \{a^n b^m c^k \mid n + m \geq k\}.$$
- Ако $\alpha \in \{a\} \cdot \mathcal{L}_G^\ell(S) \cdot \{c\}$, то от **(И.П.)** имаме, че
$$\alpha \in \{a^{n+1} b^m c^{k+1} \mid n + m \geq k\} \subseteq \{a^n b^m c^k \mid n + m \geq k\}.$$

За второто включване, нека $\alpha \in \mathcal{L}_G^{\ell+1}(B)$. Имаме три случая за думата α.

- Нека $\alpha \in \{b\} \cdot \mathcal{L}_G^\ell(B) \cdot \{c\}$. Тогава от **(И.П.)** имаме, че:
$$\alpha \in \{b^{m+1} c^{k+1} \mid m \geq k\} \subseteq \{b^m c^k \mid m \geq k\}.$$
- Нека $\alpha \in \{b\} \cdot \mathcal{L}_G^\ell(B)$. Тогава от **(И.П.)** имаме, че:
$$\alpha \in \{b^{m+1} c^k \mid m \geq k\} \subseteq \{b^m c^k \mid m \geq k\}.$$
- Нека $\alpha \in \{\varepsilon\}$. Тогава е ясно, че имаме $\alpha \in \{b^m c^k \mid m \geq k\}$.

Най-накрая заключаваме, че

Така доказахме *коректност* на граматиката.

$$\mathcal{L}_G(S) = \bigcup_\ell \mathcal{L}_G^\ell(S) \subseteq \{a^n b^m c^k \mid n + m \geq k\}$$
$$\mathcal{L}_G(B) = \bigcup_\ell \mathcal{L}_G^\ell(B) \subseteq \{a^n b^m c^k \mid n + m \geq k\}.$$

Сега трябва да докажем обратните включвания, а именно:

Сега ще докажем *пълнота* на граматиката. Тук ще използваме представянията на $\mathcal{L}_G(S)$ и $\mathcal{L}_G(B)$ според *Следствие 4.1*.

$$\{a^n b^m c^k \mid n + m \geq k\} \subseteq \mathcal{L}_G(S) \tag{4.10}$$
$$\{b^m c^k \mid m \geq k\} \subseteq \mathcal{L}_G(B). \tag{4.11}$$

Трябва да започнем първо със *Свойство 4.11*. Да разгледаме произволна дума $\alpha = b^m c^k$. Трябва да докажем, че $\alpha \in \mathcal{L}_G(B)$. Ще направим това с индукция по m.

- Нека $m = 0$. Това означава, че $\alpha = \varepsilon$. Ясно е, че $\varepsilon \in \mathcal{L}_G(B)$.
- Нека $m > 0$. Тук имаме два подслучая.
  - Нека $m = k$. Тогава $\alpha = b\gamma c$ и имаме, че $\gamma = b^{m-1} c^{k-1}$. Можем да приложим **(И.П.)** за γ и следователно $\gamma \in \mathcal{L}_G(B)$. Получаваме, че $\alpha \in \{b\} \cdot \mathcal{L}_G(B) \cdot \{c\} \subseteq \mathcal{L}_G(B)$.
  - Нека $m > k$. Тогава $\alpha = b\gamma$ и имаме, че $\gamma = b^{m-1} c^k$. Можем да приложим **(И.П.)** за γ и следователно $\gamma \in \mathcal{L}_G(B)$. Получаваме, че $\alpha \in \{b\} \cdot \mathcal{L}_G(B) \subseteq \mathcal{L}_G(B)$.

Сега преминаваме към *Свойство 4.10*. Да разгледаме произволна дума $\alpha = a^n b^m c^k$. Трябва да докажем, че $\alpha \in \mathcal{L}_G(S)$. Ще направим това с индукция по n.

- Нека $n = 0$. Тогава $\alpha = b^m c^k$ и $m \geq k$. От *Свойство 4.11* следва, че $\alpha \in \mathcal{L}_G(B) \subseteq \mathcal{L}_G(S)$.
- Нека $n > 0$. Имаме два подслучая.

- Нека $n+m=k$. Тогава $\alpha = a\gamma c$ и $\gamma = a^{n-1}b^m c^{k-1}$. Можем да приложим **(И.П.)** за γ и следователно $\gamma \in \mathcal{L}_G(S)$. Получаваме, че $\alpha \in \{a\} \cdot \mathcal{L}_G(S) \cdot \{c\} \subseteq \mathcal{L}_G(S)$.
- Нека $n+m>k$. Тогава $\alpha = a\gamma$ и $\gamma = a^{n-1}b^m c^k$. Можем да приложим **(И.П.)** за γ и следователно $\gamma \in \mathcal{L}_G(S)$. Получаваме, че $\alpha \in \{a\} \cdot \mathcal{L}_G(S) \subseteq \mathcal{L}_G(S)$.

□

Задача 4.7. Докажете, че езикът $L = \{a^n b^m c^k d^\ell \mid n+k = m+\ell\}$ е безконтекстен.

Упътване. Да разгледаме произволна дума от вида $\omega = a^n b^m c^k d^\ell$. Имаме два случая.

- Ако $n > \ell$, тогава имаме еквивалентността: $\omega \in L \Leftrightarrow (n-\ell)+k = m$.
- Ако $n \leq \ell$, тогава имаме еквивалентността: $\omega \in L \Leftrightarrow k = m+(\ell-n)$.

Това наблюдение ни подсказва, че е полезно да разгледаме следните езици:

$$L_1 = \{a^n b^m c^k \mid m = n+k\},$$
$$L_2 = \{b^m c^k d^\ell \mid k = m+\ell\}.$$

Така получаваме, че

$$L = \{a^n \cdot \omega \cdot d^n \mid n \in \mathbb{N} \,\&\, \omega \in L_1 \cup L_2\}.$$

L_1 е безконтекстен език, защото може да се опише с безконтекстната граматика G_1 със следните правила:

$$S_1 \to AC, \quad A \to aAb \mid \varepsilon, \quad C \to bCc \mid \varepsilon.$$

L_2 също е безконтекстен език, защото може да се опише с безконтекстната граматика G_2 със следните правила:

$$S_2 \to BD, \quad B \to bBc \mid \varepsilon, \quad D \to cCd \mid \varepsilon.$$

Тогава безконтекстната граматика G за L съдържа правилата на граматиките G_1 и G_2, а също и правилата

$$S \to aSd \mid S_1 \mid S_2.$$

□

Задача 4.8. Докажете, че езикът $L = \{\alpha\beta \in \{a,b\}^\star \mid \ |\alpha| = |\beta| \,\&\, \alpha \neq \beta\}$ е безконтекстен.

Ние вече знаем, че този език не е регулярен

Упътване. Да разгледаме една произволна дума ω, където $\omega = \alpha\beta$, $|\alpha| = |\beta|$ и $\alpha \neq \beta$. Знаем, че съществува индекс $i < |\alpha|$, такъв че думата ω може да се представи така:

$$\omega = \alpha[:i] \cdot \alpha[i] \cdot \alpha[i+1:] \cdot \beta[:i] \cdot \beta[i] \cdot \beta[i+1:],$$

където $\alpha[i] \neq \beta[i]$.

Нека $n = |\alpha| = |\beta|$ и да представим n като $n = i+1+k$. Имаме два случая.

- Ако $k \geq i$, то можем да преставим ω по следния начин:

$$\omega = \underbrace{\alpha[:i]}_{\text{дълж. } i} \cdot \alpha[i] \cdot \underbrace{\alpha[i+1:2i+1]}_{\text{дълж. } i} \cdot \underbrace{\alpha[2i+1:] \cdot \beta[:i]}_{\text{дълж. } k} \cdot \beta[i] \cdot \underbrace{\beta[i+1:]}_{\text{дълж. } k}.$$

- Нека $k < i$, то можем да преставим ω по следния начин:

$$\omega = \underbrace{\alpha[:i]}_{\text{дълж. } i} \cdot \alpha[i] \cdot \underbrace{\alpha[i+1:] \cdot \beta[:i-k]}_{\text{дълж. } i} \cdot \underbrace{\beta[i-k:i]}_{\text{дълж. } k} \cdot \beta[i] \cdot \underbrace{\beta[i+1:]}_{\text{дълж. } k}.$$

От тези представяния на ω е ясно, че можем да изразим езика L по следния начин:

$$L = L_a \cdot L_b \cup L_b \cdot L_a,$$

където:

$$L_a = \{\alpha \cdot a \cdot \beta \mid \alpha, \beta \in \{a,b\}^\star \,\&\, |\alpha| = |\beta|\}$$
$$L_b = \{\alpha \cdot b \cdot \beta \mid \alpha, \beta \in \{a,b\}^\star \,\&\, |\alpha| = |\beta|\}.$$

Сега разгледайте безконтекстната граматика G със следните правила:

$$S \to AB \mid BA$$
$$A \to XAX \mid a$$
$$B \to XBX \mid b$$
$$X \to a \mid b.$$

Лесно се съобразява, че $L_a = \mathcal{L}_G(A)$ и $L_b = \mathcal{L}_G(B)$ и накрая $L = \mathcal{L}_G(S)$. □

Задача 4.9. Докажете, че езикът $L = \{\alpha\sharp\beta \mid \alpha, \beta \in \{a,b\}^\star \ \&\ \alpha \neq \beta\}$ е безконтекстен.

Упътване. Разгледайте граматиката:

$$\begin{aligned}
&S \to AaR \mid BbR \mid E\\
&A \to XAX \mid bR\sharp\\
&B \to XBX \mid aR\sharp\\
&E \to XEX \mid XR\sharp \mid \sharp XR\\
&R \to XR \mid \varepsilon\\
&X \to a \mid b.
\end{aligned}$$

Имаме, че за произволни думи $\alpha, \beta, \gamma, \delta \in \{a,b\}^\star$,

$$\begin{aligned}
&S \overset{\star}{\lhd} \alpha b\gamma\sharp\beta a\delta \ \&\ |\alpha| = |\beta|,\\
&S \overset{\star}{\lhd} \alpha a\gamma\sharp\beta b\delta \ \&\ |\alpha| = |\beta|, \text{ или}\\
&S \overset{\star}{\lhd} \alpha\sharp\beta \ \&\ |\alpha| \neq |\beta|.
\end{aligned}$$

□

Задача 4.10. Докажете, че езикът $L = \{a^n b^m c^k d^\ell \mid n + k \geq m + \ell\}$ е безконтекстен.

Задача 4.11. Докажете, че езикът $L = \{\alpha\sharp\beta \mid \alpha, \beta \in \{a,b\}^\star \ \&\ |\alpha| \neq |\beta|\}$ е безконтекстен.

Задача 4.12. Да разгледаме граматиката G с правила

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

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

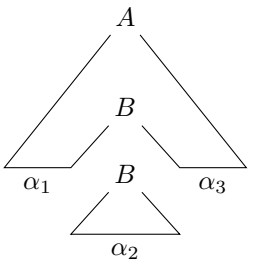
4.7 Лема за покачването

В този раздел ще разгледаме едно твърдение, което ще ни даде критерий за проверка кога един език не е безконтекстен. Ще започнем с няколко помощни твърдения.

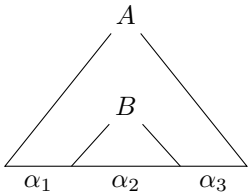
Твърдение 4.5. За произволни променливи A, B и думи $\alpha_1, \alpha_2, \alpha_3$ можем да направим извода:

$$\frac{A \overset{\ell_1}{\lhd} \alpha_1 B \alpha_3 \qquad B \overset{\ell_2}{\lhd} \alpha_2}{A \overset{\leq \ell_1+\ell_2}{\lhd} \alpha_1\alpha_2\alpha_3}$$

Упътване. Формално доказателството протича с индукция по ℓ_1.



(а) Синтактични дървета за изводите $A \overset{\ell_1}{\lhd} \alpha_1 B \alpha_3$ и $B \overset{\ell_2}{\lhd} \alpha_2$



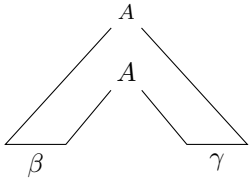
(б) Съединяваме двете дървета

□

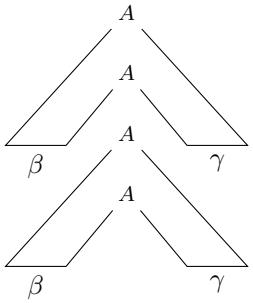
Твърдение 4.6. За произволна променлива A и произволни думи β и γ можем да направим извода:

$$\frac{A \overset{\ell}{\lhd} \beta A \gamma \qquad i \in \mathbb{N}}{A \overset{\leq i \cdot \ell}{\lhd} \beta^i A \gamma^i}$$

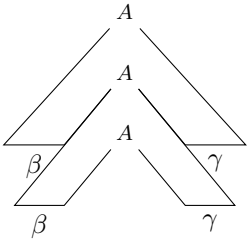
Упътване. Формално доказателството трябва да следва индукция по i. Понеже това доказателство не крие изненади, ще се задоволим с един пример. Нека P да бъде синтактично дърво съответстващо на извода $A \overset{\ell}{\lhd} \beta A \gamma$. Тогава можем да получим синтактично дърво $P^{(2)}$ съответстващо на $A \overset{\leq 2\ell}{\lhd} \beta^2 A \gamma^2$ по следния начин.



(а) Синтактично дърво P за извода $A \overset{\ell}{\lhd} \beta A \gamma$



(б) Съединяваме две копия на P



(в) Синтактично дърво $P^{(2)}$ за извода $A \overset{\leq 2\ell}{\lhd} \beta^2 A \gamma^2$

□

Следващото твърдение изолзва съществено връзката между синтактичните дървета и релацията $\overset{\ell}{\lhd}$.

Твърдение 4.7. Нека G е безконтекстна граматика и

$$b \overset{\text{деф}}{=} \max\{\, |\gamma| \mid A \to \gamma \text{ е правило в } G \,\}.$$

Тогава:

- Ако $A \overset{\leq \ell}{\lhd} \alpha$, то $|\alpha| \leq b^\ell$.
- Ако $\alpha \in \mathcal{L}(G)$ и $|\alpha| > b^\ell$, то $S \overset{\geq \ell+1}{\lhd} \alpha$.

Доказателство. Възможно е да докажем това твърдение директно, без да се позоваваме на синтактични дървета за извод, като направим индукция по ℓ. □

Доказателство.
Това твърдение е на практика следствие от *Задача 4.2* в комбинация с *Твърдение 4.2*.

Числото b е максималната разклоненост, която можем да получим за синтактично дърво на извод в граматиката G.

- Да разгледаме синтактично дърво $P = (T, \lambda)$ за извода $A \overset{\leq \ell}{\lhd} \alpha$. Според *Задача 4.2* имаме, че $|\alpha| = |\texttt{leaves}(T)| \leq b^{\texttt{height}(T)} = b^\ell$.
- Нека $\alpha \in \mathcal{L}(G)$ и $|\alpha| > b^\ell$. Това означава, че $S \overset{\star}{\lhd} \alpha$. Ако допуснем, че $S \overset{\leq \ell}{\lhd} \alpha$, то бихме имали, че $|\alpha| \leq b^\ell$, което е противоречие. Заключаваме, че $S \overset{\geq \ell+1}{\lhd} \alpha$.

□

Лема 4.3 (за покачването (безконтекстни езици)). За всеки безконтекстен език L съществува $p > 0$, такова че ако $\alpha \in L$, за която $|\alpha| \geq p$, то съществува разбиване на думата на пет части, $\alpha = xyuvw$, за което е изпълнено:

1) $|yv| \geq 1$,
2) $|yuv| \leq p$, и
3) $(\forall i \in \mathbb{N})[xy^i uv^i w \in L]$.

[21, стр. 125], [10, стр. 125], [9, стр. 275], [13, стр. 148].

Ще казваме, че p е константа на покачването. Тук отново да напомним, че $0 \in \mathbb{N}$ и $xy^0uv^0w = xuw$.

Доказателство. Нека G е безконтекстна граматика за езика L. Да положим

$$p \overset{\text{деф}}{=} b^{|V|+1}, \text{ където } b \overset{\text{деф}}{=} \max\{\, |\gamma| \mid A \to \gamma \text{ е правило в } G \,\}.$$

Числото b е максималната разклоненост на всяко дърво на извод за дума изводима от граматиката G.

Ще покажем, че този избор на p е удачен за удовлетворяването на условията на лемата. Ще наричаме p константа на покачването за граматиката G. Да разгледаме произволна дума $\alpha \in \mathcal{L}(G)$, за която $|\alpha| \geq p$. Понеже $\mathcal{L}(G)$ е безкраен език, то със сигурност можем да намерим такава дума. Щом $S \overset{\star}{\lhd} \alpha$, според *Твърдение 4.7*, винаги имаме $S \overset{\ell}{\lhd} \alpha$ за $\ell \geq |V|+1$. Нека измежду всички синтактични дървета $P = (T, \lambda)$ за извода $S \overset{\ell}{\lhd} \alpha$ сме избрали такова с *минимален* брой елементи в T. Да фиксираме *максимален* път π в T, т.е. дума $\pi \in T$ и $|\pi| = \ell$. Чрез функцията λ пътя π определя редицата $X_0, X_1, \ldots, X_\ell$, като $X_i = \lambda(\rho)$, където $\rho \prec \pi$ и $|\rho| = i$. Ясно е, че $X_i \in V$ за $i < \ell$ и $X_\ell \in \Sigma \cup \{\varepsilon\}$. Щом $\ell \geq |V| + 1$, то това означава, че по този път π се срещат

Важното е да вземем дума $\alpha \in \mathcal{L}(G)$, за която $|\alpha| > b^{|V|}$.

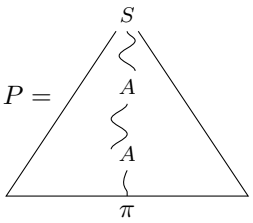
Принципът на Дирихле е известен също и като принципа на чекмеджетата.

$\ell \geq |V| + 1$ на брой променливи. От принципа на Дирихле следва, че трябва да има повторения на променливи по този път. Във вече фиксираното синтактично дърво P можем да изберем това срещане на двойка повтарящи се променливи по пътя π, както на *Фигура 4.8а*, което е възможно най-близо до края на пътя. Това означава, че можем да разбием извода $S \overset{\ell}{\lhd} \alpha$ като $S \overset{\star}{\lhd} xAw$ и $A \overset{\leq |V|+1}{\lhd} \gamma$, където $\alpha = x\gamma w$. Сега според нашия избор, изводът $A \overset{\leq |V|+1}{\lhd} \gamma$ може да се разбие като $A \overset{\leq |V|+1}{\lhd} yAv$ и $A \overset{\leq |V|}{\lhd} u$, където $\gamma = yuv$. Да обобщим. Можем да разбием думата α на пет части като $\alpha = xyuvw$ с изводите:

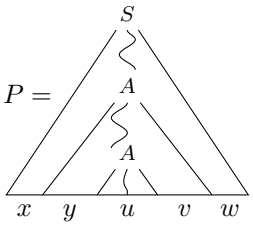
(1) $S \overset{\star}{\lhd} xAw$,

(2) $A \overset{\leq |V|+1}{\lhd} yuv$, защото в дървото P можем да изберем първата двойка повтарящи се променливи, които срещнем отзад напред по пътя π. Тук сме означили тази повтаряща се променлива с A.

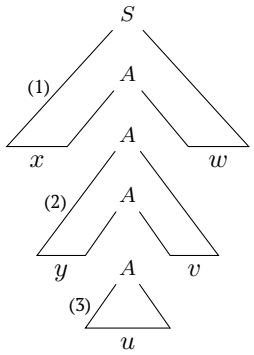
(3) $A \overset{\leq |V|}{\lhd} u$, като тук вече няма повтарящи се променливи по останалата част от пътя π.



(а) Първата двойка повтарящи се променливи, които намерим като тръгнем отдолу нагоре по пътя π.



(б) Разбиваме дървото на три части.



(в) Получаваме три отделни синтактични дървета.

Сега остава да проверим условието на лемата:

- Да допуснем, че $|yv| = 0$. Това означава, че $\alpha = xuw$ и имаме извода:

$$\frac{S \overset{\star}{\lhd} xAw \qquad A \overset{\star}{\lhd} u}{S \overset{\star}{\lhd} \underbrace{xuw}_{\alpha},} \text{ (*Твърдение 4.5*)}$$

  за който съществува дърво на извод с по-малко на брой възли отколкото P, защото махаме средната част, която съдържа поне един възел. Това е противоречие с избора на P като синтактично дърво за $S \overset{\star}{\lhd} \alpha$ с минимален брой възли. Заключаваме, че $|yv| \geq 1$.

- Понеже имаме извода $A \overset{\leq |V|+1}{\lhd} yuv$, то от *Твърдение 4.7* следва, че $|yuv| \leq b^{|V|+1} = p$.

- За произволно $i \in \mathbb{N}$ имаме извода:

$$\text{(Тв. 4.5)}\ \frac{\text{(Тв. 4.5)}\ \dfrac{\text{(Тв. 4.6)}\ \dfrac{A \overset{\star}{\lhd} yAv}{A \overset{\star}{\lhd} y^i Av^i} \qquad A \overset{\star}{\lhd} u}{A \overset{\star}{\lhd} y^i uv^i} \qquad S \overset{\star}{\lhd} xAw}{S \overset{\star}{\lhd} xy^i uv^i w}$$

Оттук заключаваме, че $(\forall i \in \mathbb{N})[xy^iuv^iw \in \mathcal{L}(G)]$.

□

Както и при Лема за покачването за регулярни езици, ние ще използваме *контрапозицията* на условието на Лема за покачването за безконтекстни езици при проверка, че даден език не е безконтекстен. [13, стр. 153].

Следствие 4.4. Нека L е произволен **безкраен** език. Нека също така е изпълнено, че:

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

$(\forall)$ за *всяко* естествено число $p \geq 1$,

$(\exists)$ можем да намерим дума $\alpha \in L$, $|\alpha| \geq p$, такава че

$(\forall)$ за *всяко* разбиване на думата на пет части, $\alpha = xyuvw$, със свойствата $|yv| \geq 1$ и $|yuv| \leq p$,

$(\exists)$ можем да намерим $i \in \mathbb{N}$, за което е изпълнено, че $xy^iuv^iw \notin L$.

Тогава L **не** е безконтекстен език.

✍ Докажете! Аналогично е на *Следствие 3.3*

Следствие 4.5. Нека G е безконтекстна граматика и p е константата на покачването за G. Тогава $|\mathcal{L}(G)| = \infty$ точно тогава, когато съществува $\alpha \in \mathcal{L}(G)$, за която $p \leq |\alpha| < 2p$.

✍ Докажете!

Забележка. Алгоритъм за проверка дали един безконтекстен език е безкраен следвайки горния критерий би имал експоненциална сложност относно $|G|$.

✍ Защо?

Пример 4.6. Да видим защо езикът $L = \{a^nb^nc^n \mid n \in \mathbb{N}\}$ не е безконтекстен.

$(\forall)$ Разглеждаме произволна константа $p \geq 1$.

$(\exists)$ Избираме дума $\alpha \in L$ със свойството, че $|\alpha| \geq p$. В случая, нека $\alpha = a^pb^pc^p$.

$(\forall)$ Разглеждаме произволно разбиване на α на пет части $\alpha = xyuvw$ със свойствата $|yuv| \leq p$ и $1 \leq |yv|$.

$(\exists)$ За всяко такова разбиване ще посочим 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.

✍ Съобразете, че може да изберете и $i = 0$. В този пример може да вземете едно и също i за всяко разбиване на думата α от предишната стъпка. В други примери ще видим, че изборът на i зависи от разглежданото разбиване на думата.

- 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. В този случай получаваме, че в xy^2uv^2w 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 4.4* следва, че езикът L не е безконтекстен.

Твърдение 4.8. Безконтекстните езици **не** са затворени относно операциите сечение и допълнение.

Упътване. Да разгледаме езика $L_0 = \{a^nb^nc^n \mid n \in \mathbb{N}\}$, за който вече знаем от *Пример 4.6*, че не е безконтекстен. Да вземем също така и безконтекстните езици

$$L_1 = \{a^nb^nc^m \mid n, m \in \mathbb{N}\},\ L_2 = \{a^mb^nc^n \mid n, m \in \mathbb{N}\}.$$

✍ Защо L_1 и L_2 са безконтекстни?

Да напомним, че с $\overline{L}$ означаваме допълнението на езика L, т.е. $\overline{L} \stackrel{\text{деф}}{=} \Sigma^\star \setminus L$.

- 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. Понеже допуснахме, че безконтекстните са затворени относно допълнение, то $\overline{L}_3$ също е безконтекстен. Но тогава получаваме, че езикът
$$L_0 = L_1 \cap L_2 = \overline{\overline{L}_1 \cup \overline{L}_2} = \overline{L}_3$$
е безконтекстен, което е противоречие.

Друг пример, с който може да се види, че безконтекстните езици не са затворени относно допълнение е като се докаже, че езикът

$\{a,b\}^\star \setminus \{\omega\omega \mid \omega \in \{a,b\}^\star\}$

е безконтекстен. Това следва лесно като се използва *Задача 4.8*.

□

Примерни задачи

Задача 4.13. Докажете, че езикът $L = \{a^i b^j c^k \mid 0 \leq i \leq j \leq k\}$ не е безконтекстен.

Упътване. Прилагаме *Следствие 4.4*.

(∀) Разглеждаме произволна константа $p \geq 1$.

(∃) Избираме конкретна дума $\alpha \in L$, $|\alpha| \geq p$. В случая, нека $\alpha = a^p b^p c^p$.

(∀) Разглеждаме произволно разбиване $xyuvw = \alpha$, за което $|yuv| \leq p$ и $1 \leq |yv|$. Знаем, че поне една от y и v не е празната дума.

(∃) Ще посочим конкретно $i \in \mathbb{N}$, за което $xy^i uv^i w \notin L$.

Обърнете внимание, че тук i съществено зависи от вида на разбиването.

- 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.

□

Задача 4.14. Докажете, че езикът $L = \{\, \alpha \cdot \alpha \mid \alpha \in \{a,b\}^\star \,\}$ не е безконтекстен.

Упътване. Разгледайте $\omega = a^p b^p a^p b^p$. □

Задача 4.15. Докажете, че езикът $L = \{\alpha\beta\alpha^{\mathtt{rev}} \mid \alpha, \beta \in \{a,b\}^\star \ \&\ |\alpha| = |\beta|\}$ не е безконтекстен.

Упътване.

- Защо не става ако разгледаме думата $\alpha = a^p b^p a^p$?
- Защо не става ако разгледаме думата $\alpha = a^p b^p a^{2p} b^p a^p$?
- Разгледайте $\alpha = a^p b^p a^p b^p b^p a^p$. Покачване с повече от p би трябвало да свърши работа.

□

Задача 4.16. Докажете, че езикът $L = \{\alpha\beta\alpha \mid \alpha, \beta \in \{a,b\}^\star \ \&\ |\alpha| \geq 1\}$ не е безконтекстен.

Упътване.

- Защо не става с $\omega = a^p b a^p b$?
- Защо не става с $\omega = a b^p a b^p$?
- Пробвайте с $\omega = a^p b^p a^p b^p$.

□

Задача 4.17. Докажете, че езикът $L = \{\alpha\sharp\beta \mid \alpha \text{ е подниз на } \beta\}$ не е безконтекстен.

Упътване.

- Защо не става ако вземем $\omega = a^p \sharp a^p$?
- Защо не става ако вземем $\omega = a^p b \sharp a^p b$?
- Разгледайте $\omega = a^p b^p \sharp a^p b^p$.

□

Задача 4.18. Вярно ли е, че следните езици са безконтекстни:

а) $L = \{\alpha\sharp\beta \mid \alpha, \beta \in \{0,1\}^\star \ \&\ \overline{\alpha}_{(2)} + 1 = \overline{\beta}_{(2)}\}$;

б) $L = \{\alpha\sharp\beta^{\mathtt{rev}} \mid \alpha, \beta \in \{0,1\}^\star \ \&\ \overline{\alpha}_{(2)} + 1 = \overline{\beta}_{(2)}\}$?

4.8 Алгоритми

Тук ще докажем, че съществуват *полиномиални* алгоритми, които за дадена безконтекстна граматика G проверяват:

- дали $\mathcal{L}(G) = \emptyset$;
- дали $|\mathcal{L}(G)| = \infty$;
- дали $|\mathcal{L}(G)| < \infty$;
- по дадена дума α дали $\alpha \in \mathcal{L}(G)$.

Нека приемем, че дължината на граматика G е

$$|G| \stackrel{\text{деф}}{=} |V| + |\Sigma| + \sum_{(A,\alpha)\in R} |A\alpha|.$$

4.8.1 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: [10, стр. 88], [9, стр. 256].

$$S \overset{\star}{\lhd}_G \lambda A \rho \overset{\star}{\lhd}_G \alpha,$$

където $\alpha \in \Sigma^\star$, а $\lambda, \rho \in (V \cup \Sigma)^\star$. 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 4.4. Нека е дадена безконтекстната граматика G и $\mathcal{L}(G) \neq \emptyset$. Съществува алгоритъм, който намира граматика G', за която $\mathcal{L}(G) = \mathcal{L}(G')$, със свойството, че за всяка променлива $A' \in V'$, $\mathcal{L}_{G'}(A') \neq \emptyset$.

Упътване. Целта ни е да намерим множеството `Gen` от променливи, които генерират думи, т.е. търсим множеството

$$\texttt{Gen} \overset{\text{деф}}{=} \{A \in V \mid \mathcal{L}_G(A) \neq \emptyset\}.$$

За целта ще построим редица от множества `Gen[n]` по следния начин:

Всяка итерация на алгоритъма отнема $\mathcal{O}(|G|)$ време. Следователно, изпълнението на целия алгоритъм отнема $\mathcal{O}(|G|^2)$ време.

- `Gen[0]` $\overset{\text{деф}}{=} \emptyset$;
- `Gen[n+1]` $\overset{\text{деф}}{=} \{A \in V \mid (\exists \alpha \in (\Sigma \cup \texttt{Gen[n]})^\star)[A \to_G \alpha]\}$.
- Спираме, когато стигнем до такова n, за което `Gen[n]` $=$ `Gen[n+1]`. Лесно се съобразява, че $(\forall k \geq n)[$ `Gen[n]` $=$ `Gen[k]` $]$.

Трябва да докажем, че `Gen` $=$ `Gen[n]`. Това ще направим като докажем, че за всяко k, е изпълнена еквивалентността: ✍ Докажете сами!

$$A \in \texttt{Gen[k]} \;\Leftrightarrow\; (\exists \alpha \in \Sigma^\star)[A \overset{\leq k}{\lhd}_G \alpha],$$

или с други думи,

$$A \in \texttt{Gen[k]} \;\Leftrightarrow\; \mathcal{L}^k_G(A) \neq \emptyset.$$

Дефинираме G' като $V' =$ `Gen` и правилата на G' са само тези правила на G, в които участват променливи от V' и букви от Σ. □

Следствие 4.6. Съществува *полиномиален* алгоритъм, който определя за всяка безконтекстна граматика G дали $\mathcal{L}(G) = \emptyset$.

Доказателство. Прилагаме алгоритъма от *Лема 4.4* и намираме множеството от променливи V'. Тогава $\mathcal{L}(G) = \emptyset$ точно тогава, когато $S \notin V'$. □

Лема 4.5. Съществува алгоритъм, който по дадена безконтекстна граматика $G = \langle V, \Sigma, R, S\rangle$, намира $G' = \langle V', \Sigma, R', S\rangle$, $\mathcal{L}(G') = \mathcal{L}(G)$, със свойството, че за всяко $B \in V'$ съществуват $\lambda, \rho \in (V' \cup \Sigma)^\star$, за които $S \overset{\star}{\lhd}_{G'} \lambda B \rho$, т.е. всяка променлива в G' е достижима от началната променлива S.

Упътване. Нашата цел е да намерим множеството

$$\texttt{Reach} \stackrel{\text{деф}}{=} \{B \in V \mid S \overset{\star}{\lhd} \lambda B \rho, \text{ за някои } \lambda, \rho \in (\Sigma \cup V)^\star\}.$$

Новата граматика G' ще има променливи $V' \stackrel{\text{деф}}{=} \texttt{Reach}$, като правилата на граматика G' са същите като тези на G, но ограничени до V'. Лесно се доказава, че $\mathcal{L}(G) = \mathcal{L}(G')$. Остава да намерим множеството Reach. За да постигнем тази цел, ще започнем да строим множествата $\texttt{Reach[i]} \subseteq V$ по следния начин:

$$\begin{aligned}\texttt{Reach[0]} &\stackrel{\text{деф}}{=} \{S\}\\ \texttt{Reach[i+1]} &\stackrel{\text{деф}}{=} \{B \in V \mid (\exists A \in \texttt{Reach[i]})(\exists \lambda, \rho \in (V \cup \Sigma)^\star)[\, A \to_G \lambda B \rho \,]\}\\ &\quad \cup \texttt{Reach[i]}.\end{aligned}$$

Докажете, че за всяко i е изпълнено:

$$\texttt{Reach[i]} = \{B \in V \mid S \overset{\leq i}{\lhd} \lambda B \rho, \text{ за някои } \lambda, \rho \in (\Sigma \cup V)^\star\}.$$

✍ Защо сме сигурни, че ще достигнем такава стъпка?

Спираме да строим тези множества, когато достигнем до стъпка n, за която

$$\texttt{Reach[n]} = \texttt{Reach[n+1]}.$$

Съобразете, че за това n е изпълнено, че $\texttt{Reach[n]} = \texttt{Reach}$, т.е.

$$\texttt{Reach[n]} = \{B \in V \mid S \overset{\star}{\lhd} \lambda B \rho, \text{ за някои } \lambda, \rho \in (\Sigma \cup V)^\star\}.$$

□

Пример 4.7. Да разгледаме безконтекстната граматика G с правила:

$$\begin{aligned}S &\to AB \mid aA\\ A &\to a \mid aAa\\ B &\to SB \mid BC\\ C &\to \varepsilon \mid cC.\end{aligned}$$

Първо да намерим променливите, от които се извеждат думи.

- $\texttt{Gen[0]} = \emptyset$;
- $\texttt{Gen[1]} = \{A, C\}$, защото $A \to a$ и $C \to \varepsilon$;
- $\texttt{Gen[2]} = \{A, C, S\}$, защото $S \to aA$;
- не можем да добавим B към Gen[3], следователно $\texttt{Gen[3]} = \texttt{Gen[2]}$.

Получаваме граматиката G':

$$\begin{aligned}S &\to aA\\ A &\to a \mid aAa\\ C &\to \varepsilon \mid cC.\end{aligned}$$

Сега премахваме променливите и буквите, които не са достижими от началната промелива S.

