

ТЕОРІЯ ФОРМАЛЬНИХ МОВ

Лекція 1

Вступ до теорії формальних мов

Алфавіт · слово · мова · операції над словами й мовами · зірка Кліні



ПЛАН ЛЕКЦІЇ

1

Поняття формальної мови

Чим формальна мова відрізняється від природної

2

Алфавіт, слово, порожнє слово

Базові будівельні елементи теорії

3

Операції над словами та мовами

Конкатенація, реверс, теоретико-множинні операції

4

Степінь слова і мови

w^n та L^n — n -разова конкатенація

5

Зірка та плюс Кліні

L^* і L^+ — ітерація мови

6

Приклади формальних мов

Дужки, паліндроми, ідентифікатори

Формальна і природна мова

Синтаксис мови – це набір правил, які визначають допустимі конструкції мови. Синтаксис визначає «форму» мови – задає набір ланцюжків символів, які належать мові.

Семантика мови – це розділ мови, що визначає значення речень мови. Семантика визначає «зміст» мови – задає значення для всіх допустимих ланцюжків мови. Семантика для більшості мов визначається неформальними методами.

Лексика – це сукупність слів мови. **Слово** або лексична одиниця (**лексема**) мови – це конструкція, яка складається з елементів алфавіту мови і не містить у собі інших конструкцій.

Формальна і природна мова

Теорія формальних мов вивчає мови, задані точним математичним способом - без опори на семантику чи «змістовне» розуміння. Належність слова до мови визначається формальним критерієм, а не змістом.

ПРИРОДНА МОВА

«Кім спить на килимі»

Речення правильне, оскільки воно граматично узгоджене й має сенс — критерій спирається на зміст і контекст.

ФОРМАЛЬНА МОВА

«0110»

Слово належить мові двійкових чисел, кратних 2, якщо останній символ дорівнює 0 — критерій суто формальний, без «сенсу».

Що таке формальна мова?

Галузь науки

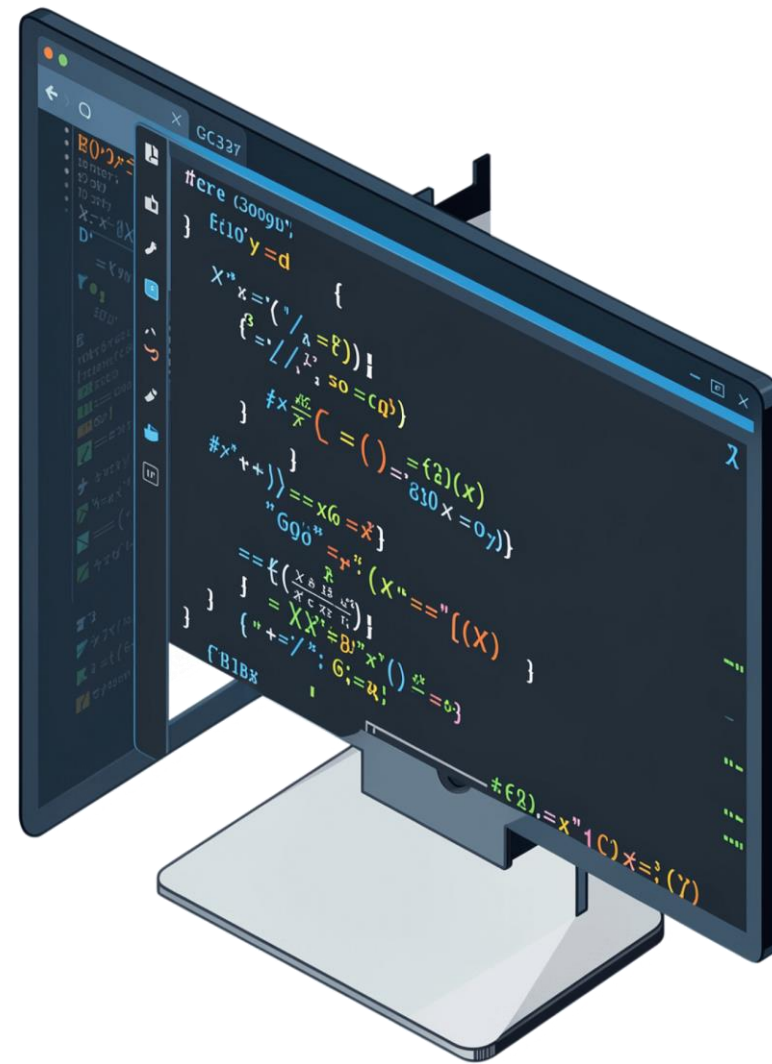
Розділ дискретної математики та теоретичної інформатики, що вивчає математичні моделі мов

Суворий синтаксис

На відміну від природних мов, формальна мова має чітко визначені правила побудови

Застосування

Мови програмування, компілятори, протоколи зв'язку, математичні моделі обчислень



Формальна і природна мова

Алфавіт - це кінцева множина символів, а символи, що входять у множину називаються **буквами** алфавіту.

Наприклад, алфавіт $A = \{2, b, c, +, !\}$ містить 5 букв, а алфавіт $B = \{00, 01, 10, 11\}$ містить 4 букви, кожна з яких складається з двох символів.

Послідовність букв алфавіту називається **словом** чи **ланцюжком** у цьому алфавіті. Число букв, що входять у слово, називається його **довжиною**.

Ланцюжок, який не має ні одного символу, називається **пустий ланцюжок** і позначається символом ϵ .

Довжину ланцюжка α будемо позначати $|\alpha|$. Довжина $|\epsilon| = 0$.

Алфавіт Σ

Алфавіт (Σ) — скінченна не порожня множина символів, з яких будуються слова.

$$\Sigma = \{a_1, a_2, \dots, a_n\}, \text{ де } n - \text{скінченне число символів}$$

$$\Sigma_1 = \{0, 1\}$$

Двійковий алфавіт

$$\Sigma_2 = \{a, b, c\}$$

Три латинські літери

$$\Sigma_3 = \{A, Б, \dots, Я\}$$

Українська абетка

$$\Sigma = \{if, else, while, id, num\}$$

Алфавіт лексем мови програмування

$$\Sigma = \{ (,), +, *, x, y \}$$

Алфавіт для арифметичних виразів

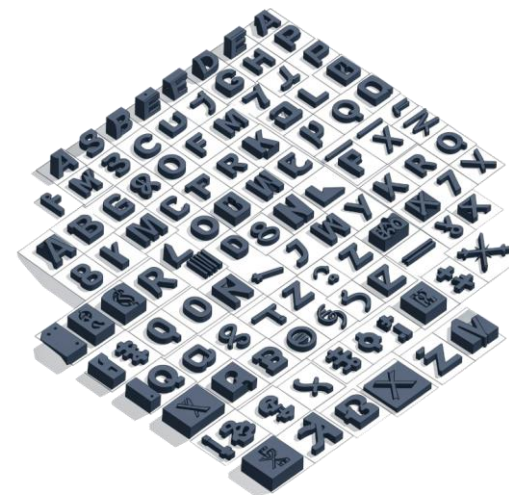
Слово (ланцюжок)

Слово (ланцюжок) - скінченна послідовність символів алфавіту, записаних без роздільників.

Над $\Sigma = \{a, b\}$ словами є: $a, b, ab, ba, aab, abba \dots$

Порожнє слово ε (епсilon) — слово без жодного символу; $|\varepsilon| = 0$.

Довжина слова $|w|$ — кількість його символів ($|abba| = 4$).



1 Слова над $\Sigma = \{a, b\}$

$a, b, ab, aab, baaba$ - усі є
словами над цим алфавітом

2 Довжина: $w = abac$

$|w| = 4$ - чотири символи у слові

3 Довжина: $w = 0110$

$|w| = 4$ - чотири двійкові
символи

Порожнє слово (ϵ)



Визначення

Порожнє слово (ϵ) - слово, яке не містить жодного символу. Його довжина: $|\epsilon| = 0$



Універсальність

Порожнє слово існує над **будь-яким** алфавітом - це фундаментальна властивість



Важлива різниця

$\epsilon \neq \emptyset$ - порожнє слово $\neq$ порожня множина.
Множина $\{\epsilon\}$ містить **один** елемент, а $\emptyset$ не містить жодного



Найпоширеніша помилка студентів:

ототожнення ϵ (порожнє слово) з $\emptyset$ (порожня множина). Це принципово різні об'єкти!

