RAJSHAHI UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Lab report: 03
Course No.: CSE 2206

Date of Experiment: 15.12.2018
Date of Submission: 22.01.2019

Submitted to:
Sadia Zaman Mishu
Assistant Professor,
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology

Submitted by:
Riyad Morshed Shoeb
Roll No: 1603013
Section: A
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology

Problem: Find ϵ − closure of each state of the given ϵ − NFA-

	ϵ	a	b	c
→ p	∅	$\{p\}$	$\{q\}$	$\{r\}$
q	$\{p\}$	$\{q\}$	$\{r\}$	∅
*r	$\{q\}$	$\{r\}$	∅	$\{p\}$

Theory:

We define the ϵ − closure ECLOSE(q) recursively, as follows:
BASIS: State q is in ECLOSE(q).
INDUCTION: If state p is in ECLOSE(q), and there is a transition from state p to state r labeled ϵ, then r is in ECLOSE(q). More precisely, if δ is the transition functions of the $\epsilon - NFA$ involved, and p is in ECLOSE(q), then ECLOSE(q) also contains all the states in $\delta(p, \epsilon)$.

Code:

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
using namespace std;
int main(void)
{
    int NUMBEROF_INPUT_SYMBOL;
    int NUMBEROF_STATES;
    int i, j;
    cout << "Enter the number of Input symbol: ";
    cin >> NUMBEROF_INPUT_SYMBOL;
    char INPUT_SYMBOL[NUMBEROF_INPUT_SYMBOL];
    cout << "Enter the input symbols: " << endl;
    for(i=1; i<=NUMBEROF_INPUT_SYMBOL; i++)
        cin >> INPUT_SYMBOL[i];
    cout << "Enter the number of states :";
    cin >> NUMBEROF_STATES;
    char STATES[NUMBEROF_STATES];
    cout << "Enter the states:" << endl;
    for(i=0; i<NUMBEROF_STATES; i++)
        cin >> STATES[i];
    char TABLE[NUMBEROF_STATES][NUMBEROF_INPUT_SYMBOL];
    cout << "Enter the transition table (type 0 for empty set):" <<
endl;
    for(i=0; i<NUMBEROF_STATES; i++)
    {
        cout << "Enter transition for " << STATES[i] << endl;
        for(j=0; j<NUMBEROF_INPUT_SYMBOL; j++)
            cin >> TABLE[i][j];
    }
    cout << "e-closure of each state:\n";
    for(i=0; i<NUMBEROF_STATES; i++)
    {
        cout << "e-close(" << STATES[i] << ") = {" << STATES[i];
        if(TABLE[i][0] != '0')
            cout << ", " << TABLE[i][0];
```

```
        for(j=0; j<NUMBEROF_STATES; j++)
        {
            if(TABLE[i][0] == STATES[j])
                break;
        }
        if(TABLE[j][0] != '0' && j<NUMBEROF_STATES)
            cout << ", " << TABLE[j][0];
        cout << "}" << endl;
    }
}
```

Output:

```
epsilon
a
b
c
Enter the number of states: 3
Enter the states:
p
q
r
Enter the transition table (type 0 for empty set):
Enter transition for p
0
p
q
r
Enter transition for q
p
q
r
0
Enter transition for r
q
r
0
p
e-closure of each state:
e-close(p) = {p}
e-close(q) = {q, p}
e-close(r) = {r, q, p}

Press any key to continue . . .
```