

## Глава 3

### Крайни автомати с памет. Машина на Тюринг и машина на Пост

#### 3.1 Безконтекстни граматика и езици

Съгласно определението, дадено в предишната глава, една граматика  $\Gamma$  е безконтекстна, ако всичките ѝ правила са от вида

$$A \rightarrow \omega, \quad \omega \in (V \cup W)^*,$$

където  $V$  и  $W$  са азбуки съответно от терминални и нетерминални символи.

**Пример 3.1.1** Да разгледаме следните граматика:

$$\Gamma_1 = \langle \{a, b\}, \{S, A, B\}, S, \{S \rightarrow AB, A \rightarrow aA|a, B \rightarrow bB|b\} \rangle;$$

$$\Gamma_2 = \langle \{a, b\}, \{S, A, B, C\}, S, \{S \rightarrow aSBC|abc, cB \rightarrow Bc, aB \rightarrow ab, bB \rightarrow bb, bC \rightarrow bc, cC \rightarrow cc\} \rangle;$$

$$\Gamma_3 = \langle \{a, b\}, \{S, A, B\}, S, \{S \rightarrow aA|bB|a|b, A \rightarrow aA|a, B \rightarrow bB|b\} \rangle;$$

$$\Gamma_4 = \langle \{a, b\}, \{S, A, B\}, S, \{S \rightarrow AB, A \rightarrow aA|a, B \rightarrow bB|b\} \rangle.$$

Без особени затруднения може да се съобрази, че езиците, които пораждат тези граматика са:

$$L(\Gamma_1) = L(\Gamma_3) = \{a^n \mid n \geq 1\} \cup \{b^n \mid n \geq 1\},$$

$$L(\Gamma_2) = \{a^n b^n c^n \mid n \geq 1\} \text{ и } L(\Gamma_4) = \{a^n b^m \mid n, m \geq 1\}.$$

Съгласно дефиницията  $L(\Gamma_1)$ ,  $L(\Gamma_3)$  и  $L(\Gamma_4)$  са безконтекстни езици, докато граматиката  $\Gamma_2$  не е безконтекстна, но това не изключва възможността да съществува безконтекстна граматика  $\Gamma'_2$ , която да поражда езика  $L(\Gamma_2)$ . Тази възможност не позволява да се произнесем дали този език  $L(\Gamma_2)$  е безконтекстен или не. Като една от целите на тази глава, ще си поставим задачата за изработване на критерии, по които да определим дали даден език  $L$  е безконтекстен или не.

Ще предполагаме, че ако  $A$  е нетерминален символ от азбуката  $W$  на граматиката  $\Gamma$ , то  $A$  може да се изведе от началния

символ  $S$ , т.е. да се изведе дума  $\omega \in (V \cup W)^*$ , която съдържа символа  $A$ .

Най-разпространеното представяне на даден извод на дума  $\omega = a_1a_2\dots a_n$  в безконтекстните граматики е чрез *дърво на извода*, което се нарича още *дърво за граматически разбор*. Коренът на такова дърво е началният символ  $S$ . Листата на дървото от ляво надясно са буквите  $a_1, a_2, \dots, a_n$ , а вътрешните възли са нетерминални символи, получени от правилата в  $\Gamma$  по следния начин: ако  $A$  е нетерминален символ и съществува правило

$$A \rightarrow x_1x_2\dots x_m, \quad x_i \in V \cup W, \quad i = 1, 2, \dots, m,$$

то върхът съответстващ на символа  $A$  е родителски за  $m$ -те върхове, отговарящи на символите  $x_1, x_2, \dots, x_m$ . Върховете, които съответствуват на терминални символи не са родителски за никои други върхове на дървото.

За пример да разгледаме дървото на извод на думата  $\omega = a^2b^3$  в граматиката  $\Gamma_4$ . (Фиг.3.1.1)

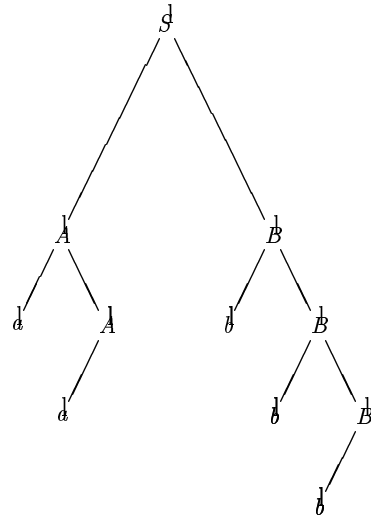
Тази дума може да се изведе по няколко схеми:

$$S \vdash AB \vdash AbA \vdash AbbB \vdash Abbb \vdash aAbbb \vdash aabbb$$

$$S \vdash AB \vdash aAB \vdash aaB \vdash aabB \vdash aabbB \vdash aabbb.$$

Макар и да имат повече от една схема на извод, думите в граматиката  $\Gamma_4$  имат единствено дърво на извод. Такива граматики се наричат *еднозначни*. Съществува възможност да се установява съответствие между дърветата и част от схемите на извод в еднозначните граматики.

