EXP NO : 2

Reg no: 192011478

Name: c .ramakrishna

NON-DETERMINISTIC FINITE AUTOMATA (NFA)

AIM :

To write a C program to simulate a Non-Deterministic Finite Automata.

ALGORTIHM :

1. Get the following as input from the user.

i. Number of states in the NFA

ii. Number of symbols in the input alphabet and the symbols

iii. Number of final states and their names

2. Declare a 3-dimensional matrix to store the transitions and initialize all the entries with -1

3. Get the transitions from every state for every input symbol from the user and store it in the matrix. For example, consider the NFA shown below.

There are two input symbols a and b. As the array index always starts with 0, we assume 0th symbol is a and 1st symbol is b. The transitions will be stored in the matrix as follows: From state 0, for input a, there are two transitions to state 0 and 1, which can be stored in the matrix as m[0][0][0]=0 m[0][0][1]=1 Similarly, the other transitions can be stored as follows: m[0][1][0]=0 (From state 0, for input b, one transition is to state 0) m[0][1][1]=2 (From state 0, for input b, next transition is to state 2) m[1][1][0]=3 (From state 1, for input b, move to state 3) m[2][0][0]=3 (From state 2, for input a, move to state 3) m[3][0][0]=3 (From state 3, for input a, move to state 3) m[3][1][0]=3 (From state 3, for input b, move to state 3) All the other entries in the matrix will be -1 indicating no moves

4. Get the input string from the user.

5. Find the length of the input string.

6. Read the input string character by character.

7. Repeat step 8 for every character

PROGRAM :

```
#include<stdio.h>

 #include<string.h>

int main()
```

```
{
 int i,j,k,l,m,next_state[20],n,mat[10][10][10],flag,p;
int num_states,final_state[5],num_symbols,num_final;
 int present_state[20],prev_trans,new_trans;
 char ch,input[20];
int symbol[5],inp,inp1;
 printf("How many states in the NFA : ");
 scanf("%d",&num_states);
 printf("How many symbols in the input alphabet : ");
 scanf("%d",&num_symbols);
 for(i=0;i<num_symbols;i++)
{
printf("Enter the input symbol %d : ",i+1);
scanf("%d",&symbol[i]);
}
printf("How many final states : ");
scanf("%d",&num_final);
for(i=0;i<num_final;i++)
{
printf("Enter the final state %d : ",i+1);
scanf("%d",&final_state[i]);
}
//Initialize all entries with -1 in Transition table
for(i=0;i<10;i++)
{
for(j=0;j<10;j++)
{
```

