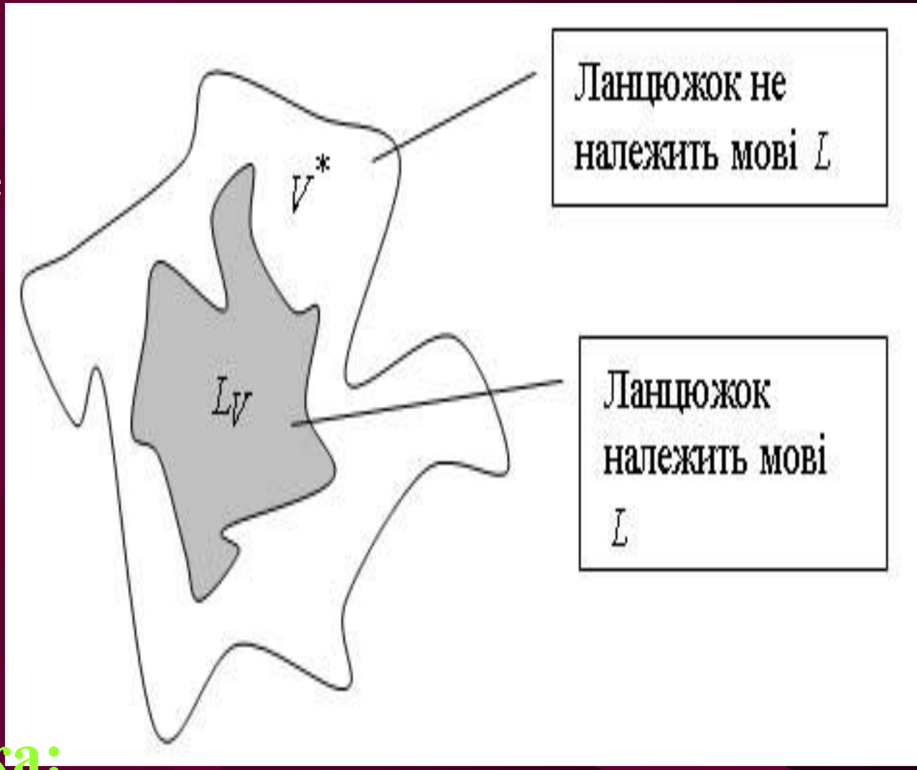
Тема 2. Елементи теорії формальних мов

1. Означення формальних мов. Ланцюжки

- Позначимо $V^{+} = V^{*} \setminus \{e\}$ – множину всіх слів, крім e (ε).
- Припустимо, що ми маємо слово ω, тоді послідовність $\underbrace{\omega\omega\omega\ldots\omega}_{n} = \omega^{n}$, а $\omega^{0} = \varepsilon$.



Формальне означення ланцюжка:

1) ε –ланцюжок в V.

2) Якщо – ω ланцюжок в алфавіті V і a є V, то – ωa ланцюжок в V.

3) α– ланцюжок в алфавіті V, якщо він ланцюжок внаслідок 1) або 2).

1.1. Приклади формальних мов

1. Множина всіх слів в алфавіті $V_1 = \{a\}$, $L_1 = \{\varepsilon, a, aa, aaa, ...\} = \{a^n | n \geq 0\}$

2. Нехай $V_2 = \{a, b, 1\}$, L_2 визначає множину ідентифікаторів:

   $L_2 = \{a, b, a1, b1, aa, ab, aa1, ...\}$.

3. $V_3 = \{a, b, c\}$, $L_3 = \{a^n bc^n | n \geq 0\}$, $aabcc \in L_3$, $abcc \notin L_3$.

