**EXP 2 : Conversion from Regular Expression to NFA**

**AIM :**

.

* To convert the Regular expression to NFA using python language.

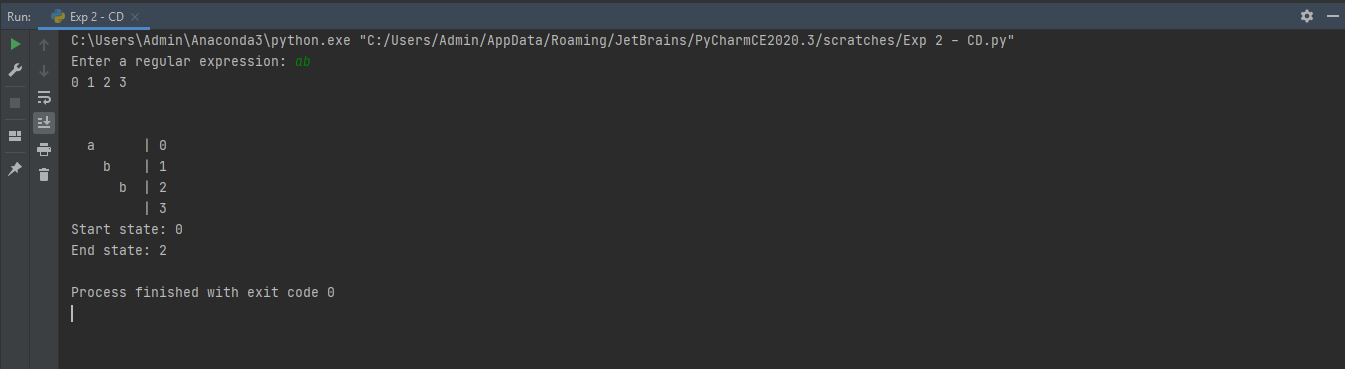
**ALGORITHM:**

* Take input as string.
* Create a list t and 2 variables r and c.
* Then loop over the expression.
* Insert the conditions for \*,| and +.
* Finally again loop over range of r and t.
* Then create a transition table for it.
* Take the first and last items as the start and final state respectively.

**SOURCE CODE:**

expr = input("Enter a regular expression: ")  
t = [[' ' for \_ in range(10)] for \_ in range(10)]  
r = 0  
c = 0  
for i in range(len(expr)):  
 if (expr[i] == "|"):  
 t[r][r + 1] = 'E'  
 t[r + 1][r + 2] = expr[i - 1]  
 t[r + 2][r + 5] = 'E'  
 t[r][r + 3] = 'E'  
 t[r + 4][r + 5] = 'E'  
 t[r + 3][r + 4] = expr[i + 1]  
 r = r + 5  
 elif(expr[i] == "\*"):  
 t[r - 1][r] = 'E'  
 t[r][r + 1] = 'E'  
 t[r][r + 3] = 'E'  
 t[r + 1][r + 2] = expr[i - 1]  
 t[r + 2][r + 1] = 'E'  
 t[r + 2][r + 3] = 'E'  
 r = r + 3  
 elif(expr[i] == "+"):  
 t[r][r + 1] = expr[i - 1]  
 t[r + 1][r] = 'E'  
 r = r + 1  
 else:  
 if(c == 0):  
 if (expr[i].isalpha() and expr[i + 1].isalpha()): t[r][r + 1] = expr[i]  
 t[r + 1][r + 2] = expr[i + 1]  
 r = r + 2  
 c = 1  
 c = 1  
 if(c == 1):  
 if (expr[i + 1].isalpha()):  
 t[r][r + 1] = expr[i + 1]  
 r = r + 1  
 c = 2  
 else:  
 if (expr[i + 1].isalpha()):  
 t[r][r + 1] = expr[i + 1]  
 r = r + 1  
 c = 3  
 break  
  
  
for num in range(r+1):  
 print(num,end=" ")  
print("\n \n")  
for i in range(r+1):  
 for j in range(r+1):  
 print(t[i][j],end=" ")  
 print(" |",i)  
print("Start state: 0\nEnd state:",i-1)

**OUTPUT:**



**RESULT :**

* Hence converted the regular expression to NFA using python.