

ТЕОРІЯ ФОРМАЛЬНИХ МОВ

Лекція 2

Формальні граматики

$G = (V, T, P, S)$ · правила виведення · ліве й праве виведення · дерева виведення



Що ми розглянемо сьогодні

1

Поняття формальної граматики

Породжувальний механізм замість переліку слів

2

Четвірка $G = (V, T, P, S)$

Структура формальної граматики

3

Термінали й нетермінали

Символи мови та допоміжні символи

4

Правила виведення

Продукції $\alpha \rightarrow \beta$ та їх застосування

5

Ліве та праве виведення

Порядок заміни нетерміналів

6

Дерева виведення й виведення слова

Структура виведення та $L(G)$

Від переліку слів до правил породження

МОВА ЯК ПЕРЕЛІК

Лекція 1: мова задається переліком слів або умовою належності. Для нескінченних мов повний перелік неможливий.

$$L = \{a, ab, abb, abbb, \dots\}$$

МОВА ЯК ГРАМАТИКА

Граматика — система правил, що породжує всі слова мови і тільки їх, виходячи з початкового символу. Компактно описує навіть нескінченні мови.

$$S \rightarrow aS / a$$

Ідея належить Ноаму Хомському (1956 р.), який запропонував ієрархію формальних граматик за складністю правил — основу сучасної теорії компіляції.

Граматика $G = (V, T, P, S)$

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

V**Нетермінали**

Допоміжні символи
(синтаксичні категорії)

T**Термінали**

Символи, з яких
складається саме слово
мови

P**Правила виведення**

Продукції вигляду $\alpha \rightarrow \beta$

S**Аксіома**

Початковий символ, $S \in V$

$V \cap T = \emptyset$ - множини не перетинаються. Мова, породжена граматикою, позначається $L(G)$.

ПРИКЛАД: граматика виразів

$V = \{S, T, F\}$, $T = \{a, +, *, (,)\}$

Т проти V: дві ролі символів

ТЕРМІНАЛИ (T)

«Фінальні» символи, з яких складається саме слово мови. Далі не замінюються і залишаються в результаті.

$$T = \{a, +, *, (,)\}$$

*У кінцевому слові «a+a*a» є лише символи з T.*

НЕТЕРМІНАЛИ (V)

Допоміжні, «службові» символи - синтаксичні категорії, що підлягають подальшій заміні.

$$V = \{S, T, F\}$$

S - «вираз», T - «доданок», F - «множник».

Продукція $\alpha \rightarrow \beta$

$$\alpha \rightarrow \beta, \quad \alpha, \beta \in (V \cup T)^*, \quad \alpha \text{ містить хоча б один символ із } V$$

Застосування правила: підланцюжок, що збігається з α , замінюється на β . Крок називають безпосереднім виведенням і позначають символом $\Rightarrow$.

$$S \rightarrow aSb$$

Символ S замінюється на ланцюжок « aSb »

$$A \rightarrow \varepsilon$$

ε -правило: нетермінал A видаляється

$$S \rightarrow a \mid b$$

Скорочення для двох правил: $S \rightarrow a$ та $S \rightarrow b$

Типи граматики за ієрархією Хомського відрізняються обмеженнями на вигляд правил $\alpha \rightarrow \beta$ (розглянемо в наступних лекціях).

Порядок заміни нетерміналів

Правила: $S \rightarrow AB$, $A \rightarrow a$, $B \rightarrow b$

ЛІВЕ ВИВЕДЕННЯ

Замінюється найлівіший нетермінал.

S

$\Rightarrow AB$

$\Rightarrow aB$ (замінили A)

$\Rightarrow ab$ (замінили B)

ПРАВЕ ВИВЕДЕННЯ

Замінюється найправіший нетермінал.