- $\texttt{Reach[0]} = \{S\}$;
- $\texttt{Reach[1]} = \{S, A\}$, защото $S \to aAa$;
- $\texttt{Reach[2]} = \{S, A\}$.

Така получаваме граматиката G'' със следните правила:

$$\begin{aligned}S &\to aA\\ A &\to a \mid aAa.\end{aligned}$$

Теорема 4.3. За всяка безконтекстна граматика G, за която $\mathcal{L}(G) \neq \emptyset$, съществува еквивалентна на нея безконтекстна граматика G' *без безполезни правила*. Освен това, граматиката G' може да се намери за полиномиално време.

Упътване. Прилагаме върху G първо процедурата от *Лема 4.4* и след това върху резултата прилагаме процедурата от *Лема 4.5*. □

Забележка. Защо е важна последователността на прилагане? Например, да разгледаме граматиката

$$\begin{aligned}S &\to AB \mid ab\\ A &\to aA \mid \varepsilon\\ B &\to bB.\end{aligned}$$

Ако първо приложим процедурата от *Лема 4.5*, то нищо няма да премахнем, защото A и B са достижими от S. След това, ако приложим процедурата от *Лема 4.4*, то ще премахнем всички правила, в които участва променливата B, т.е. получаваме граматика G' с правила $S \to ab$ и $A \to aA \mid \varepsilon$. Така A става недостижима променлива от S и трябва пак да приложим процедурата от *Лема 4.5*. Алгоритъмът описан тук е с квадратична сложност. Има и линеен такъв. Вижте [9, стр. 296].

Задача 4.19. Проверете дали $\mathcal{L}(G) = \emptyset$, където правилата на G са:

$$S \to AS \mid BC$$
$$A \to a \mid BA \mid SB$$
$$B \to b \mid BC \mid AB$$
$$C \to CB \mid SC \mid AS.$$

Премахване на ε-правила

Лема 4.6. Съществува *експоненциален* алгоритъм, такъв че превръща всяка безконтекстна граматика G в безконтекстна граматика G' без правила от вида $A \to \varepsilon$, и $\mathcal{L}(G') = \mathcal{L}(G) \setminus \{\varepsilon\}$.

Упътване. За да премахнем правилата от вида $A \to \varepsilon$ от граматиката, първо трябва да намерим множеството от променливи

$$\texttt{E} = \{A \in V \mid A \overset{\star}{\lhd} \varepsilon\}.$$

За да направим това, следваме процедурата:

1) Първо строим множествата `E[n]` по следния начин:

   На всяка стъпка трябва да обходим цялата граматика, следователно всяка итерация отнема $\mathcal{O}(|G|)$ време.

   - $\texttt{E[0]} \overset{\text{деф}}{=} \emptyset$;
   - $\texttt{E[n+1]} \overset{\text{деф}}{=} \{B \in V \mid (\exists \alpha \in \texttt{E[n]}^\star)[B \to_G \alpha]\}$.

     Обърнете внимание, че имаме $\texttt{E[n]}^\star$, а не просто `E[n]`. Също така да напомним, че $\emptyset^\star = \{\varepsilon\}$.

   - Докажете, че за всяко n е изпълнено, че
     $$\texttt{E[n]} = \{A \in V \mid A \overset{\leq n}{\lhd} \varepsilon\}.$$
   - Спираме на първото n, за което $\texttt{E[n]} = \texttt{E[n+1]}$. Лесно се съобразява, че за това n е изпълнено, че
     $$\texttt{E[n]} = \{A \in V \mid A \overset{\star}{\lhd} \varepsilon\}.$$

   Намираме множеството E за време $\mathcal{O}(|G|^2)$.

2) Дефинираме новата граматика G' така, че G' съдържа правило $A \to X_0 \cdots X_n$ точно тогава, когато $A \to \alpha_0 X_0 \alpha_1 X_1 \cdots X_n \alpha_{n+1}$ е правило в G за някои думи $\alpha_i \in \texttt{E}^\star$.

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

   Лесно се съобразява, че $\mathcal{L}(G') = \mathcal{L}(G) \setminus \{\varepsilon\}$.

   ✍ Докажете сами!

□

Пример 4.8. Нека е дадена граматиката G с правила

$$S \to D$$
$$D \to AD \mid b$$
$$A \to AB \mid BC \mid a$$
$$B \to AA \mid UC$$
$$C \to \varepsilon \mid CA \mid a$$
$$U \to \varepsilon \mid aUb.$$

Тогава можем да намерим множеството от променливи E, от които се извежда ε в граматиката G като започнем да намираме неговите апроксимации:

- $\texttt{E[0]} \overset{\text{деф}}{=} \emptyset$;
- $\texttt{E[1]} = \{C, U\}$, защото $C \to \varepsilon$ и $U \to \varepsilon$;
- $\texttt{E[2]} = \{C, U, B\}$, защото $B \to UC$;
- $\texttt{E[3]} = \{C, U, B, A\}$, защото $A \to BC$;

- $\texttt{E[4]} = \texttt{E[3]}$.

- Оттук е ясно, че $(\forall n \geq 3)[\texttt{E[n]} = \texttt{E[3]}]$. Заключаваме, че

$$\texttt{E} = \{X \in V \mid X \overset{\star}{\lhd} \varepsilon\} = \{C, U, B, A\}.$$

Това означава, че $\varepsilon \notin \mathcal{L}(G)$, защото $S \notin \texttt{E}$. Граматиката G' без ε-правила, за която $\mathcal{L}(G') = \mathcal{L}(G)$ има следните правила:

$$\begin{aligned}
&S \to D\\
&D \to AD \mid D \mid b\\
&A \to A \mid B \mid C \mid AB \mid BC \mid a\\
&B \to A \mid C \mid AA \mid U \mid UC\\
&C \to C \mid A \mid CA \mid a\\
&U \to aUb \mid ab.
\end{aligned}$$

Премахване на преименуващи правила

В [9, стр. 263] преименуващите правила се наричат *unit productions*.

Едно правило в граматиката G се нарича **преименуващо**, ако е от вида $A \to_G B$. Нека е дадена граматика $G = \langle V, \Sigma, R, S\rangle$. Ще построим еквивалентна граматика G' без преименуващи правила. В началото нека в G' да добавим всички правила от G, които не са преименуващи. След това, за всяка променлива A, за която $A \overset{\star}{\lhd} B$, ако $B \to \alpha$ е правило в граматиката G, което не е преименуващо, то добавяме към G' правилото $A \to \alpha$.

Лема 4.7. Нека G е безконтекстна граматика без ε-правила. Съществува *полиномиален* алгоритъм, такъв че превръща всяка безконтекстна граматика G в безконтекстна граматика G' без преименуващи правила и $\mathcal{L}(G') = \mathcal{L}(G)$.

Упътване. Първо намираме множеството от двойки променливи от следния вид:

$$\texttt{Unit} = \{(A, B) \in V \times V \mid A \overset{\star}{\lhd} B\}.$$

Отново, за да намерим Unit, ние строим неговите апроксимации Unit[n], където:

$$\begin{aligned}
&\texttt{Unit[0]} \overset{\text{деф}}{=} \{(A, A) \mid A \in V\}\\
&\texttt{Unit[n+1]} \overset{\text{деф}}{=} \texttt{Unit[n]} \cup \{(A, B) \mid (\exists C)[A \to_G C \;\&\; (C, B) \in \texttt{Unit[n]}]\}.
\end{aligned}$$

Лесно се съобразява, че имаме свойството:

$$\texttt{Unit[n]} = \{(A, B) \in V \times V \mid A \overset{\leq n}{\lhd} B\}.$$

Unit има големина $\mathcal{O}(|G|^2)$.

Спираме на първото n, за което $\texttt{Unit[n]} = \texttt{Unit[n+1]}$. Тогава $\texttt{Unit} = \texttt{Unit[n]}$.

Нека $R'_0 \overset{\text{деф}}{=} R \setminus (V \times V)$ са правилата на R, които не са преименуващи. Тогава правилата на новата граматика G' ще бъдат:

$$R' \overset{\text{деф}}{=} \{\, (A, \alpha) \in V \times (V \cup \Sigma)^\star \mid (\exists B)[(A, B) \in \texttt{Unit} \;\&\; (B, \alpha) \in R'_0] \,\}.$$

Обърнете внимание, че понеже $(A, A) \in \texttt{Unit}$ за всяко $A \in V$, то $R'_0 \subseteq R'$. Лесно се съобразява, че $\mathcal{L}(G) = \mathcal{L}(G')$. □

Пример 4.9. Нека е дадена граматиката G с правила

$$\begin{aligned}
&S \to B \mid CC \mid b\\
&A \to S \mid SB\\
&B \to C \mid BC\\
&C \to AB \mid a \mid b.
\end{aligned}$$

Да намерим първо множеството Unit.

$$\begin{aligned}
\texttt{Unit[0]} &= \{(S,S),(A,A),(B,B),(C,C)\};\\
\texttt{Unit[1]} &= \{(S,S),(S,B),(A,A),(A,S),\\
&\quad (B,B),(B,C),(C,C)\};\\
\texttt{Unit[2]} &= \{(S,S),(S,B),(S,C),(B,B),\\
&\quad (B,C),(C,C),(A,A),(A,S),\\
&\quad (A,B)\};\\
\texttt{Unit[3]} &= \{(S,S),(S,B),(S,C),(B,B),\\
&\quad (B,C),(C,C),(A,A),(A,S),\\
&\quad (A,B),(A,C)\};\\
\texttt{Unit[4]} &= \{(S,S),(S,B),(S,C),(B,B),\\
&\quad (B,C),(C,C),(A,A),(A,S),\\
&\quad (A,B),(A,C)\}.
\end{aligned}$$

Получихме, че `Unit[3]` = `Unit[4]`. Оттук можем да заключим следното:

$$A \overset{\star}{\lhd} A, B, S, C$$
$$B \overset{\star}{\lhd} B, C$$
$$S \overset{\star}{\lhd} S, B, C$$
$$C \overset{\star}{\lhd} C.$$

Първо добавяме към R' правилата, които не са преименуващи, а именно:

$$A \to SB$$
$$B \to BC$$
$$C \to AB \mid a \mid b$$
$$S \to CC \mid b.$$

- Понеже имаме,че $A \overset{\star}{\lhd} B, S, C$, то добавяме към R' правилата:
$$A \to BC \mid AB \mid a \mid b \mid CC.$$
- Понеже имаме,че $B \overset{\star}{\lhd}_G C$, то добавяме към R' правилата:
$$B \to AB \mid a \mid b.$$
- Понеже имаме, че $S \overset{\star}{\lhd}_G B, C$, то добавяме към R' правилата:
$$S \to BC \mid AB \mid a \mid b.$$

Накрая получаваме, че граматиката G' има правила

$$S \to BC \mid AB \mid CC \mid a \mid b$$
$$A \to BS \mid BC \mid AB \mid a \mid b \mid CC$$
$$B \to AB \mid a \mid b \mid BC$$
$$C \to AB \mid a \mid b.$$

Задача 4.20. Премахнете преименуващите правила от граматиката G, като запазите езика, ако G има следните правила:

$$S \to C \mid CC \mid b$$
$$A \to B$$
$$B \to S \mid C \mid BC$$
$$C \to a \mid AB;$$

Премахване на дългите правила

Едно правило се нарича дълго, ако е от вида $A \to \beta$, където $|\beta| \geq 3$. Да разгледаме едно дълго правило в граматиката от вида $A \to X_1X_2 \cdots X_k$, където $k \geq 3$. За да получим еквивалентна граматика без това дълго правило, добавяме нови променливи $A_1, \ldots, A_{k-2}$, и правила

Новата граматика ще има дължина $\mathcal{O}(|G|)$.

$$A \to X_1A_1,\ A_1 \to X_2A_2, \ldots,\ A_{k-2} \to X_{k-1}X_k.$$

Задача 4.21. Нека е дадена граматиката $G = \langle \{S, A, B, C\}, \{a, b\}, S, R\rangle$. Използвайте обща конструкция, за да премахнете „дългите" правила от G като при това получите безконтестна граматика G_1 с език $\mathcal{L}(G) = \mathcal{L}(G_1)$, където правилата на граматиката са:

$$S \to CC \mid b$$
$$A \to BSB \mid a$$
$$B \to ba \mid BC$$
$$C \to BaSA \mid a \mid b.$$

4.8.2 Нормална Форма на Чомски

Една безконтекстна граматика е в **нормална форма на Чомски** (НФЧ), ако всяко правило е от вида

$$A \to BC \text{ и } A \to a,$$

където A, B, C са произволни променливи и a е произволна буква.

Теорема 4.4. Всеки безконтекстен език L, който не съдържа ε, се поражда от безконтекстна граматика в нормална форма на Чомски.

Доказателство. Нека имаме безконтекстна граматика G, за която $L = \mathcal{L}(G)$. Ще построим безконтекстна граматика G' в нормална форма на Чомски, $L = \mathcal{L}(G')$. Следваме следната процедура:

- Премахваме дългите правила. Това можем да направим за време $\mathcal{O}(|G|)$ като новата граматика ще има дължина $\mathcal{O}(|G|)$.
- Премахваме ε-правилата. Това можем да направим за време $\mathcal{O}(|G|^2)$ като новата граматика ще има дължина $\mathcal{O}(|G|)$.

Редът на стъпките е важен. Трябва преди това сме премахнали дългите правила. Вижте [9, стр. 296].

- Премахваме преименуващите правила. Това можем да направим за време $\mathcal{O}(|G|^2)$ като новата граматика ще има дължина $\mathcal{O}(|G|^2)$.
- 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. Това можем да направим за време $\mathcal{O}(|G|)$ и новата граматика ще има дължина $\mathcal{O}(|G|)$.
- Ако искаме ε да бъде в езика на граматиката, то добавяме нова начална променлива S_0 и правило $S_0 \to_G \varepsilon$ и правилата $S_0 \to \alpha$ за $S \to \alpha$.

□

Теорема 4.5. При дадена безконтекстна граматика G, можем да намерим еквивалентна на нея граматика G' в нормална форма на Чомски за време $\mathcal{O}(|G|^2)$, като получената граматика е с дължина $\mathcal{O}(|G|^2)$.

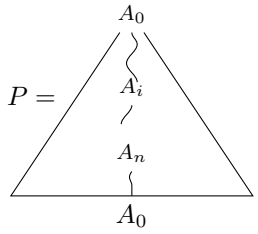
Теорема 4.6. Съществува *полиномиален* алгоритъм, който определя по дадена безконтекстна граматика G дали $\mathcal{L}(G)$ е безкраен език.

[10, стр. 137]. Важно е, че алгоритмът е полиномиален. ✍ Защо от *Лема 4.3* имаме експоненциален алгоритъм за проверка дали езикът на една граматика е безкраен?

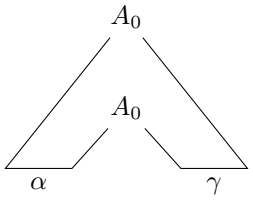
Доказателство. Нека е дадена една безконтекстна граматика G. Нека да разгледаме граматиката G' в НФЧ *без безполезни променливи*, за която $\mathcal{L}(G) = \mathcal{L}(G')$. От граматиката $G' = \langle V', \Sigma, S, R' \rangle$ строим граф с възли променливите от V' като за $A, B \in V'$ имаме ребро $A \to B$ в графа точно тогава, когато съществува $C \in V'$, за което $A \to_{G'} BC$ или $A \to_{G'} CB$ е правило в граматиката G. Сега ще съобразим, че ако в получения граф имаме цикъл, то $\mathcal{L}(G') = \infty$. Да разгледаме един такъв цикъл в графа:

$$A_0 \to A_1 \to A_2 \to \cdots \to A_n \to A_0.$$

Понеже граматиката е в нормална форма на Чомски, то съществуват думи $\lambda, \rho \in (V \cup \Sigma)^\star$, за които $A_0 \overset{n+1}{\lhd} \lambda A_0 \rho$ и $|\lambda\rho| = n + 1$



(а) $A_0 \overset{n+1}{\lhd} \lambda A_0 \rho$, където $\lambda, \rho \in (V \cup \Sigma)^\star$.



(б) $A_0 \overset{\geq n+1}{\lhd} \alpha A_0 \gamma$, където $\alpha, \gamma \in \Sigma^\star$.

Понеже в G няма безполезни променливи, то $\lambda \overset{\star}{\lhd} \alpha$, където $\alpha \in \Sigma^\star$ и $|\lambda| \leq |\alpha|$ и $\rho \overset{\star}{\lhd} \gamma$, където $\gamma \in \Sigma^\star$ и $|\rho| \leq |\gamma|$. Оттук заключаваме, че $A_0 \overset{\star}{\lhd} \alpha A_0 \gamma$ и $|\alpha\gamma| \geq 1$. Понеже A_0 не е безполезна променлива, то $A_0 \overset{\star}{\lhd} \beta$, за някоя дума $\beta \in \Sigma^\star$. Сега комбинираме *Твърдение 4.5* и *Твърдение 4.6* за да получим следния извод:

$$\frac{A_0 \overset{\star}{\lhd} \alpha A_0 \gamma \qquad A_0 \overset{\star}{\lhd} \beta \qquad i \in \mathbb{N}}{A_0 \overset{\star}{\lhd} \alpha^i \beta \gamma^i}$$

Понеже $|\alpha\beta| \geq 1$, заключаваме, че $\mathcal{L}(G)$ е безкраен език. Така доказахме, че ако в графът има цикли, то $\mathcal{L}(G)$ е безкраен език.

За обратната посока, нека в графът да няма цикли. Да разгледаме една променлива A, от която най-дългият път има k на брой възли. От принципа на Дирихле знаем, че $k \leq |V|$. Това означава, че ако $A \overset{\star}{\lhd} \alpha$, за някоя $\alpha \in \Sigma^\star$, то $A \overset{\leq k}{\lhd} \alpha$ и понеже граматиката е в нормална форма на Чомски, то $|\alpha| \leq 2^{k-1}$. Оттук следва, че всички думи, които се извеждат от променливата A са най-много 2^{k-1} на брой. Заключаваме, че ако в графът няма цикли, то $\mathcal{L}(G)$ е краен. □

Забележка. Ясно е, че ако изискваме една граматика да бъде в НФЧ, то $\varepsilon \notin \mathcal{L}(G)$. Ако все пак искаме ε да бъде в езика на граматиката G, то можем да позволим от началната променлива S да имаме правилото $S \to \varepsilon$, но тогава забраняваме S да се среща в дясна страна на правило.

За да направим това, по дадена граматика G в НФЧ, трябва да добавим нова начална променлива S' и да добавим правилата $S' \to \alpha$, където $S \to \alpha$ е правило в граматиката G. Така ще получим нова граматика G', за която $\mathcal{L}(G') = \mathcal{L}(G) \cup \{\varepsilon\}$.

Да напомним, че всички от изброените тук проблеми са алгоритмично разрешими за регулярни езици. Вижда се, че за безконтекстни езици нещата не са толкова прости.

Проблемът за празнота: Да. *Следствие 4.6* ни казва, че съществува алгоритъм, който разрешава проблема за празнота за безконтекстни езици.

Проблемът за безкрайност: Да. От *Теорема 4.6* знаем, че този проблем е алгоритмично разрешим.

Проблемът за универсалност: Не. Чак в *Теорема* **??** ще видим, че не съществува алгоритъм, който да разрешава този проблем за безконтекстни езици.

Проблемът за принадлежност: Да. Според *Теорема 4.7* проблемът за принадлежност е алгоритмично разрешим за безконтекстни езици.

Проблемът за включване: Не. Да фиксираме безконтекстна граматика G_0, за която $\mathcal{L}(G_0) = \Sigma^\star$. Съобразете, че за произволна безконтекстна граматика G имаме еквивалентността:

$$\mathcal{L}(G) = \Sigma^\star \;\Leftrightarrow\; \mathcal{L}(G_0) \subseteq \mathcal{L}(G).$$

Това означава, че ако допуснем, че проблемът за включване на безконтекстни граматики е алгоритмично разрешим, то проблемът за универсалност за безконтекстни граматики също би бил разрешим. Но това е противоречие. Заключаваме, че проблемът за включване не е разрешим.

Проблемът за равенство: Не. Аналогично се съобразява, че този проблем също не е алгоритмично разрешим, защото ако този проблем беше разрешим, то и проблемът за универсалност щеше да е разрешим.

4.8.3 Проблемът за принадлежност

Ние знаем, че съществува доста прост алгоритъм, който разрешава проблема за принадлежност за автоматни езици. Сега ще видим, че този проблем е алгоритмично разрешим и за безконтекстни езици.

Теорема 4.7. Съществува *полиномиален* алгоритъм относно дължината на входната дума, който проверява дали дадена дума принадлежни на граматиката G.

Преди да достигнем до алгоритъм за проверка за принадлежност на дума към език на граматика, ще направим някои наблюдения.

Първо, понеже входната граматика е в нормална форма на Чомски, то имаме следното полезно свойство:

$A \overset{\star}{\lhd} \alpha$ точно тогава, когато съществува правило $A \to BC$ и думата α може да се разбие като $\alpha = \beta\gamma$, където $B \overset{\star}{\lhd} \beta$ и $C \overset{\star}{\lhd} \gamma$.

Оттук веднага се вижда, че за да разберем дали една дума е в езика на граматиката, то трябва да знаем от кои променливи се извеждат различни по-къси части на тази дума.

За да бъдем по-конкретни, нека разгледаме входна дума $\alpha = a_0 a_1 \cdots a_{n-1}$ и да разгледаме множествата

$$\texttt{V[i][j]} = \{A \in V \mid A \overset{\star}{\lhd}_G a_i a_{i+1} \cdots a_j\},$$

където $0 \leq i \leq j < n$. Тогава $\alpha \in \mathcal{L}(G)$ точно тогава, когато $S \in \texttt{V[0][n-1]}$. Как можем да намерим тези множества?

Можем да пренапишем горното свойство по следния начин:

$A \in \texttt{V[i][j]}$ точно тогава, когато съществува правило $A \to BC$, съществува индекс k, $i \leq k < j$, за който $B \in \texttt{V[i][k]}$ и $C \in \texttt{V[k+1][j]}$.

Понеже по-дълги части от думата α се описват чрез по-къси части, то ще опишем как намираме множествата `V[i][j]` като постепенно увеличаваме разликата $j - i$.

- Нека $i = j$. Тогава за всяко $i < n$ е ясно, че:

$$\texttt{V[i][i]} = \{A \in V \mid A \to_G a_i\}.$$

- Нека $i < j$. Тогава за всяко k, където $i \leq k < j$, ако намерим правило $A \to BC$, за което $B \in \texttt{V[i][k]}$ и $C \in \texttt{V[k+1][j]}$, то добавяме A към множеството `V[i][j]`.

  Множествата `V[i][k]` и `V[k+1][j]` са изчислени на предишни стъпки, защото $k - i < j - i$ и $j - (k+1) < j - i$.

Това е алгоритъм на Cocke, Younger и Kasami (CYK), които откриват идеята за този алгоритъм независимо един от друг. Той е пример за динамично програмиране [13, стр. 192], [20, стр. 141]. При вход дума α, алгоритъмът работи за време $\mathcal{O}(|\alpha|^3)$.

Algorithm 5 Проблемът за принадлежност за безконтекстни езици

```
1:  procedure BELONG(G, α)
2:      n := len(α)                                   ▷ Вход дума α = a_0 a_1 ⋯ a_{n-1}
3:      for all i < n do
4:          V[i][i] := {A ∈ V | A →_G α[i]}
5:          for all j := i + 1, ..., n − 1 do
6:              V[i][j] := ∅
7:      for all 1 ≤ s < n do                          ▷ Дължина на интервала
8:          for all i < n − s do                      ▷ Начало на интервала
9:              for all i ≤ k < i + s do              ▷ Разделяне на интервала
10:                 if ∃(A, BC) ∈ R & B ∈ V[i][k] & C ∈ V[k+1][i+s] then
11:                     V[i][i+s] := V[i][i+s] ∪ {A}
12:     if S ∈ V[0][n-1] then
13:         return True                               ▷ Има извод на думата от S
14:     else
15:         return False
```

Лема 4.8. Нека е дадена безконтекстната граматика G в нормална форма на Чомски и думата α. Всеки път, точно преди да се изпълни ред (7) от програмата описана в Алгоритъм 5, е изпълнено, че за всяко $i < n - s$,

$$\texttt{V[i][i+s]} = \{A \in V \mid A \overset{\star}{\lhd}_G a_i a_{i+1} \cdots a_{i+s}\}.$$

Доказателство. Пълна индукция по s. За $s = 0$ е ясно, защото от ред (4) и от факта, че граматиката е в нормална форма на Чомски следва, че за всяко $i < n$,

$$\texttt{V[i][i]} = \{A \in V \mid A \to_G \alpha[i]\} = \{A \in V \mid A \overset{\star}{\lhd}_G a_i\}.$$

Сега ще докажем твърдението за $s > 0$, т.е. ще докажем, че за всяко $i < n - s$ е изпълнено, че:

$$\texttt{V[i][i+s]} = \{A \in V \mid A \overset{\star}{\lhd}_G a_i a_{i+1} \cdots a_{i+s}\}.$$

Да разгледаме двете посоки на това равенство.

(⊆) Нека $A \in \texttt{V[i][i+s]}$. Единствената стъпка на алгоритъма, при която може да сме добавили променливата A към множеството `V[i][i+s]` е ред (11). Тогава имаме, че съществува k, за което $i \le k < i + s$, и са изпълнени условията:

- $B \in \texttt{V[i][k]}$, като $k = i + \overbrace{(k - i)}^{s'}$;
- $C \in \texttt{V[k+1][i+s]}$, като $i + s = (k + 1) + \underbrace{(i + s - k - 1)}_{s''}$;
- $A \to BC$ е правило в граматиката G.

Понеже $s' < s$ и $s'' < s$, от индукционното предположение имаме следното:

$$B \in \texttt{V[i][k]} \overset{\text{(И.П.)}}{\Longrightarrow} B \overset{\star}{\lhd} a_i a_{i+1} \cdots a_k$$
$$C \in \texttt{V[k+1][i+2]} \overset{\text{(И.П.)}}{\Longrightarrow} C \overset{\star}{\lhd} a_{k+1} \cdots a_{i+s}.$$

Заключаваме веднага, че $A \overset{\star}{\lhd}_G a_i a_{i+1} \cdots a_{i+s}$.

(⊇) Нека $A \overset{\star}{\lhd} a_i a_{i+1} \cdots a_{i+s}$ и да разгледаме първото правило, което сме приложили в този извод. Понеже G е в нормална форма на Чомски, правилото е от вида $A \to_G BC$ и тогава съществува k, за което $i \le k < i + s$, и

$$B \overset{\star}{\lhd} a_i \cdots a_k$$
$$C \overset{\star}{\lhd} a_{k+1} \cdots a_{i+s}.$$

От **(И.П.)** получаваме, че $B \in \texttt{V[i][k]}$ и $C \in \texttt{V[k+1][i+s]}$. Тогава от ред (11) на алгоритъма е ясно, че $A \in \texttt{V[i][i+s]}$.

□

Пример 4.10. Нека е дадена безконтекстната граматика G в нормална форма на Чомски с правила

$$S \to a \mid AB \mid AC$$
$$A \to a$$
$$B \to b$$
$$C \to SB \mid AS.$$

Ще приложим CYK алгоритъма за да проверим дали $aaabb \in \mathcal{L}(G)$.

- За $s = 0$ имаме, че:
  - $\texttt{V[0][0]} = \texttt{V[1][1]} = \texttt{V[2][2]} = \{S, A\}$;
  - $\texttt{V[3][3]} = \texttt{V[4][4]} = \{B\}$.
- За $s = 1$ имаме, че:
  - $\texttt{V[0][1]} = \texttt{V[1][2]} = \{C\}$;
  - $\texttt{V[2][3]} = \{S, C\}$;
  - $\texttt{V[3][4]} = \emptyset$.
- За $s = 2$ имаме, че:
  - $\texttt{V[0][2]} = \{S\} \cup \emptyset$;
  - $\texttt{V[1][3]} = \{S, C\} \cup \emptyset$;
  - $\texttt{V[2][4]} = \emptyset \cup \{C\}$.
- За $s = 3$ имаме, че:
  - $\texttt{V[0][3]} = \{S, C\} \cup \emptyset \cup \emptyset = \{S, C\}$;
  - $\texttt{V[1][4]} = \{S\} \cup \emptyset \cup \{C\} = \{S, C\}$.
- За $s = 4$ имаме, че:
  - $\texttt{V[0][4]} = \{S, C\} \cup \emptyset \cup \emptyset \cup \{C\} = \{S, C\}$.

Понеже $S \in \texttt{V[0][4]}$, то $aaabb \in \mathcal{L}(G)$.

Сега да видим защо $aba \notin \mathcal{L}(G)$.

- За $s = 0$ имаме, че:
  - $\texttt{V[0][0]} = \texttt{V[2][2]} = \{S, A\}$.
  - $\texttt{V[1][1]} = \{B\}$.
- За $s = 1$ имаме, че:
  - $\texttt{V[0][1]} = \{S, C\}$.
  - $\texttt{V[1][2]} = \emptyset$.
- За $s = 2$ имаме, че:
  - $\texttt{V[0][2]} = \{C\}$.

Понеже $S \notin \texttt{V[0][2]}$, то $aba \notin \mathcal{L}(G)$.

4.9 Недетерминирани стекови автомати

Недетерминиран стеков автомат е седморка от вида

На англ. *Push-down automaton*. В този курс няма да разглеждаме детерминирани стекови автомати. Когато кажем стеков автомат, ще имаме предвид недетерминиран стеков автомат. Означаваме

$$\Sigma_\varepsilon \stackrel{\text{деф}}{=} \Sigma \cup \{\varepsilon\}$$
$$\Gamma^{\leq 2} \stackrel{\text{деф}}{=} \{\varepsilon\} \cup \Gamma \cup \Gamma^2.$$

$$P = \langle Q, \Sigma, \Gamma, \sharp, \Delta, q_{\texttt{start}}, q_{\texttt{accept}} \rangle,$$

където:

- Q е крайно множество от състояния;
- Σ е крайна входна азбука;
- Γ е крайна стекова азбука;
- $\sharp \in \Gamma$ е символ за дъно на стека;
- $\Delta : Q \times \Sigma_\varepsilon \times \Gamma \to \mathscr{P}(Q \times \Gamma^{\leq 2})$ е функция на преходите;
- $q_{\texttt{start}} \in Q$ е начално състояние;
- $q_{\texttt{accept}} \in Q$ е заключителното състояние.

Обикновено се взима Δ функцията да има сигнатура $\Delta : Q \times \Sigma_\varepsilon \times \Gamma \to \mathscr{P}(Q \times \Gamma^\star)$. Дефиницията на стеков автомат има много вариации, всички еквивалентни помежду си [7, стр. 131]. На англ. моментно описание - instanteneous description.

Конфигурация (или моментно описание) 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 $\vdash_P$ 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.

$$\frac{\Delta(q, x, A) \ni (p, \beta) \qquad \alpha \in \Sigma^\star \qquad \rho \in \Gamma^\star}{(q, x\alpha, A\rho) \vdash_P (p, \alpha, \beta\rho)}$$

Фигура 4.10: Правило за извършване на една стъпка в изчисление на стеков автомат

От дефиницията на релацията $\vdash_P$ веднага получаваме свойството, че можем да добавяме произволни остатъци към входната дума и стека.

Твърдение 4.9. $$\frac{(q, \alpha, \gamma) \vdash_P (q', \alpha', \gamma') \qquad \beta \in \Sigma^\star \qquad \delta \in \Gamma^\star}{(q, \alpha\beta, \gamma\delta) \vdash_P (q', \alpha'\beta, \gamma'\delta)}$$

Сега можем дефинираме бинарната релация $\vdash_P^\ell$ над $Q \times \Sigma^\star \times \Gamma^\star$, която ни казва, че конфигурацията κ се променя до конфигурация κ' след изчисление от ℓ стъпки на стековия автомат:

$$\frac{}{\kappa \vdash_P^0 \kappa} \text{ (рефлексивност)} \qquad \frac{\kappa \vdash_P \kappa'' \qquad \kappa'' \vdash_P^\ell \kappa'}{\kappa \vdash_P^{\ell+1} \kappa'} \text{ (транзитивност)}$$

Фигура 4.11: Дефиниция на релацията $\vdash_P^\ell$

Задача 4.22. Докажете, че за произволни $\ell_1 \geq 0$ и $\ell_2 \geq 0$,

$$\frac{\kappa_0 \vdash_P^{\ell_1} \kappa_1 \qquad \kappa_1 \vdash_P^{\ell_2} \kappa_2}{\kappa_0 \vdash_P^{\ell_1+\ell_2} \kappa_2}$$

Упътване. Индукция по ℓ_1. □

Ако не се интересуваме от точния брой стъпки в едно изчисление, ще използваме релацията $\vdash_P^\star$, която е дефинирата по следния начин:

$$\kappa \vdash_P^\star \kappa' \overset{\text{деф}}{\Leftrightarrow} (\exists \ell \in \mathbb{N})[\kappa \vdash_P^\ell \kappa'].$$

С други думи, бинарната релация $\vdash_P^\star$ е рефлексивното и транзитивното затваряне на бинарната релация $\vdash_P$.

Определение 4.4. Езикът $\mathcal{L}(P)$, който се разпознава от стековия автомат P дефинираме по следния начин:

$$\mathcal{L}(P) \overset{\text{деф}}{=} \{\, \omega \in \Sigma^\star \mid (q_{\texttt{start}}, \omega, \sharp) \vdash_P^\star (q_{\texttt{accept}}, \varepsilon, \varepsilon) \,\}.$$

Възможно е да се дадат и други еквивалентни дефиниции на $\mathcal{L}(P)$. В повечето обикновени учебници се прави разлика между разпознаване на дума с финално състояние и разпознаване на дума с празен стек. За нашите цели това е излишно.

Твърдение 4.10. За всяко $\ell \geq 0$ е изпълнен извода:

$$\frac{(q, \alpha, \gamma) \vdash^\ell (p, \varepsilon, \varepsilon) \qquad \beta \in \Sigma^\star \qquad \rho \in \Gamma^\star}{(q, \alpha\beta, \gamma\rho) \vdash^\ell (p, \beta, \rho)}$$

Доказателство.

