

## Зміст

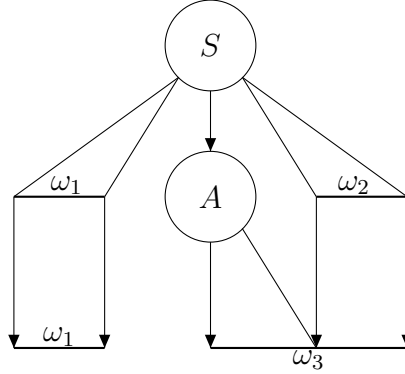
<b>9 Синтаксичний аналіз без повернення назад</b>	<b>1</b>
9.0.1 Проблеми загальних граматик . . . . .	1
9.1 $LL(k)$ -граматики . . . . .	2
9.1.1 $\text{First}_k$ . . . . .	4
9.1.2 Алгоритм пошуку $\text{First}_k$ . . . . .	4
9.2 Сильні $LL(k)$ -граматики . . . . .	6
9.2.1 Не всі грамматики сильні . . . . .	6
9.2.2 Алгоритм пошуку $\text{Follow}_k$ . . . . .	7
9.2.3 $\varepsilon$ -нетермінали . . . . .	8
9.2.4 Ліва рекурсія . . . . .	8
9.3 Контрольні запитання . . . . .	10

## 9 Синтаксичний аналіз без повернення назад

### 9.0.1 Проблеми загальних граматик

При виведенні слова  $\omega$  в  $G$  на кожному кроці безпосереднього виведення, коли ми беремо до уваги виділений нами нетермінал (в залежності від стратегії виведення), виникає питання, яку альтернативу для  $A_i$  використати. З точки зору практики, нас цікавить така стратегія виведення  $\omega$  в граматиці  $G$ , коли кожний наступний крок безпосереднього виведення наближав би нас до мети. Ця стратегія дасть можливість виконати виведення  $\omega$  в  $G$  за час  $O(n)$ , де  $n = |\omega|$ .

Зрозуміло, що не маючи інформації про структуру  $\omega$ , досягнути вибраної нами мети в більшості випадків неможливо. Але ж тримати інформацію про все слово  $\omega$  також недопустимо. З точки зору практики, отримати потрібний результат розумно при наявності локальної інформації, наприклад,  $k$  поточних вхідних лексем програми ( $k$  — наперед фіксоване число) достатньо для організації виведення  $\omega$  в  $G$  за час  $O(n)$ . З точки зору синтаксичного аналізу слова  $\omega$  мова ведеться про наступну ситуацію:



Зафіксуємо стратегію виведення: далі будемо розглядати лише лівосторонню стратегію виведення  $\omega$  в  $G$ . Тоді:

- $S \Rightarrow^* \omega_1 A \omega_2$  ( $A$  — перший зліва направо нетермінал);
- $\omega_1$  — термінальна частина слова  $\omega$ , яку вже виведено (проаналізована частина слова);
- результат  $\omega_3$ , який потрібно ще вивести, виводиться зі слова  $A \omega_2$ ;
- щоб зробити вірний крок виведення (без повернення назад) нам достатньо  $k$  поточних вхідних символів з непроаналізованої частини програми  $\omega_3$ .

Сформульовані умови забезпечує клас  $LL(k)$ -грамматик.

## 9.1 $LL(k)$ -грамматики

КС-грамматика  $G = \langle N, \Sigma, P, S \rangle$  називається  $LL(k)$ -грамматикою для деякого фіксованого  $k$ , якщо для двох лівосторонніх виведень вигляду:

1.  $S \Rightarrow^* \omega_1 A \omega_2 \Rightarrow \omega_1 \alpha \omega_2 \Rightarrow^* \omega_1 x$ ;
2.  $S \Rightarrow^* \omega_1 A \omega_2 \Rightarrow \omega_1 \beta \omega_2 \Rightarrow^* \omega_1 y$ ;

з  $\text{First}_k(x) = \text{First}_k(y)$  випливає, що  $\alpha = \beta$ , де  $A \mapsto \alpha \mid \beta$ , а

$$\text{First}_k(\alpha) = \{\omega \mid \alpha \Rightarrow^* \omega x, |\omega| = k\} \cup \{\omega \mid \alpha \Rightarrow^* \omega, |\omega| < k\}.$$

Неформально, граматика  $G$  буде  $LL(k)$ -грамматикою, якщо для слова

$$\omega_1 A \omega_2 \in (N \cup \Sigma)^*$$

достатньо  $k$  перших символів (за умови, що вони існують) решти не-проаналізованого слова щоб визначити, що з  $A\omega_2$  існує не більше однієї альтернативи виведення слова, що починається з  $\omega$  та продовжується наступними  $k$  термінальними символами.