S

$\Rightarrow AB$

$\Rightarrow Ab$ (замінили B)

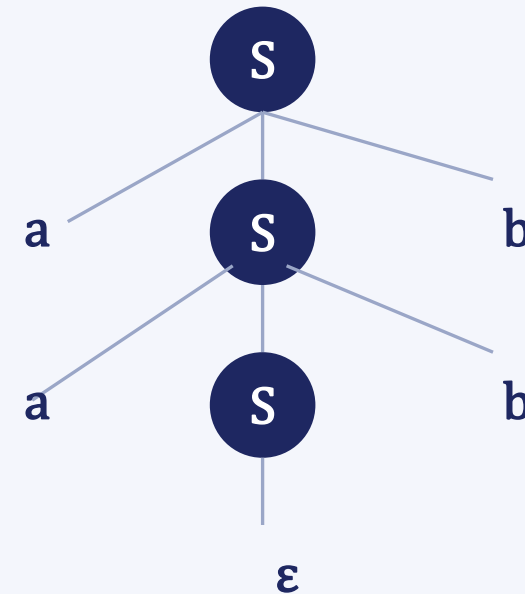
$\Rightarrow ab$ (замінили A)

Кінцевий результат однаковий («ab»), проте порядок заміни та проміжні ланцюжки відрізняються.

Структура дерева виведення

- 1 Корінь позначений початковим символом S .
- 2 Внутрішні вузли - нетермінали; дочірні вузли відповідають правій частині застосованого правила.
- 3 Листки дерева - термінальні символи (або ϵ).
- 4 Листки зліва направо утворюють виведене слово.

Приклад: $S \rightarrow aSb \mid \epsilon$, слово «aabb»



Листки зліва направо: $a, a, \epsilon, b, b \rightarrow$ «aabb»

Приклад

Формальні компоненти грамматики

Початковий нетермінал: S

Правила виводу (продукції):

$P1: S \rightarrow aSb$

$P2: S \rightarrow \varepsilon$ - порожнє слово

Крок виводу

Крок 1 (Старт): Беремо початковий символ S і застосовуємо продукцію $P1(S \rightarrow aSb)$:

(Перевірка: на даному етапі сформовано зовнішній каркас $a...b$)

Крок 2: До нетерміналу S всередині отриманого ланцюжка знову застосовуємо продукцію $P1(S \rightarrow aSb)$:

$$aSb \Rightarrow a(aSb)b = aaSbb$$

(Перевірка: додано другий рівень вкладеності $aa...bb$)

Крок 3 (Завершення): Замінюємо нетермінал S на порожнє слово ε , застосовуючи продукцію $P2(S \rightarrow \varepsilon)$:

$$aaSbb \Rightarrow aa \varepsilon bb = aabb$$

(Перевірка: нетермінали відсутні, отримано шукане слово $aabb$)

Загальний ланцюжок виводу

$$S \Rightarrow aSb \Rightarrow aaSbb \Rightarrow aa \text{ } bb$$

P1 P1 P2

Виведення ($\Rightarrow^*$)

Визначення

Послідовність прямих виведень $w_1 \Rightarrow w_2 \Rightarrow \dots \Rightarrow w_n$ записується як $w_1 \Rightarrow^* w_n$.

$\Rightarrow^*$ — рефлексивно-транзитивне замикання $\Rightarrow$ (тобто $w \Rightarrow^* w$ навіть без жодного кроку).

Приклад виведення ланцюжка aabb

За граматикою $G (N=\{S\}, T=\{a,b\}, P=\{S \rightarrow aSb, S \rightarrow ab\})$:

$$S \Rightarrow aSb \Rightarrow a(ab)b = aabb \quad \rightarrow \quad S \Rightarrow^* aabb \text{ (2 кроки)}$$

Сентенціальна форма (Sentential Form)

Сентенціальна форма — це будь-який ланцюжок символів α , який можна вивести з початкового нетермінала S за нуль або більше кроків застосування правил ($S \Rightarrow^* \alpha$)

Сентенціальна форма - будь-який ланцюжок $w \in (N \cup T)^*$, для якого $S \Rightarrow^* w$ (може містити нетермінали).

Сентенціальна форма, що складається лише з терміналів: $w \in T^*$.

Приклад

Мова, породжена граматикою G

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

$G (N=\{S\}, T=\{a,b\}, P=\{S \rightarrow aSb, S \rightarrow ab\})$ маємо $L(G) = \{a^n b^n \mid n \geq 1\}$.

