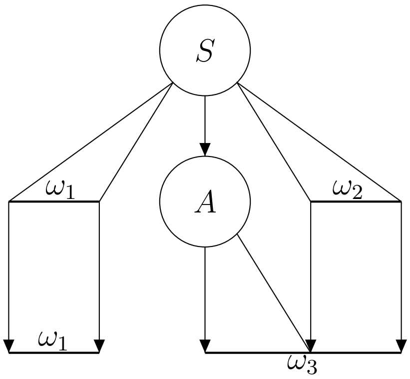
Зміст

9 Синтаксичний аналіз без повернення назад

9.0.1 Проблеми загальних граматик

При виведенні слова ω 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 ω в граматиці G, коли кожний наступний крок безпосереднього виведення наближав би нас до мети. Ця стратегія дасть можливість виконати виведення ω в G за час $O(n)$, де $n = |\omega|$.

Зрозуміло, що не маючи інформації про структуру ω, досягнути вибраної нами мети в більшості випадків неможливо. Але ж тримати інформацію про все слово ω 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 ω в G за час $O(n)$. З точки зору синтаксичного аналізу слова ω мова ведеться про наступну ситуацію:



Зафіксуємо стратегію виведення: далі будемо розглядати лише лівосторонню стратегію виведення ω в G. Тоді:

- $S \Rightarrow^\star \omega_1 A \omega_2$ (A — перший зліва направо нетермінал);
- ω_1 — термінальна частина слова ω, яку вже виведено (проаналізована частина слова);
- результат ω_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 ω_3.

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

9.1 $LL(k)$-граматики

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

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

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

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

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

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

достатньо k перших символів (за умови, що вони існують) решти непроаналізованого слова щоб визначити, що з $A\omega_2$ існує не більше однієї альтернативи виведення слова, що починається з ω 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** справедливість останнього твердження. Розглянемо граматику G з наступною схемою P: $S \mapsto Sa \mid b$.

Мова, яку породжує наведена вище граматика $L(G) = \{ba^i, i = 0, 1, \ldots\}$. 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

$$\mathrm{First}_k \left(Saa^i\right) \cap \mathrm{First}_k \left(ba^i\right) = \varnothing.$$

Втім, для $i \geq k$ маємо:

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

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 граматиці G існує вивід виду $A \Rightarrow^{\star} A\omega$), 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 називаються *розподіленою*, якщо вона задовольняє наступним умовам:

- у схемі P граматики відсутні ε-правила (правила вигляду $A \mapsto \varepsilon$);

- для нетермінала A праві частини A-правила починаються різними терміналами.

9.1.1 First$_k$

Зауважимо, що $\mathrm{First}_k(\omega_1\omega_2) = \mathrm{First}_k(\omega_1) \oplus_k \mathrm{First}_k(\omega_2)$, 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 $\omega = \alpha_1\alpha_2 \ldots \alpha_p$, де $\alpha_i \in (N \cup \Sigma)$, то

$$\mathrm{First}_k(\omega) = \mathrm{First}_k(\alpha_1) \oplus_k \mathrm{First}_k(\alpha_2) \oplus_k \ldots \oplus_k \mathrm{First}_k(\alpha_p)$$

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

9.1.2 Алгоритм пошуку First$_k$

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

Алгоритм [пошуку First$_k(A_i)$, $A_i \in N$]: 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. $F_0(A_i) = \big\{\omega \mid \omega \in \Sigma^{\star k} : A_i \mapsto \omega x, |\omega| = k\big\} \cup \big\{\omega \mid \omega \in \Sigma^{\star k} : A_i \mapsto \omega, |\omega| < k\big\}$.
3. $$\begin{aligned} &F_n(A) = F_{n-1}(A_i) \cup \\ &\cup \big\{\omega \mid \omega \in \Sigma^{\star k} : \omega \in F_{n-1}(\alpha_1) \oplus_k \ldots \oplus F_{n-1}(\alpha_p), A_i \mapsto \alpha_1 \ldots \alpha_p\big\}. \end{aligned}$$
4. $F_m(A_i) = F_{m+1}(A_i) = \ldots$ для всіх $A_i \in N$.

Очевидно, що:

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

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

Приклад: знайти множину $\mathrm{First}_k(A_i)$ 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	$\varnothing$	$\{\varepsilon\}$	$\varnothing$	$\{\varepsilon\}$	$\{a\}$
F_1	$\varnothing$	$\{\varepsilon\}$	$\{a\}$	$\{\varepsilon, \times a\}$	$\{a\}$
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 $\mathrm{First}_k(\alpha)$ сформулюємо необхідні й достатні умови, за яких КС-граматика буде $LL(k)$-граматикою: для довільного виводу в граматиці G вигляду $S \Rightarrow^\star \omega_1 A \omega_2$ та правила $A \mapsto \alpha \mid \beta$:

$$\mathrm{First}_k(\alpha\omega_2) \cap \mathrm{First}_k(\beta\omega_2) = \varnothing.$$

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

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

Оскільки $L \subseteq \Sigma^{\star k}$, 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:

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

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

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

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 ω_1, розглядаючи тільки наступні k символів які потрібно отримати після A.

