CD: COMPILER DESIGN
CD: UNIT-1 20/09/2022

20 SEPTEMBER 2022 / IT-3rd year, Vth semester
FALL SEMESTER, YEAR (Vth, 3rd)
FALL SESSION (2022-23)
(CD)
MS. SHWETA TIWARI
Published: SEPTEMBER 20, 2022

PREPARED FOR

Engineering Students

All Engineering College

CD: COMPILER DESIGN

TOPIC On : UNIT-1
Introduction to Lexical Analyzer:Regular Expression

By SHWETA TIWARI

Under On: INTRODUCTION TO COMPILER

TOPIC On : UNIT-1 Introduction to Lexical Analyzer:Regular Expression

Regular expression

❖ Regular expression

✓ A regular expression is a sequence of characters that define a pattern.

✓ Application of R.E: Validation, Searching Tools.

❑ **Notational shorthand's**

1. One or more instances: +

❖ Regular expression

- ✓ A regular expression is a sequence of characters that define a pattern.
- ✓ Application of R.E: Validation, Searching Tools.

❑ Notational shorthand's

1. One or more instances: +
2. Zero or more instances: *
3. Zero or one instances: ?
4. Alphabets: Σ

❖ Regular expression

L = One or More Occurrences of a = a^+

a
aa
aaa
aaaa
aaaaa.....

Infinite .....

❖ Regular expression (Kleene * operator)

L = Zero or More Occurrences of a = a*

ϵ
a
aa
aaa
aaaa
aaaaa.....

Infinite
.....

❖ Regular expression

1. 0 or 1

Strings: 0, 1

R.E. = (0 | 1) *also written as* (0 + 1)