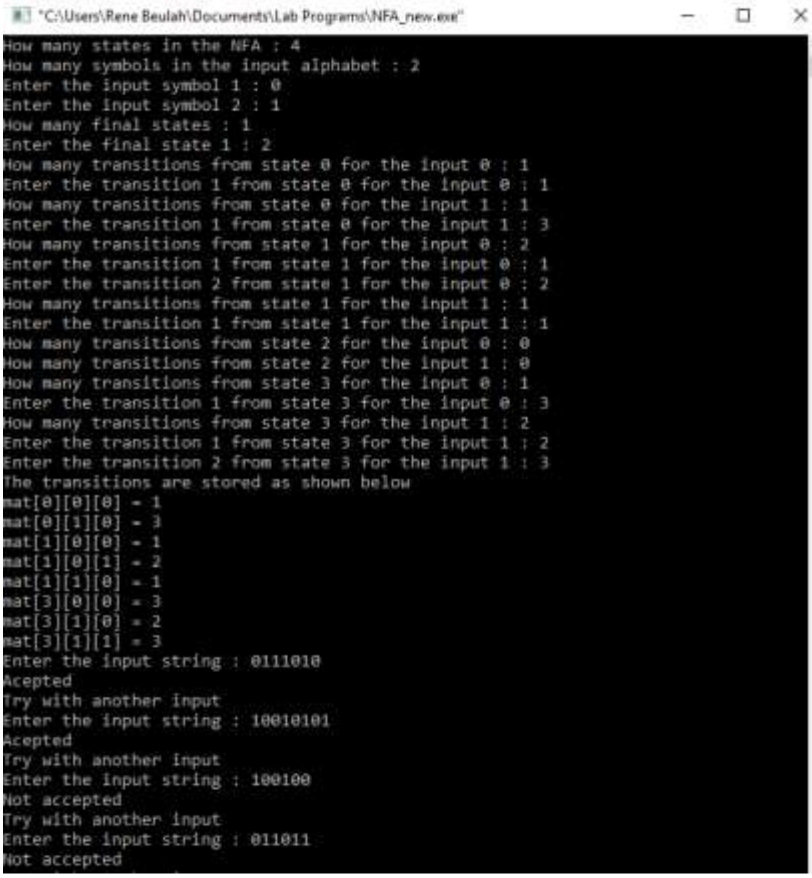
```
for(k=0;k<10;k++)
{
mat[i][j][k]=-1;
}
}
}
//Get input from the user and fill the 3D transition table
for(i=0;i<num_states;i++)
{
for(j=0;j<num_symbols;j++)
{
printf("How many transitions from state %d for the input %d :
",i,symbol[j]);
scanf("%d",&n);
for(k=0;k<n;k++)
{
printf("Enter the transition %d from state %d for the input
%d : ",k+1,i,symbol[j]);
scanf("%d",&mat[i][j][k]);
}
}
}
printf("The transitions are stored as shown below\n");
 for(i=0;i<10;i++)
{
for(j=0;j<10;j++)
{
```

```
for(k=0;k<10;k++)
{
 if(mat[i][j][k]!=-1)
 printf("mat[%d][%d][%d] = %d\n",i,j,k,mat[i][j][k]);
}
}
}
while(1)
{
printf("Enter the input string : ");
scanf("%s",input);
present_state[0]=0;
prev_trans=1;
l=strlen(input);
for(i=0;i<l;i++)
{
 if(input[i]=='0')
 inp1=0;
 else if(input[i]=='1')
 inp1=1;
 else
 {
 printf("Invalid input\n");
 exit(0);
 }
 for(m=0;m<num_symbols;m++)
{
```

```
if(inp1==symbol[m])
{
inp=m;
break;
}
}
new_trans=0;
for(j=0;j<prev_trans;j++)
{
k=0;
p=present_state[j];
while(mat[p][inp][k]!=-1)
{
next_state[new_trans++]=mat[p][inp][k];
k++;
}
}
for(j=0;j<new_trans;j++)
{
present_state[j]=next_state[j];
}
prev_trans=new_trans;
}
flag=0;
for(i=0;i<prev_trans;i++)
{
for(j=0;j<num_final;j++)
```

```
{
if(present_state[i]==final_state[j])
{
flag=1;
break;
}
}
}
if(flag==1)
printf("Acepted\n");
else
printf("Not accepted\n");
printf("Try with another input\n");
}
}
```

OUTPUT:



```
How many states in the NFA : 4
How many symbols in the input alphabet : 2
Enter the input symbol 1 : 0
Enter the input symbol 2 : 1
How many final states : 1
Enter the final state 1 : 2
How many transitions from state 0 for the input 0 : 1
Enter the transition 1 from state 0 for the input 0 : 1
How many transitions from state 0 for the input 1 : 1
Enter the transition 1 from state 0 for the input 1 : 3
How many transitions from state 1 for the input 0 : 2
Enter the transition 1 from state 1 for the input 0 : 1
Enter the transition 2 from state 1 for the input 0 : 2
How many transitions from state 1 for the input 1 : 1
Enter the transition 1 from state 1 for the input 1 : 1
How many transitions from state 2 for the input 0 : 0
How many transitions from state 2 for the input 1 : 0
How many transitions from state 3 for the input 0 : 1
Enter the transition 1 from state 3 for the input 0 : 3
How many transitions from state 3 for the input 1 : 2
Enter the transition 1 from state 3 for the input 1 : 2
Enter the transition 2 from state 3 for the input 1 : 3
The transitions are stored as shown below
mat[0][0][0] = 1
mat[0][1][0] = 3
mat[1][0][0] = 1
mat[1][0][1] = 2
mat[1][1][0] = 1
mat[3][0][0] = 3
mat[3][1][0] = 2
mat[3][1][1] = 3
Enter the input string : 0111010
Acepted
Try with another input
Enter the input string : 10010101
Acepted
Try with another input
Enter the input string : 100100
Not accepted
Try with another input
Enter the input string : 011011
Not accepted
```

RESULT: Therefore stimulation of NFA has been successfully excecuted in C program.