

## Зміст

<b>12 Побудова <math>LL(k)</math>-синтаксичного аналізатора</b>	<b>1</b>
12.1 Побудова $LL(k)$ -синтаксичного аналізатора . . . . .	1
12.1.1 $Local_k(S, A)$ . . . . .	1
12.1.2 Таблиці керування . . . . .	2
12.1.3 Приклад . . . . .	3
12.1.4 Алгоритм . . . . .	4
12.2 Контрольні запитання . . . . .	5

## 12 Побудова $LL(k)$ -синтаксичного аналізатора

### 12.1 Побудова $LL(k)$ -синтаксичного аналізатора

Повернемось до умови, при якій граматика  $G$  буде  $LL(k)$ -граматикою, а саме: для довільного виведення  $S \Rightarrow^* \omega_1 A \omega_2$  та правила  $A \mapsto \alpha \mid \beta$  маємо  $First_l(\alpha \cdot L) \cap First_k(\beta \cdot L) = \emptyset$ , де  $L = First_k(\omega_2)$ .

Оскільки  $L \subseteq \Sigma^{*k}$  — конструктивна множина, спробуємо побудувати всілякі множини  $L$ , які задовольняють попередньо сформульованій умові.

#### 12.1.1 $Local_k(S, A)$

Визначимо наступну множину:

$$Local_k(S, A) = \{L \mid \exists x, \omega : S \Rightarrow^* x A \omega, L = First_k(\omega)\}$$

Опишемо алгоритм пошуку цієї множини:

1.  $\delta_0(S, S) = \{\{\varepsilon\}\}$ , в інших випадках — невизначено.
2.  $\delta_1(S, A_i) = \delta_0(S, A_i) \cup \{L \mid S \mapsto \omega_1 A_i \omega_2, L = First_k(\omega_2)\}$ , в інших випадках — невизначено.
- 3.

$$\begin{aligned} \delta_n(S, A_i) &= \delta_{n-1}(S, A_i) \cup \\ &\cup \{L \mid A_j \mapsto \omega_1 A_i \omega_2, L = First_k(\omega_2) \oplus_k L_p, L_p \in Local_k(S, A_j)\}, \end{aligned}$$

в інших випадках — невизначено.

4.  $\delta_m(S, A_i) = \delta_{m+1}(S, A_i) = \dots, \forall A_i \in N$ .

Тоді  $\text{Local}_k(S, A_i) = \delta_m(S, A_i)$ .

Виходячи з означення  $\text{Local}_k(S, A_i)$ , умови для  $LL(k)$ -граматики будуть наступними: для довільного  $A$ -правила вигляду  $A \mapsto \omega_1 \mid \omega_2 \mid \dots \mid \omega_p$  маємо:

$$\text{First}_k(\omega_i \cdot L_m) \cap \text{First}_k(\omega_j \cdot L_m) = \emptyset, \quad i \neq k, \quad L_m \in \text{Local}_k(S, A).$$

Як наслідок, з алгоритму пошуку  $\text{Local}_k(S, A_i)$  видно, що

$$\text{Follow}_k(A_i) = \bigcup_{j=1}^m L_j, \quad L_j \in \text{Local}_k(S, A_i).$$

### 12.1.2 Таблиці керування

Для побудови синтаксичного аналізатора для  $LL(k)$ -граматики ( $k > 1$ ) необхідно побудувати множину таблиць, що забезпечать нам безтупиковий аналіз вхідного ланцюжка  $w$  (програми) за час  $O(n)$ , де  $n = |w|$ .

Побудову множини таблиць для управління  $LL(k)$ -аналізатором почнемо з таблиці, яка визначає перший крок безпосереднього виводу  $w$  в граматичі  $G$ :  $T_0 = T_{S, \{\varepsilon\}}(u) = (T_1 \alpha_1 T_2 \alpha_2 \dots T_p \alpha_p, n)$ , де  $n$  — номер правила вигляду  $S \mapsto A_1 \alpha_1 A_2 \alpha_2 \dots A_p \alpha_p$ , а  $A_i \in N$ ,  $\alpha_i \in \Sigma^*$ , і  $u = \text{First}_k(A_1 \alpha_1 A_2 \alpha_2 \dots A_p \alpha_p)$ , і нарешті  $i = \overline{1..p}$ . Зрозуміло, що в інших випадках (якщо такого правила немає абощо)  $T_0$  не визначена.