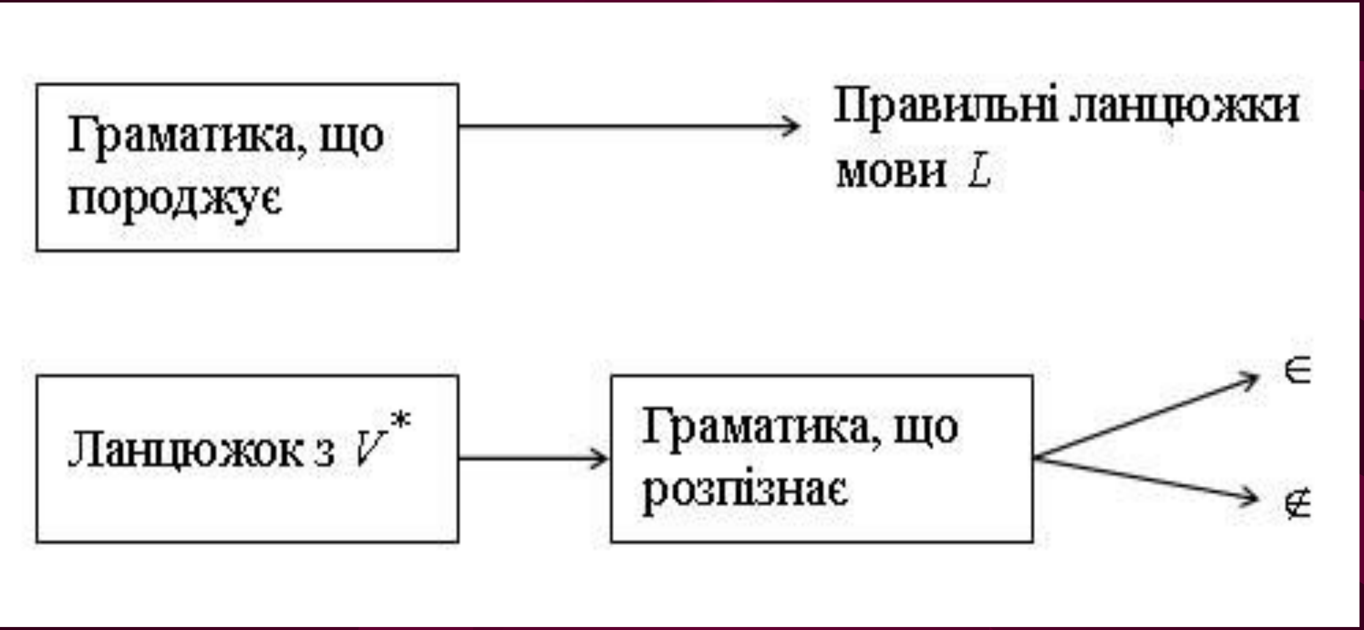
4. $V_4 = \{a\}$, $L_4 = \{\alpha \in V_4 | |\alpha| = 2^k, k = 0,1,2,...\}$, $aa \in L_4$, $aaaa \in L_4$, $aaa \notin L_4$.

5. $V_5 = \{a, b, c\}$; $L_5 = \{a^n bc^m | n, m \geq 0\}$, $aaaabcc \in L_5$, $cbaa \notin L_5$.

6. $V_6 = \{a, b, c\}$; $L_6 = \{x$ | у ланцюжку x кількості входжень a, b і c рівні$\}$,

   $cbaba \in L_6$, $ccb \notin L_6$.

1.2. Задача належності. Способи визначення мов



1.3. Регулярні операції над мовами

Нехай є три мови: L_1, L_2, L з алфавіту V.

Об'єднання $L_1 \cup L_2$ буде визначати нову мову:

$$L_1 \cup L_2 = \{\omega \mid \omega \in L_1 \text{ або } \omega \in L_2\}$$

Приклад: $\{a, ab, ba\} \cup \{a, ba, bb\} = \{a, ab, ba, bb\}$.

Конкатенацією (катенацією) мов L_1 **і** L_2 називається

$$L_1 L_2 = \{\omega_1 \omega_2 \mid \omega_1 \in L_1, \omega_2 \in L_2\}$$

Приклад: $\{a, b\}\{c, d, e\} = \{ac, ad, a, bc, bd, b\}$.

Піднесення до степеня мови L. $L^0 = \{e\}$, $L^i = L^{i-1} L$.

