قسم الامن السيبراني

DEPARTMENT OF CYBER SECURITY

SUBJECT: **COMPUTATION THEORY**
CLASS: **3rd**

LECTURER: **MSC :MUNTATHER AL-MUSSAWEE**

LECTURE: (2)

REGULAR EXPRESSION

Regular Expression (RE)

Regular languages are formal languages that can be expressed using regular expressions.

Regular languages can be generated from one-element languages by applying certain standard operations a finite number of times. These simple operations include (**concatenation, union, and Kleen closure**).

Regular expressions can be thought of as the algebraic description of a regular language. Regular expression can be defined by the following rules:

1. Every letter of the alphabet ∑ is a regular expression. L = {a, b}
2. Null string Λ and empty set Ǿ are regular expressions.
3. If r1 and r2 are regular expressions then

(i) r1, r2
(ii) r1r2 (concatenation of r1r2)
(iii) r1 + r2 (union of r1 and r2)
(iv) r1*, r2* (kleen closure of r1 and r2) are also regular expressions

4. If a string can be derived from the rules 1, 2 and 3 then it is also a regular expression.

Note that a* means zero or more occurrence of **a** in the string while **a+** means that one or more occurrence of a in the string. That means **a*** denotes language L = {Λ , a, aa, aaa, …} and a+ represents language L = {a, aa, aaa, …}. And also note that there can be more than one regular expression for a given set of strings.

<u>Example:</u>

Write the language for each of the following regular expressions,

$\sum = \{a, b\}$.

1- $(ab)^* = \{ \Lambda, ab, abab, ababab, \ldots\ldots \}$

2- $ab^*a = \{aa, aba, abba, abbba, abbbba, \ldots\ldots\}$

3- $a^*b^* = \{ \Lambda, a, b, aa, ab, bb, aaa, aab, abb, bbb, aaaa, \ldots..\}$

Notice that ba and aba are not in this language. Also we should be very careful to observe that **$a^*b^* \neq (ab)^*$**

Example: Write a regular expression for the language containing odd number of 1s, $\sum = \{0, 1\}$.

The language will contain at least one 1. It may contain any number of 0s anywhere in the string. So the language we have to write a regular expression for is 1, 01, 01101, 0111, 111, …. . This language can be represented by the following regular expression:

$$\mathbf{0^*(10^*10^*)^*\, 10^*}$$

Example:

Write the language for each of the following regular expressions, $\sum = \{x\}$.

1- $L1 = \{x^{ood}\} = x(xx)^*$ **or** $(xx)^*\, x = \{x, xxx, xxxxx, \ldots..\}$

2- $L2 = \{x^{even}\} = (xx)^*$ **or** $= \{\Lambda, xx, xxxx, \ldots..\}$

$L3 = \{x^{even > 0}\} = (xx)^*\, xx$ **or** $xx(xx)^* = \{xx.xxxx.xxxxxx\ldots\ldots\}$

Examples:

1- Consider the language L3 defined over the alphabet $\sum = \{a, b, c\}$. All the words in L3 begin with an a or c and then are followed by some number of b's. We may write this as:

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

2- Consider a finite language L4 that contains all the strings of **a's** and **b's** of length exactly three.

$$L4 = \{aaa, aab, aba, abb, baa, bab, bba, bbb\}$$

So we may write:

$$(a + b)(a + b)(a + b) \textbf{ or } (a + b)^3$$

In general, if we want to refer to the set of all possible strings of a's and b's of any length, we could write:

$$(a + b)^*$$

3- Construct **RE** for all words that begin with the letter **a** :

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

4- All words that begin with an a and end with **b** can be defined by the expression:

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

5- The language of all words that have at **least two a's** can be described by the expression:

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

6- The language of all words that have at **least one a** and at **least one b**:

$$(a + b)^*a(a + b)^*b(a + b)^* \quad \textbf{or} \quad bb^*aa^*$$

7- The words of the form some **b's** followed by some **a's**. These exceptions are all defined by the regular expression:

$$bb^*aa^* \equiv b^+a^+$$

<u>Example:</u> Write a regular expression for the language
$L = \{ab^{n} w: n >= 3, w € (a + b)^+\}$

The strings in the language begins with a followed by three bs and followed by either w, w will contain at least one a or b. The strings are like abbba, abbbb.

$$ab^3 (a+b)^+$$

<u>Homework:</u>

1- Find a regular expression over the alphabet **{a, b}:**