❖ Regular expression

1. 0 or 1

   Strings: **0, 1** ***R.E.*** **= (0 | 1)** ***also written as*** **(0 + 1)**

2. 0 or 11 or 111

   Strings: **0, 11, 111** ***R.E.*** **= (0 | 11 | 111) or (0 + 11 + 111)**

3. String having zero or more *a*.

❖ Regular expression

1. 0 or 1

   Strings: **0, 1** ***R.E.*** **= (0 | 1)** ***also written as*** **(0 + 1)**

2. 0 or 11 or 111

   Strings: **0, 11, 111** ***R.E.*** **= (0 | 11 | 111) or (0 + 11 + 111)**

3. String having zero or more *a*.

   Strings: $\epsilon, a, aa, aaa, aaaa$ ..... ***R.E.*** $= a^*$

❖ Regular expression

1. 0 or 1

   Strings: **0, 1** — ***R.E.*** **= (0 | 1)** ***also written as*** **(0 + 1)**

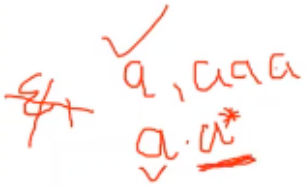
2. 0 or 11 or 111

   Strings: **0, 11, 111** — ***R.E.*** **= (0 | 11 | 111) or (0 + 11 + 111)**

3. String having zero or more *a*.

   Strings: $\epsilon, a, aa, aaa, aaaa \ldots\ldots$ — $R.E. = a^*$

4. String having one or more *a*.



❖ Regular expression

1. 0 or 1

   Strings: **0, 1** — ***R.E.*** **= (0 | 1)** ***also written as*** **(0 + 1)**

2. 0 or 11 or 111

   Strings: **0, 11, 111** — ***R.E.*** **= (0 | 11 | 111) or (0 + 11 + 111)**

3. String having zero or more *a*.

   Strings: $\epsilon, a, aa, aaa, aaaa \ldots\ldots$ — $R.E. = a^*$

4. String having one or more *a*.

   Strings: $a, aa, aaa, aaaa \ldots\ldots$ — $R.E. = a^+$

❖ Regular expression

1. 0 or 1

   Strings: **0, 1** ***R.E.*** **= (0 | 1)** ***also written as*** **(0 + 1)**

2. 0 or 11 or 111

   Strings: **0, 11, 111** ***R.E.*** **= (0 | 11 | 111) or (0 + 11 + 111)**

3. String having zero or more *a*.

   Strings: $\epsilon, a, aa, aaa, aaaa \ldots\ldots$ ***R.E.*** $= a^*$

4. String having one or more *a*.

   Strings: $a, aa, aaa, aaaa \ldots\ldots$ ***R.E.*** $= a^+ = a \cdot a^*$ ✓

❖ Regular expression

1. 0 or 1

   Strings: **0, 1** ***R.E.*** **= (0 | 1)** ***also written as*** **(0 + 1)**

2. 0 or 11 or 111

   Strings: **0, 11, 111** ***R.E.*** **= (0 | 11 | 111) or (0 + 11 + 111)**

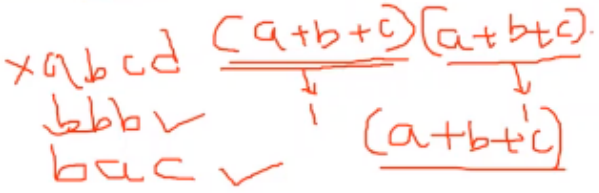
3. String having zero or more *a*.

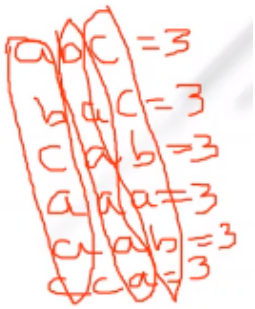
   Strings: $\epsilon, a, aa, aaa, aaaa \ldots\ldots$ ***R.E.*** $= a^*$

4. String having one or more *a*.

   Strings: $a, aa, aaa, aaaa \ldots\ldots$ ***R.E.*** $= a^+$

5. Regular expression over $\Sigma = \{a, b, c\}$ that represent all string of length 3.





❖ Regular expression

1. 0 or 1

   Strings: **0, 1** $\quad$ ***R.E.*** $= (0 \mid 1)$ ***also written as*** $(0 + 1)$

2. 0 or 11 or 111

   Strings: **0, 11, 111** $\quad$ ***R.E.*** $= (0 \mid 11 \mid 111)$ **or** $(0 + 11 + 111)$

3. String having zero or more *a*.

   Strings: $\epsilon, a, aa, aaa, aaaa \ldots\ldots$ $\quad$ ***R.E.*** $= a^*$

4. String having one or more *a*.

   Strings: $a, aa, aaa, aaaa \ldots\ldots$ $\quad$ ***R.E.*** $= a^+$

5. Regular expression over $\Sigma = \{a, b, c\}$ that represent all string of length 3.

   Strings: $abc, bca, bbb, cab, aba \ldots\ldots$ $\quad$ ***R.E.*** $= (a + b + c)(a + b + c)(a + b + c)$

❖ Regular expression

1. 0 or 1

   Strings: **0, 1** $\quad$ ***R.E.*** $= (0 \mid 1)$ ***also written as*** $(0 + 1)$

2. 0 or 11 or 111

   Strings: **0, 11, 111** $\quad$ ***R.E.*** $= (0 \mid 11 \mid 111)$ **or** $(0 + 11 + 111)$

3. String having zero or more *a*.

   Strings: $\epsilon, a, aa, aaa, aaaa \ldots\ldots$ $\quad$ ***R.E.*** $= a^*$

4. String having one or more *a*.

   Strings: $a, aa, aaa, aaaa \ldots\ldots$ $\quad$ ***R.E.*** $= a^+$

5. Regular expression over $\Sigma = \{a, b, c\}$ that represent all string of length 3.

   Strings: $abc, bca, bbb, cab, aba \ldots\ldots$ $\quad$ ***R.E.*** $= (a + b + c)(a + b + c)(a + b + c)$

