**EXP NO : 2**

# NON-DETERMINISTIC FINITE AUTOMATA (NFA)

**AIM :**

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

PROAGRAM:-

#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");

}

}

