



НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»
Кафедра «Комп'ютерної інженерії та програмування»

ФОРМАЛЬНІ МОВИ, ГРАМАТИКИ І АВТОМАТИ

Лекція 4

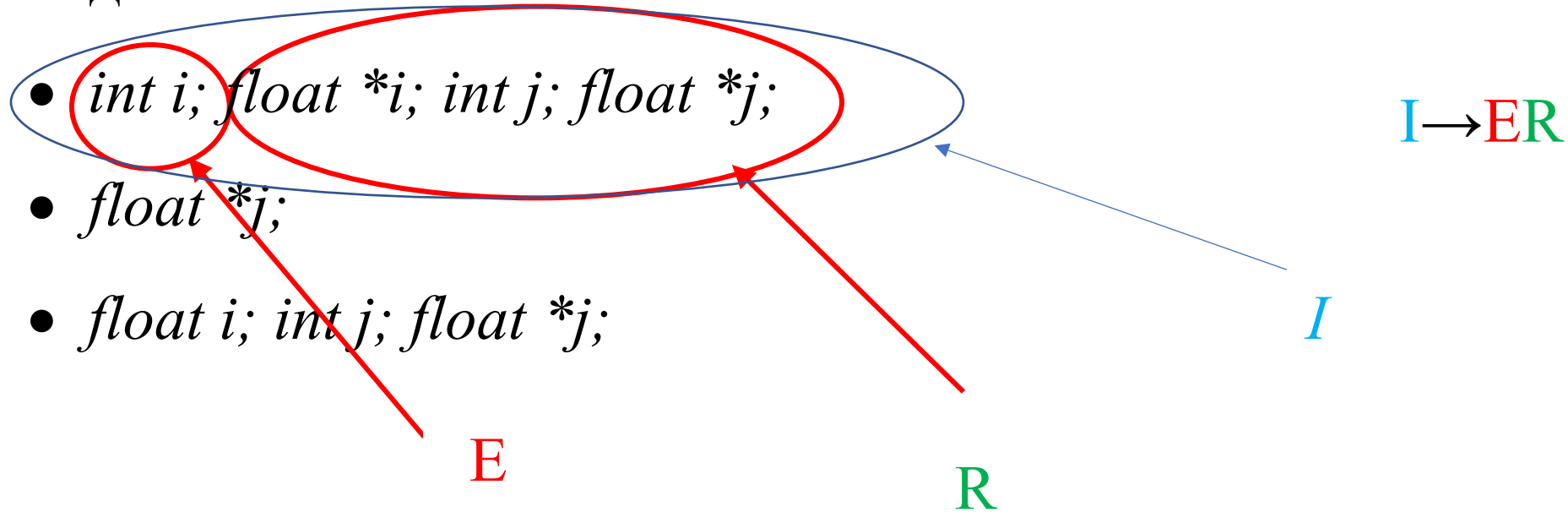
СПАДНІ РОЗПІЗНАВАЧІ



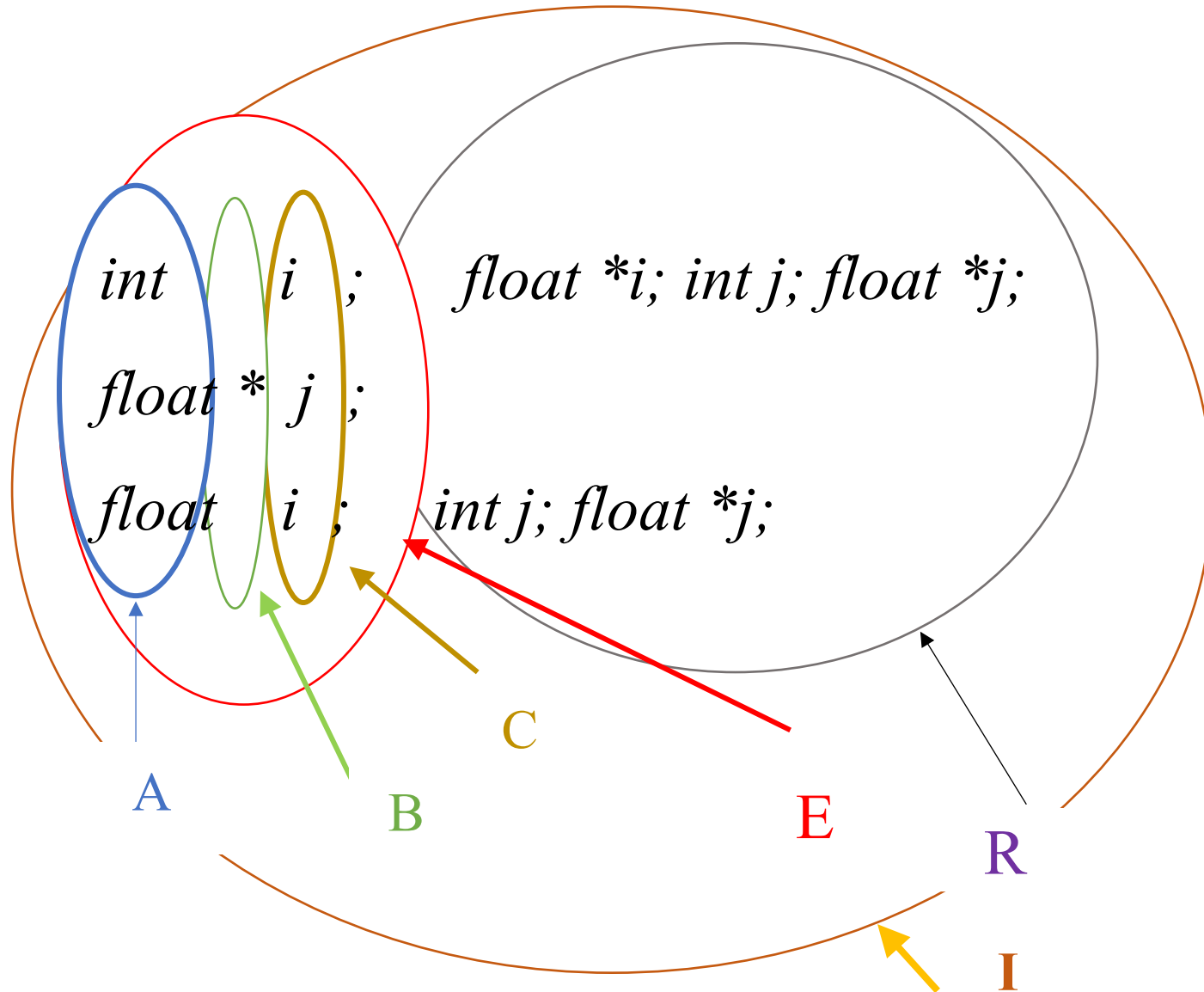
Гавриленко Світлана Юріївна
+380664088551 (Viber)
+380632864663 (Telegram)
Gavrilenko08@gmail.com
306BK

Повторення матеріалу. Частина 1

Побудувати правила грамматики для нижченаведеного коду програми. Перевірити граматику на наявність непродуктивних та недосяжних символів.



Повторення матеріалу. Частина 2



$I \rightarrow ER$
 $R \rightarrow ER \mid \$$
 $E \rightarrow ABC;$
 $A \rightarrow \text{int} \mid \text{float}$
 $B \rightarrow * \mid \$$
 $C \rightarrow i \mid j$

Повторення матеріалу. Частина 3

$I \rightarrow ER$
 $R \rightarrow ER \mid \$$
 $E \rightarrow ABC;$
 $A \rightarrow \text{int} \mid \text{float}$
 $B \rightarrow * \mid \$$
 $C \rightarrow i \mid j$

Непродуктивні символи:

- 1) R, A, B, C
- 2) R, A, B, C, E
- 3) R, A, B, C, E, I

Непродуктивних символів
немає

Недосяжні символи:

Пошук недосяжних символів:

- 1) I
- 2) I, E, R
- 3) I, E, R, A, B, C

Недосяжних символів немає.

Процедура виявлення непродуктивних символів:

1. Скласти список нетерміналів, для яких знайдеться хоча б одне правило, права частина якого не містить нетермінали.
2. Якщо знайдене таке правило, і всі нетермінали, які стоять у його правій частині вже занесені в список, то додати в список нетермінал, що стоїть в його лівій частині.
3. Якщо на кроці 2 список більше не поповнюється, то отримано список усіх продуктивних нетерміналів граматики, а всі нетермінали, які не потрапили в нього, є непродуктивними.

Процедура виявлення ненodosяжних символів:

1. Створити одноелементний список, що складається з початкового символу граматики I .
2. Якщо знайдене правило, ліва частина якого вже є в списку, то включити до списку всі символи, які містяться в його правій частині.
3. Якщо на кроці 2 нові нетермінали в список більше не додаються, то отримано список усіх досяжних нетерміналів, а нетермінали, що не потрапили в список, є недосяжними.

Типи розпізнавачів (аналізаторів)

Розпізнавачі можна поділити на дві категорії: спадні (нисхідні) і висхідні. Кожна категорія характеризується порядком, розташовування правила в дереві виведення. Спадні розпізнавачі обробляють правила зверху вниз, верхні правила раніше нижніх, у той час як висхідні аналізатори використовують нижні правила раніше тих, що розташовано вище. В основі роботи спадних розпізнавачів лежить операція заміни лівого нетермінального символу правила граматики на праву частину правила, яка йому відповідає.

У загальному випадку спадний детермінований розпізнавач можна побудувати для граматики, яка відноситься до класу LL(K) граматик.

LL(k) грамматики

Назва *LL* походить від слів Left Leftmost. Left означає що аналізатор переглядає вхідний ланцюжок зліва-направо, Leftmost - виявляє появу правила по одному чи групі символів, що утворять лівий край ланцюжка. На практиці найбільш поширеним є клас *LL(1)* граматик, для яких детермінований розпізнавач розпізнає один вхідний символ поточної позиції. До класу *LL(1)* граматик також відносяться **розділені та слабко-розділені грамматики.**

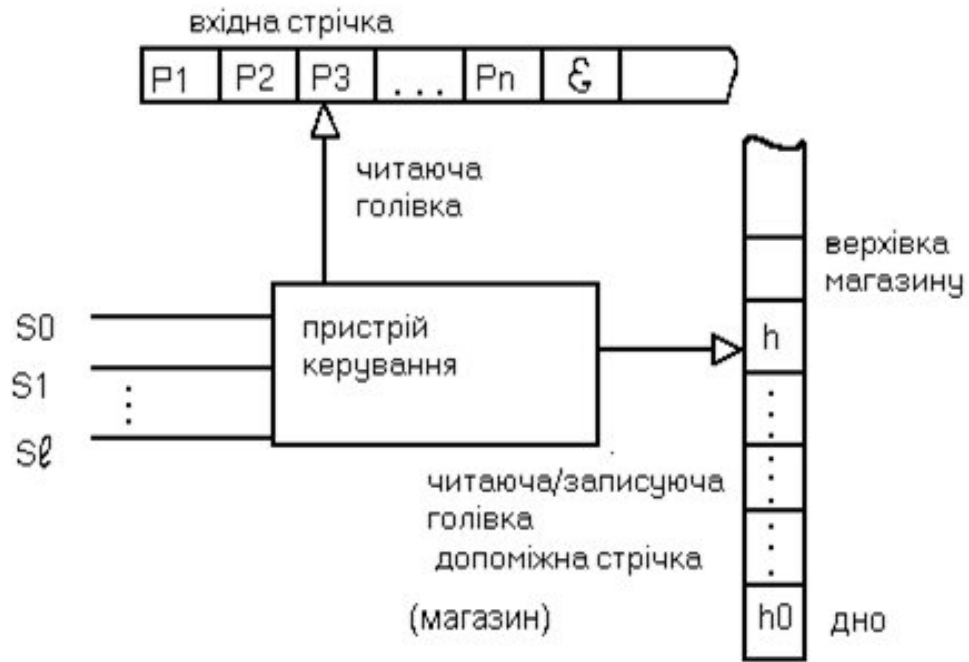
Розділені граматики

Контекстно-вільна граматика, яка **не містить правил, що анулюють**, називається **розділеною**, або **простою**, якщо виконуються наступні дві умови:

1. Права частина кожного правила починається **терміналом**;
2. Якщо два правила мають **однакові ліві частини**, то праві частини цих правил повинні починатися **різними термінальними символами**.

Важливою властивістю розділених граматик є те, що для кожної з них можна побудувати детермінований спадний розпізнавач.

Модель магазинного автомата (детальніше див. лекцію 3)



Модель магазинного **автомата** складається з вхідної стрічки, пристрою керування і допоміжної стрічки або стеку. **Вхідна стрічка** розділяється на клітини (позиції), у кожній з яких може бути записаний символ вхідного алфавіту. При цьому передбачається, що у вільних клітинах вхідної стрічки розташовані порожні символи ϵ . Допоміжна стрічка також розділена на клітини, у яких можуть розташовуватися символи магазинного алфавіту. Початок допоміжної стрічки називається **дном** магазину. Зв'язок пристрою керування зі стрічками здійснюється двома **голівками**, що можуть переміщатися уздовж стрічок.

Побудова детермінованого спадного розпізнавача для простої граматики

Побудова розпізнавача передбачає побудову для кожного правила граматики команди розпізнавача (функції переходів автомата).

1. Відповідно до загального способу побудови розпізнавача **кожному** правилу розділеної граматики, що має вигляд:

$$A \rightarrow a \alpha,$$

де α — ланцюжок символів повного словника і a належить термінальному словнику, потрібно поставити у відповідність команду:

$$f(s, a, A) = (s, \alpha'),$$

яка визначає такт роботи розпізнавача **зі зсувом вхідної голівки**, де α' є дзеркальне відображення ланцюжка α .

2. Для всіх термінальних символів, наприклад b , які розташовані в середині або в кінці правил, необхідно побудувати команди вигляду:

$$f(s, b, b) = (s, \$).$$

3. Для переходу в заключний стан використовується команда:

$$f^*(s, \$, h_0) = (s, \$).$$

Приклад 1

Задано грамматику $\Gamma_{5.1}: R = \{I \rightarrow abB,$

$I \rightarrow bBbI,$

$B \rightarrow a,$

$B \rightarrow bB\},$

$P = \{a, b\}, H = \{a, b, I, B, h_0\}, S = \{s\}, F = \{s\}, \}$.

Кожне правило починається з термінального символу, праві частини правил з однаковим лівим нетерміналом починаються **різними термінальними символами** - це проста граMATика.

Застосовуючи наведені вище правила, побудувати розпізнавач для розділеної граматики $\Gamma_{5.1}$.

Приклад 1. Продовження

$$R = \{1.I \rightarrow abB, 2.I \rightarrow bBbI, 3.B \rightarrow a, 4.B \rightarrow bB\},$$

Побудуємо команди розпізнавача:

$$1. f(s, a, I) = (s, Bb)$$

$$2. f(s, b, I) = (s, IbB)$$

$$3. f(s, a, B) = (s, \$)$$

$$4. f(s, b, B) = (s, B)$$

$$5. f(s, b, b) = (s, \$)$$

$$6. f^*(s, \$, h_0) = (s, \$)$$

Роботу побудованого автомата продемонструємо на прикладі аналізу ланцюжка *bbababa*:

$$(s, bbababa, h_0I) \vdash 2 (s, bababa, h_0IbB) \vdash 4 (s, ababa, h_0IbB) \vdash 3 (s, baba, h_0Ib) \vdash 5 (s, aba, h_0I) \vdash 1 (s, ba, h_0Bb) \vdash 5 (s, a, h_0B) \vdash 3 (s, \$, h_0) \vdash 6 (s, \$, \$)$$

Приклад 2

Задано правила грамматики: $\Gamma_{5.2} \quad R = \{1. I \rightarrow aR, 2. R \rightarrow +AR \mid *c, 3. A \rightarrow a \mid b\}$.

Побудуємо команди розпізнавача:

1. $f(s, a, I) = (s, R)$
2. $f(s, +, R) = (s, RA)$
3. $f(s, *, R) = (s, c)$
4. $f(s, a, A) = (s, \$)$
5. $f(s, b, A) = (s, \$)$
6. $f(s, c, c) = (s, \$)$
7. $f^*(s, \$, h_0) = (s, \$)$.

Роботу побудованого автомата продемонструємо на прикладі аналізу ланцюжка **$a+b+a+c$** :

$(s, a+b+a+c, h_0I) \vdash 1 (s, +b+a+c, h_0R) \vdash 2 (s, b+a+c, h_0RA) \vdash 5 (s, +a+c, h_0R) \vdash 2 (s, a+c, h_0RA) \vdash 4 (s, +c, h_0R) \vdash 2 (s, a*c, h_0RA) \vdash 4 (s, *c, h_0R) \vdash 3 (s, c, h_0c) \vdash 6 (s, \$, h_0) \vdash 7 (s, \$, \$)$

Слабко-розділеними граматиками

Граматика називається **слабко-розділеною** граматикою за умов:

- права частина кожного починається з термінального символу або є **порожнім ланцюжком**;
- якщо два правила мають однакові ліві частини, то праві частини правил повинні починатися різними термінальними символами;
- елементи множини **ВИБІР**, побудовані для правил з однаковим лівим нетермінальним символом, не повинні містити однакових символів (перетинатися).

***LL(1)*–граматики**

Граматика називається *LL(1)*- граматикою за умов:

- права частина кожного починається з **термінального, нетермінального символу** або є порожнім ланцюжком;
- якщо два правила мають однакові ліві частини, то праві частини правил повинні починатися різними символами;
- елементи множини **ВИБІР**, побудовані для правил з однаковим лівим нетермінальним символом, не повинна містити однакових символів (перетинатися).

Множина **ВИБІР**, функції **ПЕРШ**(μ) та **СЛІД** (μ)

Множина **ВИБІР** будується для кожного правила грамматики. Для визначення множини **ВИБІР** використовуються функції **ПЕРШ**(μ) і **СЛІД** (μ). Аргументом функції **ПЕРШ**(μ) може бути будь-який ланцюжок повного словника μ , а значенням функції **ПЕРШ**(μ) є множина термінальних символів, що можуть стояти на першому місці в ланцюжках, виведених з ланцюжка μ .

Побудова функції ПЕРШ(μ)

Значення функції ПЕРШ(μ) можна визначити користуючись такими правилами:

1) якщо ланцюжок $\mu = b\alpha$ починається термінальним символом b і має вигляд $A \rightarrow b\alpha$, то функція:

$$\text{ПЕРШ}(A \rightarrow b\alpha) = \{b\};$$

2) якщо ланцюжок μ є порожнім ланцюжком, $\mu \rightarrow \$$, то функція:

$$\text{ПЕРШ}(A \rightarrow \$) = \{\$ \};$$

3) якщо ланцюжок μ починається нетермінальним символом B і має вигляд $\mu = B\gamma$, а в схемі граматики є n правил, у будь-якій частині яких знаходиться символ B :

$$B \rightarrow \alpha_1 \mid \alpha_2 \mid \dots \mid \alpha_n,$$

і, якщо не існує виведення $B \rightarrow \$$, то функція $\text{ПЕРШ}(A \rightarrow B\gamma)$ є об'єднанням множин:

$$\text{ПЕРШ}(A \rightarrow B\gamma) = \text{ПЕРШ}(\alpha_1) \cup \text{ПЕРШ}(\alpha_2) \cup \dots \cup \text{ПЕРШ}(\alpha_n).$$

Побудова функції ПЕРШ(μ). Продовження

4) Якщо ланцюжок μ починається нетермінальним символом і має вигляд, $\mu = B \text{ } \color{red}{\curvearrowright}$ в схему граматики входять n правил вигляду

$$B \rightarrow \color{green}{\alpha_1} \mid \color{green}{\alpha_2} \mid \dots \mid \color{green}{\alpha_n},$$

і B є нетерміналом, що анулює, тобто існує $B \rightarrow \color{green}{\$}$, то функція

$$\text{ПЕРШ}(\mu \rightarrow B \text{ } \color{red}{\curvearrowright}) = \text{ПЕРШ}(\color{red}{\curvearrowright}) \nearrow \text{ПЕРШ}(\color{green}{\alpha_1}) \nearrow \text{ПЕРШ}(\color{green}{\alpha_2}) \nearrow \dots \nearrow \text{ПЕРВ}(\color{green}{\alpha_n}).$$

Приклад 3

Побудувати функцію ПЕРШ(μ) для грамматики $\Gamma_{5.3}$, заданої наступними правилами грамматики:

$$1. I \rightarrow \{SR\}. \quad 2. R \rightarrow SR | \$ \quad 3. S \rightarrow AB; | BA; \quad 4. A \rightarrow a | b \quad 5. B \rightarrow ++ | --.$$

Надалі, для спрощення, у якості аргументу функції ПЕРШ(μ) будемо вказувати номер правила.

$$\text{ПЕРШ}(I \rightarrow \{SR\}) = \text{ПЕРШ}(\{SR\}) = \text{ПЕРШ}(1) = \{\{\}$$

$$\text{ПЕРШ}(2.1) = \text{ПЕРШ}(S) = \text{ПЕРШ}(A) \nearrow \text{ПЕРШ}(B) = \{+, -, a, b\}$$

$$\text{ПЕРШ}(2.2) = \{\$ \}$$

$$\text{ПЕРШ}(3.1) = \text{ПЕРШ}(A) = \{a, b\}$$

$$\text{ПЕРШ}(3.2) = \text{ПЕРШ}(B) = \{+, -\}$$

$$\text{ПЕРШ}(4.1) = \{a\}$$

$$\text{ПЕРШ}(4.2) = \{b\}$$

$$\text{ПЕРШ}(5.1) = \{+\}$$

$$\text{ПЕРШ}(5.2) = \{-\}$$

Приклад 4

Побудувати функцію $\text{ПЕРШ}(\mu)$ для граматики $\Gamma_{5.4}$, заданої наступними правилами граматики:

$R = \{ 1. A \rightarrow BCc, 2. A \rightarrow gDB, 3. B \rightarrow \$, 4. B \rightarrow bCDE, 5. C \rightarrow DaB, 6. C \rightarrow ca, 7. D \rightarrow \$, 8. D \rightarrow dD, 9. E \rightarrow gAf, 10. E \rightarrow c \}$.

Спочатку знайдемо значення функції $\text{ПЕРШ}(\mu)$ для правих частин правил, що починаються термінальними символами:

$\text{ПЕРШ}(2) = \{g\}$, $\text{ПЕРШ}(4) = \{b\}$, $\text{ПЕРШ}(6) = \{c\}$, $\text{ПЕРШ}(8) = \{d\}$,
 $\text{ПЕРШ}(9) = \{g\}$, $\text{ПЕРШ}(10) = \{c\}$.