- Случаят, когато $\ell = 0$ е ясен, защото тогава $q = p$, $\alpha = \varepsilon$ и $\gamma = \varepsilon$.
- Нека $\ell > 0$. Тогава имаме следния извод:

$$\frac{(q, \alpha, \gamma) \vdash (q', \alpha', \gamma') \qquad (q', \alpha', \gamma') \vdash^{\ell-1} (p, \varepsilon, \varepsilon)}{(q, \alpha, \gamma) \vdash^\ell (p, \varepsilon, \varepsilon)}$$

Тогава за произволни $\beta \in \Sigma^\star$ и $\rho \in \Gamma^\star$ получаваме следното:

$$\frac{\dfrac{(q, \alpha, \gamma) \vdash (q', \alpha', \gamma')}{(q, \alpha\beta, \gamma\rho) \vdash (q', \alpha'\beta, \gamma'\rho)} \text{ (Тв. 4.9)} \qquad \dfrac{(q', \alpha', \gamma') \vdash^{\ell-1} (p, \varepsilon, \varepsilon)}{(q', \alpha'\beta, \gamma'\rho) \vdash^{\ell-1} (p, \beta, \rho)}}{(q, \alpha\beta, \gamma\rho) \vdash^\ell (p, \beta, \rho)} \text{ (И.П.)}$$

□

Сега ще разгледаме две важни свойства на релацията $\vdash_P^\ell$, които ще ни бъдат полезни по-нататък. Първото свойство ни казва как можем да „слепваме" две изчисления в едно голямо изчисление.

Твърдение 4.11. За произволни $\ell_1 \geq 0$ и $\ell_2 \geq 0$ е изпълнено:

$$\frac{(p, \alpha, A) \vdash_P^{\ell_1} (r, \varepsilon, \varepsilon) \qquad (r, \beta, B) \vdash_P^{\ell_2} (q, \varepsilon, \varepsilon)}{(p, \alpha\beta, AB) \vdash_P^{\ell_1+\ell_2} (q, \varepsilon, \varepsilon)}$$

Упътване. Достатъчно е да проследим извода:

$$\text{(Тв. 4.10)}\ \frac{\dfrac{(p,\alpha,A)\vdash_P^{\ell_1}(r,\varepsilon,\varepsilon)}{(p,\alpha\beta,AB)\vdash_P^{\ell_1}(r,\beta,B)}\qquad (r,\beta,B)\vdash_P^{\ell_2}(q,\varepsilon,\varepsilon)}{(p,\alpha\beta,AB)\vdash_P^{\ell_1+\ell_2}(q,\varepsilon,\varepsilon)}$$

□

Второто свойство ни казва кога можем да разбием едно изчисление на по-малки части. Това свойство ще бъде важно по-нататък.

Твърдение 4.12. Нека $(q,\alpha,A\rho)\vdash_P^{\ell}(p,\varepsilon,\varepsilon)$. Тогава съществуват ℓ_1, ℓ_2, за които $\ell_1+\ell_2=\ell$, състояние r и разбиване на α като $\alpha=\alpha_1\alpha_2$, така че:

- $(q,\alpha_1,A)\vdash_P^{\ell_1}(r,\varepsilon,\varepsilon)$;
- $(r,\alpha_2,\rho)\vdash_P^{\ell_2}(p,\varepsilon,\varepsilon)$.

Упътване. Пълна индукция по ℓ. Ясно е, че трябва $\ell\geq 1$. За $\ell=1$, то е ясно, че трябва $\rho=\varepsilon$ и единствената стъпка в изчислението е направена заради $\Delta(q,x,A)\ni(r,\varepsilon)$. Тогава е ясно, че $\alpha=x$ и $r=p$. Получаваме, че в този случай изчислението се разбива не две части по тривиален начин:

- $(q,\underbrace{x}_{\alpha_1},A)\vdash_P^1(r,\varepsilon,\varepsilon)$;
- $(r,\underbrace{\varepsilon}_{\alpha_2},\varepsilon)\vdash_P^0(r,\varepsilon,\varepsilon)$.

Нека $\ell>1$. Трябва да разгледаме няколко случая.

- Ако първата стъпка в изчислението е направена заради $\Delta(q,x,A)\ni(q',BC)$, където $x\in\Sigma_\varepsilon$, то $(q',\alpha',BC\rho)\vdash_P^{\ell-1}(p,\varepsilon,\varepsilon)$, където $\alpha=x\alpha'$. Тогава от **(И.П.)** получаваме следното:

  (1) $(q',\beta_1,B)\vdash_P^{\ell_1'}(r',\varepsilon,\varepsilon)$;

  (2) $(r',\beta_2,C\rho)\vdash_P^{\ell_2'}(p,\varepsilon,\varepsilon)$,

  за някое състояние r', където $\ell_1'+\ell_2'=\ell-1$ и $\beta_1\beta_2=\alpha'$. Понеже $\ell_2'<\ell$, отново от **(И.П.)** получаваме, че изчислението $(r',\beta_2,C\rho)\vdash_P^{\ell_2'}(p,\varepsilon,\varepsilon)$ може да се разбие така:

  (3) $(r',\gamma_1,C)\vdash_P^{k_1}(r'',\varepsilon,\varepsilon)$;

  (4) $(r'',\gamma_2,\rho)\vdash_P^{k_2}(p,\varepsilon,\varepsilon)$,

  за някое състояние r'', където $k_1+k_2=\ell_2'$ и $\gamma_1\gamma_2=\beta_2$. Можем да комбинираме първата стъпка от изчислението с частичните изчисления (1) и (3) и да заключим следното:

  – $(q,\underbrace{x\beta_1\gamma_1}_{\alpha_1},A)\vdash_P^{1+\ell_1'+k_1}(r'',\varepsilon,\varepsilon)$;

  – $(r'',\underbrace{\gamma_2}_{\alpha_2},\rho)\vdash_P^{k_2}(p,\varepsilon,\varepsilon)$.

- Вече свършихме тежката работа. Нека все пак за пълнота да разгледаме и другите случаи. Ако първата стъпка в изчислението е направена заради $\Delta(q,x,A)\ni(q',B)$, то $(q',\alpha',B\rho)\vdash_P^{\ell-1}(p,\varepsilon,\varepsilon)$, където $\alpha=x\alpha'$. Тук ведната от **(И.П.)** получаваме, че можем да разбием изчислението така:

- $(q', \alpha'_1, B) \vdash_P^{\ell'_1} (r', \varepsilon, \varepsilon)$;
- $(r', \alpha'_2, \rho) \vdash_P^{\ell'_2} (p, \varepsilon, \varepsilon)$,

за някое състояние r', където $\alpha'_1 \alpha'_2 = \alpha'$. Оттук заключаваме, че:

- $(q, \underbrace{x\alpha'_1}_{\alpha_1}, A) \vdash_P^{1+\ell'_1} (r', \vdash, \varepsilon)$;
- $(r', \underbrace{\alpha'_2}_{\alpha_2}, \rho) \vdash_P^{\ell'_2} (p, \varepsilon, \varepsilon)$.

• Ако първата стъпка в изчислението е направена заради $\Delta(q, x, A) \ni (q', \varepsilon)$, то нека $\alpha = x\alpha'$. Тогава всичко е ясно, защото можем да разбием изчислението така:

- $(q, \underbrace{x}_{\alpha_1}, A) \vdash_P^1 (q', \varepsilon, \varepsilon)$;
- $(q', \underbrace{\alpha'}_{\alpha_2}, \rho) \vdash_P^{\ell-1} (p, \varepsilon, \varepsilon)$.

□

Примери

Пример 4.11. За езика $L = \{a^n b^n \mid n \in \mathbb{N}\}$, да разгледаме стековия автомат

$$P = \langle Q, \Sigma, \Gamma, \sharp, \Delta, q_{\texttt{start}}, q_{\texttt{accept}} \rangle,$$

където:

- $Q \stackrel{\text{деф}}{=} \{q, p, f\}$;
- $q_{\texttt{start}} \stackrel{\text{деф}}{=} q$ и $q_{\texttt{accept}} \stackrel{\text{деф}}{=} f$;
- $\Sigma \stackrel{\text{деф}}{=} \{a, b\}$ и $\Gamma \stackrel{\text{деф}}{=} \{\sharp, a\}$;
- Релацията на преходите Δ има следната дефиниция:
  (1) $\Delta(q, a, \sharp) \stackrel{\text{деф}}{=} \{(q, a\sharp)\}$;
  (2) $\Delta(q, a, a) \stackrel{\text{деф}}{=} \{(q, aa)\}$; // трупаме a-та в стека
  (3) $\Delta(q, \varepsilon, \sharp) \stackrel{\text{деф}}{=} \{(f, \varepsilon)\}$; // трябва да разпознаем и думата ε
  (4) $\Delta(q, b, a) \stackrel{\text{деф}}{=} \{(p, \varepsilon)\}$; // Започваме да четем само b-та
  (5) $\Delta(p, b, a) \stackrel{\text{деф}}{=} \{(p, \varepsilon)\}$; // Чистим a-тата от стека
  (6) $\Delta(p, \varepsilon, \sharp) \stackrel{\text{деф}}{=} \{(f, \varepsilon)\}$.
  (7) За всички останали тройки (r, x, y), нека $\Delta(r, x, y) \stackrel{\text{деф}}{=} \emptyset$.

Да видим как думата a^2b^2 се разпознава от стековия автомат P:

$$\begin{aligned}
(q, a^2b^2, \sharp) &\vdash_P (q, ab^2, a\sharp) && \text{// правило (1)}\\
&\vdash_P (q, b^2, aa\sharp) && \text{// правило (2)}\\
&\vdash_P (p, b, a\sharp) && \text{// правило (4)}\\
&\vdash_P (p, \varepsilon, \sharp) && \text{// правило (5)}\\
&\vdash_P (f, \varepsilon, \varepsilon) && \text{// правило (6)}
\end{aligned}$$

Получихме, че $(q_{\texttt{start}}, a^2b^2, \sharp) \vdash_P^\star (q_{\texttt{accept}}, \varepsilon, \varepsilon)$, откъдето следва, че $a^2b^2 \in \mathcal{L}(P)$.

а) Докажете с индукция по n, че за всяко естествено число n са изпълнени свойствата:

$$(q, a^n\beta, \sharp) \vdash_P^n (q, \beta, a^n\sharp) \tag{4.12}$$
$$(p, b^n, a^n\sharp) \vdash_P^n (p, \varepsilon, \sharp). \tag{4.13}$$

Заключете, че $L \subseteq \mathcal{L}(P)$.

б) Докажете, че с индукция по n, че за всяко естествено число n са изпълнени свойствата:

$$(q, \alpha\beta, \sharp) \vdash_P^n (q, \beta, \gamma\sharp) \implies \alpha = \gamma = a^n \tag{4.14}$$
$$(p, \beta, \gamma\sharp) \vdash_P^n (p, \varepsilon, \sharp) \implies \beta = b^n \,\&\, \gamma = a^n. \tag{4.15}$$

Оттук заключете, че $\mathcal{L}(P) \subseteq L$.

Пример 4.12. Езикът $L = \{\, \omega\omega^{\texttt{rev}} \mid \omega \in \{a, b\}^\star \,\}$ се разпознава от стеков автомат

$$P = \langle Q, \Sigma, \Gamma, \sharp, \Delta, q_{\texttt{start}}, q_{\texttt{accept}} \rangle,$$

където:

- $Q \stackrel{\text{деф}}{=} \{q, p, f\}$ и $q_{\texttt{start}} \stackrel{\text{деф}}{=} q$, $q_{\texttt{accept}} \stackrel{\text{деф}}{=} f$;
- $\Sigma \stackrel{\text{деф}}{=} \{a, b\}$, $\Gamma \stackrel{\text{деф}}{=} \{a, b, \sharp\}$;
- Функцията на преходите Δ има следната дефиниция:

За всички липсващи твойки в дефиницията на Δ приемаме, че Δ връща $\emptyset$

  (1) $\Delta(q, x, \sharp) \stackrel{\text{деф}}{=} \{(q, x\sharp)\}$, където $x \in \{a, b\}$;
  (2) $\Delta(q, \varepsilon, \sharp) \stackrel{\text{деф}}{=} \{(q, \varepsilon)\}$;
  (3) $\Delta(q, x, x) \stackrel{\text{деф}}{=} \{(q, xx), (p, \varepsilon)\}$, където $x \in \{a, b\}$;
  (4) $\Delta(q, a, b) \stackrel{\text{деф}}{=} \{(q, ab)\}$;
  (5) $\Delta(q, b, a) \stackrel{\text{деф}}{=} \{(q, ba)\}$;
  (6) $\Delta(p, x, x) \stackrel{\text{деф}}{=} \{(p, \varepsilon)\}$, където $x \in \{a, b\}$;
  (7) $\Delta(p, \varepsilon, \sharp) \stackrel{\text{деф}}{=} \{(f, \varepsilon)\}$;

Основното наблюдение, което трябва да направим за да разберем конструкцията на автомата е, че всяка дума от вида $\omega\omega^{\texttt{rev}}$ може да се запише като $\omega_1 aa\omega_1^{\texttt{rev}}$ или $\omega_1 bb\omega_1^{\texttt{rev}}$. Да видим защо P разпознава думата $abaaba$ с празен стек. Започваме по следния начин:

$$\begin{aligned}
(q, abaaba, \sharp) &\vdash_P (q, baaba, a\sharp) && \text{// правило (1)}\\
&\vdash_P (q, aaba, ba\sharp) && \text{// правило (5)}\\
&\vdash_P (q, aba, aba\sharp). && \text{// правило (4)}
\end{aligned}$$

Сега можем да направим два избора как да продължим. Състоянието p служи за маркер, което ни казва, че вече сме започнали да четем $\omega^{\texttt{rev}}$. Поради тази причина, продължаваме така:

$$\begin{aligned}
(q, aba, aba\sharp) &\vdash_P (p, ba, ba\sharp) && \text{// правило (3)}\\
&\vdash_P (p, a, a\sharp) && \text{// правило (6)}\\
&\vdash_P (p, \varepsilon, \sharp) && \text{// правило (6)}\\
&\vdash_P (f, \varepsilon, \varepsilon). && \text{// правило (7)}
\end{aligned}$$

Да проиграем още един пример. Да видим защо думата aba не се извежда от автомата.

$$\begin{aligned}(q, aba, \sharp) &\vdash_P (q, ba, a\sharp) && \text{// правило } (1)\\ &\vdash_P (q, a, ba\sharp) && \text{// правило } (5)\\ &\vdash_P (q, \varepsilon, aba\sharp). && \text{// правило } (4)\end{aligned}$$

От последното моментно описание на автомата нямаме нито един преход, следователно думата aba не се разпознава от P.

✍ Докажете, че $L = \mathcal{L}(P)$!

За (4.16) приложете индукция по дължината на думата α. За индукционната стъпка разгледайте α като $\alpha = \alpha' x$.

а) Докажете с индукция по n, за всяко естествено число n са изпълнени свойствата:

$$|\alpha| = n \implies (q, \alpha\beta, \sharp) \vdash_P^n (q, \beta, \alpha^{\mathtt{rev}}\sharp) \tag{4.16}$$
$$|\beta| = n \implies (p, \beta, \beta\sharp) \vdash_P^n (p, \varepsilon, \sharp). \tag{4.17}$$

Оттук заключете, че $L \subseteq \mathcal{L}(P)$.

б) Докажете с индукця по n, че за всяко естествено число n са изпълнени свойствата:

$$(q, \alpha\beta, \sharp) \vdash_P^n (q, \beta, \gamma\sharp) \implies \gamma = \alpha^{\mathtt{rev}} \,\&\, |\alpha| = n \tag{4.18}$$
$$(p, \beta, \gamma\sharp) \vdash_P^n (p, \varepsilon, \sharp) \implies \gamma = \beta \,\&\, |\beta| = n. \tag{4.19}$$

Оттук заключете, че $\mathcal{L}(P) \subseteq L$.

От *Задача 4.25* знаем, че този език е безконтекстен.

Пример 4.13. Езикът $L = \{\, \omega \in \{a, b\}^\star \mid |\omega|_a = |\omega|_b \}$ се разпознава от стековия автомат

$$P = \langle Q, \Sigma, \Gamma, \sharp, \Delta, q_{\mathtt{start}}, q_{\mathtt{accept}} \rangle,$$

където:

- $Q = \{q, f\}$;
- $\Sigma = \{a, b\}$;
- $\Gamma = \{a, b, \sharp\}$;
- $q_{\mathtt{start}} = q$ и $q_{\mathtt{accept}} = f$;
- Можем да дефинираме релацията на преходите Δ по следния начин:
  - (1) $\Delta(q, \varepsilon, \sharp) = \{(f, \varepsilon)\}$;
  - (2) $\Delta(q, x, \sharp) = \{(q, x\sharp)\}$, където $x \in \{a, b\}$;
  - (3) $\Delta(q, x, x) = \{(q, xx)\}$, където $x \in \{a, b\}$;
  - (4) $\Delta(q, a, b) = \{(q, \varepsilon)\}$;
  - (5) $\Delta(q, b, a) = \{(q, \varepsilon)\}$.

Да видим защо думата $abbbaa \in \mathcal{L}(P)$.

$$\begin{aligned}(q, abbbaa, \sharp) &\vdash_P (q, bbbaa,\ a\sharp) && \text{// правило } (2)\\ &\vdash_P (q, bbaa,\ \sharp) && \text{// правило } (5)\\ &\vdash_P (q, baa,\ b\sharp) && \text{// правило } (2)\\ &\vdash_P (q, aa, bb\sharp) && \text{// правило } (3)\\ &\vdash_P (q, a,\ b\sharp) && \text{// правило } (4)\\ &\vdash_P (q, \varepsilon,\ \sharp) && \text{// правило } (4)\\ &\vdash_P (f, \varepsilon,\ \varepsilon). && \text{// правило } (1)\end{aligned}$$

а) Докажете с пълна индукция по дължината на думата γ, че за произволна дума $\gamma \in \{a, b\}^\star$, е изпълнено, че:

$$(\forall n)[a^n\gamma \in L \implies (q, \gamma, a^n\sharp) \vdash_P^\star (q, \varepsilon, \sharp)] \tag{4.20}$$
$$(\forall n)[b^n\gamma \in L \implies (q, \gamma, b^n\sharp) \vdash_P^\star (q, \varepsilon, \sharp)]. \tag{4.21}$$

Ще докажем едновременно *Свойство 4.20* и *Свойство 4.21* с пълна индукция по дължината на думата γ.

Да разгледаме произволна дума γ. Случаят, когато $|\gamma| = 0$ е тривиален. Нека $|\gamma| > 0$ и да приемем, че $a^n\gamma \in L$. Ясно е, че можем да представим γ като $\gamma = a^k b\gamma'$, за някое k.

- Ако $n + k = 0$, то $a^n\gamma = b\gamma' \in L$ и прилагаме **(И.П.)** за *Свойство 4.21* с думата γ' и получаваме, че:

$$\begin{aligned}(q, \overbrace{b\gamma'}^{\gamma}, \sharp) &\vdash (q, \gamma', b\sharp) && \text{// правило (2)}\\ &\vdash^\star (q, \varepsilon, \sharp) && \text{// \textbf{(И.П.)}}\end{aligned}$$

- Ако $n + k > 0$, то $a^{n+k-1}\gamma' \in L$ и прилагаме **(И.П.)** за *Свойство 4.20* с думата γ' и получаваме, че:

$$\begin{aligned}(q, \overbrace{a^k b\gamma'}^{\gamma}, a^n\sharp) &\vdash_P^\star (q, b\gamma', a^{n+k}\sharp) && \text{// правило (3)}\\ &\vdash_P^\star (q, \gamma', a^{n+k-1}\sharp) && \text{// правило (5)}\\ &\vdash_P^\star (q, \varepsilon, \sharp). && \text{// \textbf{(И.П.)}}\end{aligned}$$

Аналогично се доказва и *Свойство 4.21*. Оттук заключете, че $L \subseteq \mathcal{L}(P)$.

б) Докажете с пълна индукция по дължината на думата γ, че за произволна дума $\gamma \in \{a,b\}^\star$ е изпълнено, че:

$$(\forall n)[(q,\gamma,a^n\sharp) \vdash_P^\star (q,\varepsilon,\sharp) \implies a^n\gamma \in L] \tag{4.22}$$
$$(\forall n)[(q,\gamma,b^n\sharp) \vdash_P^\star (q,\varepsilon,\sharp) \implies b^n\gamma \in L]. \tag{4.23}$$

Нека имаме изчислението $(q,\gamma,a^n\sharp) \vdash_P^\star (q,\varepsilon,\sharp)$. Да представим думата γ като $\gamma = a^k b\gamma'$.

- Ако $n+k=0$, то можем да разбием изчислението по следния начин:

$$\begin{aligned}(q,b\gamma',\sharp) &\vdash_P (q,\gamma',b\sharp)\\ &\vdash^\star (q,\varepsilon,\sharp).\end{aligned}$$

  Тогава от **(И.П.)** за *Свойство 4.23* следва, че $a^n\gamma = b\gamma' \in L$.

- Ако $n+k>0$, то можем да разбием изчислението по следния начин:

$$\begin{aligned}(q,a^kb\gamma',a^n\sharp) &\vdash_P^\star (q,b\gamma',a^{n+k}\sharp)\\ &\vdash_P (q,\gamma',a^{n+k-1}\sharp)\\ &\vdash^\star (q,\varepsilon,\sharp).\end{aligned}$$

  Тогава от **(И.П.)** за *Свойство 4.22* следва, че $a^{n+k-1}\gamma' \in L$, но оттук веднага получаваме, че $a^na^kb\gamma' \in L$.

Аналогично се доказва и *Свойство 4.23*. Оттук заключете, че $\mathcal{L}(P) \subseteq L$.

Пример 4.14. Езикът $L = \{\, \omega \in \{a,b\}^\star \mid \omega$ е балансирана дума $\}$ се разпознава от стековия автомат

$$P = \langle Q, \Sigma, \Gamma, \sharp, \Delta, q_{\texttt{start}}, q_{\texttt{accept}}\rangle,$$

От *Задача 4.26* знаем, че този език е безконтекстен.

където:

- $Q = \{q,f\}$;
- $q_{\texttt{start}} = q$ и $q_{\texttt{accept}} = f$;
- $\Sigma = \{a,b\}$ и $\Gamma = \{a,\sharp\}$;
- Можем да дефинираме релацията на преходите Δ по следния начин:
  (1) $\Delta(q,\varepsilon,\sharp) = \{(f,\varepsilon)\}$;
  (2) $\Delta(q,a,\sharp) = \{(q,a\sharp)\}$;
  (3) $\Delta(q,a,a) = \{(q,aa)\}$;
  (4) $\Delta(q,b,a) = \{(q,\varepsilon)\}$;

✍ Докажете, че $L = \mathcal{L}(P)$!

(а) Докажете, че за произволно естествено число n и произволна дума $\beta \in \{a,b\}^\star$, е изпълнено, че:

$$a^n\beta \in L \implies (q,\beta,a^n\sharp) \vdash_P^\star (q,\varepsilon,\sharp).$$

Индукция по дължината на думата β.

Оттук заключете, че $L \subseteq \mathcal{L}(P)$.

(б) Докажете, че за произволно естествено число n и произволна дума $\beta \in \{a,b\}^\star$, е изпълнено, че:

$$(q,\beta,a^n\sharp) \vdash_P^\star (q,\varepsilon,\sharp) \implies a^n\beta \in L.$$

Индукция по броя на стъпките в изчислението на стековия автомат.

Оттук заключете, че $\mathcal{L}(P) \subseteq L$.

4.10 Теорема за еквивалентност

Доказателството на лемата следва до голяма степен [16, стр. 136].

Лема 4.9. За всяка безконтекстна граматика G, съществува стеков автомат P, такъв че $\mathcal{L}(G) = \mathcal{L}(P)$.

Тук е важно да използваме най-ляв извод в граматика.

Доказателство. Нека е дадена безконтекстната граматика $G = \langle V, \Sigma, R, S\rangle$ в нормална форма на Чомски. Нашата цел е да построим стеков автомат

$$P = \langle Q, \Sigma, \Gamma, \sharp, \Delta, q_{\texttt{start}}, q_{\texttt{accept}}\rangle,$$

който разпознава езика $\mathcal{L}(G)$.

- $Q = \{q_{\texttt{start}}, p, q_{\texttt{accept}}\}$;
- $\Gamma = \Sigma \cup V \cup \{\sharp\}$;

Понеже граматиката е в нормална форма на Чомски, то $|\alpha| \leq 2$ и удовлетворяваме дефиницията на Δ.

- Функцията на преходите Δ за стековия автомат P дефинираме по следния начин:

  (1) $\Delta(q_{\texttt{start}}, \varepsilon, \sharp) \stackrel{\text{деф}}{=} \{(p, S\sharp)\}$;

  (2) За всяка променлива $A \in V$,

  $$\Delta(p, \varepsilon, A) \stackrel{\text{деф}}{=} \{(p, \alpha) \mid A \to_G \alpha \text{ е правило в } G\}.$$

  (3) За всяка буква $a \in \Sigma$,

  $$\Delta(p, a, a) \stackrel{\text{деф}}{=} \{(p, \varepsilon)\}.$$

  (4) $\Delta(p, \varepsilon, \sharp) \stackrel{\text{деф}}{=} \{(q_{\texttt{accept}}, \varepsilon)\}$.

Ще докажем, че за всяка променлива $A \in V$ и всяка дума $\alpha \in \Sigma^\star$, е изпълнено, че:

(а) ако $A \stackrel{\star}{\lhd} \alpha$, то $(p, \alpha, A) \vdash_P^\star (p, \varepsilon, \varepsilon)$;

(б) ако $(p, \alpha, A) \vdash_P^\star (p, \varepsilon, \varepsilon)$, то $A \stackrel{\star}{\lhd} \alpha$.

Ако приемем, че (а) и (б) са изпълнени, тогава, ако вземем $A = S$, то ще получим следната верига от еквивалентности:

$$\begin{aligned}
\alpha \in \mathcal{L}(G) &\Leftrightarrow S \stackrel{\star}{\lhd} \alpha \\
&\Leftrightarrow (p, \alpha, S) \vdash_P^\star (p, \varepsilon, \varepsilon) && \text{// от (а) и (б)} \\
&\Leftrightarrow (q_{\texttt{start}}, \alpha, \sharp) \vdash_P^\star (q_{\texttt{accept}}, \varepsilon, \varepsilon) && \text{// от деф. на } P \\
&\Leftrightarrow \alpha \in \mathcal{L}(P).
\end{aligned}$$

Сега преминаваме към доказателствата на двете твърдения.

Доказателството на (а) ще проведем с *пълна индукция* по дължината ℓ на извода $A \stackrel{\ell}{\lhd} \alpha$. За целта, да разгледаме свойството

$$R(\ell) \stackrel{\text{деф}}{=} (\forall A \in V)(\forall \alpha \in \Sigma^\star)[\, A \stackrel{\ell}{\lhd} \alpha \implies (p, \alpha, A) \vdash^\star (p, \varepsilon, \varepsilon)\,].$$

Ще докажем, че $(\forall \ell \in \mathbb{N})R(\ell)$.

Нека приемем, че $A \stackrel{\ell}{\lhd} \alpha$. За да докажем, че $(p, \alpha, A) \vdash^\star (p, \varepsilon, \varepsilon)$, да видим как този извод е получен. Първо, нека имаме следния извод:

$$\frac{A \to_G a}{A \stackrel{1}{\lhd} \underbrace{a}_{\alpha}}$$

Тогава според конструкцията на стековия автомат P, имаме изчислението:

$$\dfrac{\dfrac{\dfrac{A \to_G a}{\Delta(p,\varepsilon,A) \ni (p,a)}}{(p,a,A) \vdash_P (p,a,a)} \quad \dfrac{\Delta(p,a,a) \ni (p,\varepsilon)}{(p,a,a) \vdash_P (p,\varepsilon,\varepsilon)}}{(p,a,A) \vdash_P^2 (p,\varepsilon,\varepsilon)}$$

Нека сега изводът да се разбие така:

$$\text{правило (1)} \ \dfrac{A \to_G BC \qquad B \overset{\ell_1}{\lhd} \alpha_1 \qquad C \overset{\ell_2}{\lhd} \alpha_2}{A \overset{\ell}{\lhd} \underbrace{\alpha_1\alpha_2}_{\alpha}} \ (\ell = \sup\{\ell_1,\ell_2\}+1)$$

Понеже $\ell_1 < \ell$ и $\ell_2 < \ell$, от **(И.П.)** за $R(\ell_1)$ и $R(\ell_2)$ получаваме изчислението:

$$\dfrac{\dfrac{\dfrac{A \to_G BC}{\Delta(p,\varepsilon,A) \ni (p,BC)}}{(p,\alpha_1\alpha_2,A) \vdash_P (p,\alpha_1\alpha_2,BC)} \quad \dfrac{\dfrac{B \overset{\ell_1}{\lhd} \alpha_1}{(p,\alpha_1,B) \vdash_P^\star (p,\varepsilon,\varepsilon)} \quad \dfrac{C \overset{\ell_2}{\lhd} \alpha_2}{(p,\alpha_2,C) \vdash_P^\star (p,\varepsilon,\varepsilon)}}{(p,\alpha_1\alpha_2,BC) \vdash_P^\star (p,\varepsilon,\varepsilon)}}{(p,\underbrace{\alpha_1\alpha_2}_{\alpha},A) \vdash_P^\star (p,\varepsilon,\varepsilon)}$$

Доказателството на (б) отново ще проведем с *пълна индукция* по броя на стъпките ℓ в изчислението на стековия автомат. За целта, да разгледаме свойството

$$R(\ell) \overset{\text{деф}}{=} (\forall A \in V)(\forall \alpha \in \Sigma^\star)[\,(p,\alpha,A) \vdash^\ell (p,\varepsilon,\varepsilon) \implies A \overset{\star}{\lhd} \alpha\,].$$

Нека $(p,\alpha,A) \vdash_P^\ell (p,\varepsilon,\varepsilon)$. Имаме два варианта за първата стъпка в това изчисление.
Нека да започнем с интересния случай, който е следния:

$$\dfrac{\dfrac{\Delta(p,\varepsilon,A) \ni (p,BC)}{(p,\alpha,A) \vdash_P (p,\alpha,BC)} \quad (p,\alpha,BC) \vdash_P^{\ell-1} (p,\varepsilon,\varepsilon)}{(p,\alpha,A) \vdash_P^\ell (p,\varepsilon,\varepsilon)}$$

Знаем от *Твърдение 4.12*, че можем да разбием изчислението $(p,\alpha,BC) \vdash_P^{\ell-1} (p,\varepsilon,\varepsilon)$ по следния начин:

- $(p,\alpha_1,B) \vdash_P^{\ell_1} (p,\varepsilon,\varepsilon)$;
- $(p,\alpha_2,C) \vdash_P^{\ell_2} (p,\varepsilon,\varepsilon)$,

където $\alpha_1\alpha_2 = \alpha$ и $\ell_1 + \ell_2 = \ell - 1$. Понеже $\ell_1 < \ell$ и $\ell_2 < \ell$, от **(И.П.)** за $R(\ell_1)$ и $R(\ell_2)$ получаваме импликациите:

$$(p,\alpha_1,B) \vdash_P^{\ell_1} (p,\varepsilon,\varepsilon) \overset{\textbf{(И.П.)}}{\implies} B \overset{\star}{\lhd} \alpha_1$$
$$(p,\alpha_2,C) \vdash_P^{\ell_2} (p,\varepsilon,\varepsilon) \overset{\textbf{(И.П.)}}{\implies} C \overset{\star}{\lhd} \alpha_2.$$

Сега обединяваме всичко, което имаме и получаваме извода:

$$\dfrac{A \to_G BC \qquad \dfrac{(p,\alpha_1,B) \vdash_P^{\ell_1} (p,\varepsilon,\varepsilon)}{B \overset{\star}{\lhd} \alpha_1} \qquad \dfrac{(p,\alpha_2,C) \vdash_P^{\ell_2} (p,\varepsilon,\varepsilon)}{C \overset{\star}{\lhd} \alpha_2}}{A \overset{\star}{\lhd} \alpha}$$

Сега да разгледаме случая, когато за някое $a \in \Sigma$ имаме следното:

$$\frac{\Delta(p,\varepsilon,A) \ni (p,a)}{(p,\alpha,A) \vdash_P (p,\alpha,a) \qquad (p,\alpha,a) \vdash_P^{\ell-1} (p,\varepsilon,\varepsilon)}$$
$$(p,\alpha,A) \vdash_P^{\ell} (p,\varepsilon,\varepsilon)$$

Според конструкцията на стековия автомат трябва да имаме $\ell = 2$ и $\alpha = a$, защото това е единствения начин да имаме изчислението $(p,\alpha,a) \vdash_P^{\ell-1} (p,\varepsilon,\varepsilon)$ в нашия стеков автомат. Тук всичко е ясно, защото щом $\Delta(p,\varepsilon,A) \ni (p,a)$, то $A \to_G a$ и тогава $A \overset{\star}{\lhd} \alpha$. □

Пример 4.15. Нека е дадена безконтекстната граматика G с правила

$$S \to AS \mid BS \mid \varepsilon$$
$$A \to aA \mid a$$
$$B \to Bb \mid b.$$

Ще построим стеков автомат $P = \langle Q, \Sigma, \Gamma, \sharp, \Delta, q_{\texttt{start}}, q_{\texttt{accept}} \rangle$, такъв че $\mathcal{L}(P) = \mathcal{L}(G)$.

- $\Sigma = \{a, b\}$;
- $\Gamma = \{A, S, B, a, b, \sharp\}$;
- $Q = \{q_{\texttt{start}}, q, q_{\texttt{accept}}\}$;
- Дефинираме релацията на преходите, следвайки конструкцията от *Теорема 4.8*:
  - $\Delta(q_{\texttt{start}}, \varepsilon, \sharp) = \{(q, S\sharp)\}$;
  - $\Delta(q, \varepsilon, S) = \{(q, AS), (q, BS), (q, \varepsilon)\}$;
  - $\Delta(q, \varepsilon, A) = \{(q, aA), (q, a)\}$;
  - $\Delta(q, \varepsilon, B) = \{(q, Bb), (q, b)\}$;
  - $\Delta(q, x, x) = \{(q, \varepsilon)\}$, където $x \in \{a, b\}$;
  - $\Delta(q, \varepsilon, \sharp) = \{(q_{\texttt{accept}}, \varepsilon)\}$;

Лема 4.10. За всеки стеков автомат P, съществува безконтекстна граматика G, такава че $\mathcal{L}(P) = \mathcal{L}(G)$.

Забележка. Преди да преминем към доказателството, нека да разгледаме конструкция на безконтекстна граматика G по даден ДКА $\mathcal{A}$.

- Променливите на G са $[q, p]$, за всеки $q, p \in Q_{\mathcal{A}}$.
- Имаме нужда и от начална променлива S в граматика G и правилата $S \to_G [q_{\texttt{start}}, f]$, за всяко $f \in F_{\mathcal{A}}$.
- Другите правила на граматиката G са следните: $[q, p] \to_G a[q', p]$, където $q' = \delta_{\mathcal{A}}(q, a)$.