Сформулюємо основні твердження стосовно класу  $LL(k)$ -граматик:

1. Не існує алгоритма, який перевіряє належність КС-граматики класу  $LL(k)$ -граматик.
2. Для кожного конкретного  $k$  існує алгоритм, який перевіряє, чи є задана граматика  $LL(k)$ -граматикою.
3. Якщо граматика є  $LL(k)$ -граматикою, то вона є  $LL(k+p)$ -граматикою, ( $p \geq 1$ ).
4. Клас  $LL(k)$ -граматик — це підклас КС-граматик, який не покриває його.

Продемонструємо на **прикладі** справедливність останнього твердження. Розглянемо граматика  $G$  з наступною схемою  $P: S \mapsto Sa \mid b$ .

Мова, яку породжує наведена вище граматика  $L(G) = \{ba^i, i = 0, 1, \dots\}$ . Візьмемо виведення наступного слова  $S \Rightarrow^{i+1} ba^i$ . За визначенням  $LL(k)$ -граматики якщо покласти  $A = S$ ,  $\omega_2 = a^i$ ,  $\alpha = Sa$ ,  $\beta = b$ , то маємо отримати

$$\text{First}_k(Saa^i) \cap \text{First}_k(ba^i) = \emptyset.$$

Втім, для  $i \geq k$  маємо:

$$\text{First}_k(Saa^i) = \text{First}_k(ba^i) = \{ba^{k-1}\}.$$

Таким чином, КС-граматика  $G$  не може бути  $LL(k)$ -граматикою для жодного  $k$ .

Як наслідок, КС-граматика  $G$ , яка має ліворекурсивний нетермінал  $A$  (нетермінал  $A$  називається *ліворекурсивним*, якщо в граматичі  $G$  існує вивід виду  $A \Rightarrow^* A\omega$ ), не може бути  $LL(k)$ -граматикою.

З практичної точки зору в більшості випадків ми будемо користуватися  $LL(1)$ -граматиками. У класі  $LL(1)$ -граматик існує один цікавий підклас — це розподілені  $LL(1)$ -граматики.

$LL(1)$ -граматика називається *розподіленою*, якщо вона задовольняє наступним умовам:

- у схемі  $P$  граматичи відсутні  $\varepsilon$ -правила (правила вигляду  $A \mapsto \varepsilon$ );
- для нетерміналу  $A$  праві частини  $A$ -правила починаються різними терміналами.

### 9.1.1 First<sub>k</sub>

Зауважимо, що  $\text{First}_k(\omega_1\omega_2) = \text{First}_k(\omega_1) \oplus_k \text{First}_k(\omega_2)$ , де  $\oplus_k$  — бінарна операція над словарними множинами (мовами) визначена наступним чином:

$$L_1 \oplus_k L_2 = \{\omega \mid \omega\omega_1 = xy, |\omega| = k\} \cup \{\omega \mid \omega = xy, |\omega| < k\}, \quad x \in L_1, \quad y \in L_2.$$

Звідси маємо наступний тривіальний висновок: якщо  $\omega = \alpha_1\alpha_2 \dots \alpha_p$ , де  $\alpha_i \in (N \cup \Sigma)$ , то

$$\text{First}_k(\omega) = \text{First}_k(\alpha_1) \oplus_k \text{First}_k(\alpha_2) \oplus_k \dots \oplus_k \text{First}_k(\alpha_p)$$

Для подальшого аналізу визначення  $LL(k)$ -граматики розглянемо алгоритм обчислення функції  $\text{First}_k(\alpha)$ ,  $\alpha \in (N \cup \Sigma)$ .

