Regular Expression

- The language accepted by finite automata can be easily described by simple expressions called Regular Expressions. It is the most effective way to represent any language.
- The languages accepted by some regular expression are referred to as Regular languages.
- A regular expression can also be described as a sequence of pattern that defines a string.
- Regular expressions are used to match character combinations in strings. String searching algorithm used this pattern to find the operations on a string.

What is Regular Expression?

Regular Expressions are the expressions that describe the language accepted by Finite Automata. It is the most efficient way to represent any language.

The languages accepted by some regular expressions are referred to as **Regular languages**.

Regular expressions are used to check and match character combinations in strings. The string searching algorithm uses this pattern to find the operations on a string.

Let **Σ** be an alphabet that denotes the input set.

The regular expression over Σ can be defined as follows:-

1.) Φ is a regular expression that denotes the empty set.

2.) ε is a regular expression and denotes the set {ε}, called a null string.

3.) For each 'x' in Σ 'x' is a regular expression and denotes the set {x}.

4.) If 'a' and 'b' are the regular expressions that denote the language L1 and L2, respectively, then:-

a.) a+b is equal to L1 U L2 union.

b.) ab is equal to the L1L2 concatenation.

c.) a* is equal to L1* closure.

In a regular expression, **a*** means zero or more occurrences of a. It can generate {ε, a, aa, aaa, aaaa, aaaaa, ...}.

In a regular expression, **a^+** means one or more occurrences of a. It can generate {a, aa, aaa, aaaa, aaaaa, ...}.

Operations On Regular Language

The various operations on the regular language are:

1.) Union

If R and S are two regular languages, their union R U S is also a Regular Language.

R U S = {a | a is in R or a is in S}

2.) Intersection

If R and S are two regular languages, their intersection is also a Regular Language.

L ⋂ M = {ab | a is in R and b is in S}

3.) Kleene closure

If R is a regular language, its Kleene closure R1* will also be a Regular Language.

R* = Zero or more occurrences of language R.

Examples of Regular Expressions