За да се убедим, че $\mathcal{L}(\mathcal{A}) = \mathcal{L}(G)$ е достатъчно да докажем следната еквивалентност:

$$[q, p] \overset{\star}{\lhd}_G \alpha \text{ точно тогава, когато } \delta^{\star}_{\mathcal{A}}(q, \alpha) = p. \tag{4.24}$$

Доказателство. Нека е даден стековия автомат

$$P = \langle Q, \Sigma, \Gamma, \sharp, \Delta, q_{\texttt{start}}, q_{\texttt{accept}} \rangle.$$

Тук основната идея е, че искаме променливата $[q, A, p]$ да кодира информацията, че ако стековият автомат е в състоянието q, и стекът съдържа само променливата A, то когато стековият автомат премине в състояние p стекът ще се изпразни.

Ще дефинираме безконтекстна граматика G, за която $\mathcal{L}(P) = \mathcal{L}(G)$. Променливите на граматиката са

$$V = \{[q, A, p] \mid q, p \in Q \ \&\ A \in \Gamma\}.$$

Правилата на G са следните:

- Началната променлива е $S \overset{\text{деф}}{=} [q_{\texttt{start}}, \sharp, q_{\texttt{accept}}]$;
- Нека имаме $(r, BC) \in \Delta(q, x, A)$, където $x \in \Sigma_{\varepsilon}$. Тогава за всеки две състояния q' и p добавяме правилата:
$$[q, A, p] \to_G x[r, B, q'][q', C, p].$$
- Нека имаме $(r, B) \in \Delta(q, x, A)$, където $x \in \Sigma_{\varepsilon}$. Тогава за всяко състояние $p \in Q$ добавяме правилата:
$$[q, A, p] \to_G x[r, B, p].$$

- Нека имаме $(p, \varepsilon) \in \Delta(q, x, A)$, където $x \in \Sigma_\varepsilon$. Тогава добавяме правилата:

$$[q, A, p] \to_G x.$$

След като вече сме обяснили какви правила включва граматиката G, трябва да докажем, че за произволна дума $\alpha \in \Sigma^\star$, произволни състояния q и p, и произволен символ $A \in \Gamma$, е изпълнено, че:

Сравнете с (4.24).

$$[q, A, p] \overset{\star}{\lhd} \alpha \text{ точно тогава, когато } (q, \alpha, A) \vdash_P^\star (p, \varepsilon, \varepsilon). \tag{4.25}$$

$(\Rightarrow)$ Да разгледаме свойството

Да напомним, че $\Sigma_\varepsilon \overset{\text{деф}}{=} \Sigma \cup \{\varepsilon\}$.

$$R(\ell) \overset{\text{деф}}{=} (\forall p, q \in Q)(\forall A \in V)(\forall \alpha \in \Sigma^\star)[\, (q, \alpha, A) \vdash^\ell (p, \varepsilon, \varepsilon) \implies [q, A, p] \overset{\star}{\lhd} \alpha \,].$$

Ще докажем, че $(\forall \ell \in \mathbb{N}) R(\ell)$. За целта, нека $(q, \alpha, A) \vdash^\ell (p, \varepsilon, \varepsilon)$. Да видим каква е първата стъпка в това изчисление.

Нека първата стъпка е получена благодарение на $(p, \varepsilon) \in \Delta(q, x, A)$. Тогава е ясно, че $\ell = 1$, $\alpha = x$ и според конструкцията на граматиката G имаме правилото $[q, A, p] \to_G x$, откъдето веднага следва, че $[q, A, p] \overset{1}{\lhd} x$.

Нека $\alpha = x\beta$, където $x \in \Sigma_\varepsilon$.

- Ако $\Delta(q, x, A) \ni (r, B)$, то можем да разбием изчислението по следния начин:

$$\frac{\dfrac{\Delta(q, x, A) \ni (r, B)}{(q, x\beta, A) \vdash_P (r, \beta, B)} \quad (r, \beta, B) \vdash_P^{\ell-1} (p, \varepsilon, \varepsilon)}{(q, \underbrace{x\beta}_{\alpha}, A) \vdash_P^\ell (p, \varepsilon, \varepsilon)}$$

Сега можем да приложим индукционното предположение и да получим извода:

$$\text{(деф.)}\ \frac{\dfrac{\Delta(q, x, A) \ni (r, B)}{[q, A, p] \to_G x[r, B, p]} \quad x \in \Sigma_\varepsilon \qquad \dfrac{(r, \beta, B) \vdash_P^{\ell-1} (p, \varepsilon, \varepsilon)}{[r, B, p] \overset{\star}{\lhd} \beta}\ \textbf{(И.П.)}}{[q, A, p] \overset{\star}{\lhd} \underbrace{x\beta}_{\alpha}}$$

- Ако $\Delta(q, a, A) \ni (r, BC)$, то можем да разбием изчислението по следния начин:

$$\frac{\dfrac{\Delta(q, a, A) \ni (r, BC)}{(q, a\beta, A) \vdash_P (r, \beta, BC)} \quad (r, \beta, BC) \vdash_P^{\ell-1} (p, \varepsilon, \varepsilon)}{(q, \underbrace{a\beta}_{\alpha}, A) \vdash_P^\ell (p, \varepsilon, \varepsilon)}$$

Според *Твърдение 4.12*, можем да разбием β на две части като $\beta = \beta_1\beta_2$ и да получим, че:

- $(r, \beta_1, B) \vdash_P^{\ell_1} (q', \varepsilon, \varepsilon)$;
- $(q', \beta_2, C) \vdash_P^{\ell_2} (p, \varepsilon, \varepsilon)$,

за някое състояние q', където $\ell_1 + \ell_2 = \ell - 1$. Понеже $\ell_1 < \ell$ и $\ell_2 < \ell$, от **(И.П.)** за $R(\ell_1)$ и $R(\ell_2)$ имаме следното:

$$(r, \beta_1, B) \vdash_P^{\ell_1} (q', \varepsilon, \varepsilon) \overset{\textbf{(И.П.)}}{\implies} [r, B, q'] \overset{\star}{\lhd} \beta_1$$
$$(q', \beta_2, C) \vdash_P^{\ell_2} (p, \varepsilon, \varepsilon) \overset{\textbf{(И.П.)}}{\implies} [q', C, p] \overset{\star}{\lhd} \beta_2.$$

Също така имаме, че $\Delta(q, x, A) \ni (r, BC)$ и според дефиницията на стековия автомат, в граматиката имаме правилото

$$[q, A, p] \to_G x[r, B, q'][q', C, p],$$

където $x \in \Sigma_\varepsilon$. Обединявайки всичко, получаваме извода в граматиката:

$$\dfrac{\dfrac{\Delta(q,x,A) \ni (r,BC)}{[q,A,p] \to_G x[r,B,q'][q',C,p]} \quad x \in \Sigma_\varepsilon \quad \dfrac{(r,\beta_1,B) \vdash_P^{\ell_1} (q',\varepsilon,\varepsilon)}{[r,B,q'] \overset{\star}{\lhd} \beta_1} \quad \dfrac{(q',\beta_2,C) \vdash_P^{\ell_2} (p,\varepsilon,\varepsilon)}{[q',C,p] \overset{\star}{\lhd} \beta_2}}{[q,A,p] \overset{\star}{\lhd} \underbrace{x\beta_1\beta_2}_{\alpha}}$$

($\Leftarrow$) За тази посока, да разгледаме свойството

$$R(\ell) \overset{\text{деф}}{=} (\forall q,p \in Q)(\forall A \in V)(\forall \alpha \in \Sigma^\star)[\, [q,A,p] \overset{\ell}{\lhd} \alpha \implies (q,\alpha,A) \vdash^\star (p,\varepsilon,\varepsilon)\,].$$

Ще докажем с пълна индукция, че $(\forall \ell \in \mathbb{N})R(\ell)$. Нека $[q,A,p] \overset{\ell}{\lhd} \alpha$. Да видим как е получен този извод.

- Първият случай е следния:

$$\dfrac{[q,A,p] \to_G x}{[q,A,p] \overset{1}{\lhd} \underbrace{x}_{\alpha}}$$

  Според дефиницията на граматиката, правилото $[q,A,p] \to_G x$ е добавено към граматиката, защото в стековият автомат имаме $\Delta(q,x,A) \ni (p,\varepsilon)$. Тогава е ясно, че $(q,x,A) \vdash_P (p,\varepsilon,\varepsilon)$.

Тук отново е възможно $x = \varepsilon$. Това не е проблем, защото правим индукция по дължината на извода, а не по дължината на думата α.

- Второ, да приемем, че имаме следния извод:

$$\dfrac{[q,A,p] \to_G x[r,B,p] \quad x \in \Sigma_\varepsilon \quad [r,B,p] \overset{\ell-1}{\lhd} \beta}{[q,A,p] \overset{\ell}{\lhd} \underbrace{x\beta}_{\alpha}}$$

  Понеже $\ell - 1 < \ell$, от **(И.П.)** за $R(\ell-1)$ получаваме импликацията:

$$[r,B,p] \overset{\ell-1}{\lhd} \beta \overset{\textbf{(И.П.)}}{\implies} (r,\beta,B) \vdash^\star (p,\varepsilon,\varepsilon).$$

  Сега можем да приложим индукционното предположение и да сглобим изчислението:

$$\dfrac{\dfrac{\dfrac{[q,A,p] \to_G x[r,B,p]}{\Delta(q,x,A) \ni (r,B)}\,\text{(деф. на } G)}{(q,x\beta,A) \vdash_P (r,\beta,B)} \quad \dfrac{[r,B,p] \overset{\ell-1}{\lhd} \beta}{(r,\beta,B) \vdash_P^\star (p,\varepsilon,\varepsilon)}\,\textbf{(И.П.)}}{(q,\underbrace{x\beta}_{\alpha},A) \vdash_P^\star (p,\varepsilon,\varepsilon)}$$

- Трето, да разгледаме случая:

$$\dfrac{[q,A,p] \to_G x[r,B,q'][q',C,p] \quad x \in \Sigma_\varepsilon \quad [r,B,q'] \overset{\ell_1}{\lhd} \beta_2 \quad [q',C,p] \overset{\ell_2}{\lhd} \beta_2}{[q,A,p] \overset{\ell}{\lhd} \underbrace{x\beta_1\beta_2}_{\alpha}}$$

  Понеже $\ell = 1 + \sup\{\ell_1,\ell_2\}$, то $\ell_1 < \ell$ и $\ell_2 < \ell$, от **(И.П.)** за $R(\ell_1)$ и $R(\ell_2)$ получаваме импликациите:

$$[r,B,q'] \overset{\ell_1}{\lhd} \beta_1 \overset{\textbf{(И.П.)}}{\implies} (r,\beta_1,B) \vdash_P^\star (q',\varepsilon,\varepsilon)$$
$$[q',C,p] \overset{\ell_2}{\lhd} \beta_2 \overset{\textbf{(И.П.)}}{\implies} (q',\beta_2,C) \vdash_P^\star (p,\varepsilon,\varepsilon).$$

  Правилото $[q,A,p] \to_G x[r,B,q'][q',C,p]$ е добавено в граматиката, защото $\Delta(q,x,A) \ni (r,BC)$. Обединявайки всичко, което знаем, получаваме изчислението:

$$\dfrac{\dfrac{\dfrac{[q,A,p] \to_G x[r,B,q'][q',C,p]}{\Delta(q,x,A) \ni (r,BC)}}{(q,x\beta_1\beta_2,A) \vdash_P (r,\beta_1\beta_2,BC)} \quad \dfrac{\dfrac{[r,B,q'] \overset{\ell_1}{\lhd} \beta_1}{(r,\beta_1,B) \vdash_P^\star (q',\varepsilon,\varepsilon)} \quad \dfrac{[q',C,p] \overset{\ell_2}{\lhd} \beta_2}{(q',\beta_2,C) \vdash_P^\star (p,\varepsilon,\varepsilon)}}{(r,\beta_1\beta_2,BC) \vdash_P^\star (p,\varepsilon,\varepsilon)}}{(q,\underbrace{x\beta_1\beta_2}_{\alpha},A) \vdash_P^\star (p,\varepsilon,\varepsilon)}$$

□

Пример 4.16. Да разгледаме стековия автомат от Пример 4.14.

- Началната променлива в граматиката е $S = [q,\sharp,f]$.
- Понеже $\Delta(q,\varepsilon,\sharp) = \{(f,\varepsilon)\}$, то имаме правилото $[q,\sharp,f] \to_G \varepsilon$.
- Понеже $\Delta(q,a,\sharp) = \{(q,a\sharp)\}$, то имаме правилата $[q,\sharp,p] \to_G a[q,a,r][r,\sharp,p]$, за произволни състояния $r,p \in \{q,f\}$.
- Понеже $\Delta(q,a,a) = \{(q,aa)\}$, то имаме правилата $[q,a,p] \to_G a[q,a,r][r,a,p]$, за произволни състояния $r,p \in \{q,f\}$.
- Понеже $\Delta(q,b,a) = \{(q,\varepsilon)\}$, то имаме правилото $[q,a,q] \to_G b$.

Предишните две леми ни дават следната еквивалентност.

Теорема 4.8. Класът на езиците, които се разпознават от недетерминирани стекови автомати съвпада с класа на безконтекстните езици.

Забележка. Лесно можем да съобразим, че недетерминираните крайни автомати са опростен модел на недетерминираните стекови автомати. Оттук директно следва, че всеки регулярен език е безконтекстен.

Сечение с регулярен език

От *Твърдение 4.8* знаем, че безконтекстните езици не са затворени относно операцията сечение, т.е. възможно е L_1 и L_2 да са безконтекстни езици, но $L_1 \cap L_2$ да не е безконтекстен. Оказва се обаче, че безконтекстните езици са затворени относно сечение с регулярен език.

Теорема 4.9. Нека L е безконтекстен език и R е регулярен език. Тогава тяхното сечение $L \cap R$ е безконтекстен език.

Тук адаптираме доказателството от [16, стр. 144].

Упътване. Нека имаме стеков автомат

$$P = \langle Q', \Sigma, \Gamma, \sharp, \Delta', q'_{\texttt{start}}, q'_{\texttt{accept}} \rangle, \text{ където } \mathcal{L}(P) = L,$$

и детерминиран краен автомат

$$\mathcal{A} = \langle Q'', \Sigma, q''_{\texttt{start}}, \delta'', F'' \rangle, \text{ където } \mathcal{L}(\mathcal{A}) = R.$$

Сравнете с конструкцията от *Твърдение 3.2*.

Ще определим нов стеков автомат $\mathcal{M} = \langle Q, \Sigma, \Gamma, \sharp, \Delta, q_{\texttt{start}}, q_{\texttt{accept}} \rangle$, където:

- $Q \stackrel{\text{деф}}{=} Q' \times Q''$;
- $q_{\texttt{start}} \stackrel{\text{деф}}{=} \langle q'_{\texttt{start}}, q''_{\texttt{start}} \rangle$;
- $F \stackrel{\text{деф}}{=} \{q'_{\texttt{accept}}\} \times F''$;
- Функцията на преходите Δ е дефинирана както следва:

Симулираме едновременно изчислението и на двата автомата.

  - Ако $(r_1, \gamma) \in \Delta'(q_1, a, Y)$ и $r_2 = \delta_{\mathcal{A}}(q_2, a)$, то
$$\Delta(\langle q_1, q_2 \rangle, a, Y) \ni (\langle r_1, r_2 \rangle, \gamma).$$

Нищо не четем от входната дума, следователно правим празен ход на $\mathcal{A}$

  - Ако $(r_1, \gamma) \in \Delta'(q_1, \varepsilon, Y)$ и всяко $q_2 \in Q''$, то
$$\Delta(\langle q_1, q_2 \rangle, \varepsilon, Y) \ni (\langle r_1, q_2 \rangle, \gamma).$$
  - Δ не съдържа други преходи;

Докажете, че е изпълнено свойството:

$$(\langle q_1, q_2 \rangle, \alpha, \gamma) \vdash^{\star}_{\mathcal{M}} (\langle p_1, p_2 \rangle, \varepsilon, \varepsilon) \quad \Leftrightarrow \quad (q_1, \alpha, \gamma) \vdash^{\star}_{P} (p_1, \varepsilon, \varepsilon) \text{ и } \delta^{\star}_{\mathcal{A}}(q_2, \alpha) = p_2. \tag{4.26}$$

□

[4, стр. 32]

Упътване. Можем да построим безконтекстна граматика G' директно по безконтекстна граматика G в нормална форма на Чомски и ДКА $\mathcal{A}$. Правилата на G' са следните:

- $[q, A, p] \to_{G'} [q, C, r][r, B, p]$, ако $A \to_G BC$ и $q, r, p \in Q_{\mathcal{A}}$;
- $[q, A, p] \to_{G'} a$, ако $A \to_G a$ и $\delta_{\mathcal{A}}(q, a) = p$;
- $S' \to_{G'} [q_{\texttt{start}}, S, f]$, за всяко $f \in F_{\mathcal{A}}$.

Докажете, че е изпълнено свойството:

$$[q, A, p] \stackrel{\star}{\lhd}_{G'} \alpha \text{ точно тогава, когато } A \stackrel{\star}{\lhd}_{G} \alpha \text{ и } \delta^{\star}_{\mathcal{A}}(q, \alpha) = p. \tag{4.27}$$

□

Упътване. Нека G_1 е безконтекстна граматика за L в нормална форма на Чомски и G_2 е регулярна граматика за R. Тогава дефинираме безконтекстна граматика G_3, където:

- $[X, A, Y] \to_{G_3} [X, B, Z][Z, C, Y]$, където $A \to_{G_1} BC$ и X, Y, Z са произволни променливи в G_2;
- $[X, A, Y] \to_{G_3} a$, където $A \to_{G_1} a$ и $X \to_{G_2} aY$;
- Началната променлива на G_3 е променливата S;
- Имаме и правилата $S \to_{G_3} [S_2, S_1, X]$, където S_1 е началната променлива в G_1, S_2 е началната променлива в G_2 и $X \to_{G_2} \varepsilon$.

За да се убедим, че $\mathcal{L}(G_3) = L \cap R$ е достатъчно да се убедим, че е изпълнено свойството:

$$[X, A, Y] \overset{\star}{\lhd}_{G_3} \alpha \text{ точно тогава, когато } A \overset{\star}{\lhd}_{G_1} \alpha \text{ и } X \overset{\star}{\lhd}_{G_2} \alpha Y. \tag{4.28}$$

□

Теорема 4.9 е удобна, когато искаме да докажем, че даден език не е безконтекстен. С нейна помощ можем да сведем езика до друг, за който вече знаем, че не е безконтекстен.

Пример 4.17. Да разгледаме езика

$$L = \{\omega \in \{a, b, c\}^\star \mid |\omega|_a = |\omega|_b = |\omega|_c\}.$$

Да допуснем, че L е безконтекстен език. Тогава, според *Теорема 4.9*, $L' = L \cap \mathcal{L}[\![\boldsymbol{a^\star b^\star c^\star}]\!]$ също е безконтекстен език. Но $L' = \{a^n b^n c^n \mid n \in \mathbb{N}\}$, за който знаем от *Пример 4.6*, че *не* е безконтекстен. Достигнахме до противоречие. Следователно, L не е безконтекстен език.

4.11 Допълнителни задачи

4.11.1 Равен брой леви и десни скоби

За да се приближим малко до по-реален пример, можете да си мислите, че тук искаме да разпознаем думите с равен брой леви и десни скоби, като например интерпретираме a като символа { и b като символа }.

Нека за по-голяма яснота да положим

$$\texttt{left}(\alpha) \stackrel{\text{деф}}{=} |\alpha|_a \qquad \text{// брой срещания на } a \text{ в } \alpha$$

$$\texttt{right}(\alpha) \stackrel{\text{деф}}{=} |\alpha|_b \qquad \text{// брой срещания на } b \text{ в } \alpha$$

Задача 4.23. Нека ω е произволна дума над азбуката $\{a, b\}$. Тогава:

а) ако $\texttt{left}(\omega) = \texttt{right}(\omega) + 1$, то съществуват думи ω_1, ω_2, за които е изпълнено:

- $\omega = \omega_1 a \omega_2$;
- $\texttt{left}(\omega_1) = \texttt{right}(\omega_1)$;
- $\texttt{left}(\omega_2) = \texttt{right}(\omega_2)$.

б) ако $\texttt{right}(\omega) = \texttt{left}(\omega) + 1$, то съществуват думи ω_1, ω_2, за които е изпълнено:

- $\omega = \omega_1 b \omega_2$;
- $\texttt{left}(\omega_1) = \texttt{right}(\omega_1)$;
- $\texttt{left}(\omega_2) = \texttt{right}(\omega_2)$.

Другият случай е аналогичен

Упътване. Ще се съсредоточим върху случая, когато ω е дума, за която $\texttt{left}(\omega) = \texttt{right}(\omega)+1$. Ще докажем а) с индукция по дължината на думата.

- $|\omega| = 1$. Тогава $\omega_1 = \omega_2 = \varepsilon$ и $\omega = a$.
- Да приемем, че твърдението а) е изпълнено за всички думи ω с дължина $\leq n$.
- Нека сега $|\omega| = n + 1$. Ще разгледаме два случая, в зависимост от първия символ на ω.
  - Случаят $\omega = a\omega'$ е очевиден. (Защо?)
  - Интересният случай е $\omega = b^i ba\omega'$, за някое i. 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 букви a и b, то $\texttt{left}(\omega'') = \texttt{right}(\omega'') + 1$. Според **(И.П.)** за ω'' са изпълнени свойствата:
    * $\omega'' = \omega_1'' a \omega_2''$;
    * $\texttt{left}(\omega_1'') = \texttt{right}(\omega_1'')$;
    * $\texttt{left}(\omega_2'') = \texttt{right}(\omega_2'')$.

    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 4.24. За произволна дума $\omega \in \{a, b\}^\star$, докажете, че ако $\texttt{left}(\omega) > \texttt{right}(\omega)$, то съществуват думи ω_1 и ω_2, за които са изпълнени свойствата:

- $\omega = \omega_1 a \omega_2$;
- $\texttt{left}(\omega_1) \geq \texttt{right}(\omega_1)$;
- $\texttt{left}(\omega_2) \geq \texttt{right}(\omega_2)$.

Задача 4.25. Да се докаже, че езикът $L = \{\, \alpha \in \{a, b\}^\star \mid \texttt{left}(\alpha) = \texttt{right}(\alpha) \,\}$ е безконтекстен.

Доказателство. Една възможна граматика G е следната:

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

Алтернативна граматика за езика L е

$$S \to \varepsilon \mid aSb \mid bSa \mid SS,$$

но уравнението за тази граматика има много решения.

Като следствие от *Задача 4.23* може лесно да се изведе, че за думи ω, за които $\texttt{left}(\omega) = \texttt{right}(\omega)$, е изпълнено следното:

а) ако $\omega = a\omega'$, то са изпълнени свойствата:

- $\omega = a\omega_1 b\omega_2$;
- $\texttt{left}(\omega_1) = \texttt{right}(\omega_1)$;
- $\texttt{left}(\omega_2) = \texttt{right}(\omega_2)$.

б) ако $\omega = b\omega'$, то са изпълнени свойствата:

- $\omega = b\omega_1 a\omega_2$;
- $\texttt{left}(\omega_1) = \texttt{right}(\omega_1)$;
- $\texttt{left}(\omega_2) = \texttt{right}(\omega_2)$.

Да напомним, че $\mathcal{L}^\ell_G(S) = \{\omega \in \Sigma^\star \mid S \overset{\leq \ell}{\lhd} \omega\}$ и според *Твърдение 4.3* имаме следната рекурсивна връзка:

$$\begin{aligned}
\mathcal{L}^0_G(S) =& \emptyset \\
\mathcal{L}^{\ell+1}_G(S) =& \{a\} \cdot \mathcal{L}^\ell_G(S) \cdot \{b\} \cdot \mathcal{L}^\ell_G(S) \cup \\
& \{b\} \cdot \mathcal{L}^\ell_G(S) \cdot \{a\} \cdot \mathcal{L}^\ell_G(S) \cup \\
& \{\varepsilon\}.
\end{aligned}$$

За коректност на граматиката трябва да докажем, че за всяко ℓ е изпълнено следното:

$$\mathcal{L}^\ell_G(S) \subseteq \{\omega \in \{a,b\}^\star \mid \texttt{left}(\omega) = \texttt{right}(\omega)\}. \tag{4.29}$$

Очевидно е, че *Свойство 4.29* е изпълнено за $\ell = 0$. Да приемем, че *Свойство 4.29* е изпълнено за някое ℓ. Ще докажем *Свойство 4.29* за $\ell + 1$. Да вземем произволна дума $\omega \in \mathcal{L}^{\ell+1}_G(S)$. Имаме три случая.

- Ако $\omega = \varepsilon$. Тогава е ясно, че $\texttt{left}(\omega) = \texttt{right}(\omega)$.
- Ако $\omega \in \{a\} \cdot \mathcal{L}^\ell_G(S) \cdot \{b\} \cdot \mathcal{L}^\ell_G(S)$. Тогава $\omega = a\omega_1 b\omega_2$ и $\omega_1, \omega_2 \in \mathcal{L}^\ell_G(S)$. От **(И.П.)** получаваме, че $\texttt{left}(\omega_1) = \texttt{right}(\omega_1)$ и $\texttt{left}(\omega_2) = \texttt{right}(\omega_2)$. Заключаваме, че $\texttt{left}(\omega) = \texttt{right}(\omega)$
- Случаят, когато $\omega = b\omega_1 a\omega_2$, е аналогичен.

Така доказахме коректност на граматиката, т.е. имаме следното свойство:

$$\mathcal{L}_G(S) = \bigcup_\ell \mathcal{L}^\ell_G(S) \subseteq \{\omega \in \{a,b\}^\star \mid \texttt{left}(\omega) = \texttt{right}(\omega)\}.$$

Сега за пълнота на граматиката трябва да докажем, че

$$\{\omega \in \{a,b\}^\star \mid \texttt{left}(\omega) = \texttt{right}(\omega)\} \subseteq \mathcal{L}_G(S). \tag{4.30}$$

Това ще направим с пълна индукция по дължината на думите. Да вземем произволна дума $\omega \in \{a,b\}^\star$ и $\texttt{left}(\omega) = \texttt{right}(\omega)$. Да видим защо $\omega \in \mathcal{L}_G(S)$.

- Ако $\omega = \varepsilon$, то е ясно, че $\omega \in \mathcal{L}_G(S)$.
- Нека $\omega \neq \varepsilon$. Според *Задача 4.23* имаме два случая.
  - Нека $\omega = a\omega_1 b\omega_2$, $\texttt{left}(\omega_1) = \texttt{right}(\omega_1)$ и $\texttt{left}(\omega_2) = \texttt{right}(\omega_2)$. Тогава от **(И.П.)** имаме, че $\omega_1 \in \mathcal{L}_G(S)$ и $\omega_2 \in \mathcal{L}_G(S)$. Заключаваме, че

$$\omega \in \{a\} \cdot \mathcal{L}_G(S) \cdot \{b\} \cdot \mathcal{L}_G(S) \subseteq \mathcal{L}_G(S).$$

– Ако $\omega = b\omega_1 a\omega_2$, то с аналогични разсъждения получаваме, че

$$\omega \in \{b\} \cdot \mathcal{L}_G(S) \cdot \{a\} \cdot \mathcal{L}_G(S) \subseteq \mathcal{L}_G(S).$$

Така доказахме пълнота на граматиката, т.е. имаме следното свойство:

$$\{\omega \in \{a,b\}^\star \mid \texttt{left}(\omega) = \texttt{right}(\omega)\} \subseteq \mathcal{L}_G(S).$$

□

4.11.2 Балансирани скоби

Нека α е дума над азбука, която включва буквите a и b. Ще казваме, че α е **балансирана**, ако са изпълнени свойствата:

Например, думата $aabaabbb$ е балансирана, докато $abbaaab$ не е балансирана. Практическият смисъл на тази задача е, че можем да напишем граматика, която да разпознава дали за всяка отваряща скоба {, която прочетем, по-късно ще прочетем и съответната затваряща скоба }.

- $\texttt{left}(\alpha) = \texttt{right}(\alpha)$;
- За всеки префикс γ на α, $\texttt{left}(\gamma) \geq \texttt{right}(\gamma)$.

Задача 4.26. Докажете, че $L = \{\ \alpha \in \{a,b\}^\star \mid \alpha$ е балансирана дума $\}$ е безконтекстен език.

Доказателство. Да разгледаме граматиката G с правила

$$S \to aSb \mid SS \mid \varepsilon.$$

[13, стр. 135]
✍ Докажете, че езикът L не е регулярен! Алтернативна граматика е

$$S \to aSbS \mid \varepsilon.$$

Тя по-добра в смисъл, че системата за тази граматика има единствено решение.

Ще докажем, че $L = \mathcal{L}(G)$. Имаме, че

$$\begin{aligned}\mathcal{L}^0_G(S) =& \emptyset \\ \mathcal{L}^{\ell+1}_G(S) =& \{a\} \cdot \mathcal{L}^\ell_G(S) \cdot \{b\} \cup \\ & \mathcal{L}^\ell_G(S) \cdot \mathcal{L}^\ell_G(S) \cup \\ & \{\varepsilon\}.\end{aligned}$$

Първо да разгледаме коректност на граматиката, т.е. трябва да докажем, че за всяко ℓ е изпълнено следното:

$$\mathcal{L}^\ell_G(S) \subseteq \{\ \alpha \in \{a,b\}^\star \mid \alpha \text{ е балансирана дума } \}. \tag{4.31}$$

Твърдението е очевидно за $\ell = 0$. Да приемем, че *Свойство 4.31* е изпълнено за някое ℓ. Ще докажем *Свойство 4.31* за $\ell+1$. Да разгледаме произволна дума $\omega \in \mathcal{L}^{\ell+1}_G(S)$. Имаме три случая.

- Нека $\omega = \varepsilon$. Тогава е ясно, че ω е балансирана дума.
- Нека $\omega \in \{a\} \cdot \mathcal{L}^\ell_G(S) \cdot \{b\}$. Тогава $\omega = a\omega_1 b$ и $\omega_1 \in \mathcal{L}^\ell_G(S)$. От **(И.П.)** имаме, че ω_1 е балансирана. Лесно се съобразява, че ω също е балансирана.
- Нека $\omega \in \mathcal{L}^\ell_G(S) \cdot \mathcal{L}^\ell_G(S)$. Тогава $\omega = \omega_1\omega_2$, такива че $\omega_1, \omega_2 \in \mathcal{L}^\ell_G(S)$. От **(И.П.)** имаме, че ω_1 и ω_2 са балансирани. Лесно се съобразява, че ω също е балансирана.

Така доказахме, че

$$\mathcal{L}_G(S) = \bigcup_\ell \mathcal{L}^\ell_G(S) \subseteq \{\ \alpha \in \{a,b\}^\star \mid \alpha \text{ е балансирана дума } \}.$$

Сега ще докажем пълнота на граматиката, т.е. следното свойство:

$$\{\ \alpha \in \{a,b\}^\star \mid \alpha \text{ е балансирана дума } \} \subseteq \mathcal{L}_G(S). \tag{4.32}$$

Това ще направим с *пълна* индукция по дължината на думите. Да разгледаме произволна балансирана дума ω. Имаме няколко случая, които трябва да разгледаме.

- Ако $\omega = \varepsilon$, то е ясно, че $\omega \in \mathcal{L}_G(S)$.
- Нека сега $\omega \neq \varepsilon$. Ясно е, че със сигурност $\omega = a\omega_1 b$. Проблемът е, че в общия случай не е ясно дали можем да приложим **(И.П.)** за ω_1, защото е възможно ω_1 да не е балансирана. Например, ако $\omega = abab$, то $\omega_1 = ba$ не е балансирана. Поради тази причина, трябва да сме по-внимателни и да разгледаме два допълнителни случая.
  - Нека ω има *същински* префикс ω_1, който да е балансирана дума. Понеже ω е балансирана дума, лесно се съобразява, че $\omega = \omega_1\omega_2$, ω_2 също е балансирана дума. Сега можем да приложим **(И.П.)** за ω_1 и ω_2 и да получим, че $\omega_1 \in \mathcal{L}_G(S)$ и $\omega_2 \in \mathcal{L}_G(S)$. Ясно е, че $\omega \in \mathcal{L}_G(S)$.

Тук $\omega_1 \neq \varepsilon$ и $\omega_1 \neq \alpha$. Например, ab е същински префикс на $abab$, който е балансирана дума.

Например, $aabb$ няма същински префикс, който да е балансирана дума.

– Нека ω да няма същински префикс ω_1, който е балансирана дума. Ясно е, че тогава $\omega = a\beta b$, за някое β. Да видим защо β е балансирана дума. Ако β е балансирана, то ще можем да приложим **(И.П.)** за β и ще сме готови.

  За всеки префикс γ на β имаме, че $a\gamma$ е префикс на α, и понеже α е балансирана, то $\texttt{left}(a\gamma) \geq \texttt{right}(a\gamma)$. Възможно ли е $\texttt{left}(\gamma) < \texttt{right}(\gamma)$? Това може да се случи единствено ако $\texttt{left}(a\gamma) = \texttt{right}(a\gamma)$. Но тогава $a\gamma$ е същински префикс на α, за който $a\gamma$ е балансирана дума, което противоречи на случая, който разглеждаме. Това означава, че за произволен префикс γ на β, $\texttt{left}(\gamma) \geq \texttt{right}(\gamma)$ и оттук β е балансирана дума и можем да приложим **(И.П.)** . Тогава $\beta \in \mathcal{L}_G(S)$ и следователно

$$\alpha \in \{a\} \cdot \mathcal{L}_G(S) \cdot \{b\} \subseteq \mathcal{L}_G(S).$$

Така доказахме *Свойство 4.32*, което ни дава пълнота на граматиката спрямо езика. □

4.11.3 Лесни задачи

Задача 4.27. Постройте регулярен израз за езика на следната граматика:

$$S \to S + S \mid S * S \mid A$$
$$A \to KL \mid LK$$
$$K \to 0K \mid \varepsilon$$
$$L \to 1K \mid \varepsilon.$$

Задача 4.28. Докажете, че следните езици са безконтекстни.