### 9.1.2 Алгоритм пошуку First<sub>k</sub>

Очевидно, що якщо  $\alpha_i \in \Sigma$ , то  $\text{First}_k(\alpha_i) = \{\alpha_i\}$  при  $k > 0$ . Розглянемо алгоритм пошуку  $\text{First}_k(A_i)$ ,  $A_i \in N$ .

**Алгоритм [пошуку First<sub>k</sub>(A<sub>i</sub>), A<sub>i</sub> ∈ N]:** визначимо значення функції  $F_i(x)$  для кожного  $x \in (N \cup \Sigma)$ :

1.  $F_i(a) = \{a\}$  для всіх  $a \in \Sigma$ ,  $i \geq 0$ .
2.  $F_0(A_i) = \{\omega \mid \omega \in \Sigma^{*k} : A_i \mapsto \omega x, |\omega| = k\} \cup \{\omega \mid \omega \in \Sigma^{*k} : A_i \mapsto \omega, |\omega| < k\}$ .
- 3.

$$F_n(A) = F_{n-1}(A_i) \cup \{\omega \mid \omega \in \Sigma^{*k} : \omega \in F_{n-1}(\alpha_1) \oplus_k \dots \oplus_k F_{n-1}(\alpha_p), A_i \mapsto \alpha_1 \dots \alpha_p\}.$$

4.  $F_m(A_i) = F_{m+1}(A_i) = \dots$  для всіх  $A_i \in N$ .

Очевидно, що:

- послідовність  $F_0(A_i) \subseteq F_1(A_i) \subseteq \dots$  — монотонно зростаюча;
- $F_n(A_i) \subseteq \Sigma^{*k}$  — послідовність обмежена зверху.

Тоді покладемо  $\text{First}_k(A_i) = F_m(A_i)$  для кожного  $A_i \in N$ .

**Приклад:** знайти множину  $\text{First}_k(A_i)$  для нетерміналів граматики з наступною схемою правил:

$$\begin{aligned} S &\mapsto BA, \\ A &\mapsto +BA \mid \varepsilon, \\ B &\mapsto DC, \\ C &\mapsto \times DC \mid \varepsilon, \\ D &\mapsto (S) \mid a. \end{aligned}$$

Нехай  $k = 2$ , тоді маємо наступну таблицю:

	$S$	$A$	$B$	$C$	$D$
$F_0$	$\emptyset$	$\{\varepsilon\}$	$\emptyset$	$\{\varepsilon\}$	$\{a\}$
$F_1$	$\emptyset$	$\{\varepsilon\}$	$\{a\}$	$\{\varepsilon, \times a\}$	$\{a\}$
$F_2$	$\{a\}$	$\{\varepsilon, +a\}$	$\{a, a \times\}$	$\{\varepsilon, \times a\}$	$\{a\}$
$F_3$	$\{a, a+, a \times\}$	$\{\varepsilon, +a\}$	$\{a, a \times\}$	$\{\varepsilon, \times a\}$	$\{a, (a\}$
$F_4$	$\{a, a+, a \times\}$	$\{\varepsilon, +a\}$	$\{a, a \times, (a\}$	$\{\varepsilon, \times a, \times (\}$	$\{a, (a\}$
$F_5$	$\{a, a+, a \times, (a\}$	$\{\varepsilon, +a, +(\}$	$\{a, a \times, (a\}$	$\{\varepsilon, \times a, \times (\}$	$\{a, (a\}$
$F_6$	$\{a, a+, a \times, (a\}$	$\{\varepsilon, +a, +(\}$	$\{a, a \times, (a\}$	$\{\varepsilon, \times a, \times (\}$	$\{a, (a, ((\}$
$F_7$	$\{a, a+, a \times, (a\}$	$\{\varepsilon, +a, +(\}$	$\{a, a \times, (a, ((\}$	$\{\varepsilon, \times a, \times (\}$	$\{a, (a, ((\}$
$F_8$	$\{a, a+, a \times, (a, ((\}$	$\{\varepsilon, +a, +(\}$	$\{a, a \times, (a, ((\}$	$\{\varepsilon, \times a, \times (\}$	$\{a, (a, ((\}$
$F_9$	$\{a, a+, a \times, (a, ((\}$	$\{\varepsilon, +a, +(\}$	$\{a, a \times, (a, ((\}$	$\{\varepsilon, \times a, \times (\}$	$\{a, (a, ((\}$

