Мови. Формальні породжувальні граматики. Типи граматик (ієрархія Хомські). Дерева виведення. Форми Бекуса-Наура.

Методи і положення математичної лінгвістики є теоретичною базою для створення алгоритмічних мов, для побудови систем автоматичного опрацювання мовного матеріалу в ЕОМ: машинного перекладу, інформаційного пошуку, автоматизації видавничих процесів, реферування й анотування наукової літератури, створення термінологічних банків, машинних фондів різних мов (система автоматизації трудомістких процесів у мовознавстві), автоматичного укладання словників, машинного розпізнавання і синтезу усного мовлення тощо.

Мови. Формальні породжувальні граматики.

Алфавіт (або словник) *V* – це скінченна непорожня множина елементів, які називаються символами. Слово (або речення) над *V* – це ланцюжок скінченої довжини елементів з *V*. Порожній (або нульовий) ланцюжок – це ланцюжок, який не містить символів; він позначається через Λ. Множина всіх слів над *V* позначається через V^*.

Мова над *V* – це підмножина V^*. Мови можуть бути задані різними способами. Один з них – задати всі слова мови. Інший – означити критерій, якому повинні задовольняти слова, щоб належати мові.

Розглянемо ще один важливий спосіб задати мову – через використання граматики.

Граматика складається з множини символів різного типу та множини правил побудови слів.

Точніше: граматика має алфавіт *V*, який є множиною символів, що використовуються для побудови слів мови. Деякі елементи алфавіту не можуть бути замінені іншими символами. Такі елементи називаються кінцевими (термінальними), а ті, що можуть бути замінені іншими символами, – нетермінальними. Вони позначаються через *T* та *N* відповідно.

Є спеціальний символ – елемент алфавіту – початковий символ, який позначається через *S*, з якого ми завжди починаємо.

Правила, які визначають, коли ми можемо замінити ланцюжок з V^* іншим ланцюжком, називаються продукціями граматики. Позначимо $w_0 \to w_1$ продукцію, яка означає, що ланцюжок w_0 має бути замінений на w_1. Підсумуємо сказане. Граматика із фразовою структурою (ГФС) $G = (V, T, S, P)$ містить алфавіт – множину *V*, її підмножину T термінальних елементів, початковий символ $S(S \in V)$ та множину продукцій *P*. Множина V/T позначається через N. Елементи з N називаються нетермінальними. Кожна продукція з *P* повинна містити принаймні один нетермінальний елемент у лівій частині.

Приклад 1: $G = (V, T, S, P)$, де $V = \{a, b, A, B, S\}$, $T = \{a, b\}$, *S*-початковий символ, $P = \{S \to ABa, A \to BB, B \to ab, AB \to b\}$.

Це приклад ГФС. Нехай *x* та *y* – ланцюжки над алфавітом *V*. Конкатенацією *x* та *y* називається ланцюжок *z=xy* (тобто до ланцюжка *x* дописано ланцюжок *y*).

Нехай $G = (V, T, S, P)$ – ГФС, і нехай $w_0 = lz_0r,\ z_0 \neq \Lambda$ (тобто w_0 – конкатенація $l,\ z_0\ ma\ r$) та $w_1 = lz_1r$ – ланцюжки над *V*. Якщо $z_0 \to z_1$ є продукцією граматики *G*, то кажуть, що w_1 безпосередньо виводиться з w_0 і записують $w_0 \Rightarrow w_n$.

Якщо $w_0, w_1, ..., w_n$ – ланцюжки над *V*, такі, що $w_0 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow ... \Rightarrow w_{n-1} \Rightarrow w_n$, то кажуть, що w_0 породжує w_n та використовують запис $w_0 \overset{*}{\Rightarrow} w_n$.

Послідовність кроків для отримання w_n з w_0 називається виведенням.

Приклад 2: Ланцюжок $Aaba$ безпосередньо виводиться з ABa у граматиці з прикладу 1, оскільки $B \to ab$ є продукцією граматики. Ланцюжок $abababa$ породжується ланцюжком ABa, оскільки $ABa \Rightarrow Aaba \Rightarrow BBaba \Rightarrow Bababa \Rightarrow abababa$ з допомогою продукцій $B \to ab, A \to BB, B \to ab$ *та* $B \to ab$ послідовно.

Нехай $G = (V, T, S, P)$ – ГФС. Мовою, що породжується *G*, позначається через $L(G)$, є множина всіх ланцюжків терміналів, які виводяться з початкового символу *S*, тобто $L(G) = \left\{ w \in T^* \middle| S \overset{*}{\Rightarrow} w \right\}$.

Приклад 3: Нехай *G* – граматика з алфавітом $V = \{S, A, a, b\}$, множина терміналів $T = \{a, b\}$, початковий символ *S* і множина продукцій $P = \{S \to aA, S \to b, A \to aa\}$. Знайти мову $L(G)$, яка породжується цією граматикою.

Із початкового символу *S* можна вивести *aA* використовуючи продукцію $S \to aA$; можна також використати продукцію $S \to b$, щоб вивести *b*. З *aA*, скориставшись продукцією $A \to aa$, можна вивести aaa. Ніяких інших слів вивести не можна. Отже, $L(G) = \{b, aaa\}$.

Приклад 4: Нехай *G* граматика з алфавітом $V = \{S, 0, 1\}$, $T = \{0, 1\}$, початковий символ S та множина продукцій $P = \{S \to 11S, S \to 0\}$. Знайти $L(G)$.

Отримаємо з *S* 0 $(S \to 0)$ або $11S$ $(S \to 11S)$. З $11S$ може бути отримано 110 або $1111S$. З $1111S$ виводяться 11110 або 1111110. Тобто після кожного виведення ми або додаємо дві одиниці в кінець ланцюжка або закінчуємо ланцюжок нулем. Тобто $L(G) = \{0, 110, 11110, 1111110, ...\}$ – це множина всіх ланцюжків з парною кількістю тільки 1, після яких (у кінці) один 0.

Зауваження. Ми отримали нескінченну мову ($L(G)$ складається з нескінченної кількості ланцюжків). Щоб граматика *G* породжувала нескінченну мову, в множині продукцій повинно бути принаймні одне рекурсивне правило (у прикладі 4 це правило $S \to 11S$).

Важливою є проблема побудови граматики для заданої мови.

Приклад 5: Знайти ГФС, яка породжує множину $\{0^n 1^n \mid n = 0, 1, 2, ...\}$.

Потрібно дві продукції, щоб побудувати ланцюжок, який складається з однакової кількості нулів за яким слідує така ж кількість одиниць. Перша продукція додає один 0 на початок і одну 1 у кінець ланцюжка. Друга продукція замінює *S* на порожній ланцюжок Λ. Розв'язком є граматика $G = (V, T, S, P), V = \{0, 1, S\}$, $T = \{0, 1\}$, S – початковий символ, $P = \{S \to 0S1, S \to \Lambda\}$.

Приклад 6: Знайти ГФС, яка генерує множину $\{0^n 1^m \mid n, m = 0, 1, 2, ...\}$.

Таких граматик вважаємо дві:

$G_1: V = \{S, 0, 1\}, T = \{0, 1\}, P = \{S \to 0S, \ S \to S1, \ S \to \lambda\}$ $G_2: V = \{S, A, 0, 1\}, T = \{0, 1\}$ $P = \{S \to 0S, S \to 1A, A \to 1A, A \to 1, S \to \Lambda\}$.

Цей приклад свідчить, що дві різні граматики можуть породжувати одну мову.

Іноді множина, яка легко описується, задається достатньо складною граматикою.

Приклад 7: Побудувати ГФС, яка породжує мову $\{0^n 1^n 2^n \mid n = 0,1,2,\ldots\}$.

Пропонується переконатися, що розв'язком цієї задачі є така граматика:

$G = (V, T, S, P)$, $V = \{0,1,2,S,A,B\}$, $T = \{0,1,2\}$, початковий символ S, множина продукцій $P = \{S \to 0SAB;\ S \to \Lambda;\ BA \to AB;\ 0A \to 01;\ 1A \to 11;\ 1B \to 12;\ 2B \to 22\}$.

Типи граматик (ієрархія Хомські)

Продукція, яка також називається правилом перетворення, дає можливість заміняти одну послідовність символів іншою. ГФС класифікуються за типами продукцій. Ми розглянемо класифікацію, яку запропонував Хомський (Noah Chomsky) (див. табл. 1)

Табл. 1. Типи граматик.

Тип	Обмеження на продукції $w_1 \to w_2$
0	немає обмежень
1	$\lvert w_1 \rvert \le \lvert w_2 \rvert$, або $w_2 = \Lambda$
2	$w_1 = A$, де A – нетермінальний символ
3	$w_1 = A$ та $w_2 = aB$ чи $w_2 = a$, де A, B– нетермінальні символи, a – термінальний символ, або $S \to \Lambda$

У табл. 1. через $\lvert w \rvert$ позначено довжину ланцюжка w, тобто кількість символів у ньому. Співвідношення між граматиками різних типів ілюструється діаграмою на рис. 1.

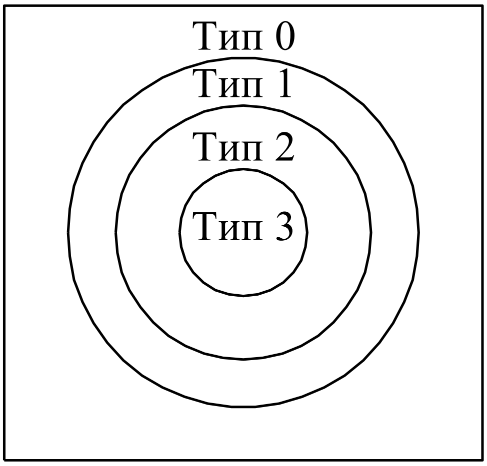

Рис. 1. Діаграма співвідношень між граматиками різних типів.

• Граматика типу 2 має продукції лише у формі $A \to w_2$, де A– нетермінальний символ. Ця граматика називається контекстно вільною, оскільки нетермінал A може бути замінений послідовністю w_2 у довільному ланцюжку щоразу, коли він зустрічається, тобто не залежно від контексту.

• Граматика типу 1 називається контекстно залежною. Коли у такій граматиці є продукція $lAr \to lw_2r$, у якій хоча б один з ланцюжків *l, r* відмінний від Λ, то нетермінал *A* може бути замінений ланцюжком w_2 лише в оточенні *l* та *r*, тобто у відповідному контексті, звідси і назва.

• Граматика типу 3 називається регулярною. Ця граматика може мати продукції лише у формі $A \to aB,\ A \to a,\ S \to \Lambda$, де *A, B* – нетермінали, *a* – термінал.

Мова називається контекстно залежною, якщо існує принаймні одна контекстно залежна граматика, яка породжує цю мову. Мова називається контекстно вільною, якщо існує принаймні одна контекстно вільна граматика, яка породжує цю мову. І, нарешті, мова називається регулярною, якщо існує принаймні одна регулярна граматика, яка породжує цю мову.

Приклад 8. Мова $\{0^m1^n | m, n = 0,1,2,...\}$ є регулярною, оскільки вона може бути породжена регулярною граматикою G_2 прикладу 6.

Приклад 9. Мова $\{0^n1^n \mid n = 0,1,2,...\}$ є контекстно вільною мовою, оскільки вона породжена граматикою з продукціями $S \to 0S1$ та $S \to \Lambda$. Проте ця мова не є регулярною: не існує регулярної граматики, яка б цю мову породжувала. Цей факт вимагає окремого доведення.

Приклад 10. Мова $\{0^n1^n2^n \mid n = 0,1,2,...\}$ є контекстно залежною мовою, оскільки вона може породжуватись граматикою типу 1 (див. приклад 7). Ця мова не може бути породжена жодною граматикою типу 2, цей факт також вимагає окремого доведення.

Дерева виведення

Виведення у мовах, породжених контекстно вільними граматиками, може зображатися графічно з використанням орієнтованих кореневих дерев. Ці дерева називають деревами виведення або синтаксичною розбору.

Кореню цього дерева відповідає початковий символ. Внутрішнім вершинам відповідають нетермінальні символи, що зустрічаються у виведенні. Листкам відповідають термінальні символи.

Нехай *w* – слово і $A \to w$ – продукція, яка використана у виведенні. Тоді вершина, яка відповідає нетермінальному символу *A* має синами вершини, які відповідають кожному символу *w* у порядку зліва направо.

Приклад 11: Визначити, чи слово *cbab* належить мові, породженій граматикою $G = (V, T, S, P)$, де $V = \{a, b, c, A, B, C, S\}$, $T = \{a, b, c\}$, а множина продукцій

$$P = \{S \to AB,\ A \to Ca,\ B \to Ba,\ B \to Cb,\ B \to b,\ C \to cb,\ C \to b\}.$$

Розв'язати цю задачу можна двома способами.

1. Розбір зверху вниз.

Оскільки є лише одна продукція з *S* у лівій частині, то починаємо з $S \to AB$. Далі використаємо продукцію $A \Rightarrow Ca$. Отже, маємо $S \Rightarrow AB \Rightarrow CaB$.

Оскільки *cbab* починається з символів *cb*, то використовуємо продукцію $C \to cb$, це дасть $S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB$. Завершуємо виведення, використавши продукцію $B \to b$:

$$S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB \Rightarrow cbab.$$

Отже, слово $cbab$ належить мові $L(G)$.

2. Розбір знизу вверх.