а) $L = \{\omega \in \{a,b\}^\star \mid \omega = \omega^{\mathtt{rev}}\}$;
б) $L = \{a^n b^{2m} c^n \mid m, n \in \mathbb{N}\}$;
в) $L = \{a^n b^m c^m d^n \mid m, n \in \mathbb{N}\}$;
г) $L = \{a^n b^{2k} \mid n, k \in \mathbb{N} \,\&\, n \neq k\}$;
д) $L = \{a^n b^k \mid n > k\}$;
е) $L = \{a^n b^k \mid n \geq 2k\}$;
ж) $L = \{a^n b^k c^m \mid n + k \geq m + 1\}$;
з) $L = \{a^n b^k c^m \mid n + k \geq m + 2\}$;
и) $L = \{a^n b^k c^m \mid n + k + 1 \geq m\}$;
к) $L = \{a^n b^m c^{2k} \mid n \neq 2m \,\&\, k \geq 1\}$;
л) $L = \{a^n b^k c^m \mid n + k \leq m\}$;
м) $L = \{a^n b^k c^m \mid n + k \leq m + 1\}$;
н) $L = \{a^n b^m c^k \mid n, m, k$ не са страни на триъгълник$\}$;
о) $L = \{a,b\}^\star \setminus \{a^{2n} b^n \mid n \in \mathbb{N}\}$;
п) $L = \{\omega \in \{a,b\}^\star \mid |\omega|_a = |\omega|_b + 1\}$;
р) $L = \{\omega \in \{a,b\}^\star \mid |\omega|_a \geq |\omega|_b\}$;
с) $L = \{\omega \in \{a,b\}^\star \mid |\omega|_a > |\omega|_b\}$;
т) $L = \{\alpha \in \{a,b\}^\star \mid$ във всеки префикс β на α, $|\beta|_b \leq |\beta|_a\}$;
у) $L = \{\alpha\sharp\beta \mid \alpha, \beta \in \{a,b\}^\star \,\&\, \alpha^{\mathtt{rev}}$ е поддума на $\beta\}$.
ф) $L = \{\omega_1\sharp\omega_2\sharp\cdots\sharp\omega_n \mid n \geq 2 \,\&\, \omega_1, \omega_2, \ldots, \omega_n \in \{a,b\}^\star \,\&\, |\omega_1| = |\omega_2|\}$;
х) $L = \{\omega_1\sharp\omega_2\sharp\cdots\sharp\omega_n \mid n \geq 2 \,\&\, \omega_1, \ldots, \omega_n \in \{a,b\}^\star \,\&\, (\exists i \neq j)[\,|\omega_i| = |\omega_j|\,]\}$;
ц) $L = \{\omega_1\sharp\omega_2\sharp\cdots\sharp\omega_n \mid n \geq 2 \,\&\, (\forall i \in [1,n])[\,\omega_i \in \{a,b\}^\star \,\&\, |\omega_i| = |\omega_{n+1-i}|\,]\}$.

Задача 4.29. Проверете дали следните езици са безконтекстни:

а) $\{a^n b^{2n} c^{3n} \mid n \in \mathbb{N}\}$;
б) $\{a^n b^k c^k a^n \mid k \leq n\}$;
в) $\{a^n b^m c^k \mid n < m < k\}$;
г) $\{a^n b^n c^k \mid n \leq k \leq 2n\}$;
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 $L = \{\alpha\alpha^{\mathtt{rev}} \mid \alpha \in \{a,b\}^\star\}$;
и) $\{\omega\omega\omega \mid \omega \in \{a,b\}^\star\}$;
к) $\{a^{n^2} b^n \mid n \in \mathbb{N}\}$;
л) $\{a^p \mid p$ е просто $\}$;
м) $\{\omega \in \{a,b\}^\star \mid \omega = \omega^{\mathtt{rev}}\}$;
н) $\{\omega^n \mid \omega \in \{a,b\}^\star \,\&\, |\omega|_b = 2 \,\&\, n \in \mathbb{N}\}$;
о) $\{\omega c^n \omega^{\mathtt{rev}} \mid \omega \in \{a,b\}^\star \,\&\, n = |\omega|\}$;
п) $\{\alpha c \beta \mid \alpha, \beta \in \{a,b\}^\star \,\&\, \alpha$ е подниз на $\beta\}$;
р) $\{\omega_1\sharp\omega_2\sharp\ldots\sharp\omega_k \mid k \geq 2 \,\&\, \omega_i \in \{a\}^\star \,\&\, (\exists i,j)[i \neq j \,\&\, \omega_i = \omega_j]\}$;
с) $\{\omega_1\sharp\omega_2\sharp\ldots\sharp\omega_k \mid k \geq 2 \,\&\, \omega_i \in \{a\}^\star \,\&\, (\forall i,j \leq k)[i \neq j \;\Leftrightarrow\; \omega_i \neq \omega_j]\}$;
т) $\{a^i b^j c^k \mid i, j, k \geq 0 \,\&\, (i = j \vee j = k)\}$;
у) $\{\omega \in \{a,b,c\}^\star \mid |\omega|_a > |\omega|_b > |\omega|_c\}$;
ф) $\{a,b\}^\star \setminus \{a^n b^n \mid n \in \mathbb{N}\}$;
х) $\{a^n b^m c^k \mid m^2 = 2nk\}$;
ц) $L = \{a^n b^m c^m a^n \mid m, n \in \mathbb{N} \,\&\, n = m + 42\}$;
ч) $L = \{\sharp a\sharp aa\sharp aaa\sharp \cdots \sharp a^{n-1}\sharp a^n\sharp \mid n \geq 1\}$;
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) $\{\omega \in \{a,b,c\}^\star \mid |\omega|_a \neq |\omega|_b \vee |\omega|_a \neq |\omega|_c \vee |\omega|_b \neq |\omega|_c\}$.

а) $S \to aSa \mid bSb \mid a \mid b \mid \varepsilon$;
г) Обединение на два езика;
д) $S \to aSb \mid aS \mid a$;
ж) $S \to aSc \mid aS \mid aB \mid bB$,
$B \to bBc \mid bB \mid \varepsilon$;
и) $S \to aSc \mid aS \mid B \mid Bc$,
$B \to bBc \mid bB \mid \varepsilon$;
н) Обединение на три езика;
п) $S \to EaE$,
$E \to aEbE \mid bEaE \mid \varepsilon$;

4.11.4 Не толкова лесни задачи

Задача 4.30. Докажете, че езикът $L = \{a^n b^{kn} \mid k, n > 0\}$ не е безконтекстен.

Упътване. Да разгледаме ситуацията $\alpha = a^p b^{p^2} \in L$ и $\alpha = xyuvw$, където $y = a^i$ и $v = b^j$. Интересният случай е когато $0 < i, j < p$.

- Нека $i = j$. Да разглеждаме думата $xy^{p^2+1}uv^{p^2+1}w$, която за да бъде в езика L означава, че трябва $p + p^2 i$ дели $p^2 + p^2 i$, т.е. $1 + pi$ трябва да дели $p + pi$.

$$p + pi = k(1 + pi), \text{ за някое } 1 \leq k < p$$
$$p = k + pi(k - 1), \text{ за някое } 1 \leq k < p$$

  Достигаме до противоречие.

- Нека $i > j$, т.е. $i \geq j + 1$. Отново разглеждаме думата $xy^{p^2+1}uv^{p^2+1}w$. За да бъде тази дума в езика L, трябва $p + p^2 i$ да дели $p^2 + p^2 j$, т.е. $1 + pi$ трябва да дели $p + pj$, но

$$1 + pi \geq 1 + p(j + 1) > p + pj.$$

  Противоречие.

- Нека $i < j$ и тогава нека $j = mi + r$, където $p > i > r \geq 0$ Разглеждаме думата $xy^{mp^2+1}uv^{mp^2+1}w$. Това означава дали $p + mp^2 i$ дели $p^2 + mp^2 j$, т.е. дали $1 + mpi$ дели $p + mpj$.

$$p + pmj = k(1 + pmi), \text{ за някое } 1 \leq k.$$

  – Възможно ли е $k \geq p$? Тогава:

$$p + pmj = k(1 + pmi) \geq p(1 + pmi) \geq p(1 + pj)$$
$$p(1 + mj) \geq p(1 + pj)$$
$$m \geq p.$$

    Достигаме до противоречие. Следователно, $1 \leq k < p$.

  – Възможно ли е $k \leq m$? Тогава:

$$p + pmj = k(1 + pmi) \leq m(1 + pmi) \leq m(1 + pj)$$
$$p + pmj \leq m + pmj$$
$$p \leq m.$$

    Достигаме до противоречие, защото $m < p$.

  – Заключаваме, че $1 \leq m < k < p$. Тогава:

$$p + pmj = k(1 + pmi)$$
$$p = k + pm(ki - j).$$

    Понеже $m < k$, то $j < (m + 1)i \leq ki$, т.е. $ki - j > 0$. Достигаме до противоречие.

□

Дефинираме функцията $\texttt{diff} : \Sigma^\star \times \Sigma^\star \to \Sigma^\star$ по следния начин:

$$\texttt{diff}(\alpha, \varepsilon) = 0$$
$$\texttt{diff}(\varepsilon, \beta) = 0$$
$$\texttt{diff}(a \cdot \alpha, b \cdot \beta) = \begin{cases} \texttt{diff}(\alpha, \beta), & \text{ако } a = b \\ 1 + \texttt{diff}(\alpha, \beta), & \text{ако } a \neq b \end{cases}$$

Задача 4.31. За всеки от следните езици, отговорете дали са безконтекстни, като се обосновете:

а) $\{\alpha\sharp\beta \mid \alpha,\beta \in \{a,b\}^\star \,\&\, |\alpha| = |\beta| \,\&\, \mathtt{diff}(\alpha,\beta^{\mathtt{rev}}) = 1\}$; Да

б) $\{\alpha\sharp\beta \mid \alpha,\beta \in \{a,b\}^\star \,\&\, |\alpha| = |\beta| \,\&\, \mathtt{diff}(\alpha,\beta^{\mathtt{rev}}) \geq 1\}$; Да

в) $\{\alpha\sharp\beta \mid \alpha,\beta \in \{a,b\}^\star \,\&\, |\alpha| = |\beta| \,\&\, \mathtt{diff}(\alpha,\beta) \geq 1\}$; Не

г) $\{\alpha\beta \mid \alpha,\beta \in \{a,b\}^\star \,\&\, |\alpha| = |\beta| \,\&\, \mathtt{diff}(\alpha,\beta) \geq 1\}$; Да

д) $\{\alpha\beta \mid \alpha,\beta \in \{a,b\}^\star \,\&\, |\alpha| = |\beta| \,\&\, \mathtt{diff}(\alpha,\beta) = 1\}$; Не

Задача 4.32. Да разгледаме езика

$$D_1 \stackrel{\text{деф}}{=} \{\alpha_1\sharp\cdots\sharp\alpha_n \mid n \geq 2 \,\&\, |\alpha_i| = |\alpha_{i+1}| \,\&\, \mathtt{diff}(\alpha_i,\alpha_{i+1}^{\mathtt{rev}}) = 1\}.$$

- Докажете, че D_1 не е безконтекстен.
- Докажете, че D_1 може да се представи като сечението на два безконтекстни езика.

Упътване. Да разгледаме безконтекстния език

$$L_1 = \{\alpha_1\sharp\alpha_2 \mid |\alpha_1| = |\alpha_2| \,\&\, \mathtt{diff}(\alpha_1,\alpha_2^{\mathtt{rev}}) = 1\}.$$

Тогава

$$D_1 = L_1\cdot(\sharp L_1)^\star\cdot(\{\varepsilon\}\cup\sharp\{a,b\}^\star)\cap \{a,b\}^\star\cdot(\sharp L_1)^\star\cdot(\{\varepsilon\}\cup\sharp\{a,b\}^\star).$$

□

Задача 4.33. За произволен език L, дефинираме езика

$$\mathtt{Diff}_n(L) = \{\alpha\in L \mid (\exists\beta\in L)[|\alpha| = |\beta| \,\&\, \mathtt{diff}(\alpha,\beta) = n]\}.$$

Вярно ли е, че:

- ако L е регулярен, то $\mathtt{Diff}_n(L)$ е регулярен?
- ако L е безконтекстен, то $\mathtt{Diff}_n(L)$ е безконтекстен?

Задача 4.34. Нека L_1 и L_2 са езици. Дефинираме [21, стр. 158]

$$L_1\triangle L_2 \stackrel{\text{деф}}{=} \{\alpha\beta \mid \alpha\in L_1 \,\&\, \beta\in L_2 \,\&\, |\alpha| = |\beta|\}.$$

Докажете, че

а) ако L_1 и L_2 са регулярни езици, то е възможно $L_1\triangle L_2$ да не е регулярен;

б) ако L_1 и L_2 са регулярни езици, то $L_1\triangle L_2$ е безконтекстен;

в) ако L_1 и L_2 са безконтекстни езици, то е възможно $L_1\triangle L_2$ да не е безконтекстен.

Вече знаем, че ако L е регулярен, то $L^{\mathtt{rev}}$ е регулярен.

Задача 4.35. Докажете, че ако L е безконтекстен език, то

$$L^{\mathtt{rev}} = \{\omega^{\mathtt{rev}} \mid \omega\in L\}$$

също е безконтекстен.

Задача 4.36. 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 4.37. 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 4.38. 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:

$$L = \{\alpha_1 \cdots \alpha_{3n} \beta_1 \cdots \beta_m \gamma_1 \cdots \gamma_n \mid \alpha_i, \gamma_j \in L_1, \beta_k \in L_2, m, n \in \mathbb{N}\}.$$

Задача 4.39. Нека $L = \{\omega \in \{a, b\}^\star \mid |\omega|_a = 2\}$. Да се докаже, че езикът $L' = \{\alpha^n \mid \alpha \in L, n \geq 0\}$ не е безконтекстен.

Задача 4.40. 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.

Знаем, че за всеки регулярен език L и произволен език M, $M^{-1}(L)$ е регулярен. [16, стр. 149]

Задача 4.41. Да фиксираме азбуката Σ. Нека L е безконтекстен език, а R е регулярен език. Докажете, че езикът

$$R^{-1}(L) = \{\alpha \in \Sigma^\star \mid (\exists \beta \in R)[\beta\alpha \in L]\}$$

е безконтекстен.

Упътване. Най-лесно се вижда с декартова конструкция на стеков автомат за L и автомат за R. □

Задача 4.42. Нека L е безконтекстен език над азбуката Σ. Докажете, че следните езици са безконтекстни:

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

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

в) $\texttt{Infix}(L) = \{\beta \in \Sigma^\star \mid (\exists \alpha)(\exists \gamma)[\alpha \cdot \beta \cdot \gamma \in L]\}$;

Задача 4.43. Докажете, че езикът

$$L = \{\, \omega_1 \sharp \omega_2 \sharp \cdots \sharp \omega_{2n} \mid n \geq 1 \;\&\; \sum_{i=1}^{n} |\omega_{2i-1}| = \sum_{i=1}^{n} |\omega_{2i}| \,\}$$

е безконтекстен.

Упътване. Най-лесно става със стеков автомат, като са необходими две допълнителни букви за азбуката на стека.

Една възможна безконтекстна граматика е следната:

$$\begin{aligned} E &\to XEX \mid \sharp O \mid O\sharp \mid \sharp E\sharp \\ O &\to OO \mid EE \mid \varepsilon \\ X &\to a \mid b, \end{aligned}$$

където началната променлива е E. □

Триене на третина от думите на език

Да напомним следните операции върху езици:

$$\texttt{Triples}_2(L) \stackrel{\text{деф}}{=} \{\omega_2 \mid (\exists\omega_1)(\exists\omega_3)[\,\omega_1\omega_2\omega_3 \in L \,\&\, |\omega_1| = |\omega_2| = |\omega_3|\,]\,\};$$

$$\texttt{Triples}_{1,3}(L) \stackrel{\text{деф}}{=} \{\omega_1\omega_3 \mid (\exists\omega_2)[\omega_1\omega_2\omega_3 \in L \,\&\, |\omega_1| = |\omega_2| = |\omega_3|]\}.$$

Вече знаем, че $\texttt{Triples}_{1,3}(L)$ не винаги е регулярен при регулярен L. Възможно е за регулярен L да се конструира и стеков автомат за $\texttt{Triples}_{1,3}(L)$ [7, стр. 140].

Задача 4.44. Докажете, че ако регулярен език L, то $\texttt{Triples}_{1,3}(L)$ е безконтекстен.

Упътване. Нека $L = \mathcal{L}(\mathcal{A})$, където $\mathcal{A}$ е ДКА. Ще построим безконтекстна граматика G за $\texttt{Triples}_{1,3}(L)$. Правилата на граматиката G са следните:

- $S \to [(q_{\texttt{start}}, q), (q, r), f]$ за произволно $f \in F$.
- $[(p, q), (r, s), t] \to a[(p', q), (r', s), t']b$, където са изпълнени условията:
  - $\delta_{\mathcal{A}}(p, a) = p'$;
  - $\delta_{\mathcal{A}}(t', b) = t$, за някое t';
  - $\delta_{\mathcal{A}}(r, x) = r'$ за някое $x \in \Sigma$.
- $[(p, p), (q, q), q] \to \varepsilon$ за произволни $p, q \in Q$.

□

Разполовяване на думи

Да напомним следната операция върху езици:

$$\texttt{Half}(L) \stackrel{\text{деф}}{=} \{\omega \mid \omega \cdot \omega \in L\}.$$

Операцията $\texttt{Double}(L)$ не е толкова интересна, защото не е трудно да се намери безконтекстен език, за който $\texttt{Double}(L)$ не е безконтекстен.

Знаем, че ако един език L е регулярен, то $\texttt{Half}(L)$ също е регулярен. Интересно е да отбележим, че това свойство не се запазва за класа на безконтекстните езици.

Задача 4.45. Дайте пример за безконтекстен език L, за който $\texttt{Half}(L)$ *не* е безконтекстен.

Упътване. Разгледайте безконтекстния език

$$L = \{a^n b^n c^k a^k b^\ell c^\ell \mid n, k, \ell \geq 1\}.$$

Съобразете, че

$$\texttt{Half}(L) = \{a^n b^n c^n \mid n \geq 1\}.$$

□

Триене на половината дума

Да напомним сега още една операция върху езици:

$$\frac{1}{2}(L) \stackrel{\text{деф}}{=} \{\alpha \mid (\exists\beta)[|\alpha| = |\beta| \,\&\, \alpha\beta \in L\}.$$

Знаем, че ако един език L е регулярен, то $\frac{1}{2}(L)$ също е регулярен.

Задача 4.46. Дайте пример за безконтекстен език L, за който $\frac{1}{2}(L)$ *не* е безконтекстен.

Упътване. Например, разгледайте безконтекстния език

$$L = \{a^n b^n a^k b a^{3k+1} \mid n, k \in \mathbb{N}\}.$$

Тогава

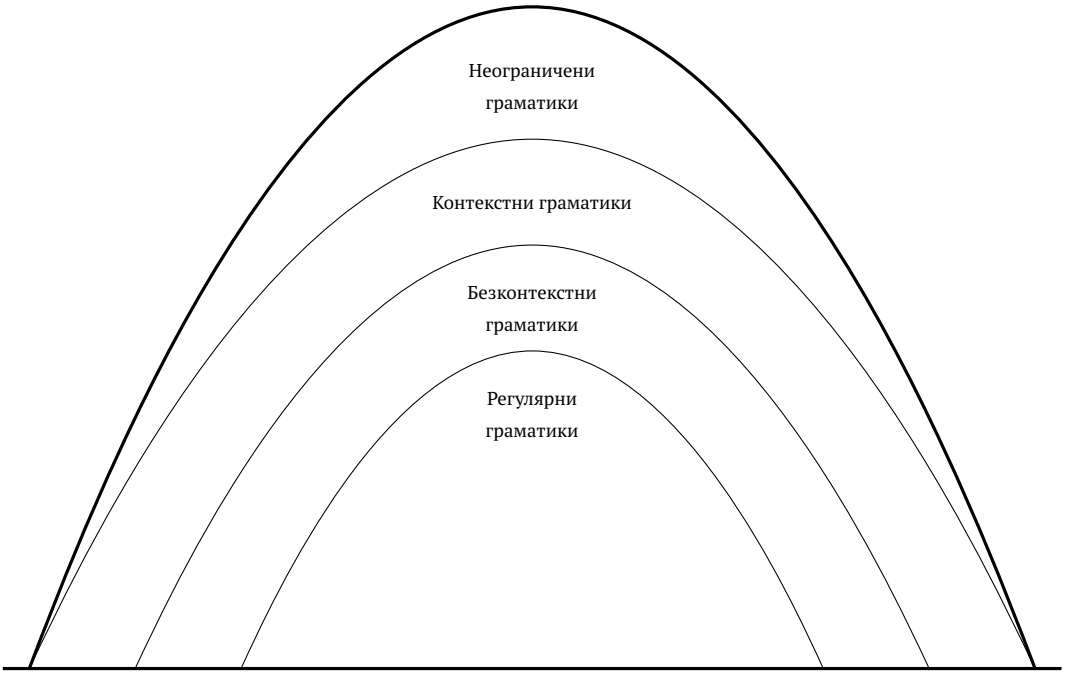
$$\frac{1}{2}(L) \cap \mathcal{L}[\![\boldsymbol{a}^+\boldsymbol{b}^+\boldsymbol{a}^+\boldsymbol{b}]\!] = \{a^n b^n a^n b \mid n \geq 1\},$$

който със сигурност не е безконтекстен. □

Глава 5

Йерархия от езици

Ще започнем като разгледаме понятието извод в граматика в най-общия му вид. Граматиките се разделят на няколко вида в зависимост от това какви *ограничения* налагаме върху правилата на граматиката. В следващите няколко глави ще разгледаме различни ограничения. След това ще разгледаме някои класове от граматики като ще се концентрираме основно върху безконтекстните граматики.



Фигура 5.1: Йерархия на Чомски

5.1 Неограничени граматики

На англ. *unrestricted grammar*. Това са тип 0 граматиките в йерархията на Чомски [10, стр. 220].

Неограничена граматика е наредена четворка от вида

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

където:

- V е крайно множество от *променливи* (нетерминали);
- Σ е крайно множество от *букви* (терминали), като $\Sigma \cap V = \emptyset$;
- $R \subseteq (V \cup \Sigma)^+ \times (V \cup \Sigma)^\star$ е крайно множество от *правила*. За по-добра яснота, обикновено правилата $(\alpha, \beta) \in R$ ще означаваме като $\alpha \to_G \beta$. Когато е ясно за коя граматика говорим, ще пишем просто $\alpha \to \beta$.
- $S \in V$ е началната променлива (нетерминал).

В [10] правилата се наричат *productions или production rules*.

Ще дефинираме бинарната релация $\Rightarrow_G$ над $(V \cup \Sigma)^\star$, така че $\gamma \Rightarrow_G \gamma'$ казва, че от думата γ се получава думата γ' чрез прилагане на някое правило $\alpha \to_G \beta$ в граматиката G.

Тук отново, когато е ясно за коя граматика говорим, ще пишем просто $\Rightarrow$ вместо $\Rightarrow_G$.

$$\frac{(\alpha, \beta) \in R \qquad \lambda, \rho \in (V \cup \Sigma)^\star}{\lambda\alpha\rho \Rightarrow_G \lambda\beta\rho}$$

Фигура 5.2: Дефиниция на релацията $\Rightarrow_G$, която казва дали от една дума може да се получи друга дума с прилагане на едно правило от граматиката.

Задача 5.1. Съобразете, че имаме свойството:

$$\frac{\alpha \Rightarrow_G \beta \qquad \lambda, \rho \in (V \cup \Sigma)^\star}{\lambda\alpha\rho \Rightarrow_G \lambda\beta\rho}$$

Сега е удобно да дефинираме **извод** на думата β от думата α в граматиката G за ℓ стъпки, което ще означаваме като $\alpha \overset{\ell}{\Rightarrow}_G \beta$, с индукция по броя на стъпките ℓ по следния начин:

В правило (1), нямаме никакви ограничения за λ и ρ. Те са произволни елементи на $(V \cup \Sigma)^\star$, което означава, че може и да са празните думи.

$$\frac{\alpha \in (V \cup \Sigma)^\star}{\alpha \overset{0}{\Rightarrow}_G \alpha} \text{ (рефлексивност)} \qquad \frac{\alpha \Rightarrow_G \beta \qquad \beta \overset{\ell}{\Rightarrow}_G \gamma}{\alpha \overset{\ell+1}{\Rightarrow}_G \gamma} \text{ (транзитивност)}$$

Фигура 5.3: Правила за извод в неограничена граматика

Пример 5.1. Да разгледаме граматиката G зададена с правилата

$$S \to_G SA \mid aSA \mid ASb \mid \varepsilon.$$

Тогава $aSbSb \Rightarrow_G aaSbbSb$, защото имаме правилото $S \to_G aSb$. Също така, $aSSb \not\Rightarrow_G ab$, но $aSSb \overset{2}{\Rightarrow}_G ab$ като приложим два пъти правилото $S \to_G \varepsilon$.

Сега дефинираме релацията $\overset{\star}{\Rightarrow}_G$ като

$$\alpha \overset{\star}{\Rightarrow}_G \beta \overset{\text{деф}}{\Leftrightarrow} (\exists \ell \in \mathbb{N})[\, \alpha \overset{\ell}{\Rightarrow}_G \beta \,].$$

Обърнете внимание, че в тази дефиниция на извод не определяме реда, в който прилагаме правилата на граматиката. Също така, понякога ,за удобство, ще пишем просто $\overset{\ell}{\Rightarrow}$ вместо $\overset{\ell}{\Rightarrow}_G$, когато се знае за коя граматика говорим. С други думи, $\overset{\star}{\Rightarrow}_G$ е рефлексивното и транзитивно затваряне на релацията $\Rightarrow_G$.

Езикът, който се поражда от граматиката G дефинираме по следния начин:

$$\mathcal{L}(G) \stackrel{\text{деф}}{=} \{\omega \in \Sigma^\star \mid S \stackrel{\star}{\Rightarrow}_G \omega\}.$$

За да можем да работим по-удобно с релацията за извод в граматика, ще започнем с няколко основни свойства.

Твърдение 5.1. За произволно естествено число ℓ имаме извода:

$$\frac{\alpha \stackrel{\ell}{\Rightarrow} \beta \qquad \lambda, \rho \in (V \cup \Sigma)^\star}{\lambda\alpha\rho \stackrel{\ell}{\Rightarrow} \lambda\beta\rho}$$

Доказателство. Индукция по ℓ.

- $\ell = 0$. Тук всичко е ясно, защото тогава $\alpha = \beta$.
- $\ell > 0$. Според правилата за извод, можем да разбием извода $\alpha \stackrel{\ell}{\Rightarrow} \beta$ по следния начин:

$$\frac{\alpha \Rightarrow \gamma \qquad \gamma \stackrel{\ell-1}{\Rightarrow} \beta}{\alpha \stackrel{\ell}{\Rightarrow} \beta}\ \text{(транзитивност)}$$

Сега като използваме **(И.П.)** получаваме следния извод:

$$\dfrac{\dfrac{\alpha \Rightarrow \gamma}{\lambda\alpha\rho \Rightarrow \lambda\gamma\rho}\ (\textit{Задача 5.1}) \qquad \dfrac{\gamma \stackrel{\ell-1}{\Rightarrow} \beta}{\lambda\gamma\rho \stackrel{\ell-1}{\Rightarrow} \lambda\beta\rho}\ \textbf{(И.П.)}}{\lambda\alpha\rho \stackrel{\ell}{\Rightarrow} \lambda\beta\rho}\ \text{(транзитивност)}$$

□

Твърдение 5.2. За произволни ℓ_1 и ℓ_2 имаме извода:

$$\frac{\alpha_1 \stackrel{\ell_1}{\Rightarrow} \beta_1 \qquad \alpha_2 \stackrel{\ell_2}{\Rightarrow} \beta_2}{\alpha_1\alpha_2 \stackrel{\ell_1+\ell_2}{\Rightarrow} \beta_1\beta_2}$$

Доказателство. Индукция по ℓ_1.

- Ако $\ell_1 = 0$, то $\alpha_1 = \beta_1$ и тогава:

$$(\textit{Твърдение 5.1})\ \frac{\alpha_1 = \beta_1 \qquad \alpha_2 \stackrel{\ell_2}{\Rightarrow}_G \beta_2}{\alpha_1\alpha_2 \stackrel{\ell_2}{\Rightarrow}_G \beta_1\beta_2}$$

- Ако $\ell_1 > 0$, то разбиваме извода $\alpha_1 \stackrel{\ell_1}{\Rightarrow} \beta_1$ по следния начин:

$$\frac{\alpha_1 \Rightarrow \gamma_1 \qquad \gamma_1 \stackrel{\ell_1-1}{\Rightarrow} \beta_1}{\alpha \stackrel{\ell_1}{\Rightarrow} \beta_1}\ \text{(транзитивност)}$$

Сега прилагаме **(И.П.)** за $\ell_1 - 1 < \ell_1$ и получаваме следния извод:

$$\dfrac{\dfrac{\alpha_1 \Rightarrow \gamma_1}{\alpha_1\alpha_2 \Rightarrow \gamma_1\alpha_2}\ (\textit{Задача 5.1}) \qquad \dfrac{\gamma_1 \overset{\ell_1-1}{\Rightarrow} \beta_1 \qquad \alpha_2 \overset{\ell_2}{\Rightarrow} \beta_2}{\gamma_1\alpha_2 \overset{\ell_1-1+\ell_2}{\Rightarrow} \beta_1\beta_2}\ \textbf{(И.П.)}}{\alpha_1\alpha_2 \overset{\ell_1+\ell_2}{\Rightarrow} \beta_1\beta_2}\ \text{(транзитивност)}$$

□

Твърдение 5.3. За всяко k е изпълнено, че:

$$\dfrac{\alpha_1 \overset{\ell_1}{\Rightarrow} \beta_1 \qquad \ldots \qquad \alpha_k \overset{\ell_k}{\Rightarrow} \beta_k}{\alpha_1 \cdots \alpha_k \overset{\ell}{\Rightarrow} \beta_1 \cdots \beta_k}\ (\ell = \textstyle\sum_{i=1}^{k} \ell_i)$$

Упътване. Индукция по k. □

Твърдение 5.4. За произволни естествени числа ℓ_1 и ℓ_2 е изпълнено, че:

$$\dfrac{\alpha \overset{\ell_1}{\Rightarrow} \beta \qquad \beta \overset{\ell_2}{\Rightarrow} \gamma}{\alpha \overset{\ell_1+\ell_2}{\Rightarrow} \gamma}$$

Доказателство. Индукция по ℓ_1.

- Нека $\ell_1 = 0$. Този случай е ясен, защото тогава $\alpha = \beta$ и имаме извода:

$$\dfrac{\dfrac{\alpha \overset{0}{\Rightarrow} \beta}{\alpha = \beta}\ \text{(рефлексивност)} \qquad \beta \overset{\ell_2}{\Rightarrow} \gamma}{\alpha \overset{0+\ell_2}{\Rightarrow} \gamma}$$

- Нека $\ell_1 > 0$. Да разбием извода $\alpha \overset{\ell_1}{\Rightarrow} \beta$ така:

$$\dfrac{\alpha \Rightarrow \alpha' \qquad \alpha' \overset{\ell_1-1}{\Rightarrow} \beta}{\alpha \overset{\ell_1}{\Rightarrow} \beta}\ \text{(транзитивност)}$$

Сега можем да приложим индукционното предположение и да получим извода:

$$\dfrac{\alpha \Rightarrow \alpha' \qquad \dfrac{\alpha' \overset{\ell_1-1}{\Rightarrow} \beta \qquad \beta \overset{\ell_2}{\Rightarrow} \gamma}{\alpha' \overset{\ell_1-1+\ell_2}{\Rightarrow} \gamma}\ \textbf{(И.П.)}}{\alpha \overset{\ell_1+\ell_2}{\Rightarrow} \gamma}\ \text{(транзитивност)}$$

□

Следствие 5.1. За произволни естествени числа ℓ_1 и ℓ_2 е изпълнено, че:

$$\frac{\alpha \overset{\ell_1}{\Rightarrow} \lambda\beta\rho \qquad \beta \overset{\ell_2}{\Rightarrow} \gamma}{\alpha \overset{\ell_1+\ell_2}{\Rightarrow} \lambda\gamma\rho}$$

5.2 Контекстни граматики

На англ. context-sensitive [10, стр. 223]. На български може да се преведат и като контекстнозависими граматики. В йерархията на Чомски това са граматиките от тип 1. В дефиницията, позволяваме и правилото $S \to \varepsilon$, ако искаме да включим ε в езика, като тогава изискваме S да не се среща в дясна страна на правило. По-проста граматика в [20, стр. 202].

Разглеждането на контекстни граматики излиза извън целите на този курс. Добавяме този раздел единствено за пълнота на изложението. Казваме, че $G = (V, \Sigma, R, S)$ е **контекстна граматика**, ако правилата на G са от вида

$$\lambda A \rho \to \lambda \alpha \rho,$$

където $\lambda, \rho \in (V \cup \Sigma)^\star$ и $\alpha \in (V \cup \Sigma)^+$.

Пример 5.2. Езикът $L = \{a^n b^n c^n \mid n > 0\}$ е контекстен.

Упътване. Разгледайте контекстната граматика G зададена със следните правила:

$$S \to aSBC \mid aBC$$
$$CB \to CZ$$
$$CZ \to WZ$$
$$WZ \to WC$$
$$WC \to BC$$
$$aB \to ab$$
$$bB \to bb$$
$$bC \to bc$$
$$cC \to cc.$$

Докажете, че за всяко $n > 0$ е изпълнено следното:

- $S \overset{n}{\Rightarrow} a^n (BC)^n$;
- $CB \overset{4}{\Rightarrow}_G BC$.
- $(BC)^n \overset{4(n-1)}{\Rightarrow} B^n C^n$;
- $aB^n \overset{n}{\Rightarrow} ab^n$;
- $bC^n \overset{n}{\Rightarrow} bc^n$.

Оттук лесно можем да докажем, че имаме включването $L \subseteq \mathcal{L}(G)$. □

5.3 Безконтекстни граматики

В Раздел 5.1 въведохме понятието неограничена граматика. След това видяхме как можем да опишем регулярните езици със специален вид граматики, които нарекохме регулярни граматики. Сега ще разгледаме още един вид граматики, които описват по-широк клас от езици.

Една граматика $G = (V, \Sigma, R, S)$ се нарича **безконтекстна**, ако имаме ограничението, че $R \subseteq V \times (V \cup \Sigma)^\star$. Да повторим дефиницията на релацията $\alpha \Rightarrow_G \beta$ от Фигура 5.2 в частния случай, когато граматиката е безконтекстна.

В [16] дефиницията е различна. Там $\Sigma \subseteq V$. На англ. *context-free grammar*. Други срещани наименования на български са *контекстносвободна*, *контекстнонезависима*. Тук всички правила са от вида $A \to \alpha$, където $\alpha \in (V \cup \Sigma)^\star$. В частност имаме, че:

$$\frac{(A, \alpha) \in R}{A \Rightarrow_G \alpha}$$

$$\frac{(A, \alpha) \in R \qquad \lambda, \rho \in (V \cup \Sigma)^\star}{\lambda A \rho \Rightarrow_G \lambda \alpha \rho}$$

Фигура 5.4: Дефиниция на релацията $\Rightarrow_G$, която казва дали от една дума може да се получи друга дума с прилагане на едно правило от граматиката.

