



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

ФОРМАЛЬНІ МОВИ, ГРАМАТИКИ І АВТОМАТИ

Лекція 5

ПРИВЕДЕНІ ГРАМАТИКИ. МАГАЗИННІ АВТОМАТИ



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

Приведені граматики

КВ-граматика називається **приведеною**, якщо вона не містить **зайвих** або **марних** (непродуктивних і недосяжних) символів.

Визначення непродуктивних символів

Символ $X \notin V_A$ називається **непродуктивним**, якщо з нього не може бути виведений кінцевий термінальний ланцюжок.

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.

Приклад 1.1

Маємо граматику Γ_1 :

$$R = \{I \rightarrow aIa,$$

$$I \rightarrow bAd,$$

$$I \rightarrow c,$$

$$A \rightarrow cBd,$$

$$A \rightarrow aAd,$$

$$B \rightarrow dAf \},$$

$$1. I + I = I$$

знаходимо, що тут непродуктивними є символи A і B . Після виключення правил, що містять непродуктивні символи, одержуємо граматику:

$$R' = \{I \rightarrow a I a,$$

$$I \rightarrow c\}.$$

Приклад 1.2

Given the grammar G_1 :

$R = \{ R = \{ 1. I \rightarrow *B**, 2. B \rightarrow AR1, 3. R1 \rightarrow *AR1, 4. R1 \rightarrow \$, 5. A \rightarrow | R, 6. R \rightarrow | R, 7. R \rightarrow \$ \} \}$,

1. $R1, R$
2. $R1, R, A$
3. $R1, R, A, B$
4. $R1, R, A, B, I$

Непродуктивних символів немає.

Визначення недосяжних символів

Символ $X \notin V_A$ називається **недосяжним** у КВ-граматиці Γ , якщо X не з'являється в жодному виведеному ланцюжку.

Процедура виявлення ненедосяжних символів:

1. Створити одноелементний список, що складається з початкового символу граматички **I**.
2. Якщо знайдене правило, ліва частина якого вже є в списку, то включити до списку всі символи, які містяться в його правій частині. Перейти до кроку 3.
3. Крок 2 повторити. 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.

Приклад 2

Маємо грамматику Γ_2 :

$$R = \{ \textcolor{green}{I} \rightarrow a I b \textcolor{red}{A}, \quad 1. \textcolor{green}{I}$$
$$I \rightarrow c, \quad 2. I, \textcolor{red}{A}$$
$$A \rightarrow b I,$$
$$A \rightarrow a \},$$

Недосяжних символів немає.

Виключення ліворекурсивних правил

Правило вигляду $A \rightarrow \alpha A$, де $A \notin V_A$, $\alpha \notin (V_T \setminus V_A)^*$, називається **праворекурсивним**, а правило вигляду $A \rightarrow A\alpha$ – **ліворекурсивним**.

Для кожної КВ-граматики Γ , що містить ліворекурсивні правила, можна побудувати еквівалентну граматичку Γ' , що не містить ліворекурсивних правил.

Приклад 3

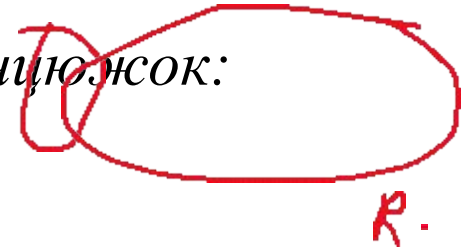
Маємо грамматику Γ_3 :

$R = \{1. E \rightarrow E + T \mid T, \Rightarrow \text{Генерує ланцюжок: } T + T + T + \dots + T$

2. $T \rightarrow T * F \mid F,$

$\Rightarrow \text{Генерує ланцюжок:}$

$F * F * F * \dots * F$



3. $F \rightarrow (E) \mid a\}.$

$T \rightarrow FR1$

Виключити ліворекурсивні правила.

$R1 \rightarrow *FR1 \mid \$$

Приклад 3 (продовження).

Перше правило $E \rightarrow E + T \mid T$ генерує ланцюжок: $T+T+\dots+T$, який може бути задано наступними правилами граматики: $E \rightarrow TR, R \rightarrow +TR \mid \$$.

Друге правило $T \rightarrow T * F \mid F$ генерує ланцюжок: $F*F*\dots*F$ перетворимо в правила

$$T \rightarrow FR_1, R_1 \rightarrow *FR_1 \mid \$.$$

В результаті отримуємо наступну граматичку Γ' , яка еквівалентна заданій Γ_3 :

$$E \rightarrow TR$$

$$R \rightarrow +TR \mid \$.$$

$$T \rightarrow FR_1,$$

$$R_1 \rightarrow *FR_1 \mid \$.$$

$$F \rightarrow (E) \mid a$$

Приклад 3 (продовження).