Потім знайдемо функцію для 3-го та 7-го правила, що є анулюючим:

$\text{ПЕРШ}(3) = \{\$\}$. $\text{ПЕРШ}(7) = \{\$\}$.

Приклад 4. Продовження

$R = \{ 1. A \rightarrow BSc, 2. A \rightarrow gDB, 3. B \rightarrow \$, 4. B \rightarrow bCDE, 5. C \rightarrow DaB, 6. C \rightarrow ca, 7. D \rightarrow \$, 8. D \rightarrow dD, 9. E \rightarrow gAf, 10. E \rightarrow c \}$.

Знайдемо значення функції ПЕРШ(μ) для 5-го та 1-го правила:

ПЕРШ(5) = ПЕРШ($C \rightarrow DaB$) = ПЕРШ(D) = ПЕРШ($D \rightarrow dD$) \cup ПЕРШ(aB) = $\{d, a\}$ по 7 правилу, якщо $D \rightarrow \$$

ПЕРШ(1) = ПЕРШ($A \rightarrow BSc$) = ПЕРШ(B) \cup ПЕРШ(C) = $\{b\} \cup$

ПЕРШ($C \rightarrow DaB$) \cup ПЕРШ($C \rightarrow ca$) = $\{b\} \cup$ ПЕРШ(D) \cup ПЕРШ(aB) \cup $\{c\}$ = $\{b, d, a, c\}$.

Побудова функції $СЛІД(\mu)$

Аргументом функції **СЛІД** є нетермінальний символ, наприклад **V**, а значення функції **СЛІД(V)** є множиною терміналів, що можуть слідувати безпосередньо за нетерміналом **V** у ланцюжках, виведених з початкового символу граматики.

Побудова функції СЛІД(μ)

Визначення функції виконується за наступними правилами:

1) якщо в схемі граматики є правила вигляду:

$$X_1 \rightarrow \mu_1 B \alpha_1, X_2 \rightarrow \mu_2 B \alpha_2, \dots, X_n \rightarrow \mu_n B \alpha_n,$$

і не існує правил $\alpha_i \rightarrow \$$, то

$$\text{СЛІД}(B) = \text{ПЕРШ}(\alpha_1) \hat{\Rightarrow} \text{ПЕРШ}(\rightarrow \alpha_2) \hat{\Rightarrow} \dots \hat{\Rightarrow} \text{ПЕРШ}(\alpha_n).$$

2) якщо ж серед приведених вище правил існує хоча б один ланцюжок

$\alpha_i \rightarrow \$$, (наприклад, нехай $\alpha_1 \rightarrow \$$, тобто є правило $X_1 \rightarrow \mu_1 B$), то функція знаходиться так:

$$\text{СЛІД}(B) = \text{СЛІД}(\mu_1) \hat{\Rightarrow} \text{ПЕРШ}(\alpha_1) \hat{\Rightarrow} \dots \hat{\Rightarrow} \text{ПЕРШ}(\alpha_n)$$

Побудова функції СЛІД(μ) для граматики $\Gamma_{5.3}$

Побудувати функцію СЛІД(μ) для граматики $\Gamma_{5.3}$, заданої наступними правилами граматики:

1. $I \rightarrow \{SR\}$. 2. $R \rightarrow SR | \$$ 3. $S \rightarrow AB; | BA;$ 4. $A \rightarrow a | b$ 5. $B \rightarrow ++ | --$.

ПЕРШ(1) = $\{\}$, ПЕРШ(2.1) = ПЕРШ(S) = ПЕРШ(A) \cup ПЕРШ(B) = $\{+, -, a, b\}$

ПЕРШ(2.2) = $\{\$$, ПЕРШ(3.1) = ПЕРШ(A) = $\{a, b\}$, ПЕРШ(3.2) = ПЕРШ(B) = $\{+, -\}$

ПЕРШ(4.1) = $\{a\}$, ПЕРШ(4.2) = $\{b\}$, ПЕРШ(5.1) = $\{+\}$, ПЕРШ(5.2) = $\{-\}$.