Нека официално да обявим, че един език L се нарича **безконтекстен**, ако съществува безконтекстна граматика G, за която

$$L = \mathcal{L}(G) = \{\omega \in \Sigma^\star \mid S \overset{\star}{\Rightarrow} \omega\}.$$

Да започнем с няколко прости примера за езици, които са безконтекстни, но за които знаем, че не са регулярни.

От тези прости примери можем да заключим, че безконтекстните граматики имат странни навици в сравнение с автоматите. Докато автоматите ядат първо супата, после основното и накрая десерта, то безконтекстните граматики ядат супата заедно с десерта.

Пример 5.3. Езикът $\{a^n b^n \mid n \in \mathbb{N}\}$ е безконтекстен, защото може да се опише с граматика с правила $S \to aSb \mid \varepsilon$.

Пример 5.4. Езикът $\{\omega\omega^{\mathtt{rev}} \mid \omega \in \{a, b\}^\star\}$ е безконтекстен, защото може да се опише с граматика с правила $S \to aSa \mid bSb \mid \varepsilon$.

По-късно ще видим как можем внимателно да докажем че езиците от примерите наистина се пораждат (генерират) от дадените граматики.

Нека сега да обърнем внимание на някои общи свойства на безконтекстните граматики. Като частен случай на *Следствие 5.1* получаваме следното свойство.

Твърдение 5.5. За произволни естествени числа ℓ_1 и ℓ_2 е изпълнено, че:

$$\frac{\alpha \overset{\ell_1}{\Rightarrow} \lambda B \rho \qquad B \overset{\ell_2}{\Rightarrow} \beta}{\alpha \overset{\ell_1+\ell_2}{\Rightarrow} \lambda \beta \rho,}$$

Пример 5.5. Да разгледаме безконтекстната граматика G, която има следните правила:

$$S \to_G AS \mid \varepsilon$$
$$A \to_G aAb \mid ab.$$

Да видим защо думата $aabbab \in \mathcal{L}(G)$. Ако следваме формално правилата за извод, получаваме следното:

За момента не е ясно как можем да проверим, че примерно думата $abbba \notin \mathcal{L}(G)$. Този въпрос ще разгледаме по-нататък.

$$\dfrac{\dfrac{(S, AS) \in R}{S \Rightarrow_G AS} \qquad \dfrac{\dfrac{\dfrac{(S, \varepsilon) \in R}{S \Rightarrow_G \varepsilon} \qquad \dfrac{}{\varepsilon \overset{0}{\Rightarrow}_G \varepsilon}}{S \overset{1}{\Rightarrow}_G \varepsilon} \qquad \dfrac{\dfrac{(S, AS) \in R}{S \Rightarrow_G AS} \qquad \dfrac{}{AS \overset{0}{\Rightarrow}_G AS}}{S \overset{1}{\Rightarrow}_G AS}\ (\textit{Тв. 5.5})}{S \overset{2}{\Rightarrow}_G A}\ (\textit{Тв. 5.2})}{AS \overset{2}{\Rightarrow}_G AA}\ (\text{транзитивност})$$
$$S \overset{3}{\Rightarrow}_G AA$$

Освен това имаме и следния формален извод:

$$\dfrac{\dfrac{(A, aAb) \in R}{A \Rightarrow_G aAb} \quad \dfrac{(A, ab) \in R}{aAb \overset{1}{\Rightarrow}_G aabb}}{A \overset{2}{\Rightarrow}_G aabb} \text{ (транзитивност)}$$

Аналогично,

$$\dfrac{\dfrac{(A, ab) \in R}{A \Rightarrow_G ab} \quad \dfrac{}{ab \overset{0}{\Rightarrow}_G ab}}{A \overset{1}{\Rightarrow}_G ab}$$

Обединявайки всичко, получаваме:

$$\dfrac{\dfrac{\dfrac{S \overset{3}{\Rightarrow}_G AA \qquad A \overset{2}{\Rightarrow}_G aabb}{S \overset{5}{\Rightarrow}_G aabbA} \text{ (Тв. 5.5)} \qquad A \overset{1}{\Rightarrow}_G ab}{S \overset{6}{\Rightarrow}_G aabbab} \text{ (Тв. 5.5)}$$

Можем да приложим правилата за извод в различен ред и пак да получим същия краен резултат. Например:

$$\dfrac{\dfrac{\dfrac{\dfrac{(S, AS) \in R}{S \Rightarrow_G AS} \quad \dfrac{A \overset{2}{\Rightarrow} aabb}{AS \overset{2}{\Rightarrow} aabbS} \text{ (Тв. 5.1)}}{S \overset{3}{\Rightarrow}_G aabbS} \qquad S \overset{2}{\Rightarrow}_G A}{S \overset{5}{\Rightarrow}_G aabbA} \text{ (Тв. 5.5)} \qquad A \overset{1}{\Rightarrow}_G ab}{S \overset{6}{\Rightarrow}_G aabbab} \text{ (Тв. 5.5)}$$

Естествено, ние няма да правим на ръка такива формални изводи в граматика, но е важно да придобием опит с механиката, чрез която се извършва един извод. Следващото твърдение ни дава едно важно свойство, което е вярно за безконтекстни граматики, но не и за неограничени граматики.

Лема 5.1 (Лема за разбиване на извода)**.** Нека G е безконтекстна граматика и нека γ_1 и γ_2 са непразни думи, за които $\gamma_1\gamma_2 \overset{\ell}{\Rightarrow} \beta$. Тогава съществуват числа ℓ_1, ℓ_2 и думи β_1 и β_2, за които: $\gamma_1 \overset{\ell_1}{\Rightarrow} \beta_1$ и $\gamma_2 \overset{\ell_2}{\Rightarrow} \beta_2$ и $\beta = \beta_1\beta_2$ и $\ell = \ell_1 + \ell_2$.

Тук $\gamma_1, \gamma_2, \beta \in (V \cup \Sigma)^\star$.

Доказателство. Индукция по дължината на извода ℓ.

- Нека $\ell = 0$. Тогава $\beta_1 = \gamma_1$ и $\beta_2 = \gamma_2$ и очевидно $\ell_1 = \ell_2 = 0$.
- Нека $\ell > 0$. Тогава да разгледаме следната ситуация:

$$\dfrac{\dfrac{(A, \alpha) \in R}{\lambda A\rho \Rightarrow_G \lambda\alpha\rho} \quad \lambda\alpha\rho \overset{\ell-1}{\Rightarrow} \beta}{\underbrace{\lambda A\rho}_{\gamma_1\gamma_2} \overset{\ell}{\Rightarrow} \beta}$$

Случаят, когато A е част от γ_2 е аналогичен.

Без ограничение на общността, нека променливата A е част от γ_1. Това означава, че можем да запишем думите γ_1 и γ_2 като $\gamma_1 = \lambda A\rho_1$ и $\rho = \rho_1\gamma_2$.

Сега можем да приложим индукционното предположение и да заключим, че съществува представяне на β като $\beta = \beta_1\beta_2$ със следния извод:

$$\dfrac{\dfrac{(A, \alpha) \in R}{\lambda A\rho_1 \Rightarrow_G \lambda\alpha\rho_1} \quad \dfrac{\overbrace{\lambda\alpha\rho_1}^{\gamma_1'}\gamma_2 \overset{\ell-1}{\Rightarrow} \beta}{\lambda\alpha\rho_1 \overset{\ell_1'}{\Rightarrow} \beta_1 \,\&\, \gamma_2 \overset{\ell_2}{\Rightarrow} \beta_2}}{\underbrace{\lambda A\rho_1}_{\gamma_1} \overset{\ell_1'+1}{\Rightarrow} \beta_1 \,\&\, \gamma_2 \overset{\ell_2}{\Rightarrow} \beta_2} \text{ (И.П.)}$$

□

Твърдение 5.6. Нека G е безконтекстна граматика и нека $X_1 \cdots X_k \overset{\ell}{\Rightarrow}_G \beta$, където $X_i \in V \cup \Sigma$ и $k \geq 2$. Тогава съществуват думи $\beta_1, \ldots, \beta_k$, такива че за $i = 1, \ldots, k$ е изпълнено, че $X_i \overset{\ell_i}{\Rightarrow} \beta_i$, където $\beta = \beta_1 \cdots \beta_k$ и $\ell = \sum_{i=1}^{k} \ell_i$.

Упътване. Пълна индукция по k като използвате *Лема 5.1*. □

Тук е възможно $X_i = a \in \Sigma$. Тогава $a \overset{0}{\Rightarrow} a$ и $\beta_i = a$.

Лема 5.2. Нека G е безконтекстна граматика, $X \in V \cup \Sigma$ и $\beta \in (V \cup \Sigma)^\star$. Тогава ако $X \overset{\star}{\Rightarrow} \beta$, то $X \overset{\star}{\lhd} \beta$.

Доказателство. С пълна индукция по ℓ ще докажем, че ако $X \overset{\ell}{\Rightarrow} \beta$, то $X \overset{\star}{\lhd} \beta$.

- $\ell = 0$, т.е. $X \overset{0}{\Rightarrow} X$. Тогава е ясно, че $X \overset{\star}{\lhd} X$.
- Нека $\ell > 0$ и $X \overset{\ell}{\Rightarrow} \beta$. Според правилата на извод в граматика имаме извода

$$\frac{X \to_G X_1 \cdots X_k \qquad X_1 \cdots X_k \overset{\ell-1}{\Rightarrow} \beta}{X \overset{\ell}{\Rightarrow} \beta} \text{ (транз.)}$$

Щом имаме, че $X_1 \cdots X_k \overset{\ell-1}{\Rightarrow} \beta$, от *Твърдение 5.6* знаем, че съществува разбиване на β на $k+1$ части, така че:

Естествено, че е възможно някои X_i да са букви от Σ. Тогава $\beta_i = X_i$ и $X_i \overset{0}{\Rightarrow}_G \beta_i$.

- $\beta = \beta_1 \cdots \beta_k$;
- $X_i \overset{\ell_i}{\Rightarrow} \beta_i$, за всяко $i = 1, \ldots, k$;
- $\ell - 1 = \sum_{i=1}^{k} \ell_i$.

Понеже $\ell_i < \ell$ за всяко $i = 1, \ldots, k$, получаваме следното:

$$\frac{X \to_G X_1 \cdots X_k \qquad \frac{X_1 \overset{\ell_1}{\Rightarrow} \beta_1}{X_1 \overset{\star}{\lhd} \beta_1} \text{ (И.П.)} \quad \cdots \quad \frac{X_k \overset{\ell_k}{\Rightarrow} \beta_k}{X_k \overset{\star}{\lhd} \beta_k} \text{ (И.П.)}}{X \overset{\star}{\lhd} \underbrace{\beta_1 \cdots \beta_k}_{\beta}} \text{ правило (1)}$$

□

Лема 5.3. Нека G е безконтекстна граматика, $X \in V \cup \Sigma$ и $\gamma \in (V \cup \Sigma)^\star$. Тогава ако $X \overset{\star}{\lhd} \gamma$, то $X \overset{\star}{\Rightarrow} \gamma$.

Доказателство. С пълна индукция по ℓ ще докажем, че ако $X \overset{\ell}{\lhd} \gamma$, то $X \overset{\star}{\Rightarrow} \gamma$.

- Нека $\ell = 0$. Това означава, че $X \overset{0}{\lhd} X$. Ясно е, че $X \overset{\star}{\Rightarrow} X$.
- Нека $\ell > 0$. Тогава имаме следното:

$$\frac{X \to_G X_1 \cdots X_n \qquad X_1 \overset{\ell_1}{\lhd} \gamma_1 \qquad \cdots \qquad X_n \overset{\ell_n}{\lhd} \gamma_n}{X \overset{\ell}{\lhd} \underbrace{\gamma_1 \cdots \gamma_n}_{\gamma}} \; (\ell = 1 + \sup\{\ell_1, \ldots, \ell_n\})$$

Понеже $\ell_i < \ell$ за всяко $i = 1, \ldots, n$, получаваме следното:

$$\dfrac{\dfrac{X_1 \overset{\ell_1}{\lhd} \gamma_1}{X_1 \overset{\star}{\Rightarrow} \gamma_1} \text{ (И.П.)} \qquad \cdots \qquad \dfrac{X_n \overset{\ell_n}{\lhd} \gamma_n}{X_n \overset{\star}{\Rightarrow} \gamma_n} \text{ (И.П.)}}{\dfrac{X \to_G X_1 \cdots X_n \qquad X_1 \cdots X_n \overset{\star}{\Rightarrow} \gamma_1 \cdots \gamma_n}{X \overset{\star}{\Rightarrow} \underbrace{\gamma_1 \cdots \gamma_n}_{\gamma}} \text{ (транз.)}} \text{ (Тв. 5.3)}$$

□

Комбинирайки предишните две леми получаваме следната теорема.

Теорема 5.1. Нека G е безконтекстна граматика, $X \in V \cup \Sigma$ и $\gamma \in (V \cup \Sigma)^\star$. Тогава $X \overset{\star}{\lhd} \gamma$ точно тогава, когато $X \overset{\star}{\Rightarrow} \gamma$.

Следващото твърдение е важно, но е трудно да не пропуснем доказателството му.

Твърдение 5.7. Безконтекстните езици са затворени относно операциите обединение, конкатенация и звезда на Клини.

Оттук можем да извлечем второ доказателство на твърдението, че всеки регулярен език е безконтекстен.

Твърдение 5.8. Всеки регулярен език е безконтекстен.

Упътване. Можем да подходим поне по два начина.

- Да проведем индукция по построението на регулярните езици и да докажем така, че всеки регулярен език е безконтекстен.
- Просто да съобразим, че всяка регулярна граматика е всъщност и безконтекстна.

□

5.4 Регулярни граматики

Сега ще разгледаме граматики с такъв вид правила, които пораждат точно регулярните (или еквивалентно автоматни) езици. Граматиката $G = \langle V, \Sigma, R, S\rangle$ се нарича **регулярна граматика**, ако всички правила са от вида

$$A \to aB,$$
$$A \to \varepsilon,$$

за произволни $A, B \in V$ и $a \in \Sigma$.

Също така се наричат граматики от тип 3 в йерархията на Чомски [10, стр. 217]. Този вид граматики понякога се нарича и дясно-регулярна граматика.

Лема 5.4. За всяка регулярна граматика G съществува недетерминиран краен автомат $\mathcal{N}$, такъв че $\mathcal{L}(G) = \mathcal{L}(\mathcal{N})$.

Упътване. Нека $G = \langle V, \Sigma, R, S\rangle$ и $V = \{A_0, \ldots, A_k\}$, където $S = A_0$. Тогава дефинираме $\mathcal{N}$ по следния начин:

- $Q \stackrel{\text{деф}}{=} \{q_0, \ldots, q_k\}$;
- $Q_{\texttt{start}} \stackrel{\text{деф}}{=} \{q_0\}$;
- $F \stackrel{\text{деф}}{=} \{q_i \mid A_i \to \varepsilon\}$;
- $\Delta(q_i, a) \stackrel{\text{деф}}{=} \{q_j \mid A_i \to aA_j$ е правило в граматиката$\}$.

Докажете, че $\mathcal{L}(\mathcal{N}) = \mathcal{L}(G)$. □

Лема 5.5. За всеки детерминиран краен автомат $\mathcal{A}$ съществува регулярна граматика G, такава че $\mathcal{L}(\mathcal{A}) = \mathcal{L}(G)$.

Упътване.
Използваме *Твърдение* **??**.
Нека $\mathcal{A} = \langle \Sigma, Q, q_{\texttt{start}}, \delta, F\rangle$ и $Q = \{q_0, \ldots, q_k\}$, където $q_{\texttt{start}} = q_0$. Тогава дефинираме $G = \langle V, \Sigma, R, S\rangle$ по следния начин:

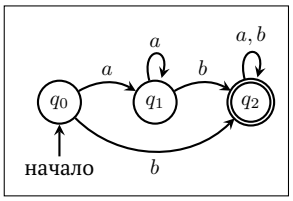
Тази конструкция може да се приложи и за недетерминиран автомат. Единствено трябва да се внимава, ако автоматът има много начални състояния.

- $V \stackrel{\text{деф}}{=} \{A_0, \ldots, A_k\}$;
- $S \stackrel{\text{деф}}{=} A_0$;
- $A_i \to aA_j \stackrel{\text{деф}}{\Leftrightarrow} \delta(q_i, a) = q_j$;
- $A_i \to \varepsilon \stackrel{\text{деф}}{\Leftrightarrow} q_i \in F$.

Докажете, че $\mathcal{L}(\mathcal{A}) = \mathcal{L}(G)$. □

Теорема 5.2. Един език е регулярен точно тогава, когато се поражда от регулярна граматика.

Пример 5.6. Да разгледаме отново автомата от *Фигура* **??**.



Фигура 5.5: $\mathcal{L}(\mathcal{A}) = \mathcal{L}[\![\boldsymbol{a^\star b(a+b)^\star}]\!]$.

Регулярна граматика G за езика $\mathcal{L}(\mathcal{A})$ можем да дефинираме така:

- $\Sigma = \{a, b\}$;
- На всяко състояние q_i на автомата ще съответства променливата A_i, т.е. $V = \{A_0, A_1, A_2\}$;
- Началната променлива е A_0, защото q_0 е началното състояние на автомата;
- Правилата следват дефиницията на δ функцията:

$$A_0 \to aA_1 \mid bA_2$$
$$A_1 \to aA_1 \mid bA_2$$
$$A_2 \to aA_2 \mid bA_2 \mid \varepsilon.$$

Допълнителни задачи

Задача 5.2. Граматиката $G = (V, \Sigma, R, S)$ се нарича **обобщено дясно-регулярна**, ако всички правила са от вида

$$A \to \omega B,$$
$$A \to \omega$$

за произволни $A, B \in V$ и $\omega \in \Sigma^\star$. Докажете, че един език L е автоматен точно тогава, когато L може да се опише с обобщена дясно-регулярна граматика.

Задача 5.3. Граматиката $G = (V, \Sigma, R, S)$ се нарича **обобщено ляво-регулярна**, ако всички правила са от вида

$$A \to B\omega,$$
$$A \to \omega$$

за произволни $A, B \in V$ и $\omega \in \Sigma^\star$. Докажете, че един език L е автоматен точно тогава, когато L може да се опише с обобщена ляво-регулярна граматика.

Глава 6

Машини на Тюринг

> Turing's 'Machines'. These machines are humans who calculate. [22, § 1096].

Както при крайните автомати, ще започнем с разглеждането на детерминираните машини на Тюринг. След това ще разгледаме по-общия модел на недетерминираните машини на Тюринг и ще видим, че те разпознават същия клас от езици.

Тук до голяма степен следваме [21, Глава 3]. Понятието за машина на Тюринг има много еквивалентни дефиниции.

6.1 Детерминирани машини на Тюринг

Детерминирана машина на Тюринг ще наричаме осморка от вида

$$\mathcal{M} = \langle Q, \Sigma, \Gamma, \delta, \sqcup, q_{\texttt{start}}, q_{\texttt{accept}}, q_{\texttt{reject}} \rangle,$$

където:

- Q - крайно множество от състояния;
- Σ - крайна азбука за входа;
- Γ - крайна азбука за лентата, $\Sigma \subseteq \Gamma$;
- $\sqcup$ - символ за празна клетка на лентата, $\sqcup \in \Gamma \setminus \Sigma$;
- $q_{\texttt{start}} \in Q$ - начално състояние;
- $q_{\texttt{accept}} \in Q$ - приемащо състояние;

Тези две състояния ще наричаме заключителни

- $q_{\texttt{reject}} \in Q$ - отхвърлящо състояние, където $q_{\texttt{accept}} \neq q_{\texttt{reject}}$;
- $\delta : Q' \times \Gamma \to Q \times \Gamma \times \{\lhd, \rhd, \square\}$ - тотална функция на преходите, където $Q' = Q \setminus \{q_{\texttt{accept}}, q_{\texttt{reject}}\}$.

Това означава, че веднъж достигнем ли заключително състояние, не можем да правим повече преходи. Тук следваме [21, стр. 169] и [9, стр. 327].

Всяка машина на Тюринг разполага с неограничено количество памет, която е представена като безкрайна (и в двете посоки) лента, разделена на клетки. Всяка клетка съдържа елемент на Γ. Сега ще опишем как $\mathcal{M}$ работи върху вход думата $\alpha \in \Sigma^\star$. Първоначално безкрайната лента съдържа само думата α. Останалите клетки на лентата съдържат символа $\sqcup$.

Освен това, $\mathcal{M}$ се намира в началното състояние $q_{\texttt{start}}$ и главата за четене е върху най-левия символ на α. Работата на $\mathcal{M}$ е описана от функцията на преходите δ.

На англ. instanteneous description. Понякога за удобство ще означаваме моментната конфигурация като $(q, \alpha\underline{x}\beta)$ вместо по-неудобното $(\alpha, q, x\beta)$.

- Формално, **моментната конфигурация** (или описание) на едно изчисление на машина на Тюринг е тройка от вида
$$(\alpha, q, x\beta) \in \Gamma^\star \times Q \times \Gamma^+,$$
като интерпретацията на тази тройка е, че машината се намира в състояние q и лентата има вида
$$\cdots \sqcup \sqcup \sqcup\ \alpha\underline{x}\beta\ \sqcup \sqcup \sqcup \cdots,$$
където четящата глава на машината е поставена върху x.

Причината за да искаме да имаме $\sqcup$ след думата α за да може лесно да работим със случая, когато $\alpha = \varepsilon$.

- Макар и да имаме безкрайна лента, моментната конфигурация, която може да се представи като *крайна* дума, описва цялото моментно състояние на машината на Тюринг.
- **Началната конфигурация** за входната дума $\omega \in \Sigma^\star$ представлява тройката
$$(\varepsilon, q_{\texttt{start}}, \omega\sqcup).$$
Удобно е да положим
$$\texttt{init}_{\mathcal{M}}(\omega) \stackrel{\text{деф}}{=} (\varepsilon, q_{\texttt{start}}, \omega\sqcup).$$
- **Приемаща конфигурация** представлява тройка от вида
$$(\lambda, q_{\texttt{accept}}, \rho),$$
за произволни $\lambda \in \Gamma^\star$ и $\rho \in \Gamma^+$. Удобно е да положим
$$\texttt{Accept}_{\mathcal{M}} \stackrel{\text{деф}}{=} \{(\lambda, q_{\texttt{accept}}, \rho) \mid \lambda \in \Gamma^\star \ \&\ \rho \in \Gamma^+\}.$$

- **Отхвърляща конфигурация** представлява тройка от вида

$$(\lambda, q_{\texttt{reject}}, \rho),$$

за произволни $\lambda \in \Gamma^\star$ и $\rho \in \Gamma^+$. Удобно е да положим

$$\texttt{Reject}_{\mathcal{M}} \stackrel{\text{деф}}{=} \{(\lambda, q_{\texttt{reject}}, \rho) \mid \lambda \in \Gamma^\star \,\&\, \rho \in \Gamma^+\}.$$

- Една конфигурация ще наричаме **заключителна**, ако тя е или приемаща или отхвърляща.

Както за автомати, удобно е да дефинираме бинарна релация $\vdash_{\mathcal{M}}$ над множеството $\Gamma^\star \times Q \times \Gamma^+$, която ще казва как моментната конфигурация на машината $\mathcal{M}$ се променя при изпълнение на една стъпка.

За да направим това, удобно е първо да дефинираме бинарната релация $\vdash_{y,d}$ над множеството $\Gamma^\star \times \Gamma^+$, която показва как една моментна конфигурация се променя, когато заменим символа на главата с y и се придвижим на посока $d \in \{\triangleleft, \triangleright, \square\}$.

Дефиницията на релацията $\vdash_{y,d}$ не зависи от конкретна машина на Тюринг!

$$\frac{x,y,z \in \Gamma \qquad \lambda,\rho \in \Gamma^\star}{(\lambda, xz\rho) \vdash_{y,\triangleright} (\lambda y, z\rho)} \qquad \frac{x,y \in \Gamma \qquad \lambda \in \Gamma^\star}{(\lambda, x) \vdash_{y,\triangleright} (\lambda y, \sqcup)}$$

$$\frac{x,y,z \in \Gamma \qquad \lambda,\rho \in \Gamma^\star}{(\lambda z, x\rho) \vdash_{y,\triangleleft} (\lambda, zy\rho)} \qquad \frac{x,y \in \Gamma \qquad \rho \in \Gamma^\star}{(\varepsilon, x\rho) \vdash_{y,\triangleleft} (\varepsilon, \sqcup y\rho)}$$

$$\frac{x,y \in \Gamma \qquad \lambda,\rho \in \Gamma^\star}{(\lambda, x\rho) \vdash_{y,\square} (\lambda, y\rho)}$$

Фигура 6.1: Дефиниция на релацията $\vdash_{y,d}$.

Сега вече сме готови да дефинираме релацията $\vdash_{\mathcal{M}}$.

Ако няма опасност да се заблудим за коя точно машина на Тюринг $\mathcal{M}$ говорим, то е възможно да пишем просто $\vdash$ вместо $\vdash_{\mathcal{M}}$.

$$\frac{\delta(q,x) = (q',y,d) \qquad (\lambda, x\rho) \vdash_{y,d} (\lambda', \rho')}{(\lambda, q, x\rho) \vdash_{\mathcal{M}} (\lambda', q', \rho')}$$

Фигура 6.2: Преход в еднолентова детерминистична машина на Тюринг $\mathcal{M}$

Сега за всяко естествено число ℓ, ще дефинираме релацията $\vdash^{\ell}_{\mathcal{M}}$, която ще казва, че от конфигурацията κ можем да достигнем до конфигурацията κ' за ℓ на брой стъпки.

$$\frac{}{\kappa \vdash^{0}_{\mathcal{M}} \kappa} \text{ (рефлексивност)} \qquad \frac{\kappa \vdash_{\mathcal{M}} \kappa'' \qquad \kappa'' \vdash^{\ell}_{\mathcal{M}} \kappa'}{\kappa \vdash^{\ell+1}_{\mathcal{M}} \kappa'} \text{ (транзитивност)}$$

- С $\vdash^{\star}_{\mathcal{M}}$ ще означаваме рефлексивното и транзитивно затваряне на релацията $\vdash$ или с други думи,

$$\kappa \vdash^{\star}_{\mathcal{M}} \kappa' \stackrel{\text{деф}}{\Leftrightarrow} (\exists \ell \in \mathbb{N})[\kappa \vdash^{\ell}_{\mathcal{M}} \kappa'].$$

- Макар и една конфигурация κ да преставлява тройка, то често ще бъде удобно да гледаме на κ като на дума от езика $\Gamma^\star Q \Gamma^+$.
- Важно свойство е, че ако $\kappa \vdash^\star_{\mathcal{M}} \kappa'$, то $|\kappa| \leq |\kappa'|$.

Важно е да имаме $\sqcup$ след думата ω, защото е възможно ω да е празната дума.

- машината на Тюринг $\mathcal{M}$ **приема** думата ω, ако

$$(\exists \kappa \in \texttt{Accept}_{\mathcal{M}})[\ \texttt{init}_{\mathcal{M}}(\omega) \vdash^\star_{\mathcal{M}} \kappa\ ].$$

- Машината на Тюринг $\mathcal{M}$ **отхвърля** думата ω, ако

$$(\exists \kappa \in \texttt{Reject}_{\mathcal{M}})[\ \texttt{init}_{\mathcal{M}}(\omega) \vdash^\star_{\mathcal{M}} \kappa\ ].$$

- Машината на Тюринг $\mathcal{M}$ **не приема** думата ω, ако $\mathcal{M}$ отхвърля ω или $\mathcal{M}$ никога не завършва при начална конфигурация $\texttt{init}_{\mathcal{M}}(\alpha)$.

На англ. такава машина на Тюринг се нарича **decider** [21, стр. 170]. Може такива машини на Тюринг да се наричат и тотални [13, стр. 213]. Да се внимава, че в Манев понятията са различни.

- Една машина на Тюринг се нарича **разрешител**, ако при всеки вход достига до заключително състояние, т.е. достига до приемаща или до отхвърляща конфигурация.
- Езикът, който се **разпознава** от машината $\mathcal{M}$ е:

$$\mathcal{L}(\mathcal{M}) \stackrel{\text{деф}}{=} \{\omega \in \Sigma^\star \mid (\exists \kappa \in \texttt{Accept}_{\mathcal{M}})[\texttt{init}_{\mathcal{M}}(\omega) \vdash^\star_{\mathcal{M}} \kappa]\}.$$

На англ. **semidecidable language**. В литературата се използва и названието **рекурсивно номеруем език**.

- Езикът L се нарича **полуразрешим**, ако съществува машина на Тюринг $\mathcal{M}$, за която $L = \mathcal{L}(\mathcal{M})$. Ако една дума $\omega \in L$, то след крайно много стъпки ще достигнем до състоянието $q_{\texttt{accept}}$. Ако $\omega \notin L$, то не е ясно какво се случва с изчислението на $\mathcal{M}$ върху ω. Възможно е да достигнем до състоянието $q_{\texttt{reject}}$, но може да попаднем в безкрайно изчисление.

На англ. **decidable language**. В литературата се използва и названието **рекурсивен език**.

- Един език L се нарича **разрешим**, ако за него съществува *разрешител* $\mathcal{M}$, за която $L = \mathcal{L}(\mathcal{M})$. В този случай се казва, че $\mathcal{M}$ разрешава езика L.

Твърдение 6.1. Ако L е разрешим език над азбуката Σ, то $\Sigma^\star \setminus L$ също е разрешим език.

Упътване. Просто разменяме $q_{\texttt{accept}}$ и $q_{\texttt{reject}}$. □

От дефинициите е ясно, че всеки разрешим език е полуразрешим. По-късно, ще видим, че съществуват полуразрешими езици, чиито допълнения не са полуразрешими, т.е. не всеки полуразрешим език е разрешим. Една от основните ни задачи ще бъде да класифицираме различни езици като (не)разрешими и (не)полуразрешими. За да придобием по-добра интуиция за тези нови понятия, ще разгледаме подробно няколко примера. Ще видим също как можем да изобразяваме функцията на преходите на $\mathcal{M}$ графично.

Примери

Пример 6.1. Да разгледаме езика $L = \{a^n b^n c^n \mid n \in \mathbb{N}\}$. Нека да видим защо този език е разрешим. Идеята на алгоритъма, който ще разгледаме е да маркира на всяка итерация по един символ a, b или c. Той завършва успешно, ако всички символи на думата са маркирани. Да въведем нов символ x, с който ще маркираме обработените от входната дума символи a, b, c. Нека първоначално думата е копирана върху лентата и четящата глава е върху първия символ на думата.

Вече сме срещали езикът L много пъти. Знаем много добре, че L не е безконтекстен, но е контекстен.
Алгоритъмът има времева сложност $\mathcal{O}(n^2)$. Ако разгледаме машина на Тюринг с три ленти, то ще получим времева сложност $\mathcal{O}(n)$.

(1) Чете x-ове надясно по лентата докато срещне първото a и го замества с x. Отива на стъпка (2). Ако символите свършат (т.е. достигне се $\sqcup$) преди да се достигне a, то алгоритъмът завършва успешно.

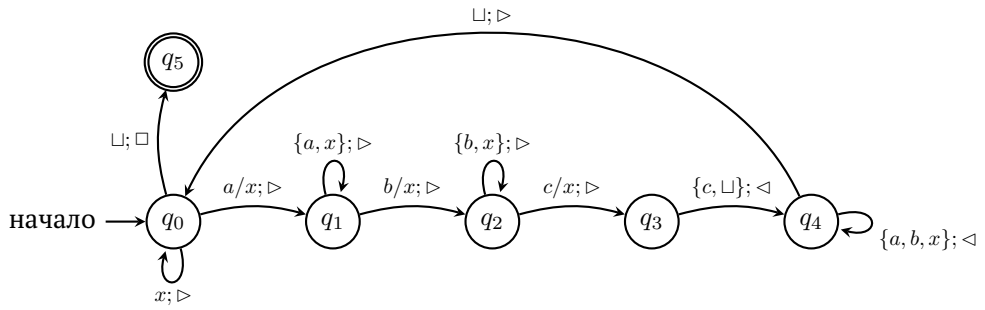
(2) Чете x-ове надясно по лентата докато срещне първото b и го замества с x. Отива на стъпка (3).

(3) Чете x-ове надясно по лентата докато срещне първото c и го замества с x.

(4) Връща четящата глава в началото на лентата, т.е. чете наляво докато не срещне символа $\sqcup$. Връща се в стъпка (1).

Нека сега да видим, че този алгоритъм може да се опише съвсем формално с машина на Тюринг. Ще построим машина на Тюринг $\mathcal{M}$, за която $L = \mathcal{L}(\mathcal{M})$, където

- $\Sigma = \{a, b, c\}$;
- $\Gamma = \{a, b, c, x, \sqcup\}$, за някой нов символ x;
- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$, като $q_{\texttt{start}} = q_0$ и $q_{\texttt{accept}} = q_5$;
- Частичната функция на преходите $\delta : (Q \setminus \{q_5\}) \times \Gamma \to Q \times \Gamma \times \{\triangleleft, \triangleright, \square\}$ е описана на схемата отдолу. Остава да добавим състоянието $q_{\texttt{reject}}$.

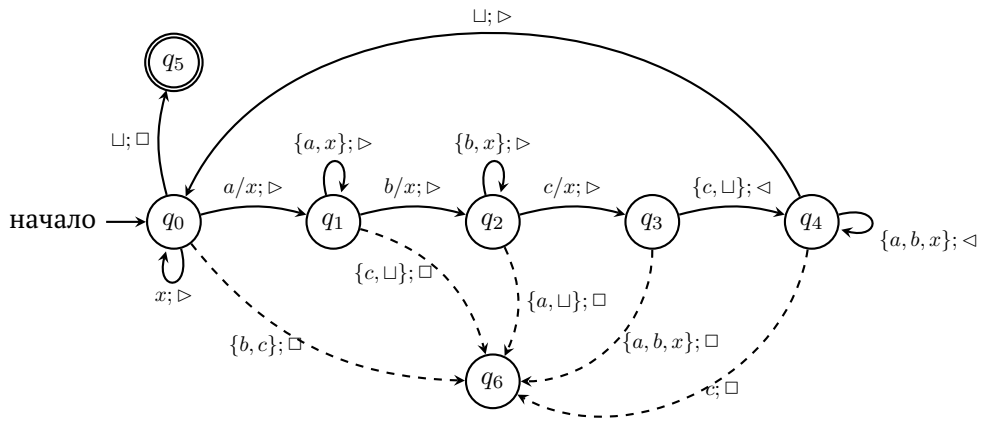


Фигура 6.4: Част от детерминистична машина на Тюринг $\mathcal{M}$, за която разрешава езика $\{a^n b^n c^n \mid n \in \mathbb{N}\}$. Остава да се добави $q_{\texttt{reject}}$ и да се попълни δ функцията до тотална.