Ще казваме, че една схема на извод в дадена безконтекстна граматиката е *лява*, ако на всяка стъпка се заменя най-левият нетерминален символ в получената дума. Така в горните две схеми на извод на думата  $a^2b^3$ , само втората е лява схема. Между дърветата на извод и левите схеми на извод в еднозначните граматики съществува взаимно еднозначно съответствие.



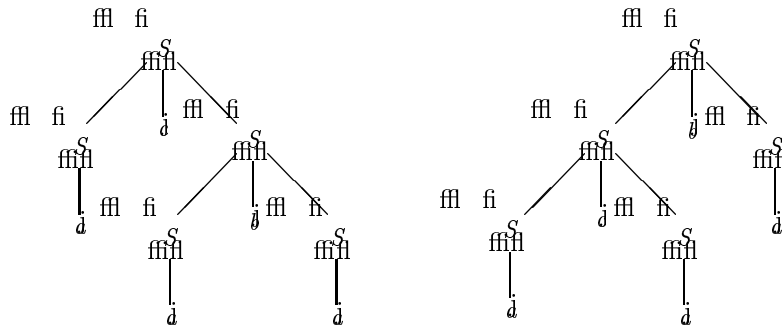
Фиг. 3.1.1

Да разгледаме граматиката  $\Gamma_5$  определена с правилата  $S \rightarrow SbS|ScS|a$  и думата  $abaca \in L(\Gamma_5)$ . На тази дума съответствуват две леви схеми на извод

$$S \vdash ScS \vdash acS \vdash acSbS \vdash acabS \vdash acaba$$

$$S \vdash SbS \vdash ScSbS \vdash acSbS \vdash acabS \vdash acaba$$

и съответно две различни дървета на извод (Фиг.3.1.2).



Фиг. 3.1.2

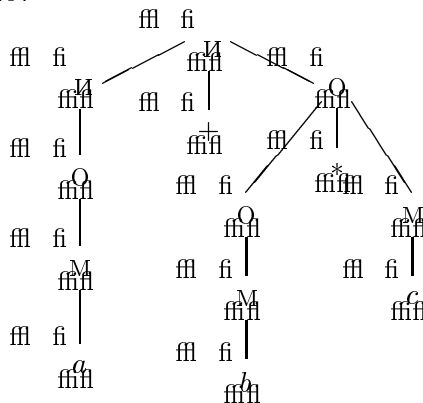
Такива граматики ще наричаме *нееднозначни*.

$$\begin{aligned} \langle \text{израз} \rangle &::= \langle \text{операнд} \rangle | \langle \text{израз} \rangle + \langle \text{операнд} \rangle | \\ &| \langle \text{израз} \rangle - \langle \text{операнд} \rangle; \\ \langle \text{операнд} \rangle &::= \langle \text{множитель} \rangle | \langle \text{операнд} \rangle * \langle \text{множитель} \rangle | \\ &| \langle \text{операнд} \rangle / \langle \text{множитель} \rangle; \\ \langle \text{множитель} \rangle &::= a | b | c | (\langle \text{израз} \rangle). \end{aligned}$$

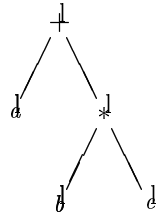
Формално тази граматика може да се представи с азбуките  $V = \{a, b, c, +, -, *, /, (, )\}$ ,  $W = \{И, О, М\}$ , начален символ И и правила

$$\begin{aligned} \mathbb{I} &\rightarrow \mathbb{O} | \mathbb{I} + \mathbb{O} | \mathbb{I} - \mathbb{O} \\ \mathbb{O} &\rightarrow \mathbb{M} | \mathbb{O} * \mathbb{M} | \mathbb{O} / \mathbb{M} \\ \mathbb{M} &\rightarrow a | b | c | ( \mathbb{I} ). \end{aligned}$$

Тази граматика е еднозначна и за всяка дума в нея съществува единствено дърво на извод. Тя може да се използва при семантичния разбор на аритметичните изрази от компилатор, например за езика *Pascal*. Дървото на извод за израза  $a + b * c$  е следното:



След граматическия разбор (т.е. построяването на дървото на извод) по това дърво, компилаторът построява друго дърво, наречено *семантично*. В него върховете са само от терминални символи. Семантичното дърво за горния израз е



След обработката на семантичното дърво, компилаторът предизвиква изпълнението на набор от команди, с които се пресмята даденият аритметичен израз.

