# Rajshahi University of Engineering & Technology

Department of Computer Science & Engineering

Lab Report 01

CSE 2206: Sessional Based on CSE 2205

Submitted to: Sadia Zaman Mishu

Assistant Professor, Dept. of CSE

Date: November 23, 2018

Submitted by
Fuad Al Abir
Roll: 1603021
Section: A
Dept. of CSE

# Rajshahi University of Engineering & Technology

Department of Computer Science & Engineering

Lab Report 02

CSE 2206: Sessional Based on CSE 2205

Submitted to: Sadia Zaman Mishu

Assistant Professor, Dept. of CSE

Date: December 15, 2018

Submitted by

Fuad Al Abir

Roll: 1603021

Section: A

Dept. of CSE

# Rajshahi University of Engineering & Technology

Department of Computer Science & Engineering

Lab Report 03

CSE 2206: Sessional Based on CSE 2205

Submitted to: Sadia Zaman Mishu

Assistant Professor, Dept. of CSE

Date: December 15, 2018

Submitted by
Fuad Al Abir
Roll: 1603021
Section: A
Dept. of CSE

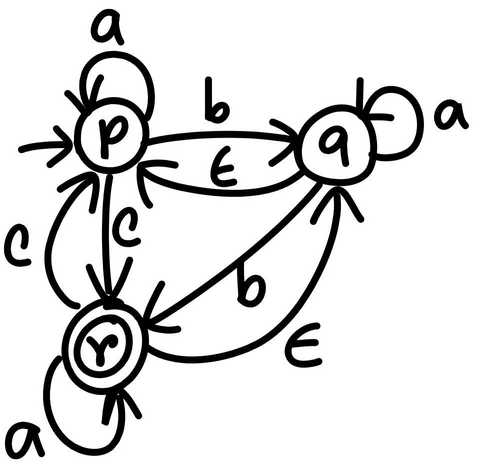
Sessional 3 – Cycle 8 – Problem A

(a) Compute the ε-closure of each state

(b) Show if any string of length three or less is accepted by the automation denoted by the following transition table.

| | ε | a | b | C |
|---|---|---|---|---|
| -> p | Φ | {p} | {q} | {r} |
| q | {p} | {q} | {r} | Φ |
| *r | {q} | {r} | Φ | {p} |

**Theory:** From the given transition table, we obtain the following ε-NFA.



Now, from the diagram, we obtain the ε-closures of each states and hence determine if a string of length of three or less is accepted or not.

Code:

```
/*-----------------------------
    I N T R O D U C T I O N
-------------------------------
Author:       Fuad Al Abir
Date:         December 16, 2018
Name:         e-closure.cpp
Objective:    (a) This program determines the ε-closures of each state and therefore
              (b) find if a string of length atmost 3 is accepted by the ε-NFA or not.
*/

#include <iostream>
#include <string.h>

using namespace std;

void _p(char i);
void _q(char i);
void _r(char i);
```

```
char p(char inp);
char q(char inp);
char r(char inp);

void ECLOSER(char i);
void checkString(string inpStr);

string input_string;
char current_state = 'p';

char p(char inp)
{
    if (inp == 'e') current_state = 'N';
    else if (inp == 'a') current_state = 'p';
    else if (inp == 'b') current_state = 'q';
    else if (inp == 'c') current_state = 'r';
    return current_state;
}

char q(char inp)
{
    if (inp == 'e') current_state = 'p';
    else if (inp == 'a') current_state = 'q';
    else if (inp == 'b') current_state = 'r';
    else if (inp == 'c') current_state = 'N';
    return current_state;
}

char r(char inp)
{
    if (inp == 'e') current_state = 'q';
    else if (inp == 'a') current_state = 'r';
    else if (inp == 'b') current_state = 'N';
    else if (inp == 'c') current_state = 'p';
    return current_state;
}

void _p(char i)
{
    cout << 'p';
    if(i == 'e')
    {
        if (p(i) == 'N') return;
        else
        {
            if (p(i) == 'p') _p(i);
            else if (p(i) == 'q') _q(i);
            else if (p(i) == 'r') _r(i);
        }
    }
}

void _q(char i)
{
    cout << 'q';
    if(i == 'e')
    {
        if (q(i) == 'N') return;
        else
        {
            if (q(i) == 'p') _p(i);
            else if (q(i) == 'q') _q(i);
            else if (q(i) == 'r') _r(i);
        }
    }
}
```

```
void _r(char i)
{
    cout << 'r';
    if(i == 'e')
    {
        if (r(i) == 'N') return;
        else
        {
            if (r(i) == 'p') _p(i);
            else if (r(i) == 'q') _q(i);
            else if (r(i) == 'r') _r(i);
        }
    }
}

void ECLOSER(char i)
{
    cout << "E-CLOSER(" << i << "): ";
    if (i == 'p') _p('e');
    else if (i == 'q') _q('e');
    else if (i == 'r') _r('e');
    cout << endl;
}

void checkString(string inpStr)
{
    int c_a = 0, c_b = 0, c_c = 0;
    for (int i = 0; i < inpStr.length(); i++)
    {
        if (inpStr[i] == 'a') c_a++;
        else if (inpStr[i] == 'b') c_b++;
        else if (inpStr[i] == 'c') c_c++;
    }
    if(c_c == 1 || (c_b >= 2 && c_c == 1 && inpStr[inpStr.length() - 1] == 'c') ||
c_b >= 2 && c_c == 0) cout << "ACCEPTED" << endl;
    else cout << "REJECTED" << endl;
}

int main()
{
    cout << endl;
    ECLOSER('p');
    ECLOSER('q');
    ECLOSER('r');

    while (1)
    {
        cout << "\nEnter string of length atmost three: ";
        cin >> input_string;
        int s = input_string.length();

        if (s > 3)
        {
            cout << "\nInput String length is greater than 3.\nProgram is
Terminating.\n";
            return -1;
        }
        else checkString(input_string);
    }

    return 0;
}
```

Input/Output:

```
E-CLOSER(p): p
E-CLOSER(q): qp
E-CLOSER(r): rqp

Enter string of length atmost three: aca
ACCEPTED

Enter string of length atmost three: acc
REJECTED

Enter string of length atmost three: aac
ACCEPTED

Enter string of length atmost three: ab
REJECTED

Enter string of length atmost three: abb
ACCEPTED

Enter string of length atmost three: bb
ACCEPTED

Enter string of length atmost three: abba

Input String length is greater than 3.
Program is Terminating.
```

Discussion: From the diagram, we've determined that the ε-NFA accepts all the string consisting of one 'c' or more than one 'b' without the string starting by the chacracter 'c' as the accepting state has no transaction when the input symbol is 'b' or more than one 'b' without and any 'c' in the string.