EXP NO 9

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

PROGRAM

#include <stdio.h>

#include <stdbool.h>

#include <string.h>

#define STATES 3 // Number of states in the NFA

// Transition function for the NFA

bool transition(int state, char symbol) {

if (state == 0 && symbol == 'o') {

return true;

} else if (state == 1 && symbol == '1') {

return true;

} else if (state == 1 && symbol == '0') {

return true; // Non-deterministic transition to handle epsilon

} else if (state == 2) {

return false; // Invalid state, the NFA should halt after reaching state 2

}

return false;

}

// NFA simulation function

bool simulateNFA(const char \*input) {

int currentStates[STATES] = {0}; // Initialize all states as reachable

for (int i = 0; input[i] != '\0'; i++) {

int nextStates[STATES] = {0}; // Temporary array to store possible next states

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

if (currentStates[j] != -1 && transition(currentStates[j], input[i])) {

if (transition(currentStates[j], '0')) {

nextStates[j] = currentStates[j]; // Keep the current state

} else {

nextStates[j] = currentStates[j] + 1; // Move to the next state

}

} else {

nextStates[j] = -1; // Mark state as unreachable

}

}

memcpy(currentStates, nextStates, sizeof(nextStates)); // Update current states

}

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

if (currentStates[i] == 1) {

return true; // String accepted if one of the states is 1

}

}

return false;

}

int main() {

char input[100];

printf("Enter a string: ");

scanf("%s", input);

if (simulateNFA(input)) {

printf("String accepted.\n");

} else {

printf("String not accepted.\n");

}

return 0;

}

OUTPUT:

