Compiler Design Lab WEEK 2

Experiment:

Implementation of regular expression to NFA

Aim:

To implement the conversion of regular expression to nfa.

Algorithm:

1.Create a menu for getting four regular expressions input as choice.

2.To draw NFA for a, a/b, ab, a\* create a routine for each regular expression.

3.For converting from regular expression to NFA, certain transitions had been made based on choice of input at the runtime.

4.Each of the NFA will be displayed in sequential order.

**Code**

#include<bits/stdc++.h>

#include<iostream>

using namespace std;

int main ()

{

char reg[20];

int q[20][3], i, j, len, a, b;

for (a = 0; a < 20; a++)

{

for (b = 0; b < 3; b++)

{

q[a][b] = 0;

}

}

// input

cout<<"Enter the Regular Expression :-"<<endl;

cin >> reg;

len = strlen (reg);

i = 0;

j = 1;

// Thompsons function

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++;

}

// Outputs

cout << "Transition functions are :- \n";

for (i = 0; i <= j; i++)

{

if (q[i][0] != 0)

cout << " q[" << i << ",a]-->" << q[i][0]<< endl;

if (q[i][1] != 0)

cout << " q[" << i << ",b]-->" << q[i][1]<< endl;

if (q[i][2] != 0)

{

if (q[i][2] < 10)

cout << " q[" << i << ",e]-->" << q[i][2] << endl;

else

cout << " q[" << i << ",e]-->" << q[i][2] /

10 << "&" << q[i][2] % 10 <<endl ;

}

}

return 0;

}

**OUTPUT**