Якщо необхідно, щоб граматика не містила ліворекурсивних та анулюючих правил, то використовуємо такий алгоритм.

Правило $E \rightarrow E + T \mid T$ перетворимо в правила $E \rightarrow T \mid TE'$ і $E' \rightarrow +T \mid +TE$, а правила $T \rightarrow T * F \mid F$ перетворимо в правила $T \rightarrow F \mid FT'$ і $T' \rightarrow *F \mid *FT'$.

В результаті отримуємо наступну граматику Γ' , яка еквівалентна заданій Γ_3 :

$$\begin{aligned} R' = \{ & E \rightarrow T, & T \rightarrow FT', \\ & E \rightarrow TE', & T' \rightarrow *F, \\ & E' \rightarrow +T, & T' \rightarrow *FT', \\ & E' \rightarrow +TE', & F \rightarrow a, \\ & T \rightarrow F, & F \rightarrow (E) \} \end{aligned}$$

Виключення ланцюгових правил

Правило грамматики виду $A \rightarrow B$, де $A, B \in V_A$, називається **ланцюговим**.

Ідея доказу полягає в наступному. Якщо схема грамматики має вигляд $R = \{..., A \rightarrow B, ..., B \rightarrow C, ..., C \rightarrow aX\}$,

то така граматики еквівалентна граматиці зі схемою

$R' = \{..., A \rightarrow aX, ...\}$,

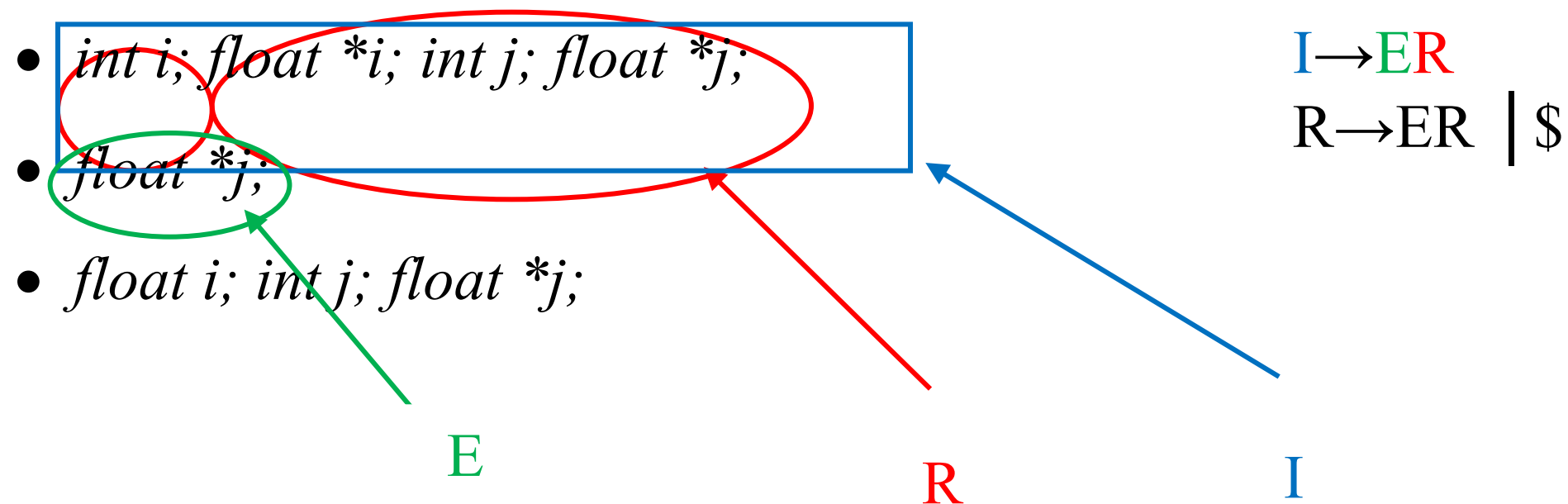
оскільки вивід у граматиці зі схемою R ланцюжка aX :

