Page No.
Class
Roll No.
Date : / /201

Practical NO 3

Title : Write a C program to recognize string under 'a*' ; 'a*b', 'abb'.

Theory :

A regular Expression can be recursively defined as follows

- ε is a regular Expression ~~is~~ indicates the language containing on empty string ($L(\varepsilon) = \{\varepsilon\}$)

- ϕ is a Regular Expression denoting an empty language ($L(\phi) = \{\}$)

- X is a Regular ~~Expression~~

- If X is a regular expression denoting the language L(X) and Y is a regular expression denoting the language L(Y) then.

→ X+Y is a regular expression corresponding to the language $L(X) + L(Y)$

→ X.Y is a regular expression corresponding to the language $(L(X).L(Y))$ where $L(X.Y) = L(X).L(Y)$

→ R* is a regular expression corresponding to the language $L(R^*)$ where $(L(R^*) = (L(R))^*$

Regular expression can be defined by the followin rules

1. Every letter of the alphabets Σ is a regular expression.

2. Null string and empty set ϕ are regular Expression.

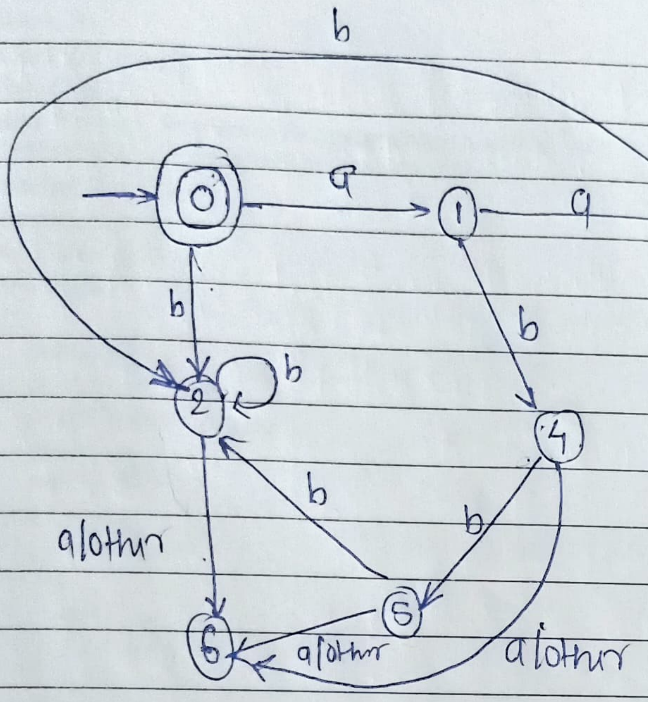
3. If r_1 and r_2 are regular expression then

i) r_1, r_2
ii) $r_1 r_2$ (concatenation of r_1, r_2)
iii) $r_1 + r_2$ (union of r_1 and r_2)
iv) $r_1{}^* r_2{}^*$ (Klean closer of r_1 and r_2)

4) If a string derived from the rules 1, 2 and 3 then it also regular expression

Some RE example.

Regular Expression	Regular Set
$(0 + 10^*)$	$L = \{0, 1, 10, 100, 1000, \ldots\}$
$(0^* + 0^*)$	$L = \{1, 01, 10, 010, 0010, \ldots\}$
$(0+\epsilon)(1+\epsilon)$	$L = \{\varepsilon, 0, 1, 01\}$



DFA of Regular Expression a*, a*b, abb

Result :-

- thus the program to recognize string under 'a*', 'a*b' 'abb' was executed and the output was verified succesfully