

2.4 Свойства на автоматните езици

Нека сега да разгледаме дефиницията на цяло двоично число с използване на Бекус-Наутова форма (БНФ).

$\langle 01 - \text{редица} \rangle ::= 0 \mid 1 \mid 0\langle 01 - \text{редица} \rangle \mid 1\langle 01 - \text{редица} \rangle;$
 $\langle \text{двоично без знак} \rangle ::= 0 \mid 1 \mid 0\langle 01 - \text{редица} \rangle \mid 1\langle 01 - \text{редица} \rangle;$
 $\langle \text{цяло двоично} \rangle ::= +\langle \text{двоично без знак} \rangle \mid -\langle \text{двоично без знак} \rangle.$

Дясната страна във всяка от по-горните форми представлява или един терминален символ или поредица от терминални и нетерминални символи. Оказва се, че символите и думите в езика *Pascal* (цели числа, идентификатори, оператори, запазени думи, специални символи) могат да се дефинират по този начин.

Разпознаването на елементите от езика, в текста на една програма заема голяма част от времето за компилация на програмата. Това определя и възможностите на граматиките, които съдържат подобни на горните правила. Езиците, които пораждаат такива граматики, се поддават на ефективно разпознаване от компютрите. Тези езици имат определени свойства, които ги характеризират и разграничават от другите типове формални езици.

Възможностите на автоматните езици, обаче са ограничени, като дори и някои прости конструкции от езика не могат да се опишат само с автоматни граматики. Ето защо ролята на автоматните езици при компилацията се ограничава само до разпознаване символите на езика използвани в програмата. Този етап от компилацията е известен като лексически анализ. По-нататък ще видим, че за синтактичен анализ при компилацията е необходима граматика с по-малко ограничения в сравнение с автоматните, а именно граматики от тип 2 и 1.

Да припомним, че автоматните граматики имат правила от вида

$$A \rightarrow a|bB$$

и евентуално правилото $S \rightarrow \varepsilon$, където a и b са терминални символи, а S, A, B – нетерминални.

Очевидно, всяко правило на автоматните граматика до-
вежда до замяна на един нетерминален символ с дума, която
има не повече от един нетерминален символ. Също така се виж-
да, че когато в получената дума има нетерминален символ, то
той е последен символ на думата.

Важен въпрос за автоматните езици е дали техните свойс-
тва се съхраняват, когато прилагаме операциите обединение и
произведение.

Ще докажем, че в известен смисъл, този въпрос има ут-
върдителен отговор.

Теорема 2.4.1 *Ако L е автоматен език, за който $\varepsilon \notin L$, то и
езикът*

$$L^+ = \bigcup_{n \geq 0} L^n$$

е автоматен.

Доказателство: Нека $\Gamma = \langle V, W, S, P \rangle$ е автоматна граматика,
която поражда L , т.е. $L = L(\Gamma)$. Да разгледаме граматиката
 $\Gamma^+ = \langle V, W, S, P^+ \rangle$, за която правилата в P^+ се получават, като
към правилата в P се добавят още правилата $A \rightarrow aS$ за всяко
правило $A \rightarrow a$ в P .

Ще докажем, че $L^+ = L(\Gamma^+)$.

Нека $\omega \in L(\Gamma^+)$. Да означим с $n(\omega)$ броя на правилата от
 P^+ , които не са правила от P и участвуват в извода на ω . Ще
покажем, чрез индукция по $n(\omega)$, че $\omega \in L^+$.

При $n(\omega) = 0$, в извода на ω се използват само правила
от P и следователно $\omega \in L$, т.е. $\omega \in L^+$.

Да допуснем, че при $n(\omega) < k$, е в сила $\omega \in L^+$.

Нека ω е дума от $L(\Gamma^+)$, чийто извод съдържа k на брой
правила от $P^+ \setminus P$. Нека последното такова правило е $A \rightarrow aS$.
Тогава схемата на извод на ω може да се представи по следния
начин:

$$S \stackrel{\Gamma^+}{\vdash} yA \stackrel{\Gamma^+}{\vdash} yaS \stackrel{\Gamma^+}{\vdash} yaz = \omega,$$

където $y, z \in V^* \setminus \{\varepsilon\}$ и $a \in V$. Ясно е, че в извода $yaS \stackrel{\Gamma^+}{=} yaz$ не се включват правила от $P^+ \setminus P$ и следователно този извод може да стане с правила от P , т.е.

$$yaS \stackrel{\Gamma}{=} yaz \text{ или } S \stackrel{\Gamma}{=} z \in L.$$

Понеже $A \rightarrow aS$ не е правило в P , следва, че в P има правило $A \rightarrow a$. Тогава

$$S \stackrel{\Gamma^+}{=} yA \stackrel{\Gamma}{=} ya$$

е извод на елемента $ya \in L(\Gamma^+)$ за който $n(ya) < k$. Следователно $ya \in L^+$ и $yaz \in L^+$, т.е. $\omega \in L^+$.