Приклад: $\{e, a, aa\}^2 = \{e, a, aa\}\{e, a, aa\} = \{e, a, aa, aaa, aaaa\}$.

Ітерація мови: $L^* = \{\omega^i \mid \omega \in L, i \ge 0\} = \{e\} \cup L \cup L^2 \cup \dots \cup L^n \cup \dots$

2. Метамова БНФ

<поняття> *має структуру* **<метавираз>**

<поняття> ::= <метавираз>

БНФ оператора присвоєння:

<оператор присвоювання> ::= <ім'я> ':=' <вираз>

<ім'я>::=<буква><послідовність букв і цифр >

Приклад 1. <оператор присвоювання> ::= <ім'я> ':=' <вираз>

<вираз> ::= <первинне> | <первинне> '+' <первинне> | <первинне> '-' <первинне>

<первинне> ::= <стала> | <ім'я>

<стала> ::= '1' | '2'

<ім'я> ::= 'x' | 'y' | 'z'

3. Розширені БНФ

Означення: Метавирази з символами "(", ")", "[", "]", "{", "}" називаються ***розширеними***, а БНФ – ***розширеними БНФ***, або ***РБНФ***.

X, Y, Z, … , T – довільні метавирази (можливо, порожні),
N – нетермінал

$N ::= X\ Z\ Y$ … $N ::= X\ T\ Y$	$N ::= X\ (\ Z \mid \ldots \mid T\ )\ Y$
<вираз> ::= <первинне> '+' <первинне> | <первинне> '-' <первинне>	<вираз> ::= <первинне> ('+' | '-') <первинне>

$N ::= X\ Z\ Y$ $N ::= X\ Y$	$N ::= X\ [\ Z\ ]\ Y$
<вираз> ::= <первинне> | <первинне> ('+'| '-') <первинне>	<вираз> ::= <первинне> [('+'| '-') <первинне>]
<оператор присвоювання> ::= <ім'я> ':=' <вираз> <вираз> ::= <первинне> | <первинне> '+' <первинне> | <первинне> '-' <первинне> <первинне> ::= <стала> | <ім'я> <стала> ::= '1' | '2' <ім'я> ::= 'x' | 'y' | 'z'	<оператор присвоювання> ::= <ім'я> ':=' ('1' | '2' | <ім'я>) [('+'| '-') ('1' | '2' | <ім'я>)] <ім'я> ::= 'x' | 'y' | 'z'

Ітераційні дужки “{“ , “}”

Якщо *X* – довільний метавираз, то *метавираз {X} позначає всі послідовності (у тому числі порожню) виразів, вивідних із X.*

Прикляд 2. Визначимо записати поняття «ідентифікатор» в алфавіті V={A,B,C,0,1}

<Ід> ::=<Б> { <Б> | <Ц> } <Б> ::= 'A' | 'B' | 'C' <Ц> ::= '0' | '1'	<Ід> ::= ('A' | 'B' | 'C'){ 'A' | 'B' | 'C' | '0' | '1' }

The syntax of C in Backus-Naur Form (fragment 1)

```
<expression> ::= <assignment-expression>
               | <expression> , <assignment-expression>

<assignment-operator> ::= =
                        | *=
                        | /=
                        | %=
                        | +=
                        | -=
                        | <<=
                        | >>=
                        | &=
                        | ^=
                        | |=
```

The syntax of C in Backus-Naur Form (fragment 2)

<selection-statement> ::= if (<expression>) <statement>
| if (<expression>) <statement> else <statement>
| switch (<expression>) <statement>

<iteration-statement> ::= while (<expression>) <statement>
| do <statement> while (<expression>) ;
| for ({<expression>}; {<expression>} ; {<expression>}) <statement>

Тут круглі дужки - термінальні символи, а не метасимволи

4. Граматики Хомського. Основні поняття

Означення: ***Граматикою Хомського*** називається четвірка

$$G = (N, T, P, S),$$

N – множина ***позначень понять*** мови, тобто ***нетермінальних символів*** (***нетерміналів***).