Изкушителна е идеята, синтаксисът на езиците за програмиране да се дефинира с безконтекстни граматика, тъй като те дават възможност за построяване на семантични дървета за различните синтактични единици. В случай, че на такава граматика се наложат някои допълнителни свойства, тя действително може да опише голяма част от синтактичните правила на даден език за програмиране. Основна част от съвременното програмно осигуряване на компютрите се състои от алгоритми за анализ на семантични дървета, при все че в тях съществуват и алгоритми за различни допълнителни проверки, таблици от параметри и др.

В безконтекстните граматика е допустимо да има и правила от вида  $A \rightarrow \varepsilon$ . Такива правила ще наричаме  $\varepsilon$ -правила. Наличието на  $\varepsilon$ -правила в дадена граматика усложнява граматическия разбор и прилагането им за описание на формалните езици. Граматика, които нямат  $\varepsilon$ -правила се наричат  $\varepsilon$ -свободни. Следващата теорема е формулирана в предишната глава за един по-общ клас от граматика.

**Теорема 3.1.1** Ако  $\Gamma = \langle V, W, S, P \rangle$  е безконтекстна граматика, в която има  $\varepsilon$ -правила, то съществува  $\varepsilon$ -свободна граматика  $\Gamma'$ , за която  $L(\Gamma') = L(\Gamma) \setminus \{\varepsilon\}$ .

**Доказателство:** Да означим с  $\Gamma' = \langle V, W, S, P' \rangle$  търсената граматика. В множеството от правила  $P'$  причисляваме всички правила от  $P$ , които не са  $\varepsilon$ -правила. Да наречем  $\varepsilon$ -пораждащ символа  $A \in W$ , ако  $A \stackrel{\Gamma}{=} \varepsilon$ . За всяко правило  $B \rightarrow \omega$  от  $P$ ,  $\omega \in (V \cup W)^*$ ,  $|\omega| \geq 2$  и за всеки  $\varepsilon$ -пораждащ символ  $A$  в думата  $\omega$  добавяме в  $P'$  правилото  $B \rightarrow \omega_1$ , където  $\omega_1$  се получава от  $\omega$ , като се пропусне символът  $A$ .

Предоставяме за самостоятелно упражнение да се покаже, че всяка непразна дума  $\omega$ , която се извежда в граматиката  $\Gamma$  може да се породи и в  $\Gamma'$  и обратно.  $\square$ .

### З а д а ч и

1. Да се определи граматика, която поражда следния език:
  - а)  $L = \{\omega \in \{a, b\}^* \mid \text{броят на } a\text{-тата е равен на броят на } b\text{-тата}\}$ ;
  - б)  $L = \{\omega \in \{a, b\}^* \mid \omega = \omega', \text{ където } \omega' \text{ е думата получена от } \omega, \text{ като буквите ѝ са записани отзад напред}\}$ .
2. Нека  $\Gamma$  е граматика с правила

$$S \rightarrow AB, \quad A \rightarrow SA|BB|bB, \quad B \rightarrow b|aA|\varepsilon.$$

Да се построи  $\varepsilon$ -свободна граматика  $\Gamma'$ , за която

$$L(\Gamma') = L(\Gamma) \setminus \{\varepsilon\}.$$

3. Да се определи езикът  $L = L(\Gamma)$ , ако  $\Gamma$  има правила

$$S \rightarrow bA|aB, \quad A \rightarrow a|aS|bAA, \quad B \rightarrow b|bS|aBB.$$

Да се докаже, че  $\Gamma$  не е еднозначна.

4. Да се докаже, че граматиката с правила

$$S \rightarrow aB|bAS|aBS, \quad A \rightarrow bAA|a, \quad B \rightarrow aBB|b$$

е еднозначна и еквивалентна на граматиката от задача 3.

**У п ъ т в а н е:** Докажете индуктивно, че във всяка сентенциална форма, общият брой на  $a$  и  $A$  е равен на общия брой на  $b$  и  $B$ .

5. Да се опише граматика, която поражда всички алгебрични изрази (допуска се и операцията коренуване) на три променливи(букви) и да се провери дали тази граматика е автоматна или е безконтекстна.