Конкатенація та реверс слів

КОНКАТЕНАЦІЯ

Якщо u та v — слова над Σ , то uv - слово, отримане дописуванням v одразу після u .

Некомутативна: $uv \neq vu$

$u = \text{«ab»}, v = \text{«ba»}$

$uv = \text{«abba»}$

$vu = \text{«baab»}$

РЕВЕРС (ОБЕРНЕННЯ)

Якщо $w = a_1a_2\dots a_n$, то $w^R = a_na_{n-1}\dots a_1$ - слово, записане у зворотному порядку.

$w = \text{«programa»}$

$w^R = \text{«amargorp»}$

Конкатенація слів

Формально:

якщо $u = a_1a_2\dots a_n$ і $v = b_1b_2\dots b_m$, то

$$uv = a_1a_2\dots a_nb_1b_2\dots b_m$$

Властивості

- $|uv| = |u| + |v|$ - довжини додаються
- $(uv)w = u(vw)$ - асоціативність
- $\epsilon w = w = w\epsilon$ - нейтральний елемент
- $uv \neq vu$ - некомутативність (у загальному випадку)

Приклади

$$u = ab, v = cab$$

$$uv = \mathbf{abcab}, vu = cabab$$

$$u = 01, v = 10$$

$$uv = \mathbf{0110}, vu = 1001$$

$$u = \text{hello}, v = \text{world}$$

$$uv = \mathbf{helloworld}$$

$$u = abc, v = \epsilon$$

$$uv = \mathbf{abc} \text{ (нейтральний елемент)}$$



Вправа: якщо $u = xy$, $v = yz$, знайдіть uv та $|uv|$

Степінь слова

Визначення

Степінь слова w^k - k -разова конкатенація слова w самого із собою.

Рекурсивне визначення

01

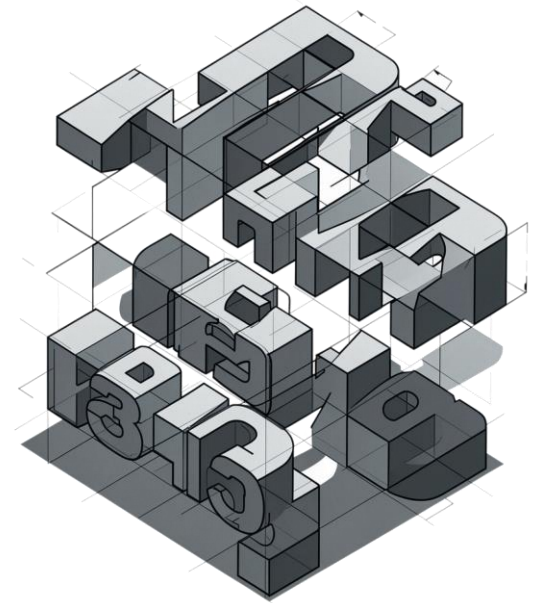
Базовий випадок

$w^0 = \epsilon$ для будь-якого слова w

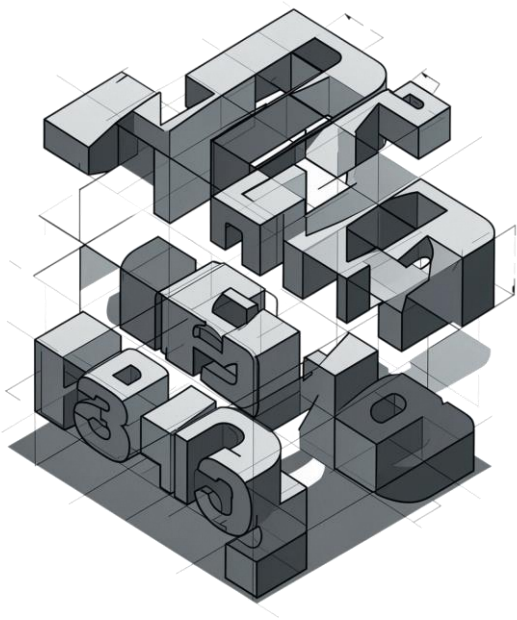
02

Рекурсивний крок

$w^{(k+1)} = w^k \cdot w$ для $k \geq 0$



Вправи: обчисліть $(abc)^3$ та знайдіть $|w^5|$, якщо $|w| = 3$



Обернення (реверс) слова

Для слова $w = a_1a_2\dots a_n$ його
обернення - це $wR = a_n\dots a_2a_1$

Властивість 1

$\epsilon R = \epsilon$ - обернення порожнього слова є порожнім словом

Властивість 2

$(uv)R = vR \cdot uR$ - обернення конкатенації

Приклади

- $w = abcd \rightarrow wR = \mathbf{dcba}$
- $w = 0110 \rightarrow wR = \mathbf{0110}$ (паліндром!)

Перевірка властивості (uv)

$$R = vR \cdot uR$$

$$u = ab, v = cd \rightarrow uv = abcd \rightarrow (uv)R = dcba$$

$$vR = dc, uR = ba \rightarrow vR \cdot uR = \mathbf{dcba} \checkmark$$

Множина всіх слів: Σ^* та Σ^+

Замикання Кліні (Σ^*)

Σ^* - множина всіх слів над алфавітом Σ , включаючи порожнє слово ϵ .

$$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \dots = \bigcup \Sigma^n \ (n \geq 0)$$

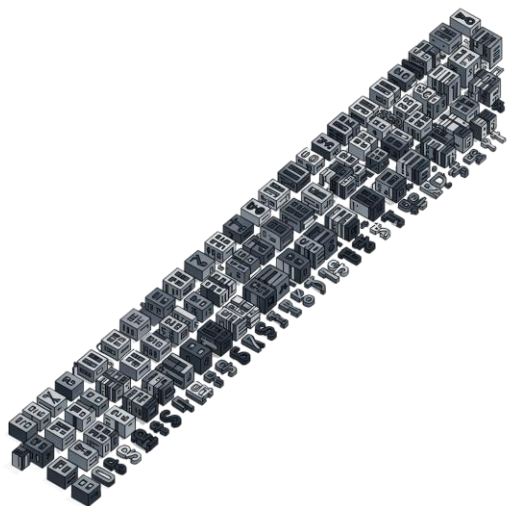
Позитивне замикання (Σ^+)

Σ^+ - множина всіх непорожніх слів над Σ .

$$\Sigma^+ = \Sigma^* \setminus \{\epsilon\} = \Sigma^1 \cup \Sigma^2 \cup \dots$$

Приклад для $\Sigma = \{a, b\}$

- $\Sigma^0 = \{\epsilon\}$
- $\Sigma^1 = \{a, b\}$
- $\Sigma^2 = \{aa, ab, ba, bb\}$
- $\Sigma^* = \{\epsilon, a, b, aa, ab, ba, bb, aaa, \dots\}$



Приклад

Нехай алфавіт складається з двох символів: $\Sigma = \{\alpha, \beta\}$.

Тоді замикання Кліні Σ^* включає:

$$\Sigma^0 = \{\varepsilon\}$$

$$\Sigma^1 = \{\alpha, \beta\}$$

$$\Sigma^2 = \{\alpha\alpha, \alpha\beta, \beta\alpha, \beta\beta\}$$

$$\Sigma^3 = \{\alpha\alpha\alpha, \alpha\alpha\beta, \alpha\beta\alpha, \alpha\beta\beta, \beta\alpha\alpha, \beta\alpha\beta, \beta\beta\alpha, \beta\beta\beta\}$$

...і так далі до безкінечності.

Отже, $\Sigma^* = \{\varepsilon, \alpha, \beta, \alpha\alpha, \alpha\beta, \beta\alpha, \beta\beta, \alpha\alpha\alpha, \dots\}$.

1.4

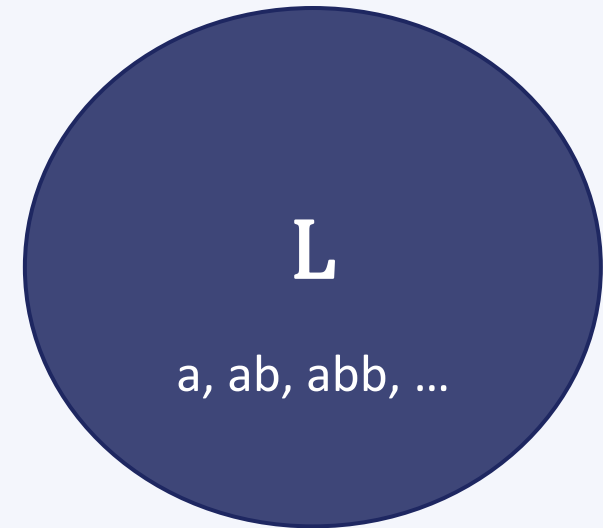
МОВА НАД АЛФАВІТОМ

Мова $L \subseteq \Sigma^*$

Мова над алфавітом Σ - довільна підмножина множини Σ^* , тобто довільна множина слів над Σ . Мова може бути порожньою ($\emptyset$) або містити лише ϵ - це різні мови: $\emptyset \neq \{\epsilon\}$.

$$L \subseteq \Sigma^*$$

Приклад: над $\Sigma = \{a, b\}$ мова $L_1 = \{a, ab, abb, abbb, \dots\}$ - усі слова, що починаються з «a», за яким слідує довільна кількість «b».

 Σ^* 

усі слова над Σ

Теоретико-множинні операції

Оскільки мови є множинами слів, до них застосовні звичайні операції над множинами:

ОБ'ЄДНАННЯ

$$L_1 \cup L_2$$

$\{a, aa, aaa, bb\}$

ПЕРЕТИН

$$L_1 \cap L_2$$

$\{aa\}$

РІЗНИЦЯ

$$L_1 \setminus L_2$$

$\{a, aaa\}$

ДОПОВНЕННЯ

$$\bar{L} = \Sigma^* \setminus L$$

усе, крім L

Приклад: $L_1 = \{a, aa, aaa\}$, $L_2 = \{aa, bb\}$

Конкатенація мов: $L_1 L_2 = \{ uv : u \in L_1, v \in L_2 \}$

$$L_1 = \{a, ab\}, L_2 = \{b, c\} \rightarrow L_1 L_2 = \{ab, ac, abb, abc\}$$

Степінь слова w^n та мови L^n

СТЕПІНЬ СЛОВА

$$w^0 = \varepsilon, \quad w^n = w \cdot w^{n-1}$$

$$w = \text{«ab»}$$

$$w^0 = \varepsilon, \quad w^1 = ab$$

$$w^2 = abab, \quad w^3 = ababab$$

СТЕПІНЬ МОВИ

$$L^0 = \{\varepsilon\}, \quad L^n = L \cdot L^{n-1}$$

$$L = \{a, b\}$$

$$L^2 = \{aa, ab, ba, bb\}$$

L^3 містить 8 слів довжини 3

Степінь n визначає, скільки разів слово (або мова) конкатенується саме із собою; L^0 завжди дорівнює $\{\varepsilon\}$ за означенням.

Ітерація мови: L^*

$$L^* = L^0 \cup L^1 \cup L^2 \cup L^3 \cup \dots = \bigcup L^n, n = 0, 1, 2, \dots$$

Зірка Кліні завжди містить ϵ , навіть якщо саме L не містить порожнього слова.

$$L = \{a\}$$

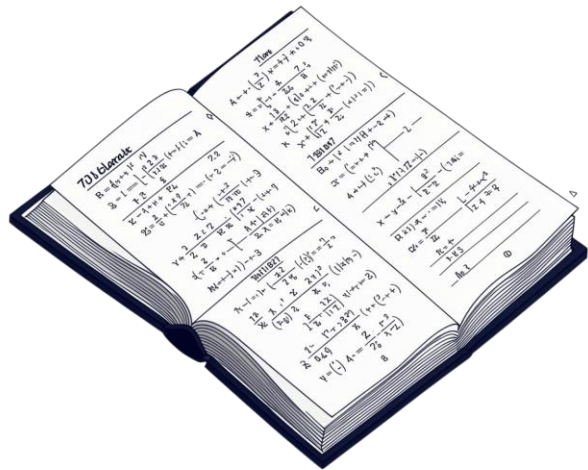
$$L^* = \{\epsilon, a, aa, aaa, aaaa, \dots\}$$

$$L = \{ab, ba\}$$

$$L^* = \{\epsilon, ab, ba, abab, baab, \dots\}$$

Плюс Кліні L^+ — те саме, але без нульового степеня (без обов'язкового ϵ):

$$L^+ = L^1 \cup L^2 \cup \dots, \quad L^* = L^+ \cup \{\epsilon\}$$



Підсумок

1

Алфавіт (Σ)

Скінченна непорожня множина символів — основа будь-якої формальної мови

2

Слово та ϵ

Скінченна послідовність символів; порожнє слово ϵ має довжину 0

3

Операції над словами

Конкатенація, степінь та обернення — базові інструменти побудови мов

4

Σ^* та мови

Мова $L \subseteq \Sigma^*$; над мовами визначені теоретико-множинні операції

Граматика

Граматика – це опис способу побудови речень деякої мови. Інакше кажучи граматика – це математична система, що визначає мову.

Фактично, визначив граматику мови, ми вказуємо правила породження ланцюжків символів, що належать цієї мові.

Граматику мови можна описати різними способами, але для синтаксичної частини мови програмування припустимо використання формального опису граматики, що побудоване на основі системи правил (або продукції).

Правило (або продукція) – це впорядкована пара ланцюжків символів (α, β) . В правилах важливий порядок ланцюжків, тому їх часто записують у виді $\alpha \rightarrow \beta$ ($\alpha ::= \beta$), Така запись читається як « α породжує β » або « β по визначенню є α ».

Приклади формальних мов

Скінченні та нескінченні

Скінченна мова: L_1

$L_1 = \{a, aa, aba\}$ над $\Sigma = \{a, b\}$

Містить рівно 3 слова.

Нескінченна мова
парної довжини: L_2

$L_2 = \{w \in \Sigma^* \mid |w| \text{ парна}\}$ над
 $\Sigma = \{a, b\}$

Слова L_2 : $\epsilon, aa, ab, ba, bb,$
 $aaaa, aaab, \dots$

Контекстно-вільна мова: L_3

$L_3 = \{a^n b^n \mid n \geq 0\}$ (баланс символів)

Слова L_3 : $\epsilon, ab, aabb, aaabbb, aaaabbbb, \dots$

Регулярні та складніші

Мова $a^n b^m$: L_4

$L_4 = \{a^n b^m \mid n, m \geq 1\} =$
 $\{a\}^+ \{b\}^+$

Слова L_4 : $ab, aab, abb, aaabb,$
 $aabbb, \dots$

Властивість: кількість a та b
незалежна, але кожного
має бути щонайменше
один.

Мова $0^n 1^{2^n}$: L_5

$L_5 = \{0^n 1^{2^n} \mid n \geq 0\}$

Слова L_5 : $\epsilon, 011, 001111,$
 $000111111, \dots$

Чотири типові мови



Парні дужки

$$\Sigma = \{ (,) \}$$

«(())», «()()» $\in L$; «)()» $\notin L$



Паліндроми

$$w = w^R$$

«aba», «abba» $\in L$; «ab» $\notin L$



Рівна к-сть символів

$$\#a(w) = \#b(w)$$

«ab», «abba» $\in L$; «aab» $\notin L$



Ідентифікатори

літера/_ ... літери/цифри/_

«index», «_temp1» $\in L$; «2value» $\notin L$

ПІДСУМОК ЛЕКЦІЇ 1

Ключові тези

- Формальна мова визначається точно, без опори на семантику.
- Алфавіт Σ — скінченна множина символів; слово — послідовність символів; ε — порожнє слово.
- Над словами й мовами визначені операції: конкатенація, реверс, об'єднання, перетин, різниця, доповнення.
- Степінь w^n , L^n — n -разова конкатенація із собою.
- Зірка Кліні L^* завжди включає ε ; L^+ — без обов'язкового ε .
- Формальні мови описують синтаксис мов програмування, протоколів, форматів даних.