RAJSHAHI UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Lab report: 01
Course No.: CSE 2206

Date of Experiment: 06.11.2018
Date of Submission: 24.11.2018

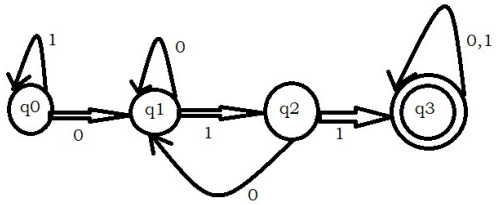
Submitted to:
Sadia Zaman Mishu
Assistant Professor,
Department of Computer Science and Engineering
Rajshahi University of Engineering and Technology

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

Problem: Design a DFA that accepts a 01 string that has a substring '011' in it.

Theory:

The DFA has a start state of q_0. After it has found '011', it reaches q_3 and then whatever the input symbol is, the transition function returns q_3. Here, q_3 is the accepting state. The transition diagram of the DFA is demonstrated below-



Code:

```
#include <iostream>
#include <cstdio>
#include <cstdlib>
#include <cstring>
#include <vector>
#include <iterator>
using namespace std;

int main(void)
{
  string input_string;
  int string_length;
  int string_iterator;
  string state;
  vector <string> path;
  vector <string>::iterator path_iterator;

  cout << "Enter string: ";
  cin >> input_string;

  cout << "Transition table:\n\
       || 0  | 1\n\
  ===============\n\
  ->q0 || q1 | q0\n\
  _______________\n\
    q1 || q1 | q2\n\
  _______________\n\
    q2 || q1 | q3\n\
  _______________\n\
   *q3 || q3 | q3" << endl;

  state = "q0";
  path.push_back(state);
  string_length = input_string.size();
  for(string_iterator=0;string_iterator<string_length;
string_iterator++)
```

```
  {
    if(state == "q0")
    {
      if(input_string[string_iterator] == '0')
        state = "q1";
      else
        state = "q0";
    }
    else if(state == "q1")
    {
      if(input_string[string_iterator] == '0')
        state = "q1";
      else
        state = "q2";
    }
    else if(state == "q2")
    {
      if(input_string[string_iterator] == '0')
        state = "q1";
      else
        state = "q3";
    }