$A \Rightarrow B \Rightarrow C \Rightarrow ax$

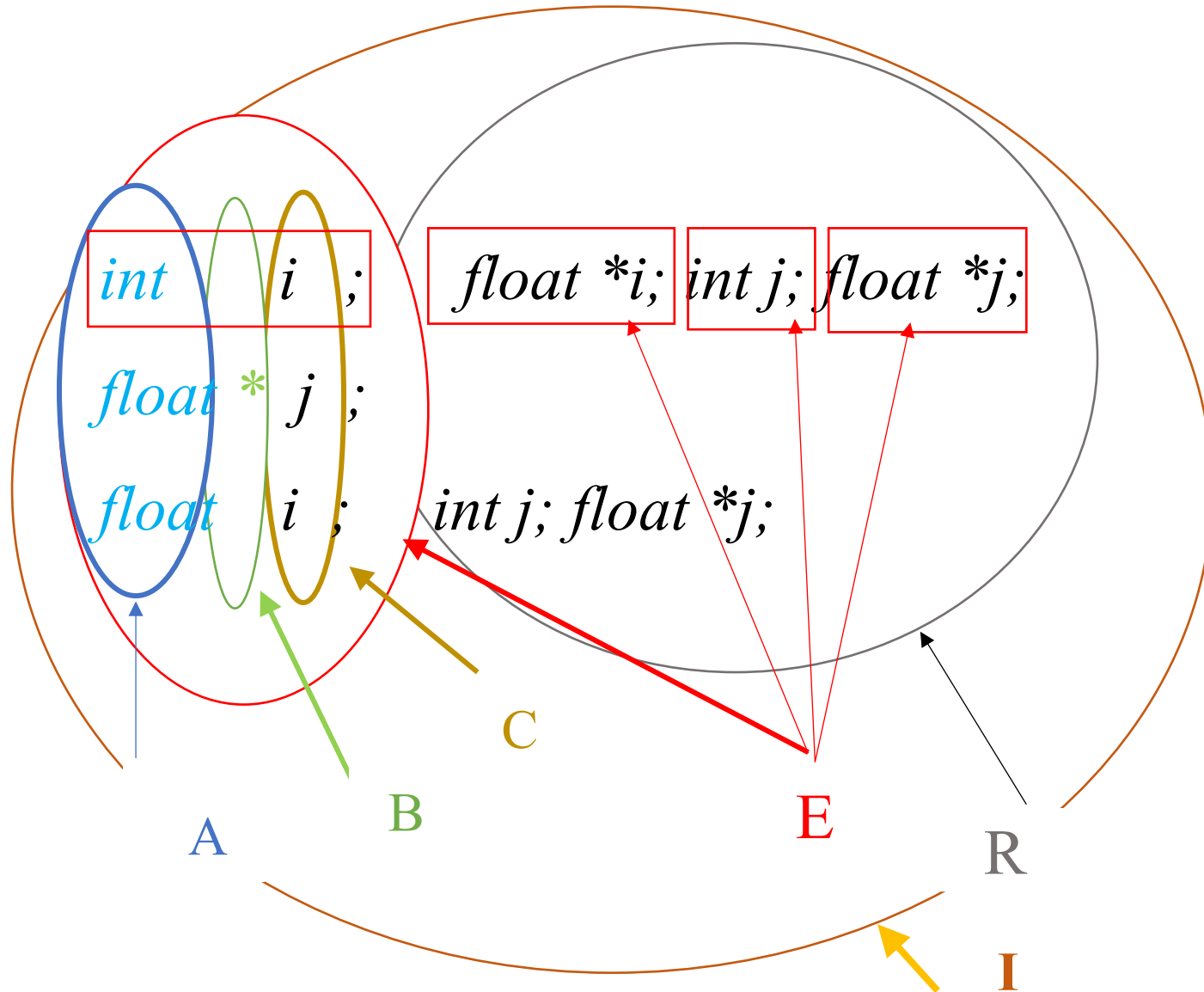
може бути отриманий у граматиці зі схемою R' за допомогою правила $A \rightarrow aX$.

Приклад 4

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



Приклад 4. Продовження


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

Перевірка граматики на наявність **непродуктивних** символів

Скласти список **нетерміналів**, для яких знайдеться хоча б одне правило, **права частина якого не містить нетермінали**.

I \rightarrow ER

R \rightarrow ER | \$

E \rightarrow ABC;

A \rightarrow int | float

B \rightarrow * | \$

C \rightarrow i | j

Перевірка грамматики на наявність непродуктивних символів

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

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.

Пошук непродуктивних символів:

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

Перевірка граматики на наявність непродуктивних символів

Якщо знайдене таке правило, і **всі нетермінали, які стоять його правій частині вже занесені в список**, то додати в список нетермінал, що стоїть в його лівій частині.

I → **E**R

R → ER | \$

E → ABC;

A → int | float

B → * | \$

C → i | j

Пошук непродуктивних символів:

1) R, A, B, C

2) R, A, B, C, **E**

3) R, A, B, C, E, **I**

Якщо список більше не поповнюється, то отримано список усіх продуктивних нетерміналів граматики, а всі нетермінали, які не потрапили в нього, є непродуктивними

Непродуктивних символів немає.

Перевірка грамматики на наявність недосяжних символів

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

Створити одноелементний список, що складається з початкового символу граматики I .

Пошук недосяжних символів:

1) I

Перевірка грамматики на наявність недосяжних символів

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

Якщо знайдене правило, ліва частина якого вже є в списку, то включити до списку всі символи, які містяться в його правій частині.

Пошук недосяжних символів:

1) I

2) I, E, R

Перевірка грамматики на наявність недосяжних символів

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

Якщо знайдене правило, ліва частина якого вже є в списку, то включити до списку всі символи, які містяться в його правій частині.

Пошук недосяжних символів:

1) I

2) I, E, R

3) I, E, R, A, B, C

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

Недосяжних символів немає.

ПРИКЛАД 5.

Перевірка граматики на наявність марних символів.

RULES:

- 1) $I \rightarrow AB(S);$
- 2) $A \rightarrow \text{void} | \text{int}$
- 3) $B \rightarrow b | d | f | g | h | k | l$
- 4) $S \rightarrow ER$
- 5) $R \rightarrow ;ER | \$$
- 6) $E \rightarrow TZ$
- 7) $T \rightarrow \text{int} | \text{char}$
- 8) $Z \rightarrow BX$
- 9) $X \rightarrow ,BX | \$$

Nonterminal Symbols: [A, B, S, I, E, R, T, Z, X]

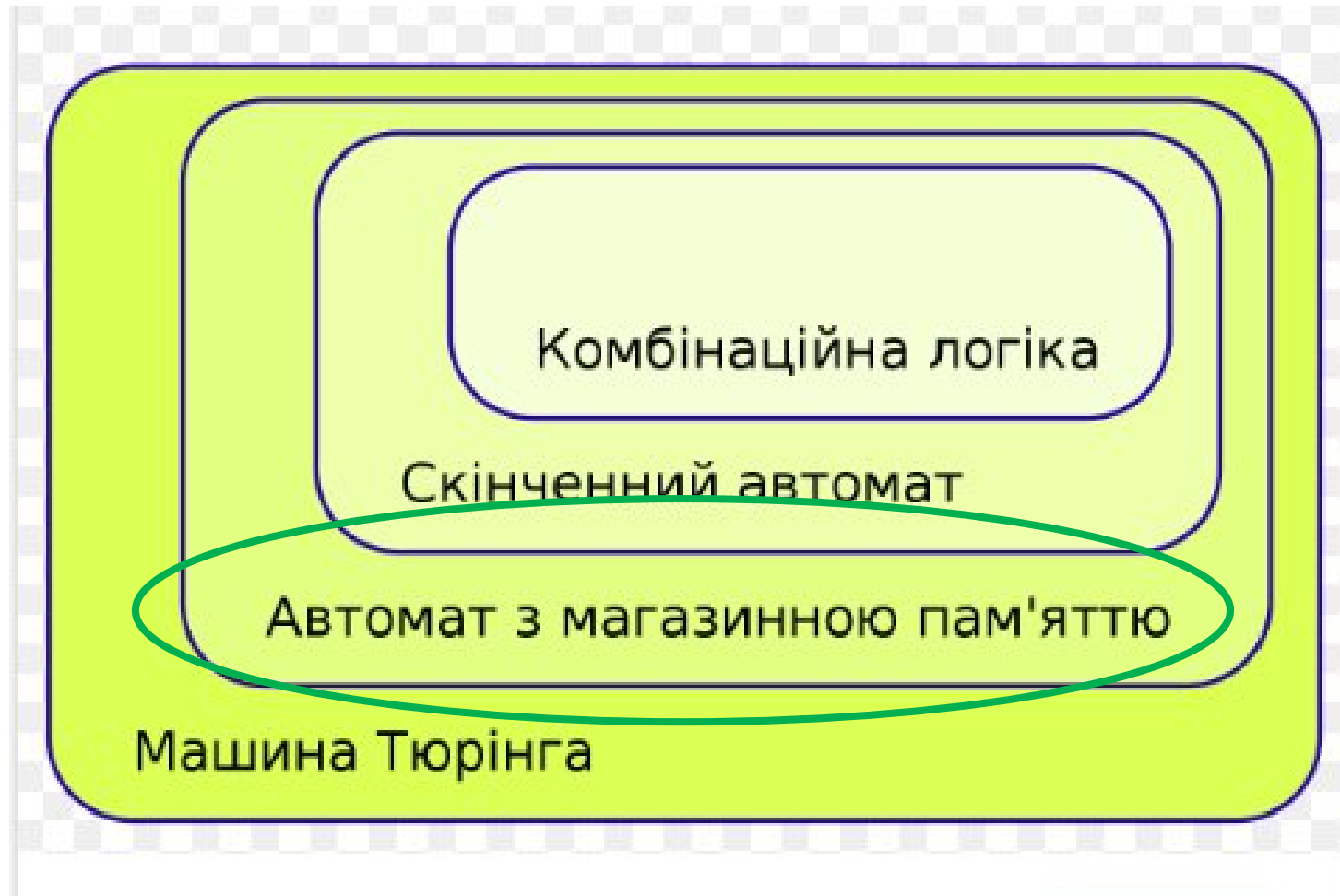
Reachable symbols: [I, A, B, S, E, R, T, Z, X]

Unreachable Symbols: []

Productive symbols: [A, B, E, R, T, X, I, S, Z]

Unproductive symbols: []

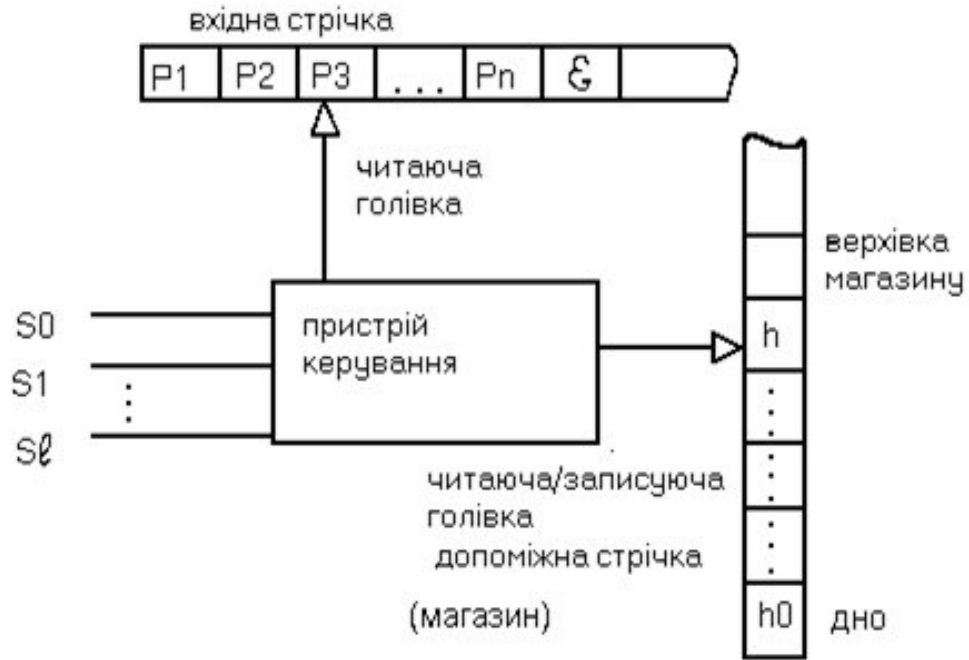
Класи автоматів



Магазинні автомати

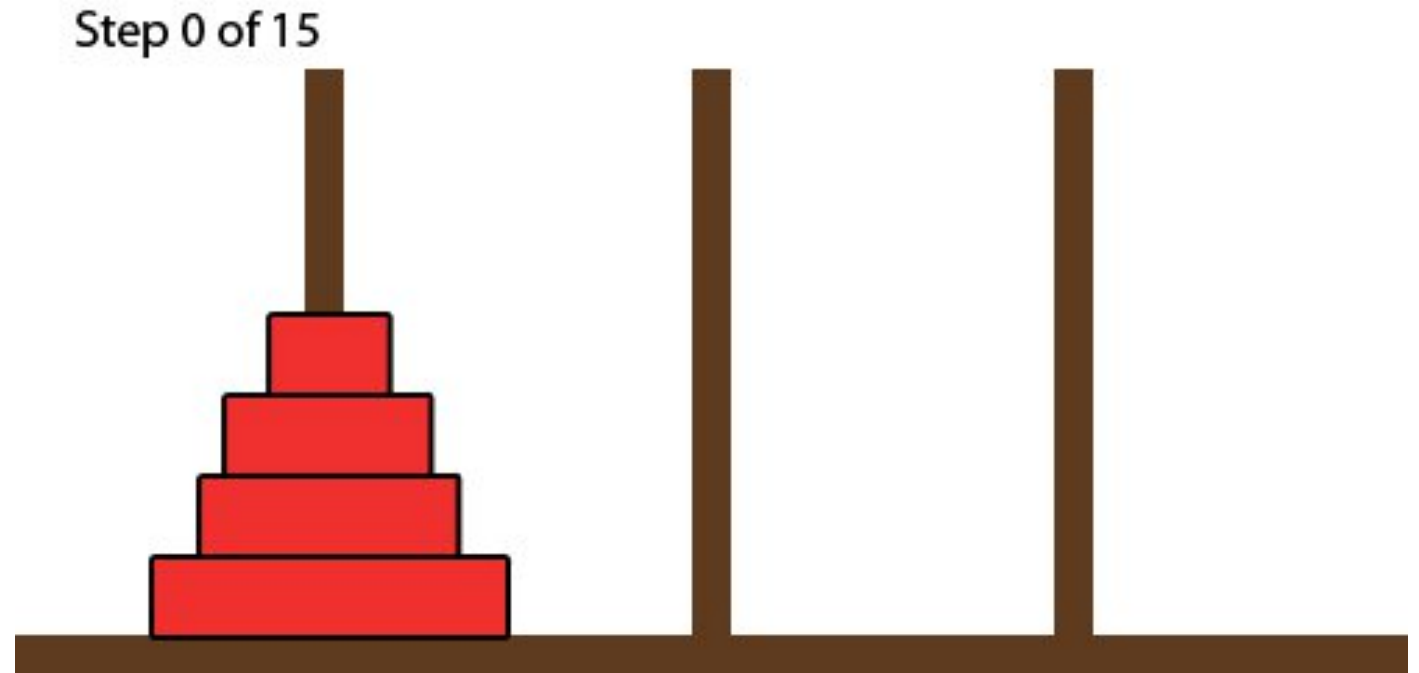
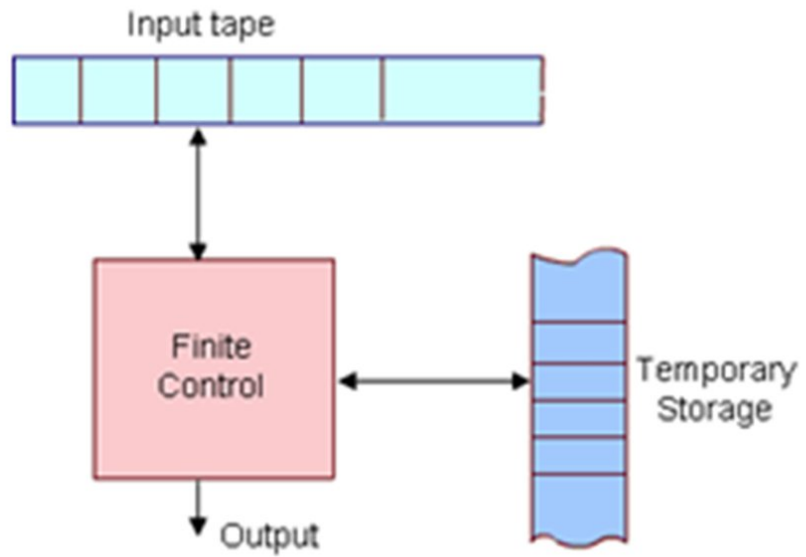
КВ-граматика визначає структуру ланцюжків і дозволяє будувати ланцюжки визначеної мови. Магазинні автомати (МА) дозволяють вирішувати для контекстно-вільних мов задачу розпізнавання, яка полягає в тому, що за заданим ланцюжком необхідно визначити, чи належить він заданій мові.

Модель магазинного автомата



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.

Автомати з магазинною пам'яттю



Зв'язок пристрою керування зі стрічками

Зв'язок пристрою керування зі стрічками здійснюється двома **головками**, що можуть переміщатися уздовж стрічок.

Головка вхідної стрічки може переміщатися тільки в один бік – вправо чи залишатися на місці. Вона може виконувати тільки операцію читання. Головка допоміжної стрічки здатна виконувати як читання, так і запис, але ці операції пов'язані з переміщенням головки певним чином:

- при записі головка попередньо зсувається на одну позицію вгору, а потім символ заноситься на стрічку;
- при читанні символ, що знаходиться під головою зчитується зі стрічки, а потім головка зсувається на одну позицію вниз, таким чином головка завжди встановлена проти останнього записаного символу. Позицію, що знаходиться в розглянутий момент часу під головою, називають вершиною

Завдання автомату

Магазинний автомат M визначається наступною сукупністю семи об'єктів:

$$M = \{P, S, s_o, f, F, H, h_o\},$$

де P – вхідний алфавіт, S – алфавіт станів, s_o – початковий стан, $s_o \in S$, F – множина кінцевих станів, $F \subseteq S$, H – алфавіт магазинних символів, записуваних на допоміжну стрічку, h_o – маркер дна, він завжди записується на дно магазину, $h_o \in H$, f – функція переходів.

Робота магазинного автомату

Робота МА зводиться до зміни конфігурацій. **Конфігурацією** автомата називають трійку об'єктів $(s, a\alpha, \gamma h)$, де S – поточний стан автомата, α – вхідний ланцюжок, лівий символ a знаходиться під головою, γ – це магазинний ланцюжок, верхній символ якого h .

Якщо $\alpha = \varepsilon$, то вважається, що вхідний ланцюжок прочитаний. Якщо $\gamma = \$$, то магазин порожній.

Початковою конфігурацією називається конфігурація (s, α, h_o) , де s – початковий стан і h_o – маркер дна магазину α – вхідний ланцюжок, а **заключною** – конфігурація $(s, \$, \$)$, де s належить до множини кінцевих станів F .

Для позначення послідовності конфігурацій, які змінюють одна одну використовується знак \vdash :

$$(s, \alpha_1, \gamma_1) \vdash (s, \alpha_2, \gamma_2) \vdash \dots \vdash (s, \alpha_n, \gamma_n)$$

Зміна конфігурацій

Один такт роботи автомата полягає у визначенні нової конфігурації за заданою. Це записується так:

$$(s, a\alpha, \gamma h) \vdash (s', \alpha, \gamma\beta)$$

У загальному виді дії, що задаються функцією переходів і виконуються автоматом, демонструє наступна форма запису:

$$f(s, \langle \text{вхідний символ} \rangle, \langle \text{магазинний символ} \rangle) = (s_1, \langle \text{ланцюжок, що заноситься} \rangle)$$

При цьому передбачається, що автомат зчитує символ a , що знаходиться під головою, і символ γ , що знаходиться у вершині магазину, визначає новий стан s' і записує ланцюжок β у магазин замість символу γ . Якщо $\beta = \$$, то верхній символ виявляється вилученим з магазину. Така зміна конфігурацій можлива, якщо функція $f(s, a, h)$ визначена і їй належить значення (s, β) . Описаний такт роботи виконується **зі зрушенням вхідної головки**.

Для опису роботи автомата також буде потрібен інший вид такту, що припускає зміну станів магазину **без зрушення вхідної голівки**. Якщо $f^*(s, a, h)$ визначена і їй належить значення (s, β) , то вона визначає зміну конфігурацій так:

$$(s, a\alpha, \gamma h) \vdash (s', a\alpha, \gamma\beta)$$

Компоненти магазинного автомату

Якщо $\Gamma = \{V_T, V_a, I, R\}$ є КС-граматикою, то за нею можливо побудувати такий магазинний автомат M , що $L(M) = L(\Gamma)$. Для спрощення, будемо використовувати магазинні автомати з одним станом s .

Отже, нехай задана граматика $\Gamma = \{V_T, V_a, I, R\}$. Визначимо компоненти автомата M наступним чином:

$$S = \{s\}, \quad P = V_T, \quad H = V_T \cup V_a \cup \{h_0\}, \quad F = \{s\},$$

де P – вхідний алфавіт (містить тільки термінальні символи), H – алфавіт магазину, який містить термінальні, нетермінальні символи та символ дна магазину h_0 .

Побудова магазинного автомату

В якості початкового стану автомата приймемо стан s і побудуємо функцію переходів так:

1. Для всіх правил $A \rightarrow \beta$, праві частини яких починаються з **нетермінального** символу, побудуємо команди виду :
 $f^*(s, \alpha, A) = (s, \beta^R)$, где β^R дзеркальне відображення β (без зрушення вхідної голівки).
2. Для всіх правил $A \rightarrow a\beta$, праві частини яких починаються з **термінального** символу, побудуємо команди виду:
 $f(s, a\alpha, A) = (s, \beta^R)$, где β^R дзеркальне відображення β (зі зрушення вхідної голівки).
3. Для всіх $a \in V_T$, розташованих в середині та в цінці правил, побудуємо команди виду:
 $f(s, a, a) = (s, \$)$
4. Для переходу до кінцевого стану побудуємо команду
 $f(s, \$, h_0) = (s, \$)$

Початкову конфігурацію автомата визначимо у вигляді:

(s, w, h_0I) , де w - заданий ланцюжок для розпізнавання, який записаний на вхідній стрічці.

