**Saket Kumar Baranwal**

**RA1911003010414 G1 COMPILER DESIGN LAB EXP – 2**

# Aim:- To study Construction of an FA from an RE.

# ALGORITHM: 1. Start

# 2. Get the input from the user

# 3. Initialize separate variables and functions for Postfix , Display and NFA

# 4. Create separate methods for different operators like +,\*, .

# 5. By using Switch case Initialize different cases for the input

# 6. For ' . ' operator Initialize a separate method by using various stack functions do the same for the other operators like ' \* ' and ' + '.

# 7. Regular expression is in the form like a.b (or) a+b

# 8. Display the output

# 9. Stop

# Code:-

# transition\_table = [ [0]\*3 for \_ in range(20) ]

# re = input("Enter the regular expression : ")

# re += " "

# i = 0

# j = 1

# while(i<len(re)):

# if re[i] == 'a':

# try:

# if re[i+1] != '|' and re[i+1] !='\*':

# transition\_table[j][0] = j+1

# j += 1

# elif re[i+1] == '|' and re[i+2] =='b':

# transition\_table[j][2]=((j+1)\*10)+(j+3)

# j+=1

# transition\_table[j][0]=j+1

# j+=1

# transition\_table[j][2]=j+3

# j+=1

# transition\_table[j][1]=j+1

# j+=1

# transition\_table[j][2]=j+1

# j+=1

# i=i+2

# elif re[i+1]=='\*':

# transition\_table[j][2]=((j+1)\*10)+(j+3)

# j+=1

# transition\_table[j][0]=j+1

# j+=1

# transition\_table[j][2]=((j+1)\*10)+(j-1)

# j+=1

# except:

# transition\_table[j][0] = j+1

# elif re[i] == 'b':

# try:

# if re[i+1] != '|' and re[i+1] !='\*':

# transition\_table[j][1] = j+1

# j += 1

# elif re[i+1]=='|' and re[i+2]=='a':

# transition\_table[j][2]=((j+1)\*10)+(j+3)

# j+=1

# transition\_table[j][1]=j+1

# j+=1

# transition\_table[j][2]=j+3

# j+=1

# transition\_table[j][0]=j+1

# j+=1

# transition\_table[j][2]=j+1

# j+=1

# i=i+2

# elif re[i+1]=='\*':

# transition\_table[j][2]=((j+1)\*10)+(j+3)

# j+=1

# transition\_table[j][1]=j+1

# j+=1

# transition\_table[j][2]=((j+1)\*10)+(j-1)

# j+=1

# except:

# transition\_table[j][1] = j+1

# 

# elif re[i]=='e' and re[i+1]!='|'and re[i+1]!='\*':

# transition\_table[j][2]=j+1

# j+=1

# elif re[i]==')' and re[i+1]=='\*':

# transition\_table[0][2]=((j+1)\*10)+1

# transition\_table[j][2]=((j+1)\*10)+1

# j+=1

# i +=1

# print ("Transition function:")

# for i in range(j):

# if(transition\_table[i][0]!=0):

# print("q[{0},a]-->{1}".format(i,transition\_table[i][0]))

# if(transition\_table[i][1]!=0):

# print("q[{0},b]-->{1}".format(i,transition\_table[i][1]))

# if(transition\_table[i][2]!=0):

# if(transition\_table[i][2]<10):

# print("q[{0},e]-->{1}".format(i,transition\_table[i][2]))

# else:

# print("q[{0},e]-->{1} & {2}".format(i,int(transition\_table[i][2]/10),transition\_table[i][2]%10))

# Output:-

Enter the regular expression : (a\*b)|abb

Transition function:

q[1,e]-->2 & 4

q[2,a]-->3

q[3,e]-->4 & 2

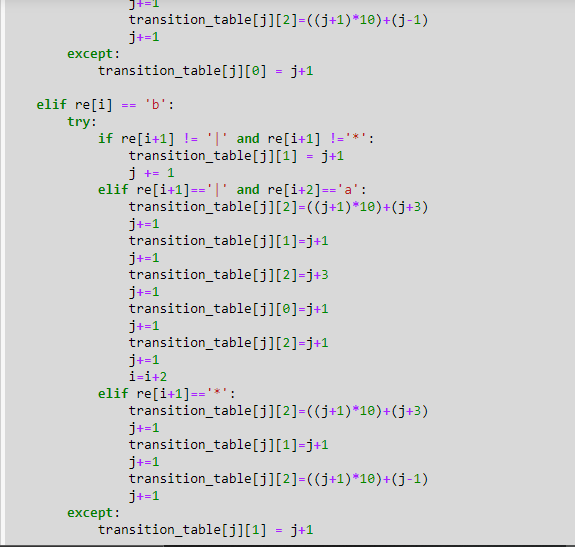
q[4,b]-->5

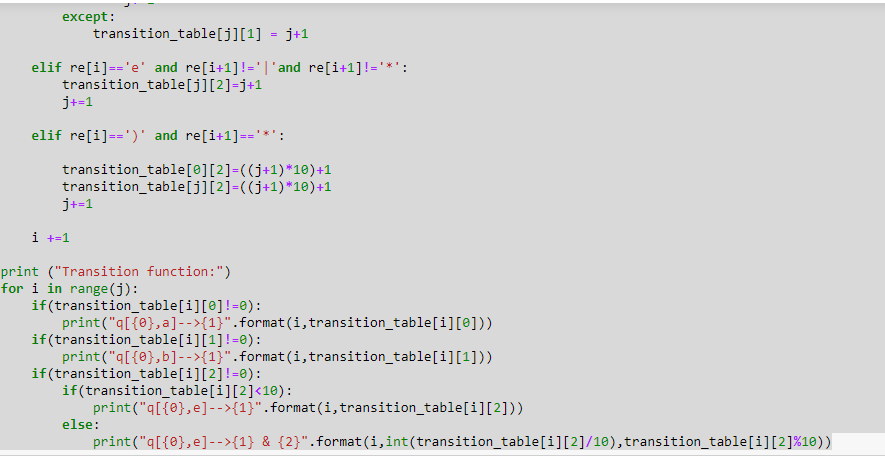
q[5,a]-->6

q[6,b]-->7

q[7,b]-->8

# 





**Output Screenshot:**