Неформально, коли в магазині автомата знаходиться аксіома  $S$ , то нас цікавить перших  $k$  термінальних символів, які можна вивести з  $S$  (аксіома — поняття “програма”) при умові, що після неї (програми) буде досягнуто EOF.

Імена інших таблиць  $T_1, T_2, \dots, T_p$  визначаються так:  $T_i = T_{A_i, L_i}$ , де  $L_i = \text{First}_k(\alpha_i A_{i+1} \alpha_{i+1} \dots A_p \alpha_p)$ ,  $i = \overline{1..p}$ .

Наступні таблиці визначаються так:  $T_i = T_{A_i, L_i}(u) = (T_1 \alpha_1 T_2 \alpha_2 \dots T_p \alpha_p, n)$ , де  $n$  — номер правила вигляду  $A_i \mapsto A_1 \alpha_1 A_2 \alpha_2 \dots A_p \alpha_p$ , а  $A_j \in N$ ,  $\alpha_j \in \Sigma^*$ , і  $u = \text{First}_k(A_1 \alpha_1 A_2 \alpha_2 \dots A_p \alpha_p) \oplus_k L_i$ , і нарешті  $j = \overline{1..p}$ . Зрозуміло, що в інших випадках (якщо такого правила немає абощо)  $T_i$  не визначена.

Імена інших таблиць  $T_1, T_2, \dots, T_p$  визначаються так:  $T_j = T_{A_j, L_j}$ , де  $L_j = \text{First}_k(\alpha_j A_{j+1} \alpha_{j+1} \dots A_p \alpha_p) \oplus_k L_i$ ,  $j = \overline{1..p}$ .

### 12.1.3 Приклад

Побудувати множину таблиць управління для  $LL(2)$ -граматики з наступною схемою правил:

$$S \mapsto abA, \quad (1)$$

$$S \mapsto \varepsilon, \quad (2)$$

$$A \mapsto Saa, \quad (3)$$

$$A \mapsto b. \quad (4)$$

Для вищенаведеної граматики множини  $\text{First}_2(A_i)$ ,  $A_i \in N$  будуть такі:  $\text{First}_2(S) = \{ab, \varepsilon\}$ ,  $\text{First}_2(A) = \{aa, ab, b\}$ , а множини  $\text{Local}_2(S, A_i)$ ,  $A_i \in N$  будуть такі:  $\text{Local}_2(S, S) = \text{Local}_2(S, A) = \{\{\varepsilon\}, \{aa\}\}$ .

Побудуємо першу таблицю  $T_0 = T_{S, \{\varepsilon\}}$ . Для  $S$ -правила відповідні множини  $u$  будуть такі:

- $S \mapsto abA$ ,  $u \in \text{First}_2(abA) = \{ab\}$ .
- $S \mapsto \varepsilon$ ,  $u \in \text{First}_2(\varepsilon) = \{\varepsilon\}$ .

Таблиця  $T_0$  визначається так:

	$aa$	$ab$	$ba$	$bb$	$a$	$b$	$\varepsilon$
$T_0 = T_{S, \{\varepsilon\}}$		$abT_1, 1$					$\varepsilon, 2$

Нова таблиця управління  $T_1 = T_{A, \{\varepsilon\}}$ . Для  $A$ -правила відповідні множини  $u$  будуть такі:

- $A \mapsto Saa$ ,  $u \in \text{First}_2(Saa) \oplus_2 \{\varepsilon\} = \{ab, aa\}$ .
- $A \mapsto b$ ,  $u \in \text{First}_2(b) \oplus_2 \{\varepsilon\} = \{b\}$ .