Нека сега $\omega \in L^+$. Следователно $\omega \in L^n$ за някое естествено число n . Ще покажем, чрез индукция по n , че $\omega \in L(\Gamma^+)$.

При $n = 1$ имаме $\omega \in L$. Понеже всички правила в P са правила и в P^+ следва, че $\omega \in L(\Gamma^+)$.

Да допуснем, че при $n < k$ е в сила $\omega \in L(\Gamma^+)$.

Да разгледаме произволна дума ω , за която $\omega \in L^k$. Следователно ω се представя, като дума от вида $\omega = yz$, където $y \in L, z \in L^{k-1}$. Нека $S \stackrel{\Gamma}{=} y$ е извод на y в Γ . Последното правило в този извод е от вида $A \rightarrow a$ и $y = y'a$, като $y' \in V^*$. Следователно $A \rightarrow aS$ е правило в P^+ и yS е сентенциална форма на Γ^+ . От индуктивното допускане следва, че съществува извод на z от S в граматиката Γ^+ . Следователно yz може да се изведе в граматиката Γ^+ , т.е. $yz \in L(\Gamma^+)$. \square

Като вземем в предвид Лема ?? от предния параграф следва, че ако $\varepsilon \in L$ и L е автоматен език, то L^+ ще бъде автоматен, а също така и L^* ще е автоматен език.

Преди да докажем, че обединението на два автоматни езика е автоматен език, да разгледаме един пример.

Пример 2.4.1 Нека $L_1 = \{a^m b^n \mid m, n \geq 1\}$ е езикът, породен от граматиката Γ_1 с правила

$$S \rightarrow aS|aB, \quad B \rightarrow bB|b.$$

Нека $L_2 = \{c^n \mid n \geq 1\}$ е езикът, породен от граматиката $\Gamma_2 = \langle \{c\}, \{S\}, S, \{S \rightarrow cS|c\} \rangle$. Тогава обединението на правилата от Γ_1 и Γ_2 ще даде следните правила

$$S \rightarrow aS|aB|cS|c, \quad B \rightarrow bB|b.$$

В новите правила може да се получат думи, които не са в езика $L_1 \cup L_2$. Например думата $ac \notin L_1 \cup L_2$, а същата може да бъде изведена по следната схема.

$$S \vdash aS \vdash ac.$$

Този пример показва, че механичното обединение на правилата от Γ_1 и Γ_2 не винаги ще доведе до правилата на новата граматика, пораждаща езика $L_1 \cup L_2$.

Това затруднение може да бъде отстранено например, ако се въведе нов начален символ S_1 и се преименуват нетерминалните символи в една от граматиките. Така получаваме

$$\Gamma_1 = \langle \{a, b\}, S, \{S \rightarrow aS|aB, B \rightarrow bB|b\} \rangle$$

и

$$\Gamma_2 = \langle \{c\}, \{S_2\}, S_2, \{S_2 \rightarrow cS_2|c\} \rangle.$$

Тогава граматиката, която поражда $L_1 \cup L_2$ е

$$\Gamma = \langle \{a, b, c\}, \{S_1, S, S_2, B\}, S_1, \{S_1 \rightarrow aS|aB|cS_2|c, \\ S \rightarrow aS|aB, B \rightarrow bB|b, S_2 \rightarrow cS_2|c\} \rangle.$$

Първото правило може да се замени с $S \rightarrow S|S_2$, но то не е правило на автоматна граматика и това налага наличието на правилата $S_1 \rightarrow \alpha$, за всяко правило $S \rightarrow \alpha$ в Γ_1 или $S_2 \rightarrow \alpha$ в Γ_2 .

Теорема 2.4.2 Ако L_1 и L_2 са автоматни езици, които не съдържат празната дума ε , то и $L_1 \cup L_2$ е автоматен език.

