3 - Regular Expression to E-NFA

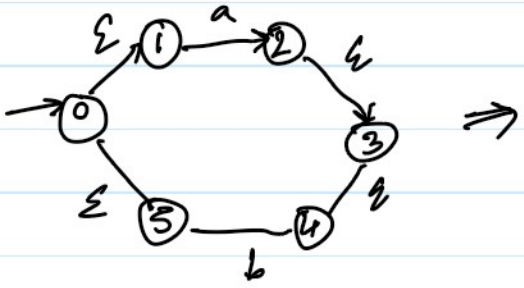
Tuesday, 30 January 2024 9:54 AM

Ashima Fatima Seik Hugiaur Raghman,
21BAI1830.

AIM: Write a c/c++/java program to convert the given regular expression to epsilon NFA. Use a regular expression with union and concatenation operation or union operation or all operations.

Sample Input:

(a|b)



Output:

	a	b	ε
0	-	-	{1,3}
1	2	-	-
2	-	-	5
3	-	4	-
4	-	-	5
5	-	-	-

ALGORITHM

1. Initialize variables:
   a. reg : an array of characters to store the regular expression
   b. q : 2D array to represent the transition Table
2. read the regular expression from the user.
3. parse the regular expression. To do so:
   a. Iterate through each character of the regular expression
   b. Based on the current character and its neighbours, determine the transition in the NFA.
   c. update the transition table accordingly.
   d. move to the next character in the regular expression.
4. Print the transition table by iterating through its transitions for each input symbol.

transitions for each input symbol.

OUTPUT

```
Ashima Fatima Seik Mugibur Raghman, 21BAI1830(a|b)
Given regular expression: (a|b)

        Transition Table
-------------------------------------
Current State | Input | Next State
-------------------------------------

  q[1]        |   e   |  q[2] , q[4]
  q[2]        |   a   |  q[3]
  q[3]        |   e   |  q[6]
  q[4]        |   b   |  q[5]
  q[5]        |   e   |  q[6]
-------------------------------------
ashima@LAPTOP-LLSNCVFU:/mnt/c/Users/Ashima/Desktop/Ashim
Ashima Fatima Seik Mugibur Raghman, 21BAI1830a.(a|b)
Given regular expression: a.(a|b)

        Transition Table
-------------------------------------
Current State | Input | Next State
-------------------------------------

  q[1]        |   a   |  q[2]
  q[2]        |   e   |  q[3] , q[5]
  q[3]        |   a   |  q[4]
  q[4]        |   e   |  q[7]
  q[5]        |   b   |  q[6]
  q[6]        |   e   |  q[7]
-------------------------------------
```

CODE

```c
#include<stdio.h>
#include<string.h>
int main()
{
    printf("Ashima Fatima Seik Mugibur Raghman, 21BAI1830");
    char reg[20]; int q[20][3],i=0,j=1,len,a,b;
    for(a=0;a<20;a++) for(b=0;b<3;b++) q[a][b]=0;
    scanf("%s",reg);
    printf("Given regular expression: %s\n",reg);
    len=strlen(reg);
    while(i<len)
    {
```

```
            if(reg[i]=='a'&&reg[i+1]!='|'&&reg[i+1]!='*') { q[j][0]=j+1; j++; }
            if(reg[i]=='b'&&reg[i+1]!='|'&&reg[i+1]!='*') { q[j][1]=j+1; j++;   }
            if(reg[i]=='e'&&reg[i+1]!='|'&&reg[i+1]!='*') { q[j][2]=j+1; j++;   }
            if(reg[i]=='a'&&reg[i+1]=='|'&&reg[i+2]=='b')
            {
              q[j][2]=((j+1)*10)+(j+3); j++;
              q[j][0]=j+1; j++;
                q[j][2]=j+3; j++;
                q[j][1]=j+1; j++;
                q[j][2]=j+1; j++;
                i=i+2;
            }
            if(reg[i]=='b'&&reg[i+1]=='|'&&reg[i+2]=='a')
            {
                q[j][2]=((j+1)*10)+(j+3); j++;
                q[j][1]=j+1; j++;
                q[j][2]=j+3; j++;
                q[j][0]=j+1; j++;
                q[j][2]=j+1; j++;
                i=i+2;
            }
            if(reg[i]=='a'&&reg[i+1]=='*')
            {
                q[j][2]=((j+1)*10)+(j+3); j++;
                q[j][0]=j+1; j++;
                q[j][2]=((j+1)*10)+(j-1); j++;
            }
            if(reg[i]=='b'&&reg[i+1]=='*')
            {
                q[j][2]=((j+1)*10)+(j+3); j++;
                q[j][1]=j+1; j++;
                q[j][2]=((j+1)*10)+(j-1); j++;
            }
            if(reg[i]==')'&&reg[i+1]=='*')
            {
                q[0][2]=((j+1)*10)+1;
                q[j][2]=((j+1)*10)+1;
                j++;
            }
            i++;
        }
        printf("\n\tTransition Table \n");
        printf("_____________________________________\n");
        printf("Current State |\tInput |\tNext State");
        printf("\n_____________________________________\n");
        for(i=0;i<=j;i++)
        {
            if(q[i][0]!=0) printf("\n  q[%d]\t      |   a   |  q[%d]",i,q[i][0]);
            if(q[i][1]!=0) printf("\n  q[%d]\t      |   b   |  q[%d]",i,q[i][1]);
            if(q[i][2]!=0)
            {
                if(q[i][2]<10) printf("\n  q[%d]\t      |   e   |  q[%d]",i,q[i][2]);
                else printf("\n  q[%d]\t      |   e   |  q[%d] , q[%d]",i,q[i]
[2]/10,q[i][2]%10);
            }
        }
        printf("\n_____________________________________\n");
        return 0;
    }
```