Теорія формальних мов і граматик - це велика область математики, що примикає до алгебри, математичної логіки та теорії автоматів. Введемо в ужиток основні поняття, які будуть використані у визначенні формальної мови.

Алфавіт — кінцева множина символів. Термін символ слід розуміти тут у найширшому значенні. Це може бути буква, цифра або розділовий знак. Але символом можна вважати і будь-який інший знак, що розглядається як щось неподільне - службове слово мови програмування, ієрогліф і т. д. Позначатимемо алфавіти буквою Σ (сигма).

Приклади алфавітів:

Σ1 = {0, 1}

Σ2 = {a, b, с}

Σ3 = {A, В, С,..., Z, а, b, с,..., z, 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, +, -, \*,..., div, ..., program }

Алфавіт - це множина, тому при перерахуванні його елементів використані фігурні дужки, як це заведено в математиці. Алфавіт Σ1 містить два символи, алфавіт Σ2 - три. Під Σ3 мається на увазі алфавіт мови Паскаль.

Ланцюжок над алфавітом Σ - довільна кінцева послідовність символів із Σ.

Приклади ланцюжків над алфавітом Σ2:

**α** = аbbса

**β** = ab

**γ** = ba

**δ** =с

Ланцюжки позначатимемо грецькими літерами.

Порожній ланцюжок - ланцюжок, що не містить символів (що містить нуль символів). Позначається буквою **ε**. Якщо **α** і **β** — ланцюжки, запис **αβ** означає їх конкатенацію (склеювання), тобто **αβ** — це ланцюжок, утворений приписуванням до ланцюжка **α** ланцюжка **β** справа. Якщо **α** — ланцюжок, то **αn** означає ланцюжок, утворений n-кратним повторенням ланцюжка **α**:

**α**n = **α α** ... **α** **α**.

В окремому випадку, якщо а - символ, то аn = а а ... а а.

Позначатимемо Σ\* — (нескінченна) множина усіх ланцюжків над алфавітом Σ, включаючи порожній ланцюжок; Σ+ - множина всіх ланцюжків над алфавітом Σ, не включаючи порожнього ланцюжка. Наприклад, якщо Σ1 = {0, 1}, то Σ1\* є безліч всіх ланцюжків, які можуть бути складені із символів 0 і 1.

У цю множину входять порожній ланцюжок, всі ланцюжки, що складаються з одного символу, всі ланцюжки, що складаються з двох символів, і т. д.: Σ1\* = { **ε**, 0,1, 00, 01,10,11, 000, 001,...}.

Σ\* = Σ+U { **ε** }, де U — знак операції об’єднання множин.

Тепер можна дати визначення формальної мови.

**Мовою** над алфавітом Σ називається довільна множина ланцюжків, складених із символів Σ. Будемо позначати мову над алфавітом (з алфавітом) Σ - **L(Σ)** або просто **L**, якщо алфавіт ясний з контексту.

Таким чином, йдеться про те, що мова - це деяке, тим чи іншим чином певне, підмножина багатьох ланцюжків, які можуть бути побудовані з символів даного алфавіту. L(Σ) ⊆ Σ\*



Мова - підмножина множин всіх ланцюжків над алфавітом Σ

Ланцюжки, що належать мові, називають також *пропозиціями* мови.

Зазначимо, що множина ланцюжків Σ \* завжди є нескінченною, тоді як множина ланцюжків, що утворюють мову, може бути і скінченною. Практичний інтерес становлять, звичайно, мови, що містять безліч ланцюжків. До таких мов належать і мови програмування.

**Приклади мов**

**Приклад 1.** Визначимо мову

L1 = {anbn | n ≥ 0}, використовуючи прийняту в теорії множин нотацію, як множину всіх ланцюжків, що спочатку містять деяку кількість символів а, а потім таку ж кількість символів b. Зауважимо, що L1 включає і порожній ланцюжок, оскільки n може дорівнювати нулю.

Записане раніше правило, що визначає мову L1 поділяє всі ланцюжки над алфавітом {а, b}, тобто які складаються із символів а і b, на які належать L1 і не належать їй.

Приклади ланцюжків, що належать мові:

ε ∈ L1 — порожній ланцюжок належить L1;

ab ∈ L1 — ланцюжок з однієї літери а, за якою слідує b;

aaabbb ∈ L1.

Ланцюжки, що не належать мові L1.

aaab ∉ L1 — неоднакова кількість символів а та b

abba ∉ L1 — порядок проходження символів не відповідає визначенню L1

**Приклад 2.** Мова L2 = {anbncn | n ≥ 0}

- множина всіх ланцюжків, що містять спочатку деяку (можливо нульову) кількість символів а, потім таку ж кількість символів b, потім стільки ж символів с.

Наприклад,

aaabbbccc ∈ L2,

в той час як

aaabbccc ∉ L2.

Не завжди вдається визначити мову, особливо якщо йдеться про мови, що становлять практичний інтерес, використовуючи нотацію, застосовану щодо L1 і L2. Значна частина наступного матеріалу буде присвячена розгляду граматик, що породжують, що дозволяють компактно і однозначно визначити великий клас формальних мов. Поки ж дамо словесний опис деяких мов, що представляють інтерес, у наступних прикладах.

**Приклад 3**. Розглянемо мову правильних дужних виразів, складених тільки з круглих дужок, відому також як мова Дика. Позначимо ії L3

Алфавіт мови Діка - це множина з двох символів - дужок, що відкриває «(» і закриває «)»: Σ = { (, ) }. Ланцюжки, що містять правильно розставлені дужки, належать мові Діка, всі інші послідовності круглих дужок, що відкривають і закривають, — ні.

Наприклад:

(())()() ∈ L3;

()(()))( ∉ L3.

**Приклад 4**. Мова L4 - множина всіх ланцюжків, що містять однакову кількість символів а та b. Незважаючи на простий «пристрій», задати мову L4 формулою, подібною до формул для L1, або L2, виявляється важко.

Можна помітити, що розглянута раніше мова L1 є підмножиною мови L4: L1 ⊆ L4, оскільки будь-який ланцюжок, що належить L1, належить і мові L4. Але не навпаки.

aabb ∈ L1, aabb ∈ L4,

abba ∈ L4, але abba ∉ L1.

**Приклад 5**. Як мову L5 розглянемо багато всіх правильних арифметичних виразів мови Паскаль, складених із символів алфавіту

Σ5 = {а, b, с, +, -, \*, /, (, )}.

Наприклад,

а\*(b + с) ∈ L5,

але

 c++ ∉ L5.