T – алфавіт означуваної мови, або множина ***термінальних символів*** (***терміналів***). $T \cap N = \varnothing$

P – множина ***правил виведення*** (***продукцій***) вигляду $v \rightarrow w$, де

$$v \in (T \cup N)^{*} N (T \cup N)^{*}, w \in (T \cup N)^{*}$$

тобто правий ланцюжок є довільною послідовністю терміналів і нетерміналів, а лівий містить принаймні один нетермінал.

S – ***початковий нетермінал*** із множини N, або ***позначення головного поняття***, яким позначаються слова мови.

How a grammar generates sentences

- $G = (N, T, P, S)$,

- Example: consider the grammar: P:

S → NP VP NP

NP → John

NP → a book

VP → reads

Terminals T: ‘John’, ‘a book’, ‘reads’. Nonterminals N: S, NP, VP.

We start with S.

We apply the first rule to replace S with NP VP NP:

S => NP VP NP

⇒ John VP NP

⇒ John reads NP

⇒ John reads a book

At this point, the string consists only of terminals and we must stop. It is a valid sentence belonging to the language described by the grammar.

How a grammar generates sentences

- Summary of the derivation:

S => NP VP NP
⇒ John VP NP
⇒ John reads NP
⇒ John reads a book

Another derivation:

S => NP VP NP
⇒ a book VP NP
⇒ a book reads NP
⇒ a book reads John

Another derivation:

S => NP VP NP
⇒ John VP NP
⇒ John reads NP
⇒ John reads John

$G = (N, T, P, S)$

Можна вважати P – скінченною підмножиною такої множини:

$$(T \cup N)^* N (T \cup N)^* \times (T \cup N)^*$$

Якщо (ν,ω) належить множині P, то серед правил граматики G існує правило вигляду ν→ ω.

Приклад: $(AB, CDE) \in P$ $\quad AB \to CDE$

Якщо $\alpha = xyABLC$, $\alpha \in G$, то $\beta = xyCDELC \in G$.

$v \to w_1$ і $v \to w_2$	$v \to w_1 \mid w_2$
$v \to w_1 w w_2$ і $v \to w_1 w_2$	$v \to w_1 [w] w_2$
$v \to w_1 u_1 w_2$ і $v \to w_1 u_2 w_2$	$v1 \to w1(u1 \mid u2)w2$

Приклади граматик Хомського

Приклад . $G_0 = (N, T, P, S)$

N: $\{ A, S \}$

T: $\{0,1\}$

P: $S \rightarrow 0\mathrm{A}1,\ 0A \rightarrow 00\mathrm{A}1,\ A \rightarrow \mathrm{e}$

Приклад . $G_1 = (\{ A, B, C, D \}, \{ a, 1, 2 \},$

$\{ A \rightarrow BC, A \rightarrow BD, A \rightarrow B, B \rightarrow a, C \rightarrow 1, D \rightarrow 2 \}, A)$

P можна переписати у вигляді

$\{ A \rightarrow B\ [\ C \mid \mathrm{D}\ ], B \rightarrow a, C \rightarrow 1, \mathrm{D} \rightarrow 2 \}$

Визначимо ряд понять:

1. На множині слів об'єднаного алфавіту $(T \cup N)^*$ означається ***відношення безпосередньої виводимості***, позначене знаком $\Rightarrow_G$ (або $\Rightarrow$, коли відомо, якою саме є G):

$$v \Rightarrow_G w, \quad \text{якщо} \quad v = x_1 u x_2, \quad w = x_1 y x_2, \quad u \to y \in P.$$

При цьому кажуть, що *w безпосередньо виводиться з v* ***застосуванням продукції u → y***.

