Exp: 2 Conversion From Regular Expression To NFA

Name: - Sai Mohit Ambekar

Reg No: - RA1911031010137

Class: - CSE-IT (L2 Section)

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 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

PROGRAM: -

```
rows, cols = (20, 3)
q = [[0]*cols]*rows

reg = input("Enter your Regular Expression: ")
```

```
len = len(reg)
i = 0
j = 1
print(q)
while(i < len):
    if reg[i] == 'a':
        try:
            if reg[i+1] != '|' and reg[i+1] != '*':
                q[j][0] = j+1
                j += 1
        except:
            q[j][0] = j+1

    elif reg[i] == 'b':
        try:
            if reg[i+1] != '|' and reg[i+1] != '*':
                q[j][1] = j+1
                j += 1
        except:
            q[j][1] = j+1

    elif reg[i] == 'e' and reg[i+1] != '|' and reg[i+1] != '*':
        q[j][2] = j+1
        j += 1

    elif reg[i] == 'a' and reg[i+1] == '|' and reg[i+2] == 'b':
        q[j][2] = ((j+1)*10)+(j+3)
        j += 1
        q[j][0] = j+1
        j += 1
        q[j][2] = j+3
        j += 1
        q[j][1] = j+1
        j += 1
        q[j][2] = j+1
        j += 1
```

```
    i = i+2

elif reg[i] == 'b' and reg[i+1] == '|' and reg[i+2] == 'a':

    q[j][2] = ((j+1)*10)+(j+3)
    j += 1
    q[j][1] = j+1
    j += 1
    q[j][2] = j+3
    j += 1
    q[j][0] = j+1
    j += 1
    q[j][2] = j+1
    j += 1
    i = i+2

elif reg[i] == 'a' and reg[i+1] == '*':

    q[j][2] = ((j+1)*10)+(j+3)
    j += 1
    q[j][0] = j+1
    j += 1
    q[j][2] = ((j+1)*10)+(j-1)
    j += 1

elif reg[i] == 'b' and reg[i+1] == '*':
    q[j][2] = ((j+1)*10)+(j+3)
    j += 1
    q[j][1] = j+1
    j += 1
    q[j][2] = ((j+1)*10)+(j-1)
    j += 1

elif reg[i] == ')' and reg[i+1] == '*':

    q[0][2] = ((j+1)*10)+1
```

```python
        q[j][2] = ((j+1)*10)+1
        j += 1

    i += 1

print("Transition Function = ")

for i in range(0, j):
    if q[i][0] != 0:

        print(f"\n {q[i]},a--> {q[i][0]}")

    elif q[i][1] != 0:
        print(f"\n {q[i]},b-->{q[i][1]}")

    elif q[i][2] != 0:

        if q[i][2] < 10:
            print(f"\n {q[i]},e-->{q[i][2]}")
        else:
            print(f"\n {q[i]},e-->{q[i][2]}/10 and {q[i][2]}%10")
```

OUTPUT: -

```
/usr/local/bin/python3 /Users/saimohitambekar/Downloads/exp_2.py
saimohitambekar@Sais-MacBook-Air ~ % /usr/local/bin/python3 /Users/saimohitambekar/Downloads/exp_2.py
Enter your Regular Expression: (a|b)*abb
[[0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [
0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]]
Transition Function =

 [4, 6, 31],a--> 4

 [4, 6, 31],a--> 4

 [4, 6, 31],a--> 4

 [4, 6, 31],a--> 4

 [4, 6, 31],a--> 4
saimohitambekar@Sais-MacBook-Air ~ %
```

RESULT: -

The program to convert regular expressions to NFA was implemented successfully.