Лекція №7

Регулярні вирази

Визначення

Регулярний вираз **R** задає (визначає) мову **L(R)**.
Мови, які можуть бути задані регулярними виразами, називаються регулярними множинами або регулярними (автоматними) мовами.

Регулярний вираз над алфавітом **T** визначається наступними правилами:

1. **ε** є регулярним виразом, що визначає множину **{ ε }**, тобто множину, що містить порожній рядок.
2. Якщо a є символом з алфавіту **T,** то цей же символ a є простим регулярним виразом, що визначає множину **{a}**, тобто множину, що містить рядок **a**. У конкретних записах суть позначення **"a"** (регулярний вираз, рядок або символ) зрозуміла з контексту.
3. Якщо A і B – регулярні вирази, що визначають мови **L(A)** і **L(B),** тоді
    1) **A | B** (інше позначення **A + B**) є регулярним виразом, що визначає мову (множину) **L(A)** ∪ **L(B).**
    2) **AB** є регулярним виразом, що визначає мову (множину) **L(A) L(B)**, тобто після речення мови **A** безпосередньо слідує речення мови **B**.
    3) **A*** є регулярним виразом, що визначає мову (множину) **(L(A)) ***.

Тобто регулярний вираз характеризується трьома операціями:
1) альтернатива;
2) конкатенація;
3) ітерація.

Алфавіт регулярного виразу складається з наступних елементів:
1) a, b, c – рядки;
2) ε – порожній рядок;
3) ∅ – порожня множина.

Якщо E_1 і E_2 – регулярні вирази, тоді:
1. Альтернатива двох регулярних виразів E_1 і E_2 позначається:
$E = E_1 \mid E_2$ або $E = E_1 + E_2$
2. Конкатенація двох регулярних виразів E_1 і E_2 позначається:
$E = E_1 E_2$
3. Ітерація регулярного виразу E_1 – це багатократне його повторення від 0 до ∞ (можливо жодного разу) і позначається:
$E = E_1^* = \varepsilon \mid E_1 \mid E_1E_1 \mid E_1E_1 \ldots E_1$.
Ітерація регулярного виразу E_1 також може позначатися як $\{E_1\}$, тобто записи E_1^* та $\{E_1\}$ є еквівалентними.

Прийнято наступні домовленості:

1. Унарний оператор * має вищий пріоритет і є лівоасоциативним.
2. Конкатенація має другий пріоритет і є лівоасоціативною.
3. Альтернатива (об'єднання) має нижчий пріоритет і є лівоасоциативною.

За таких домовленостей наступні записи є еквівалентними:

(a) | ((b)* (c)) = a | b* c.

Обидва вирази визначають (задають) множину рядків, яка є або єдиним символом a, або декількома символами b (можливо жодного), за якими слідує єдиний символ c.

Два регулярні вирази A і B називаються **еквівалентними** (A = B), якщо вони задають одну і ту ж мову.

Приклади. Розглянемо прості регулярні вирази і мови, що ними визначаються, на алфавіті

T = {a, b}.

1. Регулярний вираз a | b задає мову (множину) {a, b}.
2. Регулярний вираз (a | b) (a | b) = aa | ab | ba | bb задає мову (множину) {aa, ab, ba, bb}, тобто множину всіх рядків з a і b довжиною в два символи.
3. Регулярний вираз a* задає мову (множину) всіх рядків з будь-якого числа символів a (можливо жодного), тобто { ε, a, aa, aaa ...}.
   Регулярний вираз (a | b)* = (a* b*)* задає мову (множину) всіх рядків, що містять декілька екземплярів a і b (можливо жодного), тобто множину всіх рядків, які можна скласти з a і b.

Приклади регулярних виразів, що задають конструкції мов програмування:

1. Рядки цілих чисел без знаку (мінімум одна цифра):

   dd*, де d = {0, 1, ..., 9}

2. Цілі із знаком або без знаку (мінімум одна цифра):

   (+ | – | ε)dd*

3. Ідентифікатор:

   l (l | d)*, де l = {A, B, ..., Z, a, b, ..., z}
   d = {0, 1, ..., 9}

У підрозділі «Регулярні вирази» символ «d» для зручності позначає не перехідну функцію, а множину цифр.

Основні тотожності.