Доказателство: Нека $\Gamma_1 = \langle V_1, W_1, S_1, P_1 \rangle$ и $\Gamma_2 = \langle V_2, W_2, S_2, P_2 \rangle$ са автоматни граматики, за които $L(\Gamma_1) = L_1$

и $L(\Gamma_2) = L_2$, като $W_1 \cap W_2 = \emptyset$. Да разгледаме граматиката $\Gamma = \langle V, W, S, P \rangle$, получена по следния начин:

$V = V_1 \cup V_2$, $W = W_1 \cup W_2 \cup \{S\}$, $S \notin W_1 \cup W_2$ и правилата на Γ са $P = P_1 \cup P_2 \cup P_3$, като в P_3 са правила от вида $S \rightarrow \alpha$, за всяко правило $S_1 \rightarrow \alpha$ в P_1 или $S_2 \rightarrow \alpha$ в P_2 .

Нека $\omega \in L_1 \cup L_2$. Без ограничение на общността да приемем, че $\omega \in L_1$. Следователно $S_1 \stackrel{\Gamma_1}{=} \omega$. Понеже $P_1 \subset P$ следва, че всички правила в този извод са правила в Γ , като съществува правило $S \rightarrow \alpha$, където α непосредствено се извежда от S_1 в P_1 и $S_1 \stackrel{\Gamma_1}{\vdash} \alpha \stackrel{\Gamma_1}{=} \omega$. Следователно в Γ може да се изведе ω по следната схема

$$S \stackrel{\Gamma}{\vdash} \alpha \stackrel{\Gamma}{=} \omega.$$

Нека сега $\omega \in L(\Gamma)$ и нейния извод в Γ е $S \stackrel{\Gamma}{=} \omega$. Ако дължината на този извод е 1, то в P има правило $S_1 \rightarrow \omega$ или $S_2 \rightarrow \omega$. Понеже $W_1 \cap W_2 = \emptyset$ следва, че $\omega \in L(\Gamma_1)$ или $\omega \in L(\Gamma_2)$, т.е. $\omega \in L(\Gamma_1) \cup L(\Gamma_2) = L_1 \cup L_2$. Да предположим ω се извежда в Γ по следния начин $S \stackrel{\Gamma}{\vdash} \beta \stackrel{\Gamma}{=} \omega$, $\beta \in (V \cup W)^*$. Следователно в P_1 (или P_2) има правило $S_1 \rightarrow \beta$ (или $S_2 \rightarrow \beta$). Отново поради $W_1 \cap W_2 = \emptyset$ следва, че $\omega \in L(\Gamma_1)$ или $\omega \in L(\Gamma_2)$, т.е. $\omega \in L_1 \cup L_2$. \square .

Теорема 2.4.3 *Ако L_1 и L_2 са автоматни езици, които не съдържат празната дума ε , то и $L_1 L_2$ е автоматен език.*

Доказателството на тази теорема ще предоставим за самостоятелно упражнение, но ще построим граматиката, която поражда езика $L_1 L_2$.

Нека $\Gamma_1 = \langle V_1, W_1, S_1, P_1 \rangle$ и $\Gamma_2 = \langle V_2, W_2, S_2, P_2 \rangle$ са автоматни граматики, които пораждат съответно L_1 и L_2 .

Да разгледаме граматиката $\Gamma = \langle V, W, S_1, P \rangle$, за която:

$$V = V_1 \cup V_2, \quad W = W_1 \cup W_2, \quad W_1 \cap W_2 = \emptyset$$

и $P = P_2 \cup P_3$, а P_3 се състои от правилата в P_1 , като всяко правило $A \rightarrow a$ е заменено с правилото $A \rightarrow aS_2$.

За разгледания по-горе пример, граматиката, която поражда L_1L_2 е

$$\Gamma = \langle \{a, b, c\}, \{S, S_2, B\}, S, \{S \rightarrow aS|aB, B \rightarrow bB|bS_2, S_2 \rightarrow cS_2|c\} \rangle.$$

И тук като се използва Лема ?? от предния параграф може да се докаже, че изискването L_1 и L_2 да не съдържат празната дума ε може да бъде изоставено и от това, че L_1 и L_2 са автоматни да следва, че $L_1 \cup L_2$ и L_1L_2 са автоматни езици.

З а д а ч и

1. Нека L е автоматен език над азбуката V и h е хомоморфизъм във V^* . Да се докаже, че $h(L) = \{h(\omega) \mid \omega \in L\}$ е автоматен език.
2. Като се използва предната задача да се докаже, че ако L е автоматен език, то и L^+ е автоматен език.
3. Да се докаже, че ако L е краен език, то L е автоматен.

У п ъ т в а н е: Докажете, че ако $L' = \{\omega\}$ и $\omega = a_1a_2\dots a_s$, то L' се поражда от граматика с правила $S \rightarrow a_1A_1$, $A_i \rightarrow a_{i+1}A_{i+1}$ за $i = 1, 2, \dots, s-2$ и $A_{s-1} \rightarrow a_s$.