❖ Regular expression

1. 0 or 1

   ***Strings*: 0, 1** **$R.E. = (0 \mid 1)$ *also written as* $(0 + 1)$**

2. 0 or 11 or 111

   ***Strings*: 0, 11, 111** **$R.E. = (0 \mid 11 \mid 111)$ or $(0 + 11 + 111)$**

3. String having zero or more *a*.

   ***Strings*: $\epsilon, a, aa, aaa, aaaa$ ….** **$R.E. = a^*$**

4. String having one or more *a*.

   ***Strings*: $a, aa, aaa, aaaa$ ….** **$R.E. = a^+$**

5. Regular expression over $\Sigma = \{a, b, c\}$ that represent all string of length 3.

   ***Strings*: abc, bca, bbb, cab, aba ….** **$R.E. = (a + b + c)(a + b + c)(a + b + c)$**

6. All binary string.

+ ٧٥, ١, ٥٥, ٥١, ٥٥٥, ١١١, ٥١٥١ ✓

❖ Regular expression

1. 0 or 1

   ***Strings*: 0, 1** **$R.E. = (0 \mid 1)$ *also written as* $(0 + 1)$**

2. 0 or 11 or 111

   ***Strings*: 0, 11, 111** **$R.E. = (0 \mid 11 \mid 111)$ or $(0 + 11 + 111)$**

3. String having zero or more *a*.

   ***Strings*: $\epsilon, a, aa, aaa, aaaa$ ….** **$R.E. = a^*$**

4. String having one or more *a*.

   ***Strings*: $a, aa, aaa, aaaa$ ….** **$R.E. = a^+$**

5. Regular expression over $\Sigma = \{a, b, c\}$ that represent all string of length 3.

   ***Strings*: abc, bca, bbb, cab, aba ….** **$R.E. = (a + b + c)(a + b + c)(a + b + c)$**

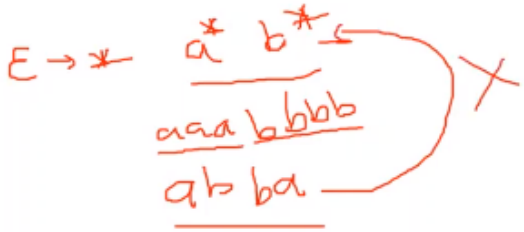
6. All binary string.

***Strings*: 0, 1, 11, 00, 101, 10101, 1111 …** **$R.E = (0 + 1)^+$**

١٠١٠١٠١٠

❖ Regular expression

7. 0 or more occurrence of either a or b or both



❖ Regular expression

7. 0 or more occurrence of either a or b or both

❖ Regular expression

7. 0 or more occurrence of either a or b or both

Strings: ϵ, *a*, *aa*, *abab*, *bab* ... ***R.E*** = (a + b)*

❖ Regular expression

7. 0 or more occurrence of either a or b or both

   Strings: $\epsilon, a, aa, abab, bab$ ... $R.E = (a + b)^*$

8. 1 or more occurrence of either a or b or both

❖ Regular expression

7. 0 or more occurrence of either a or b or both

   Strings: $\epsilon, a, aa, abab, bab$ ... $R.E = (a + b)^*$

8. 1 or more occurrence of either a or b or both

   Strings: $a, aa, abab, bab, bbbaaa$ ... $R.E. = (a + b)^+$

❖ Regular expression

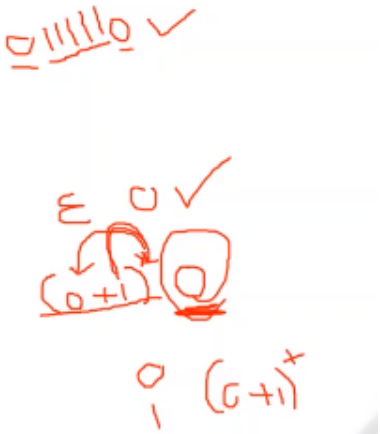
7. 0 or more occurrence of either a or b or both