Робота магазинного автомату

Автомат, побудований за наведеними вище правилами, працює наступним чином.

Якщо в вершині магазину знаходиться термінал, і символ, що считується із вхідного ланцюжка, збігається з ним, то за командою типу (3) термінал видаляється з магазину, а вхідна голівка вхідної стрічки зсувається на одну позицію вправо.

Якщо ж в вершині магазину знаходиться нетермінальний символ, то виконується команда типу (1), яка замість нетерміналу записує в магазин ланцюжок, що представляє собою праву частину правила грамматики для даного нетерміналу. Отже, автомат, послідовно замінюючи нетермінали, що з'являються в вершині магазину, будує в магазині ліве виведення вхідного ланцюжка, видаляючи отримані термінальні символи, що збігаються з символами вхідного ланцюжка. Це означає, що кожний ланцюжок, який може бути отриманим за допомогою лівого виведення в граматиці G , допускається побудованим автоматом M .

ПРИКЛАД 6

Магазинний автомат задано наступним чином:

М: $P = \{int, float, ;, *, i, j\}$, $H = \{I, E, R, A, B, C, int, float, ;, *, i, j, h_0\}$, $S = \{s\}$, $F = \{s1, s2\}$.

Робота автомату задається наступними функціями переходів (командами):

1. $f^*(s, int, I) = (s, RE)$;	9. $f(s, float, A) = (s, \$)$;
2. $f^*(s, float, I) = (s, RE)$;	10. $f(s, *, B) = (s, \$)$;
3. $f^*(s, int, R) = (s, RE)$;	11. $f^*(s, i, B) = (s, \$)$;
4. $f^*(s, float, R) = (s, RE)$;	12. $f^*(s, j, B) = (s, \$)$;
5. $f^*(s, \$, R) = (s, \$)$;	13. $f(s, i, C) = (s, \$)$;
6. $f^*(s, int, E) = (s, ;CBA)$;	14. $f(s, j, C) = (s, \$)$;
7. $f^*(s, float, E) = (s, ;CBA)$;	15. $f(s, ;, ;, ;) = (s, \$)$;
8. $f(s, int, A) = (s, \$)$;	16. $f(s_0, \$, h_0) = (s_0, \$)$.

ПРИКЛАД 6. Продовження

Перевіримо ланцюжок $int *i; float j;$ на приналежність заданій мові:

1. $f^*(s, \text{int}, I) = (s, RE);$	9. $f(s, \text{float}, A) = (s, \$);$
2. $f^*(s, \text{float}, I) = (s, RE);$	10. $f(s, *, B) = (s, \$);$
3. $f^*(s, \text{int}, R) = (s, RE);$	11. $f^*(s, i, B) = (s, \$);$
4. $f^*(s, \text{float}, R) = (s, RE);$	12. $f^*(s, j, B) = (s, \$);$
5. $f^*(s, \$, R) = (s, \$);$	13. $f(s, i, C) = (s, \$);$
6. $f^*(s, \text{int}, E) = (s, ;CBA);$	14. $f(s, j, C) = (s, \$);$
7. $f^*(s, \text{float}, E) = (s, ;CBA);$	15. $f(s, ;, ;) = (s, \$);$
8. $f(s, \text{int}, A) = (s, \$);$	16. $f(s_0, \$, h_0) = (s_0, \$).$

$(s, \text{int} *i; float j; , h_0I) \vdash 1 (s, \text{int} *i; float j;, h_0RE) \vdash 6 (s, \text{int} *i; float a j;, h_0R;CBA) \vdash$
 $8 (s, *i; float j;, h_0R;CB) \vdash 10 (s, i; float j;, h_0R;C) \vdash 13 (s,$
 $;float j;, h_0R;) \vdash 15 (s, float j;, h_0R) \vdash 4 (s, float j;, h_0RE) \vdash 7 (s, float j;, h_0RE) \vdash 7$
 $(s, float j;, h_0R;CBA) \vdash 9 (s, j;, h_0R;CB) \vdash 12$

ДЯКУЮ ЗА УВАГУ