Починаємо з рядка, який потрібно вивести: $cbab$. Можна використати продукцію $C \to cb$, отже, $Cab \Rightarrow cbab$.

Далі використаємо продукцію $A \to Ca$, тоді матимемо $Ab \Rightarrow Cab \Rightarrow cbab$. Використавши продукцію $B \to b$ отримаємо $AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$. Нарешті, використаємо продукцію $S \Rightarrow AB \Rightarrow CaB \Rightarrow cbaB \Rightarrow cbab$: $S \Rightarrow AB \Rightarrow Ab \Rightarrow Cab \Rightarrow cbab$.

Дерево виведення для рядка $cbab$ у граматиці G зображено на рис.2.

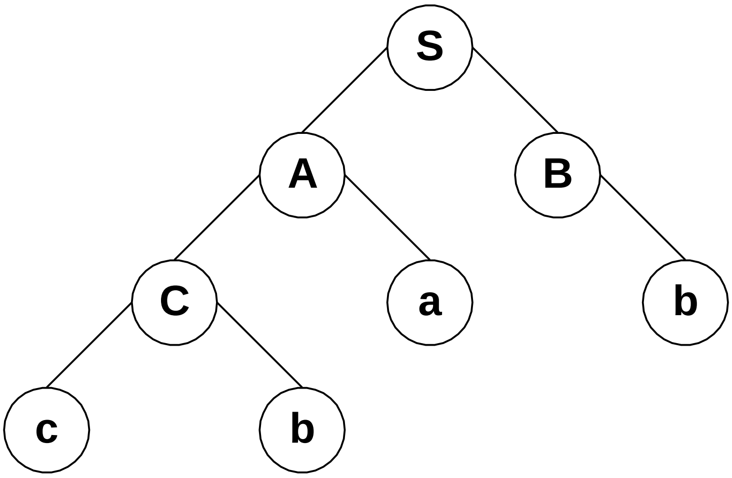

Рис. 2. Дерево виведення для рядка $cbab$.

Форми Бекуса-Наура

Для граматик типу 2 (контекстно вільних) окрім звичайного існує ще інший спосіб задання – форми Бекуса-Наура.

Продукції граматик типу 2 мають у лівій частині один символ (нетермінальний). Замість того, щоб виписувати окремо всі продукції, можна об'єднати в один вираз продукції з однаковим символом у лівій частині. У такому випадку замість символу $\to$ у продукціях використовується символ $::=$. Усі нетермінали при цьому заключаються у трикутні дужки $<>$. Праві частини продукцій в одному виразі відокремлюються одна від одної символом $|$.

Наприклад, продукції $A \to Aa, A \to a, A \to AB$ можна зобразити таким одним виразом у формі Бекуса-Наура: $\langle A \rangle ::= \langle A \rangle a \big| a \big| \langle A \rangle \langle B \rangle$

<u>Приклад 12</u>: Знайти продукції в граматиці, якщо у формі Бекуса-Наура вони записуються так:

$\langle expression \rangle ::= (\langle expression \rangle) \mid \langle expression \rangle +$

$\langle expression \rangle \mid \langle expression \rangle * \langle expression \rangle \mid \langle variable \rangle$

$\langle variable \rangle ::= x \mid y$

Зобразити дерево виведення у цій граматиці для ланцюжка $(x*y)+x$.

Для зручності використаємо позначення E для $\langle expression \rangle$ (це буде і початковий символ) та V для $\langle variable \rangle$.

Тоді правилами перетворення (продукціями граматики будуть)

$E \to (E)$, $E \to E + E$, $E \to E * E$ та $E \to V$ з першого виразу, а також $V \to x$ та $V \to y$ з другого виразу.

Дерево виведення зображено на рис. 3.

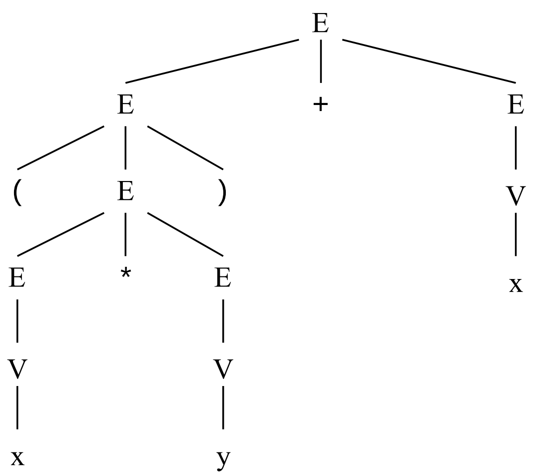

Рис. 3. Дерево виведення для рядка $(x * y) + x$.