Горната схема определя точно функцията на преходите δ. Например,

$$\delta(q_0, a) = (q_1, x, \triangleright)$$
$$\delta(q_4, \sqcup) = (q_0, \sqcup, \triangleright)$$
$$\delta(q_1, a) = (q_1, a, \triangleright)$$
$$\delta(q_1, x) = (q_1, x, \triangleright).$$

Съобразете, че тази машина на Тюринг може да се направи тотална като се добави ново състояние $q_6 = q_{\texttt{reject}}$ и за всяка двойка (q, z), за която функцията на преходите не е дефинирана, да сочи към $q_{\texttt{reject}}$. Така получаваме пълното описание на детерминистична машина на Тюринг $\mathcal{M}$, за която $\mathcal{L}(\mathcal{M}) = L$. Лесно се съобразява, че тази машина на Тюринг е *тотална*, т.е. за всеки вход $\mathcal{M}$ завършва в $q_{\texttt{accept}}$ или $q_{\texttt{reject}}$. Заключаваме, че L е не само полуразрешим, но *разрешим* език.



Фигура 6.5: Детерминирана машина на Тюринг $\mathcal{M}$ разрешаваща $\{a^n b^n c^n \mid n \in \mathbb{N}\}$. Отхвърлящото състояние е q_6, т.е. $q_{\texttt{reject}} \stackrel{\text{деф}}{=} q_6$.

Да напомним, че този език не е безконтекстен. В [10, стр. 155] е дадено по-различно решение. Тук следваме [21, стр. 173]. Там има малка грешка.

Пример 6.2. Да разгледаме езика $L = \{\omega\sharp\omega \mid \omega \in \{a, b\}^\star\}$. Първо неформално ще опишем алгоритъм, който да разпознава думите на езика L. Нека една дума е копирана върху лентата и четящата глава е поставена върху първия символ от думата.

(1) Чете x-ове надясно по лентата докато не срещне a или b и го замества с x. Запомня дали сме срещнали a или b. Ако вместо a или b срещне $\sharp$, то отива на стъпка (6). Това запаметяване става в състоянията на машината на Тюринг.

(2) Чете a-та и b-та надясно по лентата докато не стигне $\sharp$.

(3) Чете символа $\sharp$ надясно по лентата и всички следващи x-ове докато не срещне символа a или b. Той трябва да е същия символ, който сме запаметили на стъпка (1). Заместваме този символ с x.

(4) Чете x-ове наляво по лентата докато не стигне $\sharp$.

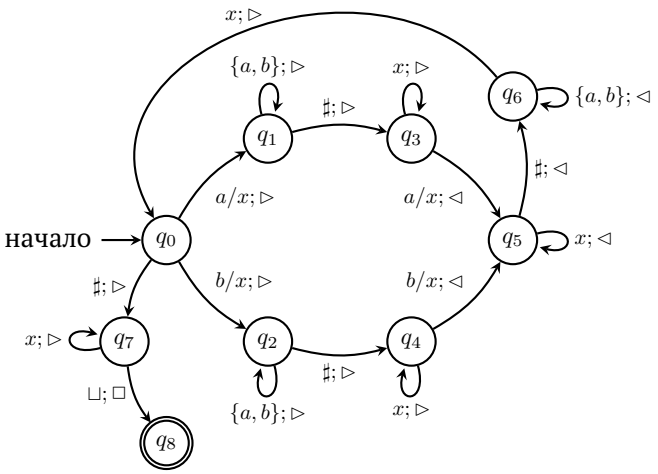
(5) Чете a-та и b-та по лентата докато не стигне x. Поставя четящата глава върху символа точно след първия x. Отива на стъпка (1).

(6) Прочита $\sharp$ надясно по лентата и чете надясно x-ове докато не срещне $\sqcup$. Алгоритъмът завършва успешно.

Обърнете внимание, че този алгоритъм има времева сложност $\mathcal{O}(n^2)$. Ще видим в Пример 6.3, че ако разгледаме двулентова машина на Тюринг, то имаме алгоритъм със сложност $\mathcal{O}(n)$.

Ще построим машина на Тюринг $\mathcal{M}$, за която $L = \mathcal{L}(\mathcal{M})$.

- $\Sigma \stackrel{\text{деф}}{=} \{a, b, \sharp\}$;
- $\Gamma \stackrel{\text{деф}}{=} \{a, b, \sharp, x, \sqcup\}$;
- $Q \stackrel{\text{деф}}{=} \{q_0, q_1, \ldots, q_8\}$, като $q_{\texttt{start}} \stackrel{\text{деф}}{=} q_0$ и $q_{\texttt{accept}} \stackrel{\text{деф}}{=} q_8$;



Фигура 6.6: Част от машина на Тюринг $\mathcal{M}$, която разрешава езика $\{\omega\sharp\omega \mid \omega \in \{a,b\}^\star\}$.

Да проследим изчислението на думата $ab\sharp ab$.

$$\begin{aligned}(q_0, \underline{a}b\sharp ab\sqcup) &\vdash_{\mathcal{M}} (q_1, x\underline{b}\sharp ab\sqcup) \vdash_{\mathcal{M}} (q_1, xb\underline{\sharp}ab\sqcup) \vdash_{\mathcal{M}} (q_3, xb\sharp\underline{a}b\sqcup)\\ &\vdash_{\mathcal{M}} (q_5, xb\underline{\sharp}xb\sqcup) \vdash_{\mathcal{M}} (q_6, x\underline{b}\sharp xb\sqcup) \vdash_{\mathcal{M}} (q_6, \underline{x}b\sharp xb\sqcup)\\ &\vdash_{\mathcal{M}} (q_0, x\underline{b}\sharp xb\sqcup) \vdash_{\mathcal{M}} (q_2, xx\underline{\sharp}xb\sqcup) \vdash_{\mathcal{M}} (q_4, xx\sharp\underline{x}b\sqcup)\\ &\vdash_{\mathcal{M}} (q_4, xx\sharp x\underline{b}\sqcup) \vdash_{\mathcal{M}} (q_5, xx\sharp\underline{x}x\sqcup) \vdash_{\mathcal{M}} (q_5, xx\underline{\sharp}xx\sqcup)\\ &\vdash_{\mathcal{M}} (q_6, x\underline{x}\sharp xx\sqcup) \vdash_{\mathcal{M}} (q_0, xx\underline{\sharp}xx\sqcup) \vdash_{\mathcal{M}} (q_7, xx\sharp\underline{x}x\sqcup)\\ &\vdash_{\mathcal{M}} (q_7, xx\sharp x\underline{x}\sqcup) \vdash_{\mathcal{M}} (q_7, xx\sharp xx\underline{\sqcup}) \vdash_{\mathcal{M}} (q_8, xx\sharp xx\underline{\sqcup}).\end{aligned}$$

Може лесно да се съобрази, че тази машина на Тюринг може да се допълни до *тотална*.

Задача 6.1. Докажете, че езикът $L = \{\, \omega \cdot \omega \mid \omega \in \{a,b\}^\star \,\}$ е разрешим.

6.2 Многолентови машини на Тюринг

Тук ще дефинираме един по-общ модел на машина на Тюринг, при който позволяваме да имаме няколко ленти, а не само една. Ще видим, че този моделът с много на брой ленти има същата изразителна сила както и модела с една лента.

Многолентови детерминирани машини на Тюринг

Детерминирана машина на Тюринг с k ленти има същата дефиниция като еднолентова машина на Тюринг с единствената разлика, че функцията на преходите δ приема следния вид:

$$\delta : Q' \times \Gamma^k \to Q \times \Gamma^k \times \{\triangleleft, \triangleright, \square\}^k,$$

където $Q' = Q \setminus \{q_{\texttt{accept}}, q_{\texttt{reject}}\}$. Това означава, че имаме k на брой ленти, всяка с отделна четяща глава, която се движи независимо от другите.

Тук означаваме $\vec{a} = (a_1, \ldots, a_k)$.

За две k-орки от думи $\vec{\alpha}$ и $\vec{\beta}$, дефинираме k-орката

$$\vec{\alpha} \cdot \vec{\beta} \stackrel{\text{деф}}{=} (\alpha_1 \cdot \beta_1, \alpha_2 \cdot \beta_2, \ldots, \alpha_k \cdot \beta_k).$$

Сега дефинираме релацията $\vdash_{\mathcal{M}}$ над множеството $(\Gamma^\star)^k \times Q \times (\Gamma^+)^k$ по следния начин:

$$\frac{\delta(q, \vec{x}) = (q', \vec{y}, \vec{d}) \qquad \forall i < k : (\lambda_i, x_i\rho_i) \vdash_{y_i, d_i} (\lambda_i', \rho_i')}{(\vec{\lambda}, q, \vec{x} \cdot \vec{\rho}) \vdash_{\mathcal{M}} (\vec{\lambda}', q', \vec{\rho}')}$$

Фигура 6.7: Едностъпков преход в k-лентова детерминистична машина на Тюринг

Приемаме, че първоначално входната дума α е записана върху първата лента, а всички останали ленти са празни, т.е. запълнени са със символа $\sqcup$. Удобно да положим

$$\texttt{init}_{\mathcal{M}}(\omega) \stackrel{\text{деф}}{=} (\underbrace{\varepsilon, \ldots, \varepsilon}_{k}, q_{\texttt{start}}, \underbrace{\omega\sqcup, \sqcup, \ldots, \sqcup}_{k}),$$

с което означаваме началната конфигурация при входна дума ω. Удобно е да положим също така и множествтото на приемащите конфигурации.

$$\texttt{Accept}_{\mathcal{M}} \stackrel{\text{деф}}{=} \{(\lambda_1, \ldots, \lambda_k, q_{\texttt{accept}}, \rho_1, \ldots, \rho_k) \mid \lambda_i \in \Gamma^\star \ \&\ \rho_i \in \Gamma^+ \text{ за } i = 1, \ldots, k\}.$$

Аналогично, можем да дефинираме и множеството на отхвърлящите конфигурации.

$$\texttt{Reject}_{\mathcal{M}} \stackrel{\text{деф}}{=} \{(\lambda_1, \ldots, \lambda_k, q_{\texttt{reject}}, \rho_1, \ldots, \rho_k) \mid \lambda_i \in \Gamma^\star \ \&\ \rho_i \in \Gamma^+ \text{ за } i = 1, \ldots, k\}.$$

Тогава

$$\mathcal{L}(\mathcal{M}) \stackrel{\text{деф}}{=} \{\, \omega \in \Sigma^\star \mid (\exists \kappa \in \texttt{Accept}_{\mathcal{M}})[\, \texttt{init}_{\mathcal{M}}(\omega) \vdash_{\mathcal{M}}^\star \kappa \,] \,\}.$$

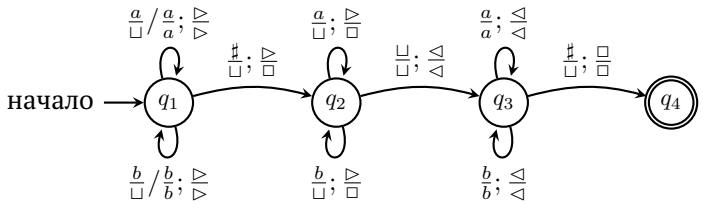
Примери

Пример 6.3. Да видим как двулентова машина на Тюринг разрешава езика

$$L = \{\, \omega\sharp\omega \mid \omega \in \{a, b\}^\star \,\}.$$

- $Q = \{q_1, q_2, q_3, q_4, q_5\}$;
- $q_{\texttt{start}} = q_1$, $q_{\texttt{accept}} = q_4$ и $q_{\texttt{reject}} = q_5$;

- $\Sigma \stackrel{\text{деф}}{=} \{a, b, \sharp\}$;
- $\Gamma \stackrel{\text{деф}}{=} \{a, b, \sharp, \sqcup\}$;
- $\delta : Q' \times \Gamma^2 \to Q \times \Gamma^2 \times \{\triangleleft, \triangleright, \square\}^2$, където $Q' = \{q_1, q_2, q_3\}$.



Фигура 6.8: Непълно описание на двулентова детерминирана машина на Тюринг $\mathcal{M}$, за която $\mathcal{L}(\mathcal{M}) = \{\omega\sharp\omega \mid \omega \in \{a, b\}^\star\}$. Всички неописани преходи трябва да сочат към отхвърлящото състояние q_5.

В началото втората лента е празна. Имаме две глави, които се движат независимо една от друга. При вход думата $\omega\sharp\omega$, $\mathcal{M}$ първо копира ω върху втората лента. След това сравнява това, което е записано на втората лента с думата, която следва след символа $\sharp$. Лесно се съобразява, че сега сложността на изчислението е $\mathcal{O}(n)$, докато при еднолентова машина на Тюринг то беше $\mathcal{O}(n^2)$. Да разгледаме един пример:

$$\begin{aligned}
(q_1, \tfrac{\hat{a}}{\hat{\sqcup}}\tfrac{b}{\sqcup}\tfrac{\sharp}{\sqcup}\tfrac{a}{\sqcup}\tfrac{b}{\sqcup}\tfrac{\sqcup}{\sqcup}) &\vdash (q_1, \tfrac{a}{a}\tfrac{\hat{b}}{\hat{\sqcup}}\tfrac{\sharp}{\sqcup}\tfrac{a}{\sqcup}\tfrac{b}{\sqcup}\tfrac{\sqcup}{\sqcup}) \vdash (q_1, \tfrac{a}{a}\tfrac{b}{b}\tfrac{\hat{\sharp}}{\hat{\sqcup}}\tfrac{a}{\sqcup}\tfrac{b}{\sqcup}\tfrac{\sqcup}{\sqcup}) \\
&\vdash (q_2, \tfrac{a}{a}\tfrac{b}{b}\tfrac{\sharp}{\hat{\sqcup}}\tfrac{\hat{a}}{\sqcup}\tfrac{b}{\sqcup}\tfrac{\sqcup}{\sqcup}) \vdash (q_2, \tfrac{a}{a}\tfrac{b}{b}\tfrac{\sharp}{\hat{\sqcup}}\tfrac{\hat{a}}{\sqcup}\tfrac{b}{\sqcup}\tfrac{\sqcup}{\sqcup}) \\
&\vdash (q_2, \tfrac{a}{a}\tfrac{b}{b}\tfrac{\sharp}{\hat{\sqcup}}\tfrac{a}{\sqcup}\tfrac{\hat{b}}{\sqcup}\tfrac{\sqcup}{\sqcup}) \vdash (q_2, \tfrac{a}{a}\tfrac{b}{b}\tfrac{\sharp}{\hat{\sqcup}}\tfrac{a}{\sqcup}\tfrac{b}{\sqcup}\tfrac{\hat{\sqcup}}{\sqcup}) \\
&\vdash (q_3, \tfrac{a}{a}\tfrac{b}{\hat{b}}\tfrac{\sharp}{\sqcup}\tfrac{a}{\sqcup}\tfrac{\hat{b}}{\sqcup}\tfrac{\sqcup}{\sqcup}) \vdash (q_3, \tfrac{a}{\hat{a}}\tfrac{b}{b}\tfrac{\sharp}{\sqcup}\tfrac{\hat{a}}{\sqcup}\tfrac{b}{\sqcup}\tfrac{\sqcup}{\sqcup}) \\
&\vdash (q_3, \tfrac{\sqcup}{\hat{\sqcup}}\tfrac{a}{a}\tfrac{b}{b}\tfrac{\hat{\sharp}}{\sqcup}\tfrac{a}{\sqcup}\tfrac{b}{\sqcup}\tfrac{\sqcup}{\sqcup}) \vdash (q_4, \tfrac{\sqcup}{\hat{\sqcup}}\tfrac{a}{a}\tfrac{b}{b}\tfrac{\hat{\sharp}}{\sqcup}\tfrac{a}{\sqcup}\tfrac{b}{\sqcup}\tfrac{\sqcup}{\sqcup}).
\end{aligned}$$

Твърдение 6.2. За всяка k-лентова машина на Тюринг $\mathcal{M}$ съществува еднолентова машина на Тюринг $\mathcal{M}'$, такава че $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{M}')$.

Упътване. Нека $\mathcal{M}$ е k-лентова машина на Тюринг. Ще построим еднолентова машина на Тюринг $\mathcal{M}'$, за която $\mathcal{L}(\mathcal{M}) = \mathcal{L}(\mathcal{M}')$. Да означим $\hat{\Gamma} = \{\hat{x} \mid x \in \Gamma\}$. Тогава азбуката на лентата на $\mathcal{M}'$ ще бъде $\Gamma' = \Sigma \cup (\hat{\Gamma} \cup \Gamma)^k$. Сега вместо да имаме k ленти със символи от Γ, ще имаме една лента със символи k-орки от $\hat{\Gamma} \cup \Gamma$. За да симулираме $\mathcal{M}$, използваме символите $\hat{x}$ за да маркират позицията на главите на $\mathcal{M}$, като във всяка компонента на лентата има точно по един символ от вида $\hat{x}$. При вход думата $\alpha = a_1 a_2 \cdots a_n$, която е поставена върху единствената лента на $\mathcal{M}'$, в случая на $k = 2$, първата работа на $\mathcal{M}'$ е да замени α с думата

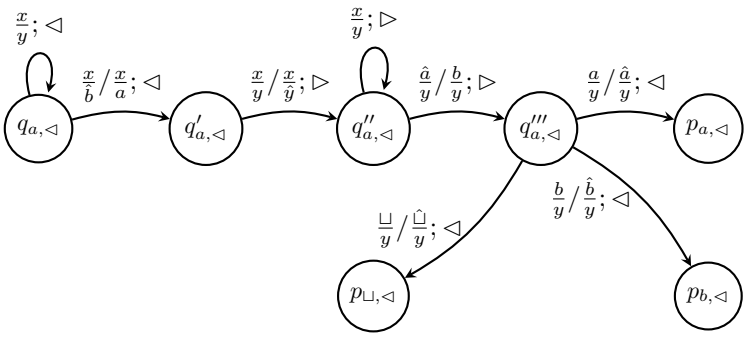
В [21, стр. 177] конструкцията е малко по-различна. Там съдържанието на всяка лента се поставя последователно върху една лента, като се разделят със специален символ. Тук следваме [10, стр. 162].

$$\frac{\hat{a}_1}{\hat{\sqcup}}\frac{a_2}{\sqcup}\cdots\frac{a_n}{\sqcup}$$

За да определим следващия ход на машината $\mathcal{M}'$, трябва да сканираме лентата докато не открием разположението на всичките k на брой маркирани клетки. Тогава симулираме ход на

$\mathcal{M}$ и отново трябва да променим маркираните клетки. □

Да видим по-подробно как можем да симулираме изчислението на двулентова машина на Тюринг $\mathcal{M}$ с еднолентовата машина на Тюринг $\mathcal{M}'$.



Фигура 6.9: Симулация на прехода $\delta_{\mathcal{M}}(q, \frac{a}{b}) = (p, \frac{b}{a}, \frac{\triangleright}{\triangleleft})$ при положение, че главата на втората лента е наляво спрямо главата на първата лента

- В еднолентовата машина на Тюринг, за удобство пишем $\frac{x}{y}$ вместо наредената двойка (x, y).

✍ Помислете как да обобщите тази идея за машина на Тюринг с n ленти. Важното е, че във всяко състояние кодираме крайна информация.

- За всяко състояние q на $\mathcal{M}$ и всяко a от Γ, в машината $\mathcal{M}'$ ще имаме състояния от вида $q_{a,\triangleleft}, q_{a,\triangleright}, q_{a,\square}$, които носят информацията, че в състояние q на симулираната двулентова машина на Тюринг $\mathcal{M}$, главата на първата лента е върху символа a и главата на втората лента е разположена наляво/надясно/на същата позиция спрямо главата на първата лента.
- Във Фигура 6.9, $\frac{x}{y}$ е съкратен запис за $\{\frac{x}{y} \mid x, y \in \Gamma\}$;
- Възможно е на всяка симулирана стъпка, главите на двете ленти да се раздалечават. Това означава, че след симулацията на s стъпки, в най-лошия случай, двете глави са на разстояние $2s$. Тогава за да симулираме $(s+1)$-вата стъпка на $\mathcal{M}$, първо трябва да отидем $2s$ стъпки наляво, после $2s$ стъпки надясно. Следователно $(s+1)$-вата стъпка на $\mathcal{M}$ се симулира за приблизително $4s$ стъпки.
- Ако изчислението на $\mathcal{M}$ върху вход α отнема s стъпки, то симулираното изчисление върху α ще отнеме в най-лошия случай приблизително

$$\sum_{i=0}^{s} 4i = 2s^2 + 2s$$

стъпки. Заключаваме, че за s стъпки от изчисленито на $\mathcal{M}$, симулацията върху $\mathcal{M}'$ отнема време $\mathcal{O}(s^2)$, т.е. имаме квадратично забавяне.

6.3 Изчислими функции

Преди да дефинираме какво означава една функция да бъде изчислима с машина на Тюринг, удобно е да положим

$$\texttt{Accept}_{\mathcal{M}}(\omega) = \{(\vec{\lambda}, \sqcup^m, q_{\texttt{accept}}, \vec{\rho}, \omega\sqcup^{n+1}) \mid \vec{\lambda} \in (\Gamma^\star)^{k-1} \;\&\; \vec{\rho} \in (\Gamma^+)^{k-1} \;\&\; m, n \in \mathbb{N}\}$$

Една *тотална* функция $f : \Sigma^\star \to \Sigma^\star$ се нарича изчислима с машина на Тюринг $\mathcal{M}$, ако за всяка дума $\omega \in \Sigma^\star$,

$$\texttt{init}_{\mathcal{M}}(\omega) \vdash^\star_{\mathcal{M}} \kappa, \text{ за някое } \kappa \in \texttt{Accept}_{\mathcal{M}}(f(\omega)).$$

Възможно е да се дефинира и за двулентова машина на Тюринг, като $f(\alpha)$ ще бъде върху втората лента.

Това означава, че машината на Тюринг $\mathcal{M}$ винаги завършва. Лесно се съобразява, че езикът $\texttt{Graph}(f) = \{\, \alpha\sharp f(\alpha) \mid \alpha \in \Sigma^\star \,\}$ е разрешим.

Задача 6.2. Докажете, че съществуват функции от вида $f : \Sigma^\star \to \Sigma^\star$, които не са изчислими с машина на Тюринг.

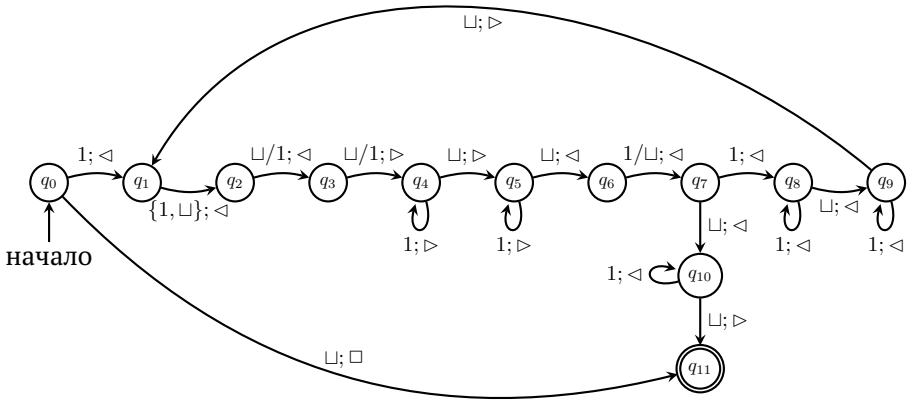
Упътване. Всяка машина на Тюринг може да се кодира с естествено число. Това означава, че съществуват изброимо безкрайно много машини на Тюринг. От друга страна, съществуват неизброимо много функции от вида $f : \Sigma^\star \to \Sigma^\star$. □

Пример 6.4. Да разгледаме функцията $f : \{1\}^\star \to \{1\}^\star$, където

$$f(1^n) \stackrel{\text{деф}}{=} 1^{2n}.$$

При двулентова машина на Тюринг тази задача е много по-лесна и сложността ще бъде $\mathcal{O}(n)$ вместо $\mathcal{O}(n^2)$.

Да видим защо f е изчислима с машина на Тюринг.



Фигура 6.10: Машина на Тюринг за функцията $f(1^n) = 1^{2n}$.

Пример 6.5. Да разгледаме тоталната функция $f : \{a, b\}^\star \to \{a, b\}^\star$, където

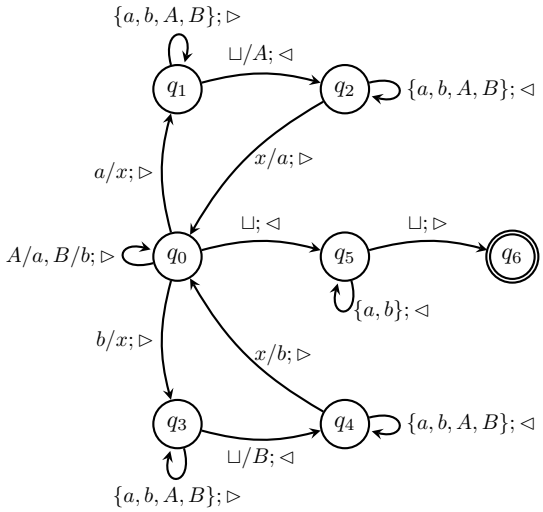
$$f(\alpha) \stackrel{\text{деф}}{=} \alpha \cdot \alpha.$$

Това пак става много по-лесно с двулентова машина на Тюринг и сложността пак ще бъде $\mathcal{O}(n)$ вместо $\mathcal{O}(n^2)$.

Да видим защо е изчислима с машина на Тюринг.

- $\Sigma \stackrel{\text{деф}}{=} \{a, b\}$;

- $\Gamma \stackrel{\text{деф}}{=} \{a, b, x, A, B\}$;
- $q_{\texttt{start}} \stackrel{\text{деф}}{=} q_0$ и $q_{\texttt{accept}} \stackrel{\text{деф}}{=} q_6$



Фигура 6.11: Машина на Тюринг за $f(\alpha) = \alpha \cdot \alpha$.

Да проследим работата на $\mathcal{M}$ върху думата ab. Първо копираме добавяме AB и така лентата съдържа $abAB$. След това заменяме A с a и B с b. Така най-накравя полуваме върху лентата думата $abab$.

$$\begin{aligned}(q_0, \underline{a}b\sqcup) &\vdash_{\mathcal{M}} (q_1, x\underline{b}\sqcup) \vdash_{\mathcal{M}} (q_1, xb\underline{\sqcup}) \vdash_{\mathcal{M}} (q_2, x\underline{b}A) \vdash_{\mathcal{M}} (q_2, \underline{x}bA)\\ &\vdash_{\mathcal{M}} (q_0, a\underline{b}A) \vdash_{\mathcal{M}} (q_3, ax\underline{A}) \vdash_{\mathcal{M}} (q_3, axA\underline{\sqcup}) \vdash_{\mathcal{M}} (q_4, ax\underline{A}B)\\ &\vdash_{\mathcal{M}} (q_4, a\underline{x}AB) \vdash_{\mathcal{M}} (q_0, ab\underline{A}B) \vdash_{\mathcal{M}} (q_0, aba\underline{B}) \vdash_{\mathcal{M}} (q_0, abab\underline{\sqcup})\\ &\vdash_{\mathcal{M}} (q_5, aba\underline{b}) \vdash_{\mathcal{M}} (q_5, ab\underline{a}b) \vdash_{\mathcal{M}} (q_5, a\underline{b}ab) \vdash_{\mathcal{M}} (q_5, \underline{a}bab)\\ &\vdash_{\mathcal{M}} (q_5, \sqcup abab) \vdash_{\mathcal{M}} (q_6, \underline{a}bab).\end{aligned}$$

Не можем директно да започнем да копираме α, защото така няма да знаем къде е края на първото копие на α. Това можем да направим като първо запишем на лентата $\alpha\sharp\alpha$ и след това второто копие на α го изместим с една позиция наляво.

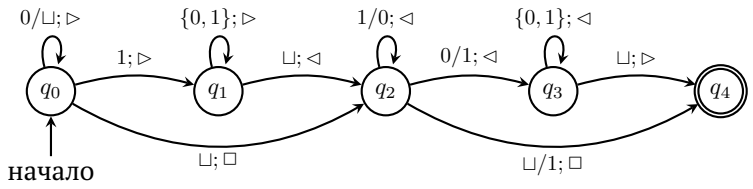
Изискваме $f(\alpha)$ да започва с 1 за да може f да бъде функция, т.е. $f(\alpha)$ е най-късият двоичен запис на числото $\overline{\alpha}_{(2)} + 1$.

Пример 6.6. Да разгледаме тоталната функция $f : \{0,1\}^\star \to 1 \cdot \{0,1\}^\star$, където

$$\overline{f(\alpha)}_{(2)} \stackrel{\text{деф}}{=} \overline{\alpha}_{(2)} + 1.$$

Нека да видим, че тази функция е изчислима с машина на Тюринг.

- $\Sigma \stackrel{\text{деф}}{=} \{0,1\}$;
- $\Gamma \stackrel{\text{деф}}{=} \{0,1,\sqcup\}$;
- $q_{\texttt{start}} = q_0$ и $q_{\texttt{accept}} \stackrel{\text{деф}}{=} q_4$.



Фигура 6.12: Машина на Тюринг изчисляваща f, за която $\overline{f(\alpha)}_{(2)} = \overline{\alpha}_{(2)} + 1$.

Да проследим изчислението на $\mathcal{M}$ върху вход 01011.

$$\begin{aligned}(q_0, 0\underline{1}011\sqcup) &\vdash_{\mathcal{M}} (q_0, \sqcup\underline{1}011\sqcup) \vdash_{\mathcal{M}} (q_1, \sqcup 1\underline{0}11\sqcup) \vdash_{\mathcal{M}} (q_1, \sqcup 10\underline{1}1\sqcup)\\ &\vdash_{\mathcal{M}} (q_1, \sqcup 101\underline{1}\sqcup) \vdash_{\mathcal{M}} (q_1, \sqcup 1011\underline{\sqcup}) \vdash_{\mathcal{M}} (q_2, \sqcup 101\underline{1}\sqcup)\\ &\vdash_{\mathcal{M}} (q_2, \sqcup 10\underline{1}0\sqcup) \vdash_{\mathcal{M}} (q_2, \sqcup 1\underline{0}00\sqcup) \vdash_{\mathcal{M}} (q_3, \sqcup\underline{1}100\sqcup)\\ &\vdash_{\mathcal{M}} (q_3, \underline{\sqcup}1100\sqcup) \vdash_{\mathcal{M}} (q_4, \sqcup\underline{1}100\sqcup).\end{aligned}$$

Задача 6.3. Да разгледаме азбуката $\Sigma = \{0, 1, \ldots, k-1\}$, където $k > 2$. Да разгледаме тоталната функция

$$f : \Sigma^\star \to (\Sigma \setminus \{0\}) \cdot \Sigma^\star,$$

където

$$\overline{f(\alpha)}_{(k)} \stackrel{\text{деф}}{=} \overline{\alpha}_{(k)} + 1.$$

Дефинирайте машина на Тюринг $\mathcal{M}$, която изчислява функцията f.

Канонична наредба на $\Sigma^\star$

За доказателството, че всяка НМТ е еквивалентна на ДМТ, е необходимо да фиксираме канонична подредба на думите над дадена азбука.

Нека $\Sigma = \{a_0, a_1, \ldots, a_{k-1}\}$. Подреждаме думите по ред на тяхната дължина. Думите с еднаква дължина подреждаме по техния числов ред, т.е. гледаме на буквите a_i като числото i в k-ична бройна система. Тогава думите с дължина n са числата от 0 до $k^n - 1$ записани в k-ична бройна система. Ще означаваме с ω_i i-тата дума в $\Sigma^\star$ при тази подредба.

Ако $\Sigma = \{0, 1\}$, то наредбата започва така:

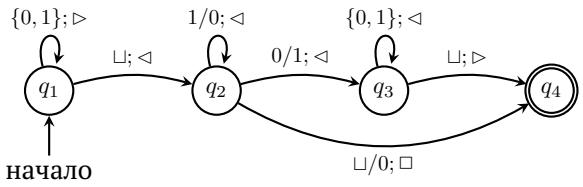
$$\varepsilon, 0, 1, \underbrace{00, 01, 10, 11}_{\text{от 0 до 3}}, \underbrace{000, 001, 010, 011, 100, 101, 110, 111}_{\text{от 0 до 7}}, 0000, 0001, \ldots$$

В този случай, $\omega_0 = \varepsilon$, $\omega_7 = 000$, $\omega_{13} = 110$. Обърнете внимание, че тази наредба отговаря на обхождане в широчина на едно пълно наредено двоично дърво. Можем да дефинираме и релацията $<_{\text{can}}$ по следния начин:

$$\alpha <_{\text{can}} \beta \overset{\text{деф}}{\Leftrightarrow} |\alpha| < |\beta| \vee (|\alpha| = |\beta| \,\&\, \alpha <_{\text{lex}} \beta).$$

Задача 6.4. Нека $\Sigma = \{a_0, \ldots, a_{k-1}\}$. Да разгледаме функцията $f : \Sigma^\star \to \Sigma^\star$, за която $f(\alpha)$ е думата веднага след α в каноничната подредба на $\Sigma^\star$. Докажете, че f е изчислима с еднолентова детерминираната машина на Тюринг.

Упътване. Ако $\Sigma = \{0, 1\}$, то машината на Тюринг има следния вид:



Фигура 6.13: Генериране на следващата дума в каноничната наредба.

□

6.4 Недетерминирани машини на Тюринг

Една машина на Тюринг $\mathcal{N}$ се нарича недетерминирана, ако функцията на преходите има вида

$$\Delta : Q' \times \Gamma \to \mathscr{P}(Q \times \Gamma \times \{\triangleleft, \triangleright, \square\}),$$

където да напомним, че $Q' = Q \setminus \{q_{\texttt{accept}}, q_{\texttt{reject}}\}$.

Отново можем да дефинираме бинарна релация $\vdash_{\mathcal{N}}$ над множеството $\Gamma^\star \times Q \times \Gamma^+$, която ще казва как моментното описание на машината $\mathcal{N}$ се променя при изпълнение на една стъпка.

$$\frac{\Delta(q,x) \ni (q', y, d) \qquad (\lambda, x\rho) \vdash_{y,d} (\lambda', \rho')}{(\lambda, q, x\rho) \vdash_{\mathcal{N}} (\lambda', q', \rho')}$$

Фигура 6.14: Преход в еднолентова недетерминирана машина на Тюринг $\mathcal{N}$

Сега за всяко естествено число ℓ, ще дефинираме релацията $\vdash^{\ell}_{\mathcal{N}}$, която ще казва, че от конфигурацията κ можем да достигнем до конфигурацията κ' за ℓ на брой стъпки.