Приклад . $G_1 = (\{ A, B, C, D \}, \{ a, 1, 2 \},$

$\{ A \to BC, A \to BD, A \to B, B \to a, C \to 1, D \to 2 \}, A)$

$BC \Rightarrow aC$ застосуванням продукції $B \to a$,

$aC \Rightarrow a1$ застосуванням $C \to 1$

2. На множині $(T \cup N)^*$ означається ***відношення виводимості***, позначене $\Rightarrow^*_G$ (або $\Rightarrow^*$, коли відомо, якою саме є G): $\boldsymbol{v} \Rightarrow^* \boldsymbol{w}$, якщо $v=w$ або існує послідовність $w_1, w_2, \ldots, w_n$ слів, де $n \geq 1$, така, що $v \Rightarrow w_1$, $w_1 \Rightarrow w_2$, … , $w_{n-1} \Rightarrow w_n$, $w_n=w$. Послідовність $v \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \ldots \Rightarrow w_n$ називається ***виведенням*** w_n із v, а n – ***довжиною виведення***.

$\boldsymbol{v} \Rightarrow^* \boldsymbol{w}$ можна уточнити $v \Rightarrow^n w$.

Приклад . $G_1 =(\{ A, B, C, D \},\{ a, 1, 2 \},$

$\{ A \to BC, A \to BD, A \to B, B \to a, C \to 1, D \to 2 \}, A)$

$BC \Rightarrow^* a1$, оскільки $BC \Rightarrow aC$, $aC \Rightarrow a1$. $(BC \Rightarrow^2 a1)$

3. Якщо $S \Rightarrow_G^* w$, то послідовність $S \Rightarrow \ldots \Rightarrow w$ називається ***виведенням слова w у граматиці G***, а слово w – ***вивідним***.

Приклад .Слова A, BC, aC, $a1$ вивідні в граматиці G_1.

4. Вивідні слова в алфавіті T називаються ***породжуваними***, а множина їх усіх – ***мовою, що задається*** (***породжується***) ***граматикою G***:

$$L(G) = \{w \mid w \in T^* \text{ та } S \Rightarrow^* w\}.$$

Приклад . $G_0 = (N, T, P, S)$

P: $S \to 0A1,\ 0A \to 00A1,\ A \to e$

N: $\{A, S\}$

T: $\{0,1\}$

$L(G_0) = \{0^n1^n \mid n$ □ $1\}$

$$S \Rightarrow 0A1 \Rightarrow 00A11 \Rightarrow 0011$$

Приклад . $G_1 = (\{A, B, C, D\}, \{a, 1, 2\},$

$\{A \to BC,\ A \to BD,\ A \to B,\ B \to a,\ C \to 1,\ D \to 2\},\ A)$

$L(G_1) = \{a,\ a1,\ a2\}$

$$A \to BC \Rightarrow aC \Rightarrow a1$$

5. Вивідний ланцюжок граматики , що не містить нетермінальних символів називається ***термінальним ланцюжком***, що породжується граматикою , а мова, що породжується граматикою – це множина всіх термінальних ланцюжків, що породжуються граматикою.

6. Граматики називаються ***еквівалентними***, якщо задають ту саму мову.

Приклад . $G_1 =(\{ A, B, C, D \},\{ a, 1, 2 \},$

$$\{ A \to BC, A \to BD, A \to B, B \to a, C \to 1, D \to 2 \}, A)$$

$\underline{G}_1 = (\{ A \}, \{ a, 1, 2 \}, \{ A \to a [1 | 2] \}, A)$ $L(G_1) = \{a, a\,1, a\,2 \}$

Приклад . $G_2 = (\{ I, L, D \},\{ a, \ldots, \mathrm{z}, 0, \ldots, 9 \},$

$$\{ I \to L \mid IL \mid ID, L \to a \mid \ldots \mid \mathrm{z}, D \to 0 \mid ... \mid 9 \}, I)$$

$\underline{G}_2 = (\{I, C\}, \{a, \ldots, \mathrm{z}, 0, \ldots, 9\},$

$$\{I \to (a|\ldots|\mathrm{z})C, C \to \varepsilon \mid C(a| \ldots|\mathrm{z}|0|\ldots|9)\}, I)$$

Введемо позначення:

1. Будемо позначати a,b, …,0,1, …9 – термінали.
2. A, B, C, D, E, S – нетермінали.