СЛІД(S) = ПЕРШ(R) \cup ПЕРШ($\{$) = ПЕРШ(S) \cup $\{\}$ = ПЕРШ(A) \cup ПЕРШ(B) \cup $\{\}$ = $\{+, -, a, b, \}$

СЛІД(R) = $\{\}$ \cup СЛІД(R) = $\{\}$

СЛІД(A) = ПЕРШ(B) \cup $\{;$ = $\{+, -, ;\}$

СЛІД(B) = ПЕРШ(A) \cup $\{;$ = $\{;, a, b\}$.

Побудувати функції ПЕРШ(μ) та СЛІД(μ) для граматики $\Gamma_{5.5}$

1. $I \rightarrow \text{var } S : T;$
2. $S \rightarrow ABR$
3. $A \rightarrow \wedge \mid \$$
4. $B \rightarrow i \mid j$
5. $R \rightarrow , ABR \mid \$$
6. $T \rightarrow \text{char} \mid \text{integer}$

Приклад $\text{:var } i, j, i : \text{integer};$
 $\text{var } j, i : \text{char};$

ПЕРШ (1) = {var}

ПЕРШ (2) = ПЕРШ (A) \cup ПЕРШ (B) = { \wedge , i, j}

A \rightarrow \$

ПЕРШ (3.1) = { \wedge }

ПЕРШ (3.2) = {\$}

ПЕРШ (4.1) = {i}

ПЕРШ (4.2) = {j}

ПЕРШ (5.1) = {,}

ПЕРШ (5.2) = {\$}

ПЕРШ (6.1) = {char}

ПЕРШ (6.2) = {integer}

СЛІД (S) = {::}

СЛІД (T) = {;}

СЛІД (A) = ПЕРШ(B) = {i, j}

СЛІД (B) = ПЕРШ(R) $\not\rightarrow$ СЛІД(S) = {',', '::'}
 $R \rightarrow \$$

СЛІД (R) = СЛІД(S) = {::}

Побудова функції СЛІД(μ) для граматики $\Gamma_{5.4}$

Побудувати функцію СЛІД(μ) для граматики $\Gamma_{5.4} : R = \{ 1. A \rightarrow B C c, 2. A \rightarrow g D B, 3. B \rightarrow \$, 4. B \rightarrow b C D E, 5. C \rightarrow D a B, 6. C \rightarrow c a, 7. D \rightarrow \$, 8. D \rightarrow d D, 9. E \rightarrow g A f, 10. E \rightarrow c \}$.

СЛІД(A) = ПЕРШ(f) = $\{f\}$.

СЛІД(C) = ПЕРШ(c) $\hat{=}$ ПЕРШ(D) $\hat{=}$ ПЕРШ(E) $\hat{=}$ $\{c, d, g\}$.

СЛІД(B) = ПЕРШ(C) $\hat{=}$ СЛІД(A) $\hat{=}$ СЛІД(C) = $\{a, c, d\} \hat{=} \{f\} \hat{=} \{c, d, g\} = \{a, c, d, g, f\}$.

СЛІД(E) = СЛІД(B) = $\{a, d, c, g, f\}$.

СЛІД(D) = ПЕРШ(B) $\hat{=}$ СЛІД(A) $\hat{=}$ ПЕРШ(E) $\hat{=}$ ПЕРШ(a) = ПЕРШ(B) $\hat{=}$

СЛІД(A) $\hat{=}$ ПЕРВ(E) $\hat{=}$ $\{a\} = \{b\} \hat{=} \{f\} \hat{=} \{c, g\} \hat{=} \{a\} = \{a, b, c, g, f\}$.

ДЯКУЮ ЗА УВАГУ