Module-2 :- Regular Expression:-

1) The language accepted by FA can be easily described by simple expressions called R.E.
2) It is the most effective way to represent any language.
3) The languages accepted by some regular expressions are referred as regular languages.
4) A R.E can be described as a sequence of pattern that defined a string.

* **Regular Set:-** Any set represented by a regular expression is called as regular set.

Eg:- x^* denotes the set $\{\varepsilon, x, xx, \dots\}$

x^+ denotes the set $\{x, xx, \dots\}$.

If a, b are the elements of Σ then regular expression

1) a denotes the set $\{a\}$.
2) $a+b$ denotes the set $\{a, b\}$
3) ab denotes the set $\{ab\}$
4) a^* denotes the set $\{\varepsilon, a, aa \dots\}$
5) $(a+b)^*$ denotes the set $\{\varepsilon, a, b, aa, ab, ba, bb \dots\}$

$\boxed{R = R_1 + R_2}$

* **Operations:-**

⇒ **Union** of 2 RE R_1 and R_2 is RE $R \rightarrow \boxed{R_1 \cup R_2 = R}$.

Eg:- a is RE R_1 ; b is RE $R_2 \Rightarrow R = a+b$

⇒ **Concatenation** of 2 RE R_1, R_2 written as $R_1 \cdot R_2$. is also a RE. $(R \Rightarrow R_1 \cdot R_2)$ Eg:- a, b are RE of R_1, R_2 then $R = a \cdot b$

⇒ **Iteration (closure)**:-

Kleene closure → $\bigcup_{i=0}^{\infty} L^i = L^0 \cup L^1 \cup L^2 \dots$

Positive closure → $\bigcup_{i=1}^{\infty} L^i = L^1 \cup L^2 \cup L^3 \dots$

These are the operations on Regular Expression

* **Regular Set:** Any set represented by a regular expression is called a regular set.

If a, b are the elements of Σ then regular expressions

→ a denotes the set {a}.

→ a+b denotes the set {a, b}.

→ ab denotes the set {ab}.

→ a^* denotes the set {ε, a, aa ...}

→ $(a+b)^*$ denotes the set {ε, a, b, aa, ab, ba, bb, aaa ...}.

* **Representing the regular set by regular expression:-**

Regular Set	Regular Expression
{101}	101
{ε, a}	a
{ε, a, aa, ab, ba, bb...}	$(a+b)^*$
{ab, ba}	ab+ba

* **Describing the following sets by Regular Expression:-**

1) All strings of 0's and 1's:- The language has the elements {ε, 0, 1, 00, 01, 10, 11, 000, ...}. Hence the regular expression is $(0+1)^*$.

2) Set of all strings of 0's and 1's ending in 00:- The language has the elements {00, 000, 100, 0000, 0100, 1000, 1100, 00000, ...}.

This can be written as {ε, 0, 1, 00, 01, 10, 11, 000, ... } 00. Hence the regular expression is $(0+1)^*00$.

3) Set of all strings 0's and 1's begin with 0 and ending with 1 :- The language has the elements {01, 001, 0001, 0011, 0101, 0111, 00001, ... }. This can be written as 0 {ε, 0, 1, 00, 01, 11, 000, ... } 1. Hence the regular expression is $0(0+1)^*1$.

4) Set of all strings having even no. of 1's:- The language has the elements. {ε, 11, 1111, 111111, 11111111, ... }. Hence the regular expression is $(11)^*$.

5) Set of all strings having odd number of 1's:- The language has the elements {1, 111, 11111, 1111111, ... }. This can be written as {ε, 11, 1111, 111111, 11111111, ... }1. Hence the regular expression is $1(11)^*$ or $(11)^*1$.

→ $\boxed{\text{Note}}$: 1^* is wrong as $(11)^* + 1(11)^* = 1^*$

6) Strings of 0's and 1's with at least two connective 00:- The language has the elements {00, 000, 001, 100, 0000, 0010, 0011, 0100, 1000, 1100, 00001, ... }. These strings can be generated by a regular expression $(0+1)^*00(0+1)^*$.

7) All strings of 0's and 1's beginning with 1 or 0 and not having two consecutive 0's :- 11111...1 1^* is allowed.

10101010...$(10)^*$ is allowed.

Regular expression for strings with 0's and 1's that do not have two consecutive 0's is $(1+10)^*$ but here string starts with only 1 hence alter expression to start string even with 0. Hence the regular expression is $(0+\epsilon)(1+10)^*$.

8) <u>Set of all strings ends with 011</u> :- The language has the elements {011, 0011, 1011, 00011, 010111, 1011, 11011, ...}. This can be written as {ε, 0, 1, 00, 01, 10, 11, 000 ....}011. Hence the regular expression is $(0+1)^*011$.

9) <u>Set of all strings with atleast one 0, 1 one 2</u> :- The language has the elements {012, 0012, 0112, 0122, 00112, 00122, 01122, 001122, ...}. Hence the regular expression is $0^+1^+2^+$.

10) <u>Set of all strings of 0's and 1's whose last two symbols are the same</u> : The language has the elements {00, 11, 011, 000, 0011, 0111, 0100, 0000 ....}. Hence the regular expression is $(0+1)^*(00+11)$.

11) <u>Set of strings in which every 0 is immedeately followed by atleast two 1's</u> :-

→ $\boxed{\text{Note}}$: To find the RE consider two possibilities :-

(i) strings with only 1's.

(ii) Every 0 preeded by 11 i.e., 011

Hence $(1+011)^*$.

12) <u>Set of all strings with 1100 as sub string is $(0+1)^*1100(0+1)^*$</u> :- The language has the elements {1100, 11000, 11001, 11100, 11000, 110010, 110011, 011100, 111000, 111100, 1100001, ...}. These strings can be generated by a regular expression $(0+1)^*00(0+1)^*$.