Regular Expression

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.

- In a regular expression, x* means zero or more occurrence of x.
- It can generate {e, x, xx, xxx, xxxx, ...}
- In a regular expression, x^+ means one or more occurrence of x.
- It can generate {x, xx, xxx, xxxx, ...}

Ex

- Write the regular expression for the language accepting all combinations of a's, over the set ∑ = {a}
- Ans: R = a*
- That is Kleen closure of a.

Ex

- Write the regular expression for the language accepting all combinations of a's except the null string, over the set ∑ = {a}

- Ans: L = {a, aa, aaa, ...}

- This set indicates that there is no null string.

- R = a^+

Ex

- Write the regular expression for the language accepting all the strings containing any number of a's and b's.

- All combinations of a,b

- r.e. = (a + b)*

- This will give the set as L = {ε, a, aa, b, bb, ab, ba, aba, bab, ...}, any combination of a and b, even a null string.

Ex

- Write the regular expression for the language accepting all the strings which are starting with 1 and ending with 0, over ∑ = {0, 1}.

Ex

- Write the regular expression for the language accepting all the strings which are starting with 1 and ending with 0, over ∑ = {0, 1}.

- Plz note: 1+0+
- Word which has 1's in beginning followed by 0's in the end

Ex

- Write the regular expression for the language accepting all the string which are starting with 1 and ending with 0, over ∑ = {0, 1}.

- R = 1 (0+1)* 0

Ex

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

Ex

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

- R = a (b)* a

Ex

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

Ex

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

- a*+ a*ba*   ??

Ex

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

- **Solution:**

- L = {a, aa, ab, aba, aab, aaa, abab, ...}

- The regular expression for the above language is:

- R = {a + ab}+

Ex

- Write the regular expression for the language accepting all the string in which any number of a's is followed by any number of b's is followed by any number of c's.

Ex

- Write the regular expression for the language accepting all the string in which any number of a's is followed by any number of b's is followed by any number of c's.

- R = a* b* c*

Ex

- Write the regular expression for the language over ∑ = {0} having even length of the string.

Ex

- Write the regular expression for the language over ∑ = {0} having even length of the string.

- R = (00)*

- Write the regular expression for the language having a string which should have atleast one 0 and alteast one 1.

- R = [(0 + 1)* 0 (0 + 1)* 1 (0 + 1)*] + [(0 + 1)* 1 (0 + 1)* 0 (0 + 1)*]
- {01,10,101,010,110,001,....}

Regular Expressions	Regular Set
(0 + 10*)	L = { 0, 1, 10, 100, 1000, 10000, ... }
(0*10*)	L = {1, 01, 10, 010, 0010, ...}
(0 + ε)(1 + ε)	L = {ε, 0, 1, 01}
(a+b)*	Set of strings of a's and b's of any length including the null string. So L = { ε, a, b, aa , ab , bb , ba, aaa.......}
(a+b)*abb	Set of strings of a's and b's ending with the string abb. So L = {abb, aabb, babb, aaabb, ababb, ...}
(11)*	Set consisting of even number of 1's including empty string, So L= {ε, 11, 1111, 111111, ...}
(aa)*(bb)*b	Set of strings consisting of even number of a's followed by odd number of b's , so L = {b, aab, aabbb, aabbbbb, aaaab, aaaabbb, ...}
(aa + ab + ba + bb)*	String of a's and b's of even length can be obtained by concatenating any combination of the strings aa, ab, ba and bb including null, so L = {aa, ab, ba, bb, aaab, aaba, ...}