Date 01/02/23

# EXPERIMENT- 2

## CONVERTING REGULAR EXPRESSION TO NFA

AIM: To write program for converting regular expression to NFA.

ALGORITHM:

1. Start
2. Get input from the user.
3. Initialize seperate variable and functions for postfix, Dfa and NFA.
4. Create seperate methods for diffent operations like +,*.
5. By using switch case initialize different cases for the input.
6. For ' ' operator initialize different a seperate method by using various stack functions. do the same for + & *.
7. Regular expresions is in form a.b or a+b.
8. Display the output
9. Stop.

PROGRAM:

```
#include <stdio.h>
# include <string.h>
int main()
{
```

```
char reg[20]; int q[20][3], i=0, j=1, len, a, b;
for (a=0; a<20; a++) for (b=0; b<3; b++) q[a][b]=0;
scanf("%s", reg);
printf("Given regular exp: %s\n", reg);
len = strlen(reg);
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[j+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[i][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; }
```

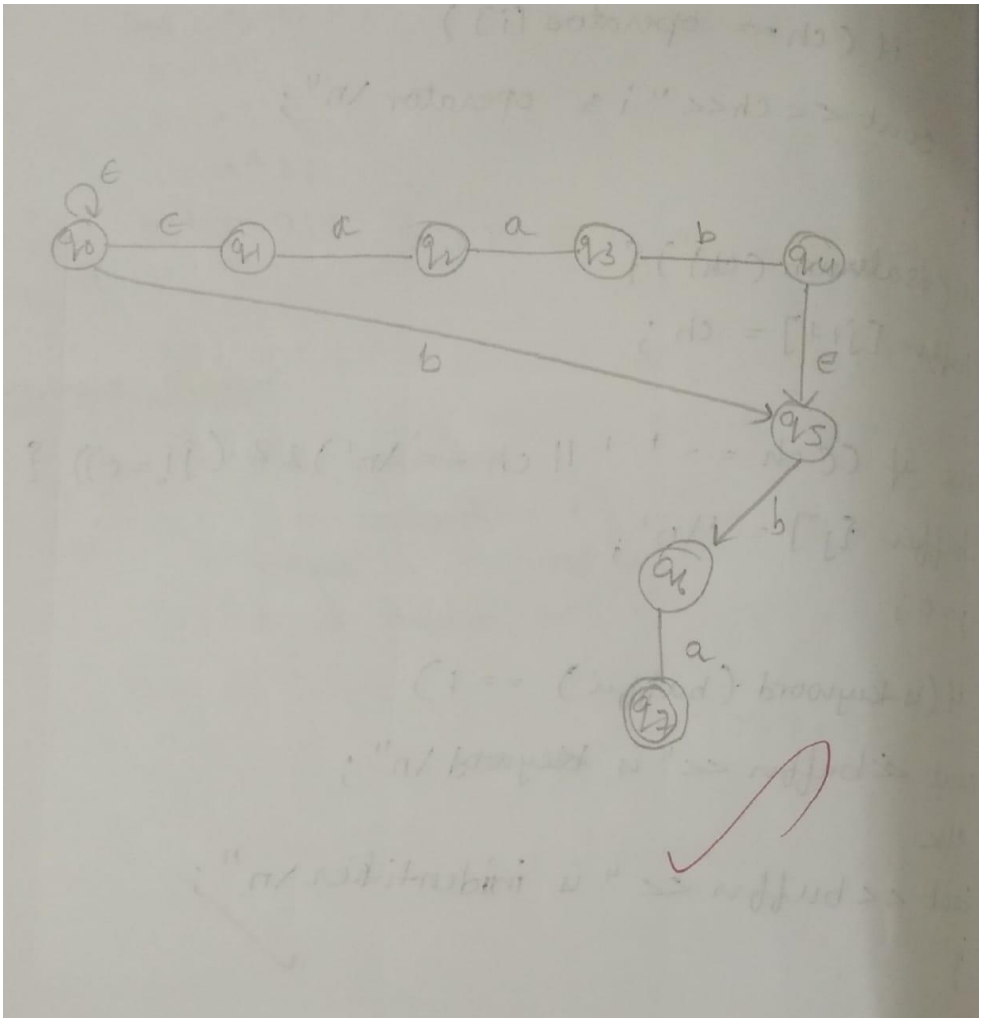
RA2011028010099
DURGA CHANDANA SREE M

output :

$a(a+b)^* ba$

Transition table

| Current state | Input | Next state |
|---|---|---|
| q[0] | e | q[5], q[1] |
| q[1] | a | q[2] |
| q[2] | a | q[3] |
| q[3] | b | [q4] |
| q[4] | e | q[5], q[1] |
| q[5] | b | q[6] |
| q[6] | a | q[7] |




```
#include<stdio.h>
#include<string.h>
int main()
{
char reg[20]; int q[20][3],i=0,j=1,len,a,b;
for(a=0;a<20;a++) for(b=0;b<3;b++) q[a][b]=0;
scanf("%s",reg);
printf("Given regular expression: %s\n",reg);
len=strlen(reg);
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++;
```

```
/tmp/DLS8L4twhv.o
a(a+b)*ba
Given regular expression: a(a+b)*ba
Transition Table
_____________________________________
Current State | Input | Next State
_____________________________________

q[0]    | e | q[5] , q[1]
q[1]    | a | q[2]
q[2]    | a | q[3]
q[3]    | b | q[4]
q[4]    | e | q[5] , q[1]
q[5]    | b | q[6]
q[6]    | a | q[7]
_____________________________________
```

RA2011028010099
DURGA CHANDANA SREE M

```
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++; }
printf("\n\t\tTransition table\n");
printf("Current state\t Input\t Next state");
for(i=0; i<=j; i++)
{ if(q[i][0]!=0) Printf("\n q[%d]\t", i);
if(q[i][1]!=0) Printf("\n q[%d]\t b\t q[%d]", i, q[i][1]);
if(q[i][2]!=0)
{ if(q[i][2]<10) Printf("\n q[%d]\t e\t q[%d]", i, q[i][2]);
else printf("\n q[%d]\t e\t q[%d], q[%d]", i, q[i][2]);
}}
return 0; }
```

RESULT: Hence the program to convert regular expression to NFA is implemented successfully.

RA2011028010099
DURGA CHANDANA SREE M