Еквівалентність граматик

Визначення

Граматиками G_1 та G_2 називаються **еквівалентними**, якщо вони породжують одну і ту саму мову:

$$L(G_1) = L(G_2)$$

Приклад

$P_2 = \{S \rightarrow aSb, S \rightarrow \epsilon\}$ і $P_3 = \{S \rightarrow aAb, A \rightarrow aAb \mid \epsilon\}$ — різні за виглядом, але обидві породжують

$L = \{a^n b^n \mid n \geq 0\}$: граматиками еквівалентні.

А от $P_1 = \{S \rightarrow aSb, S \rightarrow ab\}$ породжує $\{a^n b^n \mid n \geq 1\}$ — іншу мову (без порожнього слова), тож P_1 і P_2 не еквівалентні.

Формальне визначення $L(G)$

$$S = \gamma_0 \Rightarrow \gamma_1 \Rightarrow \gamma_2 \Rightarrow \dots \Rightarrow \gamma_n = w, \quad n \geq 0$$

Скорочено записують: $S \Rightarrow^* w$ — «виводиться за скінченну (можливо, нульову) кількість кроків».

$$L(G) = \{ w \in T^* : S \Rightarrow^* w \}$$

Мова граматики - множина всіх термінальних слів (лише із символів T), які можна вивести із символу S за скінченну кількість застосувань правил P .

Неоднозначна граматика (ambiguous)

Якщо для одного слова можна побудувати кілька різних дерев виведення — граматика неоднозначна. Це ускладнює синтаксичний аналіз мов програмування.

2.7

ПРИКЛАД ВИВЕДЕННЯ

Повний розбір: мова $a^n b^n$

$G = (V, T, P, S)$, $V = \{S\}$, $T = \{a, b\}$, $P: S \rightarrow aSb, S \rightarrow ab$

Виведемо слово «aaabbbb»:

$\Rightarrow aSb$

застосували $S \rightarrow aSb$

$\Rightarrow aaSbb$

застосували $S \rightarrow aSb$ до внутрішнього S

$\Rightarrow aaabbbb$

застосували $S \rightarrow ab$ до останнього S

$S \Rightarrow^* aaabbbb$, тому $L(G) = \{ a^n b^n : n \geq 1 \}$

усі слова з рівною кількістю «a», за якими йде стільки ж «b»

Граматика арифметичних виразів

$V = \{S\}$, $T = \{a, +, *, (,)\}$, $P: S \rightarrow S+S \mid S*S \mid (S) \mid a$

Ліве виведення слова « $a+a*a$ »:

$S \Rightarrow S+S \Rightarrow a+S \Rightarrow a+S*S \Rightarrow a+a*S \Rightarrow a+a*a$

ВАРІАНТ 1

$S \Rightarrow S+S \Rightarrow \dots$

Спочатку виконується «+», потім «*» — трактування « $a + (a*a)$ ».

ВАРІАНТ 2

$S \Rightarrow S*S \Rightarrow \dots$

Інший порядок застосування правил дає інше дерево виведення для того самого слова.

Ця граматика неоднозначна — саме тому мови програмування проєктують граматики виразів так, щоб врахувати пріоритет операцій.



Чотири типи граматик

У 1956 році Ноам Хомський запропонував класифікувати граматики за ступенем обмеження на вигляд правил Р. Що менше обмежень — то ширший клас мов граматика здатна породити.

Тип 0	Граматики загального вигляду	$\alpha \rightarrow \beta, \alpha \neq \varepsilon$ (без інших обмежень)
-------	------------------------------	--

Тип 1	Контекстно-залежні (КЗ)	$\alpha A \beta \rightarrow \alpha \gamma \beta, \gamma \neq \varepsilon$ (довжина не зменшується)
-------	-------------------------	--

Тип 2	Контекстно-вільні (КВ)	$A \rightarrow \beta$ (ліва частина — один нетермінал)
-------	------------------------	--

