<u>**Compiler Construction (CS-636)**</u>

Dr. Naseer Ahmed Sajid **email id: naseer@biit.edu.pk** **Whatsapp# 0346-5100010**

(Week 4) Lectures 7 & 8

Objectives: Learning objectives of this lecture are

- **Students will able to understand the regular expression of the languages and these expressions can be written for the regular languages.**

Text Book & Resources:

1. Compilers Principles Techniques and Tools (2nd Edition) by Alfread V. Aho, Ravi Sethi.
2. Introduction to Computer Theroy By Daniel I.A. Cohen.

Video Links:

https://youtu.be/tdQbzy1p0jE (Part 1)
https://youtu.be/G2d8dTJeVFA (Part 2)
https://youtu.be/401fEOqYYtY (Part 3)
https://youtu.be/UJuSAbsmRrk (Part 4)

Dr. Naseer Ahmed Sajid **email id: naseer@biit.edu.pk** **Whatsapp# 0346-5100010**

Regular Expressions

Regular Languages

- All those languages are regular for which we can write regular expression

Symbol Used in Regular Expression:

Multiple Sign

- We now introduce the use of the Kleene star, applied not to a set, but directly to the letter x and written as a superscript: x*.
- This simple expression indicates some sequence of x's (may be none at all):

$$\mathbf{x^*} = \Lambda \text{ or } x \text{ or } x^2 \text{ or } x^3 \ldots$$
$$= x^n \text{ for some } n = 0, 1, 2, 3, \ldots$$

Plus Sign

- Let us introduce another use of the plus sign. By the expression

$$x + y$$

where x and y are strings of characters from an alphabet, we mean **either** x **or** y.

- Care should be taken so as not to confuse this notation with the notation + (as an exponent).

Rules to Build Regular Expression

Definition

The set of **regular expressions** is defined by the following rules:

- **Rule 1:** Every letter of the alphabet Σ can be made into a regular expression by writing it in **boldface, ε** itself is a regular expression.
- **Rule 2:** If r_1 and r_2 are regular expressions, then so are:

(a) (r_1)

(b) r_1r_2

(c) $r_1 + r_2$

(d) r_1^*

- **Rule 3**: Nothing else is a regular expression.
- Note: If $\mathbf{r_1 = aa + b}$ then when we write r_1^* , we really mean $(r_1)^*$,

that is $r_1^* = (r_1)^* = (aa + b)^*$

Dr. Naseer Ahmed Sajid **email id: naseer@biit.edu.pk** **Whatsapp# 0346-5100010**

Example #1:

- Consider the language defined by the expression

  $(a + b)^*a(a + b)^*$

- At the beginning of any word in this language we have

$(a + b)^*$, which is any string of ***a*’s** and ***b*’s**, then comes an ***a***, then another any string.

- For example, the word **abbaab** can be considered to come from this expression by 3 different choices:

  (ε)a(bbaab) or (abb)a(ab) or (abba)a(b)

Example #2:

- The language of all words that have at least **one a** and at least **one b** is somewhat trickier. If we write

  $(a + b)^*a(a + b)^*b(a + b)^*$

then we are requiring that an a must precede **a b** in the word. Such words as **ba** and **bbaaaa** are not included in this language.

- Since we know that either the **a** comes before the **b** or the **b** comes before the **a**, we can define the language by the expression

  $(a + b)^*a(a + b)^*b(a + b)^* + (a + b)^*b(a + b)^*a(a + b)^*$

- Note that the only words that are omitted by the first term

$(a + b)^*a(a + b)^*b(a + b)^*$ are the words of the form some **b’s** followed by some **a’s**. They are defined by the expression bb^*aa^*

- We can add these specific exceptions. So, the language of all words over the alphabet $\Sigma = \{a, b\}$ that contain at least **one a** and at least **one b** is defined by the expression:

  $(a + b)a(a + b)b(a + b) + bb^*aa^*$

- Thus, we have proved that

  $(a + b)^*a(a + b)^*b(a + b)^* + (a + b)^*b(a + b)^*a(a + b)^*$

$= (a + b)^*a(a + b)^*b(a + b)^* + bb^*aa^*$

Dr. Naseer Ahmed Sajid **email id: naseer@biit.edu.pk** **Whatsapp# 0346-5100010**

Example #3:

Language having words {aa, bb} over ∑ = { a b}

- **aa** can be generated occurrence of a twice and can be generated by using rule 1 and **2b**.
- Similarly for **bb**
- Now there is a choice between aa and bb.
- According to rule **2c** we get the following regular expression

R.E = aa+bb

Example #4:

Language

{ε, a, aa, aaa, aaaa ……}

over ∑ = { a b}

- Here a is repeating itself 0 to n times
- So rule 1 and 2d will be executed

R.E = a*

Example #5:

Language

{ε, b, bb, bbb, bbbb ……}

over ∑ = { a b}

- Here b is repeating itself 0 to n times
- So rule 1 and 2d will be executed

R.E = b*

Example #6:

Language

{b, ab, aab, aaab, aaaab, aaaaab, ……}

over ∑ = { a b}

- Here a is repeating itself 0 to n times and then b appear at the and
- So rule 1, 2b and 2d will be executed

R.E = a*b

Dr. Naseer Ahmed Sajid **email id:** naseer@biit.edu.pk **Whatsapp# 0346-5100010**

Example #7:

The language contain substring ab or ba defined over, ∑={a,b}

This language must contain 'ab' or 'ba' and it is minimum condition which must be meeting. Before or after these symbols it may contain any symbols.

RE=$(a+b)^*a$b$(a+b)^*$+$(a+b)^*ba(a+b)^*$

Or

RE=$(a+b)^*(a$b+ba)$(a+b)^*$

Example #8:

The Language contain exact two a's defined over, ∑={a,b}

Minimum requirement for this language is two a's which may be at any position but condition on b's symbol is nothing. It may or may not contain b's.

RE=b*ab*ab*

Example #9:

The Language contain at most two a's defined over, ∑={a,b}

Minimum requirement for this language is zero, one or two a's which may be at any position but condition on b's symbol is nothing. It may or may not contain b's. Necessary condition is no more than two a's.

RE=b*+b*ab*+b*ab*ab*

Example #10:

The language cannot end on 'ab' defined over, ∑={a,b}

This language must end with symbols 'ab'. It may contains any symbols before this.

RE=$(a+b)^*(ba+aa+bb)$ +Λ+a+b

Example #11:

The language contain even length of word defined over, ∑={0,1,2}

This language must contain words of odd length and there is no condition on symbols.

RE=$((0+1+2)(0+1+2))^*$

Example #12:

The language contain word of length multiple of 3 defined over, ∑={a,b}

This language must contain words of length which is divisible by 3..

RE=$((a+b)(a+b)(a+b))^*$

Dr. Naseer Ahmed Sajid **email id: naseer@biit.edu.pk** **Whatsapp# 0346-5100010**

Example #13:

The language contain substring 'aa' defined over, ∑={a,b}

RE=$(a+b)^*$(aa) $(a+b)^*$

Example #14:

The Language contains even length of words defined over, ∑={a,b}

RE= ((a+b)(a+b))*

Example #15:

The Language having words of odd length defined over, ∑={a,b}

RE= (a+b)((a+b)(a+b))*

Example #16:

The Language having even length & every even position letter must be 'b' over, ∑={a,b}

RE= ((a+b)b)*

Example #17:

The language contain even number of a's AND even number of b's , defined over, ∑={a,b}

RE=(aa+bb+(ab+ba)(aa+bb)*(ab+ba))*

Example #18:

The language cannot contain substring 'ab' or 'ba' defined over, ∑={a,b}

RE=a*+b*

Example #19:

The language contain 'aa' exactly one's defined over, ∑={a,b}

RE=(ʌ+b)(ab+b)*aa(ʌ+b)(ba+b)*(ʌ+a)

Example #20:

The language starts with 'ab' if contain odd length of word, start with 'ba' if contain even length of word defined over, ∑={a,b}

RE=ab(a+b)**(**(a+b)(a+b)*+ba((a+b)(a+b)**)***

Example #21:

The language which start & end with same letters defined over, ∑={a,b}

RE=Ɛ+a+b+a(a+b)*a+ b(a+b)*b

Example #22:

The language which start & end with different letters defined over, ∑={a,b}

RE= a(a+b)*b+ b(a+b)*a

Dr. Naseer Ahmed Sajid **email id: naseer@biit.edu.pk** **Whatsapp# 0346-5100010**

Example #23:

Write the regular expression for the language starting and ending with a and having any having any combination of b's in between.

RE = a b* a

Example #24:

Write the regular expression for the language starting with a but not having consecutive b's.

R E= a(a + ab)*

Example #25:

Write the regular expression for the language having a string which should have at least one 0 and at least one 1.

RE = [(0 + 1)* 0 (0 + 1)* 1 (0 + 1)*] + [(0 + 1)* 1 (0 + 1)* 0 (0 + 1)*]

Example #26:

Write the regular expression for the language L over ∑ = {0, 1} such that all the string do not contain the substring 01.

R E= (1* 0*)

Example #27:

Write the regular expression for the language containing the string in which every 0 is immediately followed by 11

RE = (011 + 1)*