Тази дефиниция на релацията $\vdash^{\ell}_{\mathcal{N}}$ вече се повтаря няколко пъти.

$$\frac{}{\kappa \vdash^{0}_{\mathcal{N}} \kappa} \text{ (рефлексивност)} \qquad \frac{\kappa \vdash_{\mathcal{N}} \kappa'' \qquad \kappa'' \vdash^{\ell}_{\mathcal{N}} \kappa'}{\kappa \vdash^{\ell+1}_{\mathcal{N}} \kappa'} \text{ (транзитивност)}$$

С $\vdash^{\star}_{\mathcal{N}}$ ще означаваме рефлексивното и транзитивно затваряне на релацията $\vdash_{\mathcal{N}}$ или с други думи,

$$\kappa \vdash^{\star}_{\mathcal{N}} \kappa' \overset{\text{деф}}{\Leftrightarrow} (\exists \ell \in \mathbb{N})[\kappa \vdash^{\ell}_{\mathcal{N}} \kappa'].$$

Езикът, който се разпознава от една недетерминирана машина на Тюринг $\mathcal{N}$ е

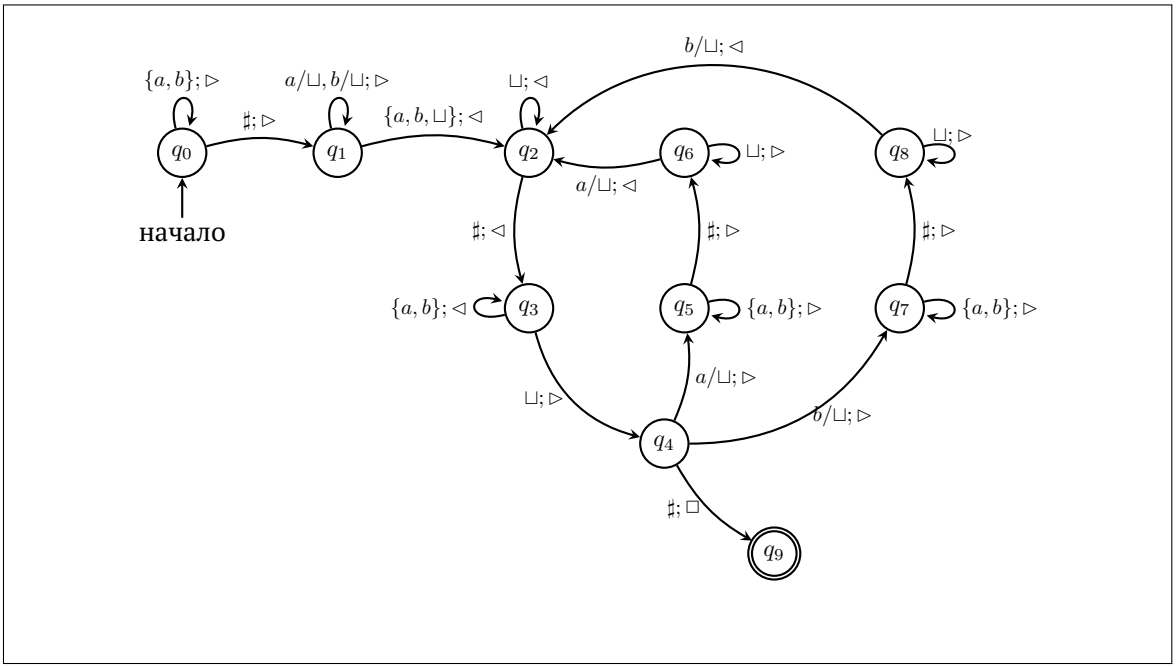
$$\mathcal{L}(\mathcal{N}) \overset{\text{деф}}{=} \{\, \omega \in \Sigma^\star \mid (\exists \kappa \in \texttt{Accept}_{\mathcal{N}})[\, \texttt{init}_{\mathcal{N}}(\omega) \vdash^{\star}_{\mathcal{N}} \kappa \,]\}.$$

Забележка. Върху дадена дума ω, недетерминираната машина на Тюринг $\mathcal{N}$ може да има много различни изчисления. Думата ω принадлежи на $\mathcal{L}(\mathcal{N})$ ако съществува *поне едно* изчисление, което завършва в състоянието $q_{\texttt{accept}}$. Възможно е много други изчисления при вход ω да завършват в $q_{\texttt{reject}}$ или никога да не завършват.

Аналогично, дефинираме една недетерминираната машина на Тюринг $\mathcal{N}$ да бъде **разрешител**, ако за всяка дума ω и *всяко изчисление* на $\mathcal{N}$ върху ω завършва в $q_{\texttt{accept}}$ или $q_{\texttt{reject}}$.

Пример 6.7. Нека да видим, че $L = \{\alpha\sharp\beta \mid \alpha, \beta \in \{a,b\}^\star \;\&\; \alpha \text{ е подниз на } \beta\}$ е разрешим език като построим недетерминираната машина на Тюринг $\mathcal{N}$, която разрешава този език.

Не е обяснено защо е разрешим.



Да видим, че $\mathcal{M}$ успешно разпознава думата $ab\sharp aabb$, която принадлежи на L.

$$
\begin{aligned}
(q_0, \underline{a}b\sharp aabb\sqcup) &\vdash (q_0, a\underline{b}\sharp aabb\sqcup) \vdash (q_0, ab\underline{\sharp}aabb\sqcup) \vdash (q_1, ab\sharp\underline{a}abb\sqcup) \\
&\vdash (q_1, ab\sharp\sqcup\underline{a}bb\sqcup) \vdash (q_2, ab\sharp\underline{\sqcup}abb\sqcup) \vdash (q_2, ab\underline{\sharp}\sqcup abb\sqcup) \\
&\vdash (q_3, a\underline{b}\sharp\sqcup abb\sqcup) \vdash (q_3, \underline{a}b\sharp\sqcup abb\sqcup) \vdash (q_3, \underline{\sqcup}ab\sharp\sqcup abb\sqcup) \\
&\vdash (q_4, \underline{a}b\sharp\sqcup abb\sqcup) \vdash (q_5, \sqcup\underline{b}\sharp\sqcup abb\sqcup) \vdash (q_5, \sqcup b\underline{\sharp}\sqcup abb\sqcup) \\
&\vdash (q_6, \sqcup b\sharp\underline{\sqcup}abb\sqcup) \vdash (q_6, \sqcup b\sharp\sqcup\underline{a}bb\sqcup) \vdash (q_2, \sqcup b\sharp\underline{\sqcup}\sqcup bb\sqcup) \\
&\vdash (q_2, \sqcup b\underline{\sharp}\sqcup\sqcup bb\sqcup) \vdash (q_3, \sqcup\underline{b}\sharp\sqcup\sqcup bb\sqcup) \vdash (q_3, \underline{\sqcup}b\sharp\sqcup\sqcup bb\sqcup) \\
&\vdash (q_4, \sqcup\underline{b}\sharp\sqcup\sqcup bb\sqcup) \vdash (q_7, \sqcup\sqcup\underline{\sharp}\sqcup\sqcup bb\sqcup) \\
&\vdash (q_8, \sqcup\sqcup\sharp\underline{\sqcup}\sqcup bb\sqcup) \vdash (q_8, \sqcup\sqcup\sharp\sqcup\underline{\sqcup}bb\sqcup) \\
&\vdash (q_8, \sqcup\sqcup\sharp\sqcup\sqcup\underline{b}b\sqcup) \vdash (q_2, \sqcup\sqcup\sharp\sqcup\underline{\sqcup}\sqcup b\sqcup) \\
&\vdash \cdots \vdash (q_4, \sqcup\sqcup\underline{\sharp}\sqcup\sqcup\sqcup b\sqcup) \vdash (q_9, \sqcup\sqcup\underline{\sharp}\sqcup\sqcup\sqcup b\sqcup)
\end{aligned}
$$

Теорема 6.1. Ако L се разпознава от *недетерминирана* машина на Тюринг $\mathcal{N}$, то L е разпознава и от *детерминирана* машина на Тюринг $\mathcal{D}$.

В [10, стр. 164] не е добре обяснено.

Доказателство. Нека имаме недетерминирана машина на Тюринг $\mathcal{N}$, за която $L = \mathcal{L}(\mathcal{N})$. Една дума α принадлежи на $\mathcal{L}(\mathcal{N})$ точно тогава, когато съществува изчисление, което започва с думата α върху лентата и след краен брой стъпки, следвайки функцията на преходите $\Delta_{\mathcal{N}}$, достига до състоянието $q_{\texttt{accept}}$. Сложността идва от факта, че за думата α може да имаме много различни изчисления, като само някои от тях завършват в $q_{\texttt{accept}}$. Ще построим детерминирана машина на Тюринг $\mathcal{D}$, която последователно ще симулира всички възможни *крайни* изчисления за думата α, докато намери такова, което завършва в състоянието $q_{\texttt{accept}}$.

На практика това, което правим е да представим всички възможни изчисления на $\mathcal{N}$ като r-разклонено дърво и да го обходим в широчина, докато не достигнем до $q_{\texttt{accept}}$

Лесно се съобразява, че всяко изчисление на $\mathcal{N}$ може да се представи като крайна редица от елементи на $Q \times \Gamma \times \{\triangleleft, \triangleright, \square\}$. Понеже това множество е крайно, то можем на всяка такава тройка да съпоставим естествено число $< r$, където

$$r = |Q| \cdot |\Gamma| \cdot 3.$$

Например, нека $Q = \{q_0, q_1\}$, $\Gamma = \{a, b\}$. Тогава можем да направим следната съпоставка:

$$(q_0, a, \square) \to 0,\ (q_0, a, \lhd) \to 1,\ (q_0, a, \rhd) \to 2,$$
$$(q_0, b, \square) \to 3,\ (q_0, b, \lhd) \to 4,\ (q_0, b, \rhd) \to 5,$$
$$(q_1, a, \square) \to 6, (q_1, a, \lhd) \to 7, (q_1, a, \rhd) \to 8,$$
$$(q_1, b, \square) \to 9, (q_1, b, \lhd) \to 10,\ (q_1, b, \rhd) \to 11.$$

Ясно е, че всяко изчисление на $\mathcal{N}$ може да се представи като дума над азбуката $\Sigma = \{x_0, x_1, \ldots, x_{r-1}\}$. Например, изчислението от три стъпки

$$(\sqcup, q_0, aba) \vdash_N (b, q_1, ba) \vdash_{\mathcal{N}} (b, q_1, aa) \vdash_{\mathcal{N}} (ba, q_0, a)$$

може да се опише като думата $x_{11}x_6x_2$ над азбуката $\Sigma = \{x_0, x_1, \ldots, x_{11}\}$.

Детерминираната машина на Тюринг $\mathcal{D}$ има три ленти.

- На първата лента съхраняваме входящия низ и *тя никога не се променя*.
- На втората лента ще записваме последователно думи следвайки каноничната наредба на думите над азбуката $\{x_0, x_1, \ldots, x_{r-1}\}$. От *Задача 6.4* знаем как последователно да генерираме тези думи върху една лента.
- На третата лента симулираме изчислението на $\mathcal{N}$ върху думата от първата лента, използвайки изчислението, което е описано на втората лента. Например, ако съдържанието на втората лента е $x_{11}x_6x_2$, това означава, че симулираме изчисление от три стъпки като на първата стъпка избираме дванайсетата възможна тройка, на втората стъпка избираме седмата възможна тройка, на третата стъпка избираме третата възможна тройка.

  Ако симулацията завърши в състоянието $q_{\texttt{accept}}$ на $\mathcal{N}$, то машината $\mathcal{D}$ завършва успешно. В противен случай, на втората лента генерираме чрез функцията от *Задача 6.4* следващия низ относно каноничната наредба на $\{x_0, x_1 \ldots, x_{r-1}\}$; изтриваме третата лента, копираме първата лента на третата и започваме нова детерминистична симулация като думата върху втората лента ни ръководи какъв преход да правим на всяка стъпка.

□

Следствие 6.1. Ако L се разпознава от *недетерминиран* разрешител $\mathcal{N}$, то L също се разпознава от *детерминиран* разрешител $\mathcal{D}$.

Доказателство. Да разгледаме дървото T с крайно разклонение r, което представя всички изчисления на разрешителя $\mathcal{N}$ при вход думата ω. От *Лема 4.1* следва, че T е крайно дърво, да кажем с височина h, защото ако допуснем, че T е безкрайно, то ще има безкрайно дълго изчисление на $\mathcal{N}$, което е невъзможно, понеже $\mathcal{N}$ винаги достига до заключително състояние ($q_{\texttt{accept}}$ или $q_{\texttt{reject}}$).

- Ако $\mathcal{N}$ приема дадена дума ω, то детерминистичната ни симулация на $\mathcal{N}$ ще достигне до изчисление, кодирано като път в T, което завършва в състояние $q_{\texttt{accept}}$.
- Ако $\mathcal{N}$ не приема дадена дума ω, то детерминистичната ни симулация на $\mathcal{N}$ ще покаже, че всяко изчисление, кодирано като път в T, завършва в състояние $q_{\texttt{reject}}$. Един начин да направим това е да имаме една допълнителна лента, която използваме за брояч колко от възможните изчисления на $\mathcal{N}$ са завършили. Спираме, когато този брояч достигне r^h, където h е дължината на думата на втората лента, т.е. дълбочината на дървото на изчисленията на $\mathcal{N}$.

□

Многолентови недетерминирани машини на Тюринг

Аналогично, тук разликата е в дефиницията на функцита Δ. Сега тя е следната:

$$\Delta : Q' \times \Gamma^k \to \mathcal{P}(Q \times \Gamma^k \times \{\triangleleft, \triangleright, \square\}^k),$$

където $Q' = Q \setminus \{q_{\texttt{accept}}, q_{\texttt{reject}}\}$.

$$\frac{\Delta(q, \vec{x}) \ni (q', \vec{y}, \vec{d}) \qquad \forall i < k : (\lambda_i, x_i\rho_i) \vdash_{y_i, d_i} (\lambda_i', \rho_i')}{(\vec{\lambda}, q, \vec{x} \cdot \vec{\rho}) \vdash_{\mathcal{N}} (\vec{\lambda}', q', \vec{\rho'})}$$

Фигура 6.16: Едностъпков преход в k-лентова недетерминистична машина на Тюринг

Вече би трябвало да е ясно, че езиците, които се разпознават с k-лентови недетерминирани машини на Тюринг съвпадат с езиците, които се разпознават от еднолентови детерминирани машини на Тюринг.

6.5 Основни свойства

Твърдение 6.3. Ако езикът L е разрешим, то $\overline{L}$ също е разрешим език.

Упътване. Нека $L = \mathcal{L}(\mathcal{M})$, където $\mathcal{M}$ е разрешител. Нека $\mathcal{M}'$ е същата като $\mathcal{M}$, само със сменени $q_{\texttt{accept}}$ и $q_{\texttt{reject}}$ състояния. Тогава $\mathcal{M}'$ също е разрешител и $\overline{L} = \mathcal{L}(\mathcal{M}')$. □

Означаваме $\overline{L} = \Sigma^* \setminus L$. С други думи, твърдението ни казва, че разрешимите езици са затворени относно операцията допълнение. След малко в *Твърдение 6.7* ще видим, че това твърдение не е изпълнено за полуразрешими езици.

Твърдение 6.4. Ако езиците L_1 и L_2 са разрешими , то $L_1 \cup L_2$ е разрешим език.

Упътване. Нека $L_1 = \mathcal{L}(\mathcal{M}_1)$ и $L_2 = \mathcal{L}(\mathcal{M}_2)$. Строим нова машина на Тюринг $\mathcal{M}$, която при вход думата α симулира едновременно изчисленията на $\mathcal{M}_1$ и $\mathcal{M}_2$ върху α. Това можем да направим като приемем, че $\mathcal{M}$ има две ленти - една за лентата на $\mathcal{M}_1$ и една за лентата на $\mathcal{M}_2$, като състоянията на $\mathcal{M}$ ще бъдат елементи на $Q_1 \times Q_2$. Ако една от двете машини достигне своето приемащо състояние, то $\mathcal{M}$ приема думата α. Ако и двете машини достигнат своите отхвърлящи състояния, то $\mathcal{M}$ отхвърля думата α. □

С други думи, разрешимите езици са затворени относно операцията обединение. Като следствие получаваме, че всяко *крайно* обединение на разрешими езици е разрешим език.
✍ Съобразете, че това твърдение е изпълнено и за полуразрешими езици.

Твърдение 6.5. Ако L_1 и L_2 са разрешими езици, то $L_1 \cap L_2$ е разрешим език.

Упътване. Нека $L_1 = \mathcal{L}(\mathcal{M}_1)$ и $L_2 = \mathcal{L}(\mathcal{M}_2)$. Строим нова машина на Тюринг $\mathcal{M}$, която при вход думата α симулира едновременно изчисленията на $\mathcal{M}_1$ и $\mathcal{M}_2$ върху α. Ако и двете машини достигнат до приемащите си състояния, то $\mathcal{M}$ приема думата α. Ако поне една от двете машини достигне до отхвърлящо състояние, то $\mathcal{M}$ отхвърля думата α. □

С други думи, разрешимите езици са затворени относно операцията сечение. Като следствие получаваме, че всяко *крайно* сечение на разрешими езици е разрешим език. ✍ Съобразете, че това твърдение е изпълнено и за полуразрешими езици.

Теорема 6.2 (Клини-Пост). Езиците L и $\overline{L}$ са полуразрешими точно тогава, когато L е разрешим език.

Упътване. Посоката ($\Leftarrow$) е ясна. За посоката ($\Rightarrow$), нека $L = \mathcal{L}(\mathcal{M}_1)$ и $\overline{L} = \mathcal{L}(\mathcal{M}_2)$. Строим разрешител $\mathcal{M}$, която при вход думата α симулира едновременно изчисленията на $\mathcal{M}_1$ и $\mathcal{M}_2$ върху α. Например, може $\mathcal{M}$ да има две ленти за симулацията на $\mathcal{M}_1$ и $\mathcal{M}_2$. Знаем със сигурност, че точно едно от двете симулирани изчисления ще завърши в приемащо състояние. Ако това е $\mathcal{M}_1$, то $\mathcal{M}$ приема α. Ако това е $\mathcal{M}_2$, то $\mathcal{M}$ отхвърля α. □

✍ Дефинирайте сами новата машина на Тюринг $\mathcal{M}$.

Кодиране на машина на Тюринг

Тук е удобно да индексираме от 1 вместо от 0.

Да приемем, че:

- $Q = \{q_1, q_2, \ldots, q_n\}$, където $n \geq 2$;
- $q_1 = q_{\texttt{start}}$, $q_2 = q_{\texttt{accept}}$ и $q_3 = q_{\texttt{reject}}$.
- $\Sigma = \{x_1, \ldots, x_\ell\}$;
- $\Gamma = \{x_1, x_2, \ldots, x_s\}$, където $\ell < s$ и $x_s = \sqcup$;
- $d_1 = \square$, $d_2 = \triangleleft$, $d_3 = \triangleright$;

Ясно е, че тук съвсем спокойно можем да разгледаме и недетерминистична машина на Тюринг.

Така можем да кодираме преходите на детерминистична машина на Тюринг като думи. Да разгледаме прехода $\delta(q_i, x_j) = (q_k, x_t, d_m)$. Кодираме този преход със следната дума:

$$0^i 1 0^j 1 0^k 1 0^t 1 0^m.$$

Да обърнем внимание, че в този двоичен код няма последователни единици и той започва и завършва с нула.

За да кодираме една машина на Тюринг $\mathcal{M}$ е достатъчно да кодираме функцията на преходите δ. Понеже δ е крайна функция, нека с числото r да означим броя на всички възможни преходи. По описания по-горе начин, нека $\texttt{code}_i$ е числото в двоичен запис, получено за i-тия преход на δ. Тогава кодът на $\mathcal{M}$ е следното число в двоичен запис:

Ясно е, че не всяка дума над азбуката $\{0, 1\}$ е код на машина на Тюринг.

$$\ulcorner \mathcal{M} \urcorner \stackrel{\text{деф}}{=} 1110^\ell 11\ \texttt{code}_1\ 11\ \texttt{code}_2\ 11\ \cdots\ 11\ \texttt{code}_r\ 111.$$

По същия начин можем да дефинираме и код на краен автомат и стеков автомат.

Ще означаваме с $\mathcal{M}_\omega$ машината на Тюринг, чийто код е ω, т.е. искаме $\ulcorner \mathcal{M}_\omega \urcorner = \omega$. Важно свойство, което ще използваме често по-нататък, е че съществува алгоритъм, който при вход произволна дума $\omega \in \{0,1\}^\star$, може да определи дали тази дума ω представлява код на машина на Тюринг.

Задача 6.5. Докажете, че езикът

$$L_{\texttt{code}} \stackrel{\text{деф}}{=} \{\ \omega \in \{0,1\}^\star \mid \omega \text{ е код на машина на Тюринг}\ \}$$

е разрешим.

Твърдение 6.6. Съществуват следните функции:

- тотална изчислима функция f_1 на два аргумента, която има свойството:
$$\mathcal{L}(\mathcal{M}_{f_0(\omega,\rho)}) = \mathcal{L}(\mathcal{M}_\omega) \cap \mathcal{L}(\mathcal{M}_\rho).$$
- тотална изчислима функция f_2 на два аргумента, която има свойството:
$$\mathcal{L}(\mathcal{M}_{f_1(\omega,\rho)}) = \mathcal{L}(\mathcal{M}_\omega) \cup \mathcal{L}(\mathcal{M}_\rho).$$
- тотална изчислима функция f_3, която има свойството:
$$\mathcal{L}(\mathcal{M}_{f_3(\omega)}) = \{\alpha \in \{0,1\}^\star \mid \mathcal{M}_\omega \text{ не завършва върху } \omega \text{ за } \leq |\alpha| \text{ стъпки}\ \}.$$
- тотална изчислима функция f_4, която има свойството:
$$\mathcal{L}(\mathcal{M}_{f_4(\omega)}) = \{\alpha \in \{0,1\}^\star \mid \mathcal{M}_\omega \text{ завършва върху } \omega \text{ за } \leq |\alpha| \text{ стъпки}\ \}.$$
- тотална изчислима функция f_5, която има свойството:
$$\mathcal{L}(\mathcal{M}_{f_5(\omega)}) = \{\alpha \in \{0,1\}^\star \mid \omega \in \mathcal{L}(\mathcal{M}_\omega)\}.$$

Диагоналният език

Теорема 6.3. Диагоналният език

$$L_{\texttt{diag}} \stackrel{\text{деф}}{=} \{\, \omega \in L_{\texttt{code}} \mid \omega \notin L(\mathcal{M}_\omega) \,\}$$

не се разпознава от машина на Тюринг, т.е. $L_{\texttt{diag}}$ **не** е полуразрешим език.

Доказателство. Да допуснем, че $L_{\texttt{diag}}$ се разпознава от машина на Тюринг $\mathcal{M}$, т.е. $L_{\texttt{diag}} = \mathcal{L}(\mathcal{M})$. Тогава да видим какво имаме за думата $\ulcorner\mathcal{M}\urcorner$:

$$\ulcorner\mathcal{M}\urcorner \in L_{\texttt{diag}} \implies \ulcorner\mathcal{M}\urcorner \in \mathcal{L}(\mathcal{M}) \implies \ulcorner\mathcal{M}\urcorner \notin L_{\texttt{diag}},$$
$$\ulcorner\mathcal{M}\urcorner \notin L_{\texttt{diag}} \implies \ulcorner\mathcal{M}\urcorner \notin \mathcal{L}(\ulcorner\mathcal{M}\urcorner) \implies \ulcorner\mathcal{M}\urcorner \in L_{\texttt{diag}}.$$

Достигаме до противоречие. □

Това е версия на диагоналния метод на Кантор, с чиято помощ се доказва, че реалните числа са неизброимо много, т.е. има повече реални числа отколкото естествените.

Тук е добре една безкрайна таблица да се нарисува.

Твърдение 6.7. Езикът

$$L_{\texttt{accept}} \stackrel{\text{деф}}{=} \{\, \omega \in L_{\texttt{code}} \mid \omega \in \mathcal{L}(\mathcal{M}_\omega) \,\}$$

е полуразрешим, но не е разрешим.

Упътване. Лесно се съобразява, че $L_{\texttt{accept}}$ е полуразрешим. Дефинираме машина на Тюринг $\mathcal{M}'$, която, при вход произволна дума ω, работи по следния начин:

(1) $\mathcal{M}'$ проверява дали ω е код на машина на Тюринг $\mathcal{M}_\omega$.

(2) Ако ω е код на машина на Тюринг $\mathcal{M}_\omega$, то $\mathcal{M}'$ симулира работата на $\mathcal{M}_\omega$ върху ω.

(3) Ако след краен брой стъпки симулацията завърши с резултат, че $\mathcal{M}_\omega$ приема думата ω, то $\mathcal{M}'$ също завършва като приеме ω.

Това можем да го направим, защото знаем, че $L_{\texttt{code}}$ е разрешим.

Получаваме, че за всяка дума ω е изпълнена еквивалентността

$$\omega \in L_{\texttt{accept}} \;\Leftrightarrow\; \omega \in \mathcal{L}(\mathcal{M}'),$$

откъдето следва, че $L_{\texttt{accept}}$ е полуразрешим език.

Ако допуснем, че $L_{\texttt{accept}}$ е разрешим, то езикът $L_{\texttt{code}} \setminus L_{\texttt{accept}} = L_{\texttt{diag}}$ би бил разрешим, което е противоречие, защото $L_{\texttt{diag}}$ не е е дори полуразрешим. □

Следствие 6.2. Съществува полуразрешим език L, за който $\overline{L}$ не е полуразрешим.

Упътване. Вземете езика $L = L_{\texttt{accept}}$. Знаем, че L е полуразрешим, но не е разрешим. Ако допуснем, че $\overline{L}$ е полуразрешим, то тогава от теоремата на Клини-Пост би следвало, че L е разрешим, което е противоречие. □

Твърдение 6.8. Докажете, че езикът

$$L_{\texttt{halt}} = \{\omega \in L_{\texttt{code}} \mid \mathcal{M}_\omega \text{ спира върху } \omega\}$$

е полуразрешим, но не е разрешим.

Упътване. Лесно се вижда, че $L_{\texttt{halt}}$ е полуразрешим. Да допуснем, че съществува машина на Тюринг $\mathcal{M}_{\texttt{halt}}$, която е разрешител за $L_{\texttt{halt}}$. Ще покажем, че тогава можем да построим разрешител $\mathcal{M}'$ за $L_{\texttt{accept}}$, което би било противоречие. Машината на Тюринг $\mathcal{M}'$ би работила така върху произволен вход ω:

(1) Ако ω не е код на машина на Тюринг, то $\mathcal{M}'$ директно отхвърляме ω.

(2) Ако ω е код на машина на Тюринг $\mathcal{M}_\omega$, то $\mathcal{M}'$ симулира работата на $\mathcal{M}_{\texttt{halt}}$ върху ω.

(3) Ако симулацията завърши с резултат, че $\mathcal{M}_{\texttt{halt}}$ отхвътрля думата ω, то $\mathcal{M}'$ завършва като отхвърля думата ω.

Щом $\omega \in L_{\texttt{halt}}$, то знаем, че рано или късно тази симулация ще завърши в някое състояние.

(4) Ако симулацията завърши с резултат, че $\mathcal{M}_{\texttt{halt}}$ приема думата ω, то $\mathcal{M}'$ симулира работата на $\mathcal{M}_\omega$ върху ω, докато тя завърши в състояние q.

- Ако $q = q_{\texttt{accept}}$ на $\mathcal{M}_\omega$, то $\mathcal{M}'$ завършва като приеме думата ω.
- В противен случай, $\mathcal{M}'$ завършва като отхвърля думата ω.

□

Универсална машина на Тюринг

> A man provided with paper, pencil, and rubber, and subject to strict discipline, is in effect a universal machine. (Turing 1948: 416)

Можем за простота да считаме, че всички разглеждани машини на Тюринг са дефинирани над азбуката $\{0, 1\}$.

Теорема 6.4. Универсалният език

$$L_{\mathtt{univ}} \stackrel{\text{деф}}{=} \{\, \ulcorner \mathcal{M} \urcorner \sharp \omega \mid \mathcal{M} \text{ е машина на Тюринг и } \omega \in \mathcal{L}(\mathcal{M}) \,\}$$

е полуразрешим, но **не** е разрешим.

Упътване. Първо да съобразим защо $L_{\mathtt{univ}}$ е полуразрешим език. Дефинираме (многолентова) машина на Тюринг $\mathcal{U}$, която работи по следния начин:

Разсъждението е много сходно с това защо $L_{\mathtt{accept}}$ полуразрешим.
Ще наричаме *$\mathcal{U}$ универсална машина на Тюринг.*

- вход дума α;
- $\mathcal{U}$ проверява дали α има вида $\ulcorner \mathcal{M} \urcorner \sharp \omega$, за някоя машина на Тюринг $\mathcal{M}$ и дума ω. Това става лесно, защото ω започва веднага след второ срещане на 111 в α.
- Ако α е от вида $\ulcorner \mathcal{M} \urcorner \sharp \omega$, то $\mathcal{U}$ симулира работата на $\mathcal{M}$ върху ω.
  - Ако $\mathcal{M}$ завърши след краен брой стъпки като приеме ω, то $\mathcal{U}$ приема α.
  - Ако $\mathcal{M}$ завърши след краен брой стъпки като отхвърли ω, то $\mathcal{U}$ отхвърля α.
  - Ако $\mathcal{M}$ никога не завършва върху ω, то очевидно $\mathcal{U}$ също никога не завършва върху α.
- Ако α няма вида $\ulcorner \mathcal{M} \urcorner \sharp \omega$, то $\mathcal{U}$ завършва веднага като отхвърля думата α.

Получаваме, че

$$\alpha \in L_{\mathtt{univ}} \;\Leftrightarrow\; \alpha \in \mathcal{L}(\mathcal{U}).$$

Сега да съобразим защо $L_{\mathtt{univ}}$ не е разрешим език. За произволна дума ω имаме:

$$\omega \in L_{\mathtt{accept}} \;\Leftrightarrow\; \omega\sharp\omega \in L_{\mathtt{univ}}.$$

Ако допуснем, че $L_{\mathtt{univ}}$ е разрешим, то тогава $L_{\mathtt{accept}}$ е разрешим език, което е противоречие. □

Следствие 6.3. Езикът

$$L'_{\mathtt{univ}} \stackrel{\text{деф}}{=} \{\omega\sharp\alpha \mid \omega \in L_{\mathtt{code}} \,\&\, \alpha \notin \mathcal{L}(\mathcal{M}_\omega)\}$$

не е полуразрешим.

Библиография

[1] Alfred Aho и др. *Compilers: principles, techniques and tools*. 2-е изд. Pearson Education, 2007.

[2] D. N. Arden. “Delayed-logic and finite-state machines”. В: *2nd Annual Symposium on Switching Circuit Theory and Logical Design (SWCT 1961)*. 1961, с. 133—151.

[3] Y. Bar-Hillel, M. Perles и E. Shamir. “On formal properties of simple phrase structure grammars”. В: *STUF - Language Typology and Universals* 14.1-4 (1961), с. 143—172. doi: 10.1524/stuf.1961.14.14.143.

[4] Jean Berstel. “Context-Free Languages”. В: *Transductions and Context-Free Languages*. Wiesbaden: Vieweg+Teubner Verlag, 1979, с. 22—50. isbn: 978-3-663-09367-1. doi: 10.1007/978-3-663-09367-1_2. url: https://doi.org/10.1007/978-3-663-09367-1_2.

[5] Janusz A. Brzozowski. “Derivatives of Regular Expressions”. В: *Journal of the ACM* 11.4 (окт. 1964), с. 481—494. issn: 0004-5411. doi: 10.1145/321239.321249.

[6] John H. Conway. *Regular algebra and finite machines*. Chapman и Hall, 1971. isbn: 0412106205.

[7] Ding-Zhu Du и Ker-I. Ko. *Problem Solving in Automata, Languages, and Complexity*. OCLC: 475923550. John Wiley & Sons, 2004. isbn: 978-0-471-46408-2.

[8] Javier Esparza. *Automata theory. An algorithmic approach*. 2017. url: https://www7.in.tum.de/~esparza/automatanotes.html.

[9] John E. Hopcroft, Rajeev Motwani и Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. second. Addison-Wesley, 2001.

[10] John E. Hopcroft и Jeffrey D. Ullman. *Introduction to Automata Theory, Languages, and Computation*. first. Addison-Wesley, 1979.

[11] Bakhadyr Khoussainov и Anil Nerode. *Automata Theory and its Applications*. Springer, 2001.

[12] S. C. Kleene. “Representation of events in nerve nets and finite automata”. В: *Automata Studies*. Под ред. на C. E. Shannon и J. McCarthy. Princeton University Press, 1956, с. 3—42. doi: 10.1515/9781400882618-002.

[13] Dexter Kozen. *Automata and Computability*. Springer, 1997.

[14] Anil Maheshwari и Michael Smid. *Introduction to Theory of Computation*. 2019. url: https://cglab.ca/~michiel/TheoryOfComputation/.

[15] Anil Nerode и Richard Shore. *Logic for Applications*. Springer-Verlag, 1993. doi: 10.1007/978-1-4612-0649-1.

[16] Christos Papadimitriou и Harry Lewis. *Elements of the Theory of Computation*. Prentice-Hall, 1998.

[17] Alberto Pettorossi. *Automata Theory and Formal Languages: Fundamental Notions, Theorems, and Techniques*. Undergraduate Topics in Computer Science. Springer International Publishing, 2022. doi: 10.1007/978-3-031-11965-1.

[18] M. O. Rabin и D. Scott. “Finite automata and their decision problems”. В: *IBM Journal of Research and Development* 3 (1959), pp. 114 —125. url: http://www.research.ibm.com/journal/rd/032/ibmrd0302C.pdf.

[19] Jacques Sakarovitch. *Elements of Automata Theory*. Cambridge University Press, 2009. doi: 10.1017/CBO9781139195218.

[20] Jeffrey Shallit. *A Second Course in Formal Languages and Automata Theory*. CUP, 2008.

[21] Michael Sipser. *Introduction to the Theory of Computation*. 3-е изд. Cengage Learning, 2012.

[22] L. Wittgenstein, G.H.V. Wright и G.E.M. Anscombe. *Remarks on the philosophy of psychology*. Т. 1. University of Chicago Press/B. Blackwell, 1980.

Азбучен указател