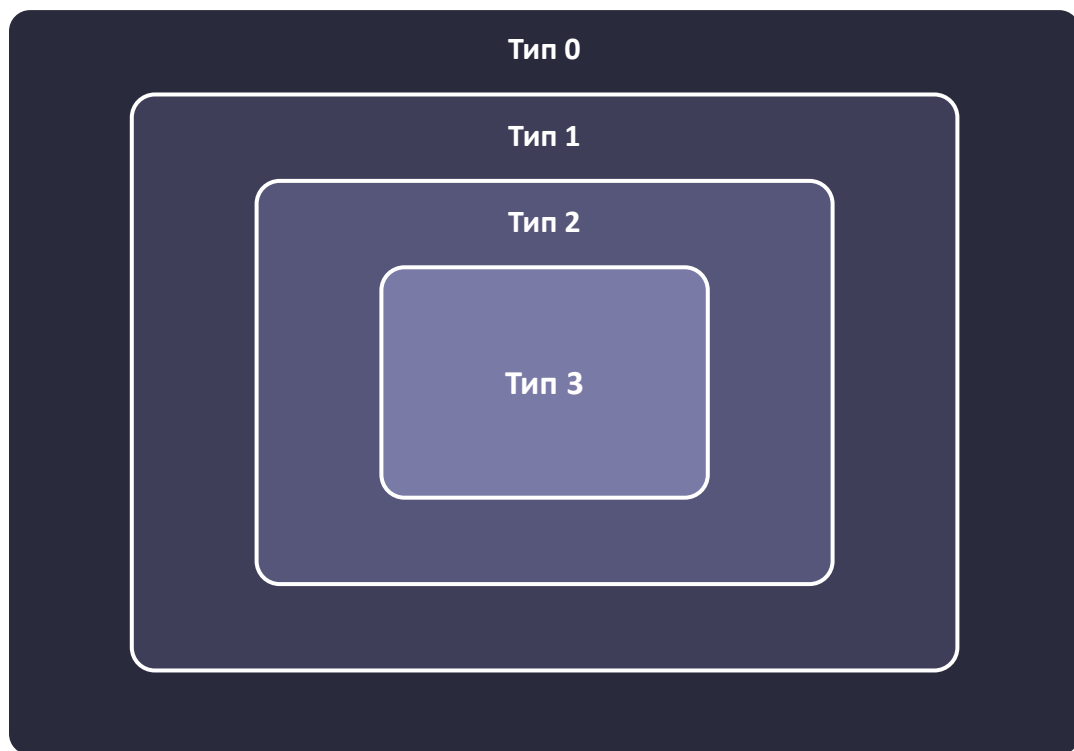
Тип 3	Регулярні	$A \rightarrow aB$ або $A \rightarrow a$ (право-/ліволінійні)
-------	-----------	---



Вкладеність класів мов

У 1956 році Ноам Хомський запропонував класифікувати граматики за ступенем обмеження на вигляд правил Р. Що менше обмежень — то ширший клас мов граматика здатна породити.

Тип 3 $\subset$ Тип 2 $\subset$ Тип 1 $\subset$ Тип 0



Відповідність класам автоматів

Тип 3 (регулярні) $\leftrightarrow$ скінченні автомати

Тип 2 (контекстно-вільні) $\leftrightarrow$ автомати з магазинною пам'яттю

Тип 1 (контекстно-залежні) $\leftrightarrow$ лінійно обмежені автомати

Тип 0 (загального вигляду) $\leftrightarrow$ машини Тюрінга



Регулярні та контекстно-вільні граматики

Тип 3 — регулярна граматика

$N = \{S\}, T = \{a, b\}$

$P = \{$

$S \rightarrow aS,$

$S \rightarrow b \}$

$L(G) = \{a^n b \mid n \geq 0\}$

Регулярна мова — усі правила праволінійні: $A \rightarrow aB$ або $A \rightarrow a$.

Тип 2 — контекстно-вільна граматика

$N = \{S\}, T = \{a, b\}$

$P = \{$

$S \rightarrow aSb,$

$S \rightarrow \epsilon \}$

$L(G) = \{a^n b^n \mid n \geq 0\}$

Контекстно-вільна мова — ліва частина кожного правила складається з одного нетерміналу S .



Контекстно-залежні та граматики загального вигляду

Тип 1 — контекстно-залежна граматика

Класичний приклад: $L = \{a^n b^n c^n \mid n \geq 1\}$.

Ця мова не є контекстно-вільною, але породжується контекстно-залежною граматиною.

Правила мають вигляд $\alpha A \beta \rightarrow \alpha \gamma \beta$ (заміна A залежить від контексту α, β) і ніколи не зменшують довжину ланцюжка.

