1. Write a C program to simulate a Non-Deterministic Finite Automata (NFA) for the given language representing strings that start with b and end with a

AIM : To write a C program to find ε-closure of a Non-Deterministic Finite Automata with ε-moves

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 including ε

iii. Input symbols iv. 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 3 states 0, 1, and 2 There are three input symbols ε, 0 and

1. As the array index always starts with 0, we assume 0th symbol is ε, 1st symbol is 0 and 2nd symbol is 1. The transitions will be stored in the matrix as follows: From state 0, for input ε, there is one transition to state 1, which can be stored in the matrix as m[0][0][0]=1 From state 0, for input 0, there is no transition. From state 0, for input 1, there is one transition to state
2. 1, whichcan be stored in the matrix as m[0][2][0]=1 Similarly, the other transitions can be stored as follows: m[1][0][0]=2 (From state 1, for input ε, the transition is to state

2) m[1][1][0]=1 (From state 1, for input 0, the transition is to state 1) All the other entries in the matrix will be -1 indicating no moves 4. Initialize a two-dimensional matrix e\_closure with -1 in all the entries.

5. ε-closure of a state q is defined as the set of all states that can be reached from state q using only ε-transitions. Example: Consider the NFA with ε-transitions given below: ε-closure(0)={0,1,2) ε-closure(1)={1,2} ε-closure(2)={2} Here, we see that ε-closure of every state contains that state first. So initialize the first entry of the array e\_closure with the same state. e\_closure(0,0)=0; e\_closure(1,0)=1; e\_closure(2,0)=2;

6. For every state i, find ε-closure as follows: If there is an ε-transition from state i to state j, add j to the matrix e\_closure[i]. Call the recursive function find\_e\_closure(j) and add the other states that are reachable from i using ε

7. For every state, print the ε-closure values The function find\_e\_closure(i) This function finds ε-closure of a state recursively by tracing all the εtransitions

PROGRAM:

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<ctype.h>

void validate();

int s\_table[3][2]={{5,1},{2,1},{2,1}},i,f;

char str[10];

int main(){

printf("enter the string :");

scanf("%s",str);

for(i=0;str[i]!='\0';i++)

if(!isalpha(str[i])){

f=1;

break;}

for(i=0;i<strlen(str);i++){

if((str[i]!='a')&&(str[i]!='b')){

f=1;

break;

}

}

if(f==1)

printf("INVALID CHARECTERS");

else

validate();

}

void validate(){

int l=strlen(str),c\_state=0,f\_state=2,x;

for(i=0;i<l;i++){

if(str[i]=='a')

x=0;

if(str[i]=='b')

x=1;

c\_state=s\_table[c\_state][x];

printf("-->%d",c\_state);

if(c\_state==5)

exit(0);

}

if(c\_state==f\_state)

printf("VALID STRING");

else

printf("INVALID STRING");

}

OUTPUT;