Таблиця  $T_1$  визначається так:

	$aa$	$ab$	$ba$	$bb$	$a$	$b$	$\varepsilon$
$T_1 = T_{A, \{\varepsilon\}}$	$T_2aa, 3$	$T_2aa, 3$				$b, 4$	

Нова таблиця управління  $T_2 = T_{S, L}$  де  $L = \text{First}_2(aa) \oplus_2 \{\varepsilon\} = \{aa\}$ . Для таблиці  $T_2$  та  $S$ -правила множини  $u$  будуть такі

- $S \mapsto abA$ ,  $u \in \text{First}_2(abA) \oplus_2 \{aa\} = \{ab\} \oplus_2 \{aa\} = \{ab\}$ .
- $S \mapsto \varepsilon$ ,  $u \in \text{First}_2(\varepsilon) \oplus_2 \{aa\} = \{aa\}$ .

	$aa$	$ab$	$ba$	$bb$	$a$	$b$	$\varepsilon$
$T_2 = T_{S, \{aa\}}$	$\varepsilon, 2$	$abT_3, 1$					

Наступна таблиця  $T_3 = T_{A, L}$  де  $L = \text{First}_2(\varepsilon) \oplus_2 \{aa\} = \{aa\}$ . Для таблиці  $T_3$  та  $A$ -правила множини  $u$  будуть такі:

- $A \mapsto Saa, u \in \text{First}_2(Saa) \oplus_2 \{aa\} = \{ab, aa\}$ .
- $A \mapsto b, u \in \text{First}_2(b) \oplus_2 \{aa\} = \{ba\}$ .

Таблиця  $T_3$  визначається так:

	$aa$	$ab$	$ba$	$bb$	$a$	$b$	$\varepsilon$
$T_3 = T_{A, \{aa\}}$	$T_2aa, 3$	$T_2aa, 3$	$b, 4$				

Нова таблиця  $T_4 = T_{S, L} = T_2$ , оскільки  $L = \text{First}_2(aa) \oplus_2 \{aa\} = \{aa\}$ .

Ми визначили чотири таблиці-рядки (а їх кількість для довільної  $LL(k)$ -граматики визначається як  $\sum_{i=1}^n n_i$ , де  $n_i$  — кількість елементів множини  $\text{Local}_k(S, A_i)$ ,  $m = |N|$ ).

Об'єднаємо рядки-таблиці в єдину таблицю та виконаємо перейменування рядків:

	$aa$	$ab$	$ba$	$bb$	$a$	$b$	$\varepsilon$
$T_0$		$abT_1, 1$					$\varepsilon, 2$
$T_1$	$T_2aa, 3$	$T_2aa, 3$				$b, 4$	
$T_2$	$\varepsilon, 2$	$abT_3, 1$					
$T_3$	$T_2aa, 3$	$T_2aa, 3$	$b, 4$				

#### 12.1.4 Алгоритм

Синтаксичний аналізатор для  $LL(k)$ -граматики ( $k > 1$ ).

1. Прочитати  $k$  лексем з вхідного файла програми (звичайно, інколи менше ніж  $k$ ). В магазин занести таблицю  $T_0$ .
2. Загальний крок:
  - Якщо на вершині магазину знаходиться таблиця  $T_i$ , то елемент таблиці  $M(T_i, \langle k \text{ вхідних лексем} \rangle)$  визначає ланцюжок, який  $T_i$  заміщає на вершині магазину.

- Якщо на вершині магазину  $a_i \in \Sigma$  перша поточна лексема з  $k$  прочитаних лексем рівна  $a_i$ , то з вершини магазину зняти  $a_i$  та прочитати з вхідного файлу додатково одну лексему (звичайно, якщо це можливо).
- Якщо досягли кінця вхідного файлу програми та магазин порожній, то програма не має синтаксичних помилок.
- В інших випадках — синтаксична помилка.

## 12.2 Контрольні запитання

1. Наведіть визначення множини  $\text{Local}_k(S, A)$ .
2. Опишіть алгоритм побудови  $\text{Local}_k(S, A)$ .
3. Опишіть алгоритм побудови таблиць керування (або рядків великої результуючої таблиці керування).
4. Якою формулою визначається кількість рядків таблиці керування?
5. Опишіть алгоритм синтаксичного аналізу для  $LL(k)$ -граматики.