Тип 0 — граматика загального вигляду

На відміну від типу 1, правила типу 0 можуть і зменшувати довжину ланцюжка (наприклад, $AB \rightarrow \epsilon$ — допустиме правило).

Це найширший клас: граматики загального вигляду породжують усі рекурсивно-перелічувані мови.

Серед них є мови, для яких не існує алгоритму розпізнавання.



Приклади граматик і мов

Мова типу 0:

$$L(G) = \{a^{2^n} \mid n \geq 1\}$$

P:{

$S \rightarrow aaCFD$

$F \rightarrow AFB \mid AB$

$AB \rightarrow bBA$

$Ab \rightarrow bA$

$AD \rightarrow D$

$Cb \rightarrow bC$

$CB \rightarrow C$

$bCD \rightarrow \varepsilon$ }

Мова типу 1:

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

P:{

$S \rightarrow aSBC \mid abC$

$CB \rightarrow BC$

$bB \rightarrow bb$

$bC \rightarrow bc$

$cC \rightarrow cc$

}



Приклади граматик і мов

Мова типу 2:

$$L(G) = \{(ac)^n (cb)^n \mid n > 0\}$$
$$P: \{ S \rightarrow aQb \mid accb$$
$$Q \rightarrow cSc$$
$$\}$$

Мова типу 3:

$$L(G) = \{\omega \perp \mid \omega \in \{a,b\}^+, \text{ де немає двох } a, \text{ що стоять поруч} \}$$
$$P: \{ S \rightarrow A \perp \mid B \perp$$
$$A \rightarrow a \mid Ba$$
$$B \rightarrow b \mid Bb \mid Ab$$
$$\}$$

Співвідношення між типами мов

- (1) кожна регулярна мова є КВ-мовою, але існують КВ- мови, що не є регулярними (наприклад, $L = \{a^n b^n \mid n > 0\}$).
- (2) кожна КВ- мова є КЗ- мовою, але існують КЗ-мови, що не є КВ-мовами (наприклад, $L = \{a^n b^n c^n \mid n > 0\}$).
- (3) кожна КЗ-мова є мовою типу 0.

Варто підкреслити, що якщо мова задана граматикою типу k , то це не означає, що не існує граматик типу k' ($k' > k$), що описує ту ж мову. Тому, коли говорять про мову типу k , звичайно мають на увазі максимально можливе k .



Приклад

КЗ- граматика $G_1 = (\{0,1\}, \{A,S\}, P_1, S)$ та

КВ- граматика $G_2 = (\{0,1\}, \{S\}, P_2, S)$,

де $P_1: S \rightarrow 0A1$ $P_2: S \rightarrow 0S1 \mid 01$

$0A \rightarrow 00A1$

$A \rightarrow \varepsilon$

описують ту ж саму мову $L = L(G_1) = L(G_2) = \{0^n 1^n \mid n > 0\}$.

Мову L називають КВ-мовою, тому що існує КВ-граматика, що її описує. Але вона не є регулярною мовою, тому що не існує регулярної граматики, що її описує.

Ключові тези

- Формальна граматика $G = (V, T, P, S)$ породжує мову набором правил, а не переліком слів.
- V — нетермінали, T — термінали, P — правила виведення, S — аксіома.
- Правило $\alpha \rightarrow \beta$ дозволяє замінювати підланцюжок α на β у процесі виведення.
- Ліве виведення замінює найлівіший нетермінал, праве — найправіший.
- Дерево виведення унаочнює структуру слова незалежно від порядку заміни.
- $L(G) = \{ w \in T^* : S \Rightarrow^* w \}$ — мова, породжена граматикою G .

Контрольні запитання

Перевірте свої знання та поглибте розуміння формальних граматик, розв'язавши наступні завдання:

1

Що таке сентенціальна форма і чим вона відрізняється від речення мови?

2

Наведіть граматику, яка породжує мову $\{a^n b^{2n} \mid n \geq 0\}$, і побудуйте виведення для $n = 2$.

3

У чому полягає різниця між контекстно-вільною та контекстно-залежною граматиною?

4

Чи можуть дві граматики з різною кількістю правил бути еквівалентними? Наведіть приклад.

5

Чому клас регулярних мов є власною підмножиною класу контекстно-вільних мов?