   E, S – початкові нетермінали.
3. U, V, …, Z – або термінал, або нетермінал.
4. α, β – ланцюжки, що містять термінали і нетермінали.
5. u, v, …,z – ланцюжки, що містять тільки термінали.

5. Класифікація граматик Хомського. Приклади

Граматики можна класифікувати за виглядом їх правил. Класифікація нижче називається ієрархією за Хомським. Нехай $G=(N,T,P,S)$.

Означення 1. Граматика G називається праволінійною (ліволінійною), якщо

$A\to x, A\to xB$, де $A,B\in N, x\in T^*$

(якщо $A\to x, A\to Bx$, де $A,B\in N, x\in T^*$).

Приклад. $G_3=(\{S\},\{0,1\},\{S\to 0S\mid 1S\mid e\},S)$.

Означення 2. Граматика G називається контекстно-вільною (безконтекстною), якщо кожне правило з P має вигляд: $A \to \alpha, A \in N, \alpha \in (T \cup N)^*$.

Застосування продукції $A \to w$ до ланцюжка uAv не залежить, тобто є ***вільним*** від сусідніх з A символів, які утворюють ***контекст*** uv.

Приклад контекстно-вільної граматики:

$G_4 = (\{ E, T, F\}, \{ a, *, +, (,)\}, P, E)$

P: $E \to E+T \mid T$

$T \to T*F \mid F$

$F \to (E) \mid a$

$\underline{E} \Rightarrow \underline{E}+T \Rightarrow \underline{T}+T \Rightarrow \underline{F}+T \Rightarrow a+\underline{T} \Rightarrow a+\underline{T}*F \Rightarrow a+a*F \Rightarrow a+a*a$

Означення 3. Граматика G називається контекстно-залежною або нескоротною, якщо кожне правило з P має вигляд: $\alpha \to \beta, \alpha, \beta \in (T \cup N)^*, \ |\alpha| \leq |\beta|$.

Контекстно-залежні граматики не допускають правила вигляду A →e.

Приклад контекстно-залежної граматики:

G_5 = ({ B,C,S},{ a, b, c} ,P,S)

P:
- S→aSBC | abC
- CB→BC
- bB→bb
- bC→bc
- cC→cc

$$L(G_5) = \{a^n b^n c^n \mid n \geq 1\}.$$

S=>aSBC=>aabCBC=> aabBCC=> aabbCC => aabbcC => aabbcc

Означення 4. Граматика, яка не підлягає жодним обмеженням з означень 1,2,3 називається необмеженою або граматикою загального вигляду.

Приклад необмеженої граматики:

$G_6 = (\{A,B,C,D,S\}, \{a, b\}, P, S)$

P:
1) $S \rightarrow CD$
2) $C \rightarrow aCA$
3) $C \rightarrow bCB$
4) $AD \rightarrow aD$
5) $BD \rightarrow bD$
6) $Aa \rightarrow aA$
7) $Ab \rightarrow bA$
8) $Ba \rightarrow aB$
9) $Bb \rightarrow bB$
10) $C \rightarrow e$
11) $D \rightarrow e$

$$S \Rightarrow^{(1)} CD \Rightarrow^{(2)} aCAD \Rightarrow^{(3)} abCBAD \Rightarrow^{(4)} abCBaD \Rightarrow^{(8)} abCaBD \Rightarrow^{(5)}$$

$$\Rightarrow^{(5)} abCabD \Rightarrow^{(10),(11)} abab.$$

$$L(G_6) = (\omega\omega \mid \omega \in \{a, b\}^*).$$

$$L(G_6) = (\omega\omega \mid \omega \in \{a,b\}^*).$$

$$L(G_6) \supseteq (\omega\omega \mid \omega \in \{a,b\}^*). \qquad L(G_6) \subseteq (\omega\omega \mid \omega \in \{a,b\}^*).$$

1) S→CD
2) C→aCA
3) C→bCB
4) AD→aD
5) BD→bD
6) Aa→aA
7) Ab→bA
8) Ba→aB
9) Bb→bB
10) C→e
11) D→e

$$S \Rightarrow CD$$

Для $n \geq 1: \ C \Rightarrow^{(2)-(3),n} c_1 c_2 .. c_n C X_n X_{n-1} .. X_1$

$$\Rightarrow^{(10)} c_1 c_2 .. c_n X_n X_{n-1} .. X_1.$$

$c_i = a$ тоді і тільки тоді, коли $X_i = A.$

$c_i = b$ тоді і тільки тоді, коли $X_i = B.$

$$X_n X_{n-1} .. X_1 D \Rightarrow^{(4)-(5)} X_n X_{n-1} .. X_2 c_1 D$$

$$\Rightarrow^{(6)-(9),n-1} c_1 X_n X_{n-1} .. X_2 D$$

$$\Rightarrow^* c_1 c_2 X_n X_{n-1} .. X_3 D$$

$$\Rightarrow^{*(4)-(9)} c_1 c_2 .. c_n D \Rightarrow^{(11)} c_1 c_2 .. c_n.$$

$$S \Rightarrow^* c_1 c_2 .. c_n c_1 c_2 .. c_n$$

Означення 5. Праволінійна граматика $G = (T, N, P, S)$ називається *регулярною (автоматною)*, якщо

- кожне правило із P, за виключенням $S \to e$, має вигляд $A \to aB$ або $A \to a$, де $A, B \in N$, $a \in T$;
- якщо $S \to e$ належить P, то S не зустрічається в правих частинах правил.

Приклади:

$G_7 = (\{ A,S\}, \{ a, b\}, P, S)$ P: $S \to abA \mid e$, $A \to Saa \mid b$

$G_8 = (\{ A,C,S\}, \{ a, b\}, P, S)$ P: $S \to aC \mid e$, $A \to a$, $C \to bA \mid bC \mid e$

$G_9 = (\{S\}, \{ 0, 1\}, P, S)$ P: $S \to 0S \mid 1S \mid e$

<u>Означення 6</u>. Граматика G називається *розширеною граматикою*, якщо вона задається списком пар $A_i \to r_i$, де A_i — різні символи нетермінального алфавіту N; r_i — регулярні вирази в алфавіті $N \cup T$. Нетермінал першої пари вважається головним (початковим).

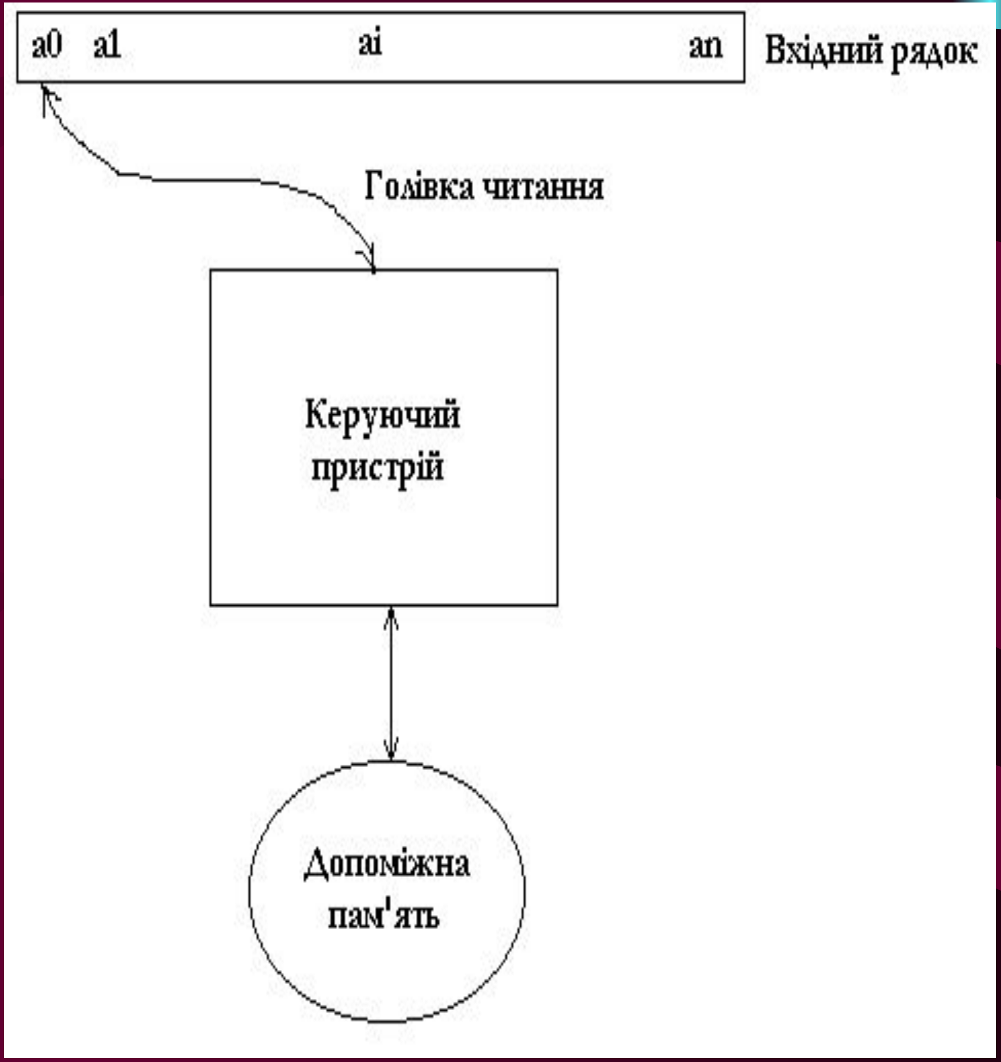
Приклади розширених граматик:

- $G_{10}=\{S\to AB,\ B \to x|y,\ A \to z|\omega\}$;
- $G_{11}=\{S\to (z|\omega)(x|y)\}$. G_{10} □ G_{11}

Адреса://fpm.chnu
Гіперпосилання “Системне програмування”, Вкладинка “Програмний супровід”

5. Розпізнавачі

Розпізнавач – це схематизований алгоритм, який визначає деяку множину ланцюжків.



Розпізнавач складається з трьох частин:

- вхідна стрічка;
- керуючий пристрій із скінченню пам'яттю;
- робоча (допоміжна) пам'ять.

Означення 1. Керуючий пристрій називається ***детермінованим***, якщо для кожної конфігурації існує не більше одного наступного детермінованого кроку.

Означення 2. Конфігурація називається ***початковою***, якщо керуючий пристрій знаходиться в заданому початковому стані, вхідна голівка розглядає найлівіший символ, а пам'ять має початкове вмістиме.

Означення 3. Конфігурація називається ***заключною***, якщо керуючий пристрій знаходиться в одному із заключних станів, а вхідна голівка оглядає правий кінець стрічки.

Означення 4. Кажуть, що розпізнавач ***допускає вхідний рядок***, якщо починаючи із початкової кофігурації, розпізнавач може виконати послідовність кроків, що закінчиться заключною конфігурацією.

Означення 5. ***Мова***, що дозволяється розпізнавачем – це множина вхідних ланцюжків, які він допускає.

Для кожного класу граматик із ієрархії Хомського існує свій клас розпізнавачів, які визначають ті самі класи мов, що і граматики, наприклад:

1) Мова L *праволінійна* тоді і тільки тоді, коли вона розпізнається одностороннім детермнiованим скінченним розпізнавачем.

2) Мова L *контекстно-вільна* тоді і тільки тоді, коли вона визначається одностороннім недетермнiованим розпізнавачем з магазинною пам’яттю.

3) Мова L *контекстно-залежна* тоді і тільки тоді, коли вона розпізнається двостороннім недетермнiованим обмеженим розпізнавачем.

Тема 4