11.Write a C program to find ε -closure for all the states in a Non-Deterministic Finite Automata (NFA) with ε -moves.

PROGRAM

#include <stdio.h>

#include <stdbool.h>

#define MAX\_STATES 100

// Structure to represent a state in the NFA

struct State {

int transitions[MAX\_STATES]; // Transitions on input symbols

int epsilonTransitions[MAX\_STATES]; // Transitions on epsilon

};

// Function to compute the ε-closure of a state in the NFA

void computeEpsilonClosure(int state, struct State nfa[], bool visited[], int numStates) {

// Mark the current state as visited

visited[state] = true;

// Process ε-transitions

for (int i = 0; i < nfa[state].epsilonTransitions[0]; ++i) {

int nextState = nfa[state].epsilonTransitions[i + 1];

if (!visited[nextState]) {

computeEpsilonClosure(nextState, nfa, visited, numStates);

}

}

}

// Function to print the ε-closure for all states in the NFA

void printEpsilonClosures(struct State nfa[], int numStates) {

for (int i = 0; i < numStates; ++i) {

bool visited[MAX\_STATES] = {false};

computeEpsilonClosure(i, nfa, visited, numStates);

printf("ε-closure(q%d): { ", i);

for (int j = 0; j < numStates; ++j) {

if (visited[j]) {

printf("q%d ", j);

}

}

printf("}\n");

}

}

int main() {

int numStates;

// Get the number of states in the NFA

printf("Enter the number of states in the NFA: ");

scanf("%d", &numStates);

struct State nfa[MAX\_STATES];

// Input transitions for each state

for (int i = 0; i < numStates; ++i) {

printf("Enter the number of transitions from state q%d on epsilon: ", i);

scanf("%d", &nfa[i].epsilonTransitions[0]);

printf("Enter the states reached from q%d on epsilon: ", i);

for (int j = 0; j < nfa[i].epsilonTransitions[0]; ++j) {

scanf("%d", &nfa[i].epsilonTransitions[j + 1]);

}

}

// Print the ε-closure for all states in the NFA

printEpsilonClosures(nfa, numStates);

return 0;

}