**OBJECT-05**

**Object:- Write a C program to calculate NFA with null transition into NFA without null transition.**

#include <stdio.h>

#include <stdlib.h>

#define MAX\_STATES 100

#define MAX\_ALPHABET 26

int n\_states, n\_alphabet;

int transition[MAX\_STATES][MAX\_ALPHABET][MAX\_STATES];

int epsilon\_closure[MAX\_STATES][MAX\_STATES];

void initialize() {

for (int i = 0; i < MAX\_STATES; i++) {

for (int j = 0; j < MAX\_ALPHABET; j++) {

for (int k = 0; k < MAX\_STATES; k++) {

transition[i][j][k] = -1;

}

}

}

}

void add\_transition(int from, char symbol, int to) {

int symbol\_index = symbol - 'a'; // Assuming alphabet is 'a' to 'z'

transition[from][symbol\_index][to] = 1;

}

void add\_epsilon\_closure(int state, int closure\_state) {

epsilon\_closure[state][closure\_state] = 1;

}

void compute\_epsilon\_closure() {

for (int state = 0; state < n\_states; state++) {

for (int closure\_state = 0; closure\_state < n\_states; closure\_state++) {

if (epsilon\_closure[closure\_state][state]) {

for (int new\_state = 0; new\_state < n\_states; new\_state++) {

if (epsilon\_closure[state][new\_state]) {

epsilon\_closure[closure\_state][new\_state] = 1;

}

}

}

}

}

}

void convert\_to\_nfa() {

for (int state = 0; state < n\_states; state++) {

for (int symbol = 0; symbol < n\_alphabet; symbol++) {

for (int to\_state = 0; to\_state < n\_states; to\_state++) {

if (transition[state][symbol][to\_state]) {

for (int closure\_state = 0; closure\_state < n\_states; closure\_state++) {

if (epsilon\_closure[state][closure\_state]) {

add\_transition(closure\_state, symbol + 'a', to\_state);

}

}

}

}

}

}

}

void print\_nfa() {

printf("NFA without epsilon transitions:\n");

for (int state = 0; state < n\_states; state++) {

for (int symbol = 0; symbol < n\_alphabet; symbol++) {

for (int to\_state = 0; to\_state < n\_states; to\_state++) {

if (transition[state][symbol][to\_state] == 1) {

printf("q%d -- %c --> q%d\n", state, symbol + 'a', to\_state);

}

}

}

}

}

int main() {

initialize();

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

scanf("%d", &n\_states);

printf("Enter the number of alphabets (a-z): ");

scanf("%d", &n\_alphabet);

for (int state = 0; state < n\_states; state++) {

printf("Enter transitions from state q%d (e.g., a 0 b for 'a' to q0, '0' to q1): ", state);

char symbol;

int to\_state;

while (1) {

scanf(" %c", &symbol); // Read alphabet symbol

if (symbol == 'e') {

add\_epsilon\_closure(state, state);

continue;

}

if (symbol == '$') {

break;

}

scanf("%d", &to\_state); // Read destination state

add\_transition(state, symbol, to\_state);

}

}

compute\_epsilon\_closure();

convert\_to\_nfa();

print\_nfa();

return 0;

}

**OUTPUT**

Enter the number of states: 2

Enter the number of alphabets (a-z): 2

Enter transitions from state q0 (e.g., a 0 b for 'a' to q0, '0' to q1): a 0 b e $

Enter transitions from state q1 (e.g., a 0 b for 'a' to q0, '0' to q1): b 0 e $

NFA without epsilon transitions:

q0 -- a --> q0

q0 -- a --> q1

q0 -- b --> q0

q0 -- b --> q1

q1 -- a --> q0

q1 -- a --> q1

q1 -- b --> q0

q1 -- b --> q1