Нехай A, B, C, E – регулярні вирази, тоді мають місце наступні тотожності:

1. A | B = B | A або A + B = B + A – комутативність альтернативи.
2. ∅* = ε – тобто ітерацією порожньої множини є порожній рядок.
3. A | (B | C) = (A | B) | C або A + (B + C) = (A + B) + C – асоціативність альтернативи.
4. A (B C) = (A B) C – асоціативність конкатенації.
5. A (B | C) = A B | A C або A (B + C) = A B + A C – дистрибутивність конкатенації над альтернативою.
6. Eε = εE = E – ε є «одиничним» елементом по відношенню до конкатенації.
7. E + ∅ = E – ∅ є «нульовим» елементом по відношенню до альтернативи.
8. (E | ε)* = E*.
9. E | E* = E*.
10. E** = E*.
11. Якщо E = A E | B, то E = A* B.

Перетворення регулярних виразів до скінченного автомата

Нехай E, E_1, E_2 – регулярні вирази.

ε – порожній рядок;

a – рядок регулярного виразу;

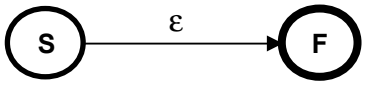
S – початковий стан автомата;

F – завершальний стан автомата.

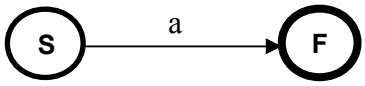
Тоді перехід від регулярного виразу до кінцевого автомата буде наступним:

1. ***Якщо*** $E = \varnothing$

2. $E = \varepsilon$

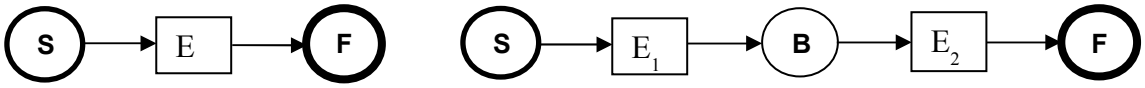


3. $E = a$



4. $E = E_1 + E_2$

5. $E = E_1 E_2$

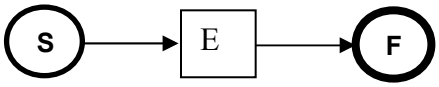


6. $E = E_1^*$

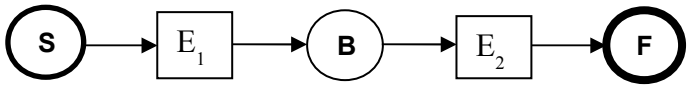
Приклад. Побудова автомата за регулярним виразом.
Побудуємо автомат для регулярного виразу **E = (+ | – | ε) d d***.
Позначимо частину (+ | – | ε) регулярного виразу як E_1, а частину d d* – як E_2, тобто

E_1 = (+ | – | ε),
E_2 = d d*.

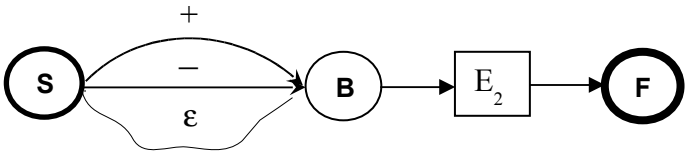
0.



1.



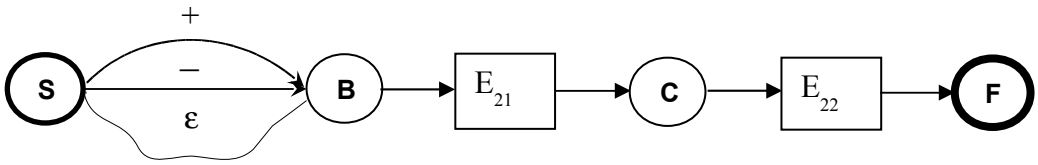
2.



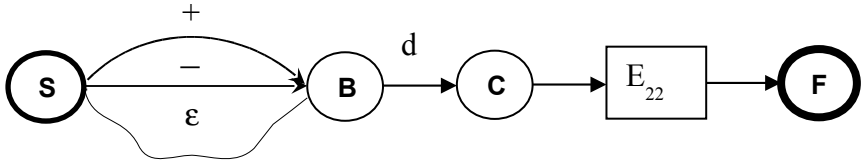
3. Позначимо частину **d** регулярного виразу E_2 як E_{21}, а частину d* – як E_{22}, тобто

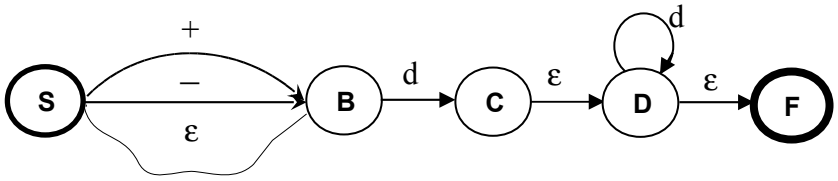
E_{21} = **d**,
E_{22} = **d***.

4.



5.





Регулярні визначення

Для зручності запису регулярним виразам можна давати імена і використовувати ці імена як символи в інших регулярних виразах.

Сукупність декількох (одного або більше) регулярних виразів, яким були дані імена, називається **регулярним визначенням**, загальний вид якого такий:

$D_1 \rightarrow R_1$
$D_2 \rightarrow R_2$
...
$D_n \rightarrow R_n$,

де D_i – імена регулярних виразів R_i.

Відмітимо, що в регулярних виразах R_i можуть використовуватися вже визначені раніше імена D_i, тобто кожне R_i визначене на множині

$T \cup \{D_1, D_2, \ldots, D_{i-1}\}$

Щоб відрізняти імена регулярних визначень від символів, записуватимемо ці імена великими буквами.

Приклад 1. Регулярне визначення ідентифікатора.
LETTER → A | B | ... | Z | a | b | ... | z
DIGIT → 0 | 1 | 2 | ... | 9
ID → LETTER (LETTER | DIGIT)*

Приклад 2. Ціле число.
DIGIT → 0 | 1 | 2 | ... | 9
INTNUM → DIGIT DIGIT*

Додаткові позначення в регулярних виразах

1. Унарний постфіксний оператор + означає "один або більше екземплярів".

Якщо R – регулярний вираз, що визначає мову L(R), то R^+ є регулярним виразом, що визначає мову $(L(R))^+$. Оператор $^+$ має той же пріоритет і асоціативність, що й оператор *.

Мають місце дві тотожності:

1) $R^* = R^+ | \varepsilon$
2) $R^+ = R R^*$

Використовуючи цей оператор визначення цілого числа можна записати так:
DIGIT → 0 | 1 | 2 | … | 9
INTNUM → DIGIT $^+$

2. Унарний постфіксний оператор ? означає "один екземпляр або жодного екземпляра".

Позначення R? є скороченим записом R | ε.

Якщо R – регулярний вираз, то R? – регулярний вираз, що описує мову L(R) U {ε}.

3. Класи символів.

Для скороченого запису деякої множини символів можна використовувати позначення у вигляді **класів символів**.

Клас символів [abc] позначає регулярний вираз a | b | c, тобто

[abc] = a | b | c,

де a, b, c – символи алфавіту.

Клас символів [a-z] позначає регулярний вираз a | b | .... | z, тобто

[a-z] = a | b | ... |z,

де a, b … z – символи алфавіту.

Приклад. Регулярний вираз ідентифікатора з використанням класів символів:

[A-Za-z] [A-Za-z0-9]*

Обмеженість регулярних виразів

Регулярні вирази мають обмежені описові можливості, тому не всі мови можуть бути описані регулярними виразами.

Характерні конструкції, які не можуть бути описані регулярними виразами:

1. Рядки зі збалансованими символами.
   Наприклад, рядок, який завжди містить однакове число відкриваючих і закриваючих дужок.
2. Рядки з повторним входженням одного і того ж підрядка.
   Наприклад, $\alpha\ \beta\ \alpha$.
   Більш того, така конструкція не може бути описана навіть контекстно-вільною граматикою.
3. Рядки, в яких один з символів обчислюється по інших символах того ж рядка.
   Наприклад, так звані рядки Холлеріта:
   $n\ H\ a_1\ a_2\ a_3\ ...\ a_n$,
   де кількість символів a_i повинна відповідати десятковому числу n, що стоїть перед символом H.

Висновки:

1. Регулярні вирази можуть використовуватися для опису тільки фіксованої або невизначеної кількості повторень якої-небудь конструкції.
2. Два довільні числа або два довільні підрядки не можуть порівнюватися в контексті регулярних виразів для визначення, рівні вони чи ні.