   Strings: $\epsilon, a, aa, abab, bab$ ... $R.E = (a + b)^*$

8. 1 or more occurrence of either a or b or both

   Strings: $a, aa, abab, bab, bbbaaa$ ... $R.E. = (a + b)^+$

9. Binary no. ends with 0



❖ Regular expression

7. 0 or more occurrence of either a or b or both

   Strings: $\epsilon, a, aa, abab, bab$ ... $R.E = (a + b)^*$

8. 1 or more occurrence of either a or b or both

   Strings: $a, aa, abab, bab, bbbaaa$ ... $R.E. = (a + b)^+$

9. Binary no. ends with 0

   Strings: 0, 10, 100, 1010, 11110 ... $R.E = (0 + 1)^*0$

❖ Regular expression

7. 0 or more occurrence of either a or b or both

   Strings: $\epsilon, a, aa, abab, bab$ ... $R.E = (a + b)^*$

8. 1 or more occurrence of either a or b or both

   Strings: $a, aa, abab, bab, bbbaaa$ ... $R.E. = (a + b)^+$

9. Binary no. ends with 0

   Strings: $0, 10, 100, 1010, 11110$ ... $R.E = (0 + 1)^* 0$

10. Binary no. ends with 1

    Strings: $1, 101, 1001, 10101$, ... $R.E = (0 + 1)^* 1$

11. Binary no. starts and ends with 1

    Strings: $11, 101, 1001, 10101$, ... $R.E = 1(0 + 1)^* 1$

12. String starts and ends with same character

    Strings: $00, 101, aba, baab$ ... $R.E = 1(0 + 1)^* 1$ OR $0\,(0 + 1)^* 0$
    $a(a + b)^* a$ OR $b(a + b)^* b$

❖ Regular expression

13. All string of a and b starting with a

    Strings: a, ab, aab, abb... $R.E = a(a + b)^*$

14. String of 0 and 1 ends with 00

    Strings: $00, 100, 000, 1000, 1100$... $R.E = (0 + 1)^* 00$

15. String ends with abb

    Strings: $abb, babb, ababb$... $R.E. = (a + b) * abb$

16. String starts with 1 and ends with 0

    Strings: $10, 100, 110, 1000, 1100$... $R.E. = 1(0 + 1)^* 0$

17. All binary string with at least 3 characters and 3rd character should be zero

    Strings: $000, 100, 1100, 1001$... $R.E. = (0 + 1)(0 + 1)0(0 + 1)^*$

18. Language which consist of exactly two b's over the set $\Sigma = \{a, b\}$

    Strings: $bb, bab, aabb, abba$... $R.E. = a^* b a^* b a^*$

❖ Regular expression

19. The language with $\Sigma = \{a, b\}$ such that 3rd character from right end of the string is always a

    Strings: aaa, aba, aaba, abb… $R.E = (a + b)^*a(a + b)\ (a + b)$

19. Any no. of a followed by any no. of b followed by any no. of c

    Strings: ϵ, abc, aabbcc, aabc, abb… $R.E. = a^*\ b^*\ c^*$

20. String should contain at least three 1

    Strings: 111, 01101, 0101110…. $R.E. = (0 + 1)^*1\ (0 + 1)^*1\ (0 + 1)^*1\ (0 + 1)^*$

21. String should contain exactly two 1

    Strings: 11, 0101, 1100, 010010, 100100…. $R.E. = 0^*10^*10^*$

22. Length of string should be at least 1 and at most 3

    Strings: 0, 1, 11, 01, 111, 010, 100…. $R.E. = (0 + 1) + (0 + 1)(0 + 1) + (0 + 1)(0 + 1)(0 + 1)$

❖ Regular expression

23. The language with $\Sigma = \{a, b, c\}$ where a should be multiple of 3

    Strings: aaa, baaa, bacaba, aaaaaa.. $R.E. = ((b + c)^*a(b + c)^*a(b + c)^*a(b + c)^*)^*$

24. Even no. of 0

    Strings: 00, 0101, 0000, 100100…. $R.E. = (1^*01^*01^*)^*$

25. String should have odd length

    Strings: 0, 010, 110, 000, 10010…. $R.E. = (0 + 1)\ ((0 + 1)\ (0 + 1))^*$

26. String should have even length

    Strings: 00, 0101, 0000, 100100…. $R.E. = ((0 + 1)\ (0 + 1))^*$

27. String start with 0 and has odd length

    Strings: 0, 010, 010, 000, 00010…. $R.E. = 0\ ((0 + 1)(0 + 1))^*$

❖ Regular expression

29. All string begins or ends with 00 or 11

Strings: 00101, 10100, 110, 01011 ... $R.E = (00 + 11)(0 + 1)^* + (0 + 1)^*(00 + 11)$

30. Language of all string containing both 11 and 00 as substring

Strings: 0011, 1100, 100110, 010011 ...

$R.E. = ((0 + 1)^*00(0 + 1)^*11(0 + 1)^*) + ((0 + 1)^*11(0 + 1)^*00(0 + 1)^*)$

31. String ending with 1 and not contain 00

Strings: 011, 1101, 1011 .... $R.E = (1 + 01)^+$

32. Language of C identifier

Strings: *area*, *i*, *redious*, *grade*1 .... $R.E. = (\_ + L)(\_ + L + D)^*$

where L is Letter & D is digit

THANK YOU