Операції First_k та Follow_k можна узагальнити для словарної множини L, тоді:

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

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 граматики сильні

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

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

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

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

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

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

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

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

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

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

9.2.2 Алгоритм пошуку Follow_k

Алгоритм [обчислення $\mathrm{Follow}_k(A_i)$, $A_i \in N$]: 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 ε. У інших випадках $\sigma_0(S, A_i)$ — невизначено, $A_i \in (N \setminus \{S\})$.
2. $\sigma_1(S, A_i) = \sigma_0(S, A_i) \cup \{\omega \mid S \mapsto \omega_1 A_i \omega_2, \omega \in \mathrm{First}_k(\omega_2)\}$. В інших випадках $\sigma_1(S, A_i)$ — невизначено.
3. $\sigma_n(S, A_i) = \sigma_{n-1}(S, A_i) \cup \{\omega \mid A_j \mapsto \omega_1 A_i \omega_2, \omega \in \mathrm{First}_k(\omega_2 \cdot \sigma_{n-1}(S, A_j))\}$. В інших випадках $\sigma_n(S, A_i)$ — невизначено.

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

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

Очевидно, що:

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

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

9.2.3 ε-нетермінали

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

Алгоритм [пошуку ε-нетерміналів]:

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

Тоді множина S_m — множина ε-нетерміналів.

Приклад. Для граматики G з схемою правил P знайдемо множину ε-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 ε-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 $S_0, S_1, \ldots$:

1. $S_0 = \{A_i \mid A_i \mapsto \omega_1 A_i \omega_2, \omega_1 \Rightarrow^\star \varepsilon\}$, починаємо з нетерміналу A_i.

2. $S_1 = S_0 \cup \{A_i \mid A_i \mapsto \omega_1 A_j \omega_2, \omega_1 \Rightarrow^\star \varepsilon, A_j \in S_0\}$.

3. $S_n = S_{n-1} \cup \{A_i \mid A_i \mapsto \omega_1 A_j \omega_2, \omega_1 \Rightarrow^\star \varepsilon, A_j \in S_{n-1}\}$.

4. $S_m = S_{m+1} = \ldots$.

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 нетермінала окремо, наприклад, для нетермінала 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 $LL(k)$-граматик уникнути лівої рекурсії. Розглянемо граматику зі схемою правил $S \mapsto Sa \mid b$, яка має ліворекурсивний нетермінал S. Замінимо 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 нетермінала знайдемо множину $\mathrm{Follow}_1(A)$ $(k = 1)$:

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

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

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

	S	A	B	C	D
δ_0	$\{\varepsilon\}$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
δ_1	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{+, \varepsilon\}$	$\varnothing$	$\varnothing$
δ_2	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{+, \varepsilon\}$	$\{+, \varepsilon\}$	$\varnothing$
δ_3	$\{\varepsilon\}$	$\{\varepsilon\}$	$\{+, \varepsilon\}$	$\{+, \varepsilon\}$	$\{\times, +, \varepsilon\}$
δ_4	$\{\varepsilon,)\}$	$\{\varepsilon\}$	$\{+, \varepsilon\}$	$\{+, \varepsilon\}$	$\{\times, +, \varepsilon\}$
δ_5	$\{\varepsilon,)\}$	$\{\varepsilon,)\}$	$\{+, \varepsilon\}$	$\{+, \varepsilon\}$	$\{\times, +, \varepsilon\}$
δ_6	$\{\varepsilon,)\}$	$\{\varepsilon,)\}$	$\{+, \varepsilon,)\}$	$\{+, \varepsilon,)\}$	$\{\times, +, \varepsilon,)\}$
δ_7	$\{\varepsilon,)\}$	$\{\varepsilon,)\}$	$\{+, \varepsilon,)\}$	$\{+, \varepsilon,)\}$	$\{\times, +, \varepsilon,)\}$

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

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 $\mathrm{First}_k(\alpha)$?
6. Опишіть алгоритм пошуку 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 $\mathrm{Follow}_k(\alpha)$?
10. Опишіть алгоритм пошуку Follow_k і доведіть його збіжність.
11. Який нетермінал $A_i \in N$ називається ε-нетерміналом?

12. Опишіть алгоритм перевірки нетерміналу $A_i \in N$ на ε-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.