Скористаємося визначенням  $\text{First}_k(\alpha)$  сформулюємо необхідні й достатні умови, за яких КС-граматика буде  $LL(k)$ -граматикою: для довільного виводу в граматиці  $G$  вигляду  $S \Rightarrow^* \omega_1 A \omega_2$  та правила  $A \mapsto \alpha \mid \beta$ :

$$\text{First}_k(\alpha \omega_2) \cap \text{First}_k(\beta \omega_2) = \emptyset.$$

Вище сформульована умова для  $LL(k)$ -граматик може бути перефразована з урахуванням визначення множини  $\text{First}_k$ : для довільного виведення в граматиці  $G$  вигляду  $S \Rightarrow^* \omega_1 A \omega_2$  та правила  $A \mapsto \alpha \mid \beta$ :

$$\text{First}_k(\alpha \cdot L) \cap \text{First}_k(\beta \cdot L) = \emptyset, \quad L = \text{First}_k(\omega_2).$$

Оскільки  $L \subseteq \Sigma^{*k}$ , то остання умова є конструктивною умовою і може бути використана для перевірки, чи КС-граматика є  $LL(k)$ -граматикою для фіксованого  $k$ .

## 9.2 Сильні $LL(k)$ -граматики

КС-граматика називається *сильною  $LL(k)$ -граматикою*, якщо для кожного правила вигляду  $A \mapsto \alpha \mid \beta$  виконується умова:

$$\text{First}_k(\alpha \cdot \text{Follow}_k(A)) \cap \text{First}_k(\beta \cdot \text{Follow}_k(A)) = \emptyset,$$

де  $\text{Follow}_k(\alpha)$ ,  $\alpha \in (N \cup \Sigma)^*$  визначається так:

$$\text{Follow}_k(\alpha) = \{\omega \mid S \Rightarrow^* \omega_1 \alpha \omega_2, \omega \in \text{First}_k(\omega_2)\}.$$

Неформально, відмінність сильних  $LL(k)$ -граматик від звичайних  $LL(k)$ -граматик полягає у тому, що наступне правило безпосереднього виведення, яке буде застосовано до  $A$  можна визначити абстраговано від уже виведеної частини слова  $\omega_1$ , розглядаючи тільки наступні  $k$  символів які потрібно отримати після  $A$ .

Операції  $\text{First}_k$  та  $\text{Follow}_k$  можна узагальнити для словарної множини  $L$ , тоді:

$$\begin{aligned} \text{First}_k(L) &= \{\omega \mid \exists \alpha_i \in L : \omega \in \text{First}_k(\alpha_i)\}. \\ \text{Follow}_k(L) &= \{\omega \mid \exists \alpha_i \in L : S \Rightarrow^* \omega_1 \alpha_i \omega_2, \omega \in \text{First}_k(\omega_2)\}. \end{aligned}$$

Без доведення зафіксуємо наступні твердження:

- кожна  $LL(1)$ -граматика є сильною  $LL(1)$ -граматикою;
- існують  $LL(k)$ -граматики ( $k > 1$ ), які не є сильними  $LL(k)$ -граматиками.

### 9.2.1 Не всі граматиками сильні

На **прикладі** продемонструємо останнє твердження. Нехай граматика  $G$  визначена наступними правилами:  $S \mapsto aAaa \mid bAba$ ,  $A \mapsto b \mid \varepsilon$ .

Відповідні множини  $\text{First}_2(S) = \{ab, aa, bb\}$ ,  $\text{First}_2(A) = \{b, \varepsilon\}$ ,  $\text{Follow}_2(A) = \{aa, ba\}$ ,  $\text{Follow}_2(S) = \{\varepsilon\}$ .

Перевіримо умову для сильної  $LL(2)$ -граматики:

1. виконаємо перевірку  $LL(2)$ -умови для правила  $S \mapsto aAaa \mid bAba$ :

$$\begin{aligned} &\text{First}_2(aAaa \cdot \text{Follow}_2(S)) \cap \text{First}_2(bAba \cdot \text{Follow}_2(S)) = \\ &= (\text{First}_2(aAaa) \oplus_2 \text{Follow}_2(S)) \cap (\text{First}_2(bAba) \oplus_2 \text{Follow}_2(S)) = \\ &= (\{ab, aa\} \oplus_2 \{\varepsilon\}) \cap (\{bb\} \oplus_2 \{\varepsilon\}) = \{ab, aa\} \cap \{bb\} = \emptyset. \end{aligned}$$

2. виконаємо перевірку  $LL(2)$ -умови для правила  $A \mapsto b \mid \varepsilon$ :

$$\begin{aligned} \text{First}_2(b \cdot \text{Follow}_2(A)) \cap \text{First}_2(\varepsilon \cdot \text{Follow}_2(A)) &= \\ &= \{ba, bb\} \cap \{aa, ba\} = \{ba\}. \end{aligned}$$

**Висновок:** вище наведена граматика не є сильною  $LL(2)$ -граматикою. Перевіримо цю ж граматiku на властивість  $LL(2)$ -граматики. Тут ми маємо два різні варіанти виводу з  $S$ :

1.  $S \Rightarrow^* aAaa$ :  $\text{First}_2(b \cdot aa) \cap \text{First}_2(\varepsilon \cdot aa) = \{ba\} \cap \{aa\} = \emptyset$ .
2.  $S \Rightarrow^* bAba$ :  $\text{First}_2(b \cdot ba) \cap \text{First}_2(\varepsilon \cdot ba) = \{bb\} \cap \{ba\} = \emptyset$ .

Висновок: наведена вище граматика є  $LL(2)$ -граматикою.

### 9.2.2 Алгоритм пошуку $\text{Follow}_k$

**Алгоритм [обчислення  $\text{Follow}_k(A_i)$ ,  $A_i \in N$ ]:** будемо розглядати всілякі дерева, які можна побудувати, починаючи з аксіоми  $S$ :

1.  $\sigma_0(S, S) = \{\varepsilon\}$ . Очевидно, за 0 кроків ми виведемо  $S$ , після якої знаходиться  $\varepsilon$ . У інших випадках  $\sigma_0(S, A_i)$  — невизначено,  $A_i \in (N \setminus \{S\})$ .
2.  $\sigma_1(S, A_i) = \sigma_0(S, A_i) \cup \{\omega \mid S \mapsto \omega_1 A_i \omega_2, \omega \in \text{First}_k(\omega_2)\}$ . В інших випадках  $\sigma_1(S, A_i)$  — невизначено.
3.  $\sigma_n(S, A_i) = \sigma_{n-1}(S, A_i) \cup \{\omega \mid A_j \mapsto \omega_1 A_i \omega_2, \omega \in \text{First}_k(\omega_2 \cdot \sigma_{n-1}(S, A_j))\}$ . В інших випадках  $\sigma_n(S, A_i)$  — невизначено.

Настане крок  $m$ , коли  $\sigma_m(S, A_i) = \sigma_{m+1}(S, A_i) = \dots, \forall A_i \in N$ .

Тоді покладемо  $\text{Follow}_k(A_i) = \sigma_m(S, A_i), \forall A_i \in N$ .

Очевидно, що:

- послідовність  $\sigma_0(S, A_i) \subseteq \sigma_1(S, A_i) \subseteq \dots$  монотонно зростаюча;
- $\sigma_n(S, A_i) \subseteq \Sigma^{*k}$  — послідовність обмежена зверху.

Разом ці умови гарантують збіжність послідовності  $\{\sigma_n(S, A_i)\}$ , а отже і алгоритму пошуку  $\text{Follow}_k(A_i)$ .

### 9.2.3 $\varepsilon$ -нетермінали

Нетермінал  $A_i$  КС-граматики  $G$  називається  $\varepsilon$ -нетерміналом, якщо  $A_i \Rightarrow^* \varepsilon$ .

**Алгоритм [пошуку  $\varepsilon$ -нетерміналів]:**

1.  $S_0 = \{A_i \mid A_i \mapsto \varepsilon\}$ .
2.  $S_1 = S_0 \cup \{A_i \mid A_i \mapsto \alpha_1 \alpha_2 \dots \alpha_p, \alpha_j \in S_0, j = \overline{1..p}\}$ .
3.  $S_n = S_{n-1} \cup \{A_i \mid A_i \mapsto \alpha_1 \alpha_2 \dots \alpha_p, \alpha_j \in S_{n-1}, j = \overline{1..p}\}$ .
4.  $S_m = S_{m+1} = \dots$

Тоді множина  $S_m$  — множина  $\varepsilon$ -нетерміналів.

**Приклад.** Для граматики  $G$  з схемою правил  $P$  знайдемо множину  $\varepsilon$ -нетерміналів:

$$\begin{aligned} S &\mapsto aBD \mid D \mid AC \mid b, \\ A &\mapsto SCB \mid SABC \mid CbD \mid \varepsilon, \\ B &\mapsto CA \mid d, \\ C &\mapsto ADC \mid a \mid \varepsilon, \\ D &\mapsto EaC \mid SC, \\ E &\mapsto BCS \mid a. \end{aligned}$$

$$\begin{aligned} S_0 &= \{A, C\}, \\ S_1 &= \{A, C\} \cup \{B, S\}, \\ S_2 &= \{A, B, C, S\} \cup \{D\}, \\ S_3 &= \{A, B, C, S, D\} \cup \{E\}, \\ S_4 &= \{A, B, C, S, D, E\} \cup \{E\}. \end{aligned}$$

Таким чином, множина  $\varepsilon$ -нетерміналів для наведеної вище граматики —  $\{S, A, B, C, D, E\}$ .

### 9.2.4 Ліва рекурсія

До того, як перевірити граматику на  $LL(k)$ -властивість необхідно перевірити її на наявність ліворекурсивних нетерміналів та спробувати уникнути лівої рекурсії.

**Алгоритм [тестування нетермінала  $A_i$  на ліву рекурсію]:** для кожного нетермінала  $A_i$  побудуємо наступну послідовність множин  $S_0, S_1, \dots$ :



1.  $S_0 = \{A_i \mid A_i \mapsto \omega_1 A_i \omega_2, \omega_1 \Rightarrow^* \varepsilon\}$ , починаємо з нетерміналу  $A_i$ .
2.  $S_1 = S_0 \cup \{A_i \mid A_i \mapsto \omega_1 A_j \omega_2, \omega_1 \Rightarrow^* \varepsilon, A_j \in S_0\}$ .
3.  $S_n = S_{n-1} \cup \{A_i \mid A_i \mapsto \omega_1 A_j \omega_2, \omega_1 \Rightarrow^* \varepsilon, A_j \in S_{n-1}\}$ .
4.  $S_m = S_{m+1} = \dots$

Тоді якщо  $A_i \in S_m$ , то  $A_i$  — ліворекурсивний нетермінал.

**Приклад.** Для граматики  $G$  зі схемою правил  $P$  знайдемо множину ліворекурсивних нетерміналів:

$$\begin{aligned}
 S &\mapsto AbS \mid AC, \\
 A &\mapsto BD, \\
 B &\mapsto BC \mid \varepsilon, \\
 C &\mapsto Sa \mid \varepsilon, \\
 D &\mapsto aB \mid BA.
 \end{aligned}$$

Виконаємо процедуру тестування для кожного нетерміналу окремо, наприклад, для нетерміналу  $S$ :

$$\begin{aligned}
 S_0 &= \{A\}, \\
 S_1 &= \{A, B, D\}, \\
 S_2 &= \{A, B, D, C\}, \\
 S_3 &= \{A, B, D, C, S\}.
 \end{aligned}$$

Запропонуємо декілька прийомів, що дають можливість при побудові  $LL(k)$ -грамматик уникнути лівої рекурсії. Розглянемо граматику зі схемою правил  $S \mapsto Sa \mid b$ , яка має ліворекурсивний нетермінал  $S$ . Замінімо схему правил новою схемою з трьома правилами  $S \mapsto bS_1$ ,  $S_1 \mapsto aS_1 \mid \varepsilon$ .

**Приклад:** для граматики  $G$  з схемою правил  $P$  для кожного нетерміналу знайдемо множину  $\text{Follow}_1(A)$  ( $k = 1$ ):

$$\begin{aligned}
 S &\mapsto BA, \\
 A &\mapsto +BA \mid \varepsilon, \\
 B &\mapsto DC, \\
 C &\mapsto \times DC \mid \varepsilon, \\
 D &\mapsto (S) \mid a.
 \end{aligned}$$

З прикладу, що наведено раніше множини  $\text{First}_1(A)$ , будуть такими:

$$\begin{aligned}\text{First}_1(S) &= \text{First}_1(B) = \text{First}_1(D) = \{ (, a \}, \\ \text{First}_1(A) &= \{ +, \varepsilon \}, \\ \text{First}_1(C) &= \{ \times, \varepsilon \}.\end{aligned}$$

	$S$	$A$	$B$	$C$	$D$
$\delta_0$	$\{\varepsilon\}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset$
$\delta_1$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{+, \varepsilon\}$	$\emptyset$	$\emptyset$
$\delta_2$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{+, \varepsilon\}$	$\{+, \varepsilon\}$	$\emptyset$
$\delta_3$	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{+, \varepsilon\}$	$\{+, \varepsilon\}$	$\{\times, +, \varepsilon\}$
$\delta_4$	$\{\varepsilon, )\}$	$\{\varepsilon\}$	$\{+, \varepsilon\}$	$\{+, \varepsilon\}$	$\{\times, +, \varepsilon\}$
$\delta_5$	$\{\varepsilon, )\}$	$\{\varepsilon, )\}$	$\{+, \varepsilon\}$	$\{+, \varepsilon\}$	$\{\times, +, \varepsilon\}$
$\delta_6$	$\{\varepsilon, )\}$	$\{\varepsilon, )\}$	$\{+, \varepsilon, )\}$	$\{+, \varepsilon, )\}$	$\{\times, +, \varepsilon, )\}$
$\delta_7$	$\{\varepsilon, )\}$	$\{\varepsilon, )\}$	$\{+, \varepsilon, )\}$	$\{+, \varepsilon, )\}$	$\{\times, +, \varepsilon, )\}$

Таким чином,  $\text{Follow}_1(S) = \{\varepsilon, )\}$ ,  $\text{Follow}_1(A) = \{\varepsilon, )\}$ ,  $\text{Follow}_1(B) = \{+, \varepsilon, )\}$ ,  $\text{Follow}_1(C) = \{+, \varepsilon, )\}$ ,  $\text{Follow}_1(D) = \{\times, +, \varepsilon, )\}$ .

### 9.3 Контрольні запитання

1. Яка граматика називається  $LL(k)$ -граматика?
2. Чи кожна КС-граматика є  $LL(k)$ -граматикою для деякого  $k$ ?
3. Яка  $LL(1)$ -граматика називається розподіленою?
4. Яку бінарну операцію над мовами позначає символ  $\oplus_k$ ?
5. Яку мову (множину слів) позначає запис  $\text{First}_k(\alpha)$ ?
6. Опишіть алгоритм пошуку  $\text{First}_k$  і доведіть його збіжність.
7. Яка  $LL(k)$ -граматика називається сильною?
8. Чи кожна  $LL(k)$ -граматика є сильною  $LL(k)$ -граматикою?
9. Яку мову (множину слів) позначає запис  $\text{Follow}_k(\alpha)$ ?
10. Опишіть алгоритм пошуку  $\text{Follow}_k$  і доведіть його збіжність.
11. Який нетермінал  $A_i \in N$  називається  $\varepsilon$ -нетерміналом?

12. Опишіть алгоритм перевірки нетерміналу  $A_i \in N$  на  $\varepsilon$ -нетермінал і доведіть його збіжність.
13. Який нетермінал  $A_i \in N$  називається ліворекурсивним?
14. Опишіть алгоритм перевірки нетерміналу  $A_i \in N$  на ліву рекурсію і доведіть його збіжність.