EXP 2: Conversion from RE to NFA AIM: To write a program for converting regular expression to NFA ALGORITHM: 1. Start 2. Get the input from the user 3. Initialize separate variables 4 fuctions for portfin. 4. Create separate methode for different operators like +, *, -5. By using snitch case initialize different cases for the input. 6- For ', operator initialize a separate method by using various stack functions, do me same for me other Operators like '+' and 7. RE is en the form like a. b (or) atb. 8. Display the output 9. Stop. PROGIRAM: Clausition-table = [[0]*3 for in range (20)] re = input ("Enter tre regular exponessions") Re = + = " " while (ix luncre)): if recij == 'a' try: if re EIH] != " |" and re [iH] != "*" transfrom table [i][0] = JH elif re[iH] == i" and re[i+2] == 'b'; transfrom _ table [j][2] = ((j+1)*10) + j+3) 1+=1

```
transition - table [j] [0] = (+1
   j +=1
transition - table [j][2]:j+3
  j+ = 1
Lauration-table[j+7]=j+1
  1 + =1
transition_table [j][2]=j+1
  1+=1
  i = i + 2
elf re City ] == 1 x':
transfron - table [j][2] = ((j+1)*10 + (j+3)
   j+=1
transition-table [j][0]= j+1
  1+=1
transition - table tj][2] = ((y+1)*10) + (j-1)
  1+=1
except:
transtren -table [j][0]=j+1
elif reti] = 161:
try:
if reciti7!= 1' and reciti7!= '+':
transition_talshe [j] [2] = ((j+1)*10) + (j+3)
 1+=1
teamsition_table[j][1]=j+1
 j+ = 1
tansition_table [j][2]=j+3
 j+ =1
teausition-table GICO] = j + 1
j+ = 1
transfron-table [j ] [2] = j+1
```

```
j+ = 1
  i=i+2
elif recitiJ== 1 * ':
teamention_table [j][2]= ((j+1)*10) + (j+3)
  1+ =1
transition-table GICiI=j+1
 j+==1
transitven - table [j][2]=((j+1)*10)+(j-1)
  1+=1
                    Transition de la Co
except:
transtrum- Lable GJCi ]=j+1
                             may townin
elif re Ci] == ie' and re[i+1]! = [i] and
        re [i+1]!= 1 * 1:
transition-table GJ [2] = 171
    j+=-1
                               1111
clif re[i] == ) and re[i+1] == '+';
    transition_table cozerz = (cj+1) * 10) +1
    Leavestien - table [j][2] = ((j+1) * 10) +1
       j+ = 1
   it=1
 point (" Transition fretion")
   for un i in samge (j):
    if (transition_talsh [i][0] !=0):
```

Start with a and with ba

Fig: a (a+b) * ba

Cgiven regular expression: a abba

Transition talde

Current	clabe	0 1 - 7	. !
		Supert	Nent state
910		(a)	9[2]
9 [27	laj	.9 [2]
9 (3]	[6]	7[3]
	4]	16 (9[3]
	[a]	121	9[2]

DUT PUT:

Enter the regular expressions (a/6) # ass transition function:

Nent State 9[7], 9[1] 9[0] 1 le Col 9.[1] 9[2],9[4] rel 2637 2[2] 121 2[3] 101 9 [6] [[4] 161 : (1) +... 9 [5] 9 [5] 121 9[6] 9[6] 1e1 9[7], 2[1] 9[7] lal 2 [8] 9[8]

[[8] [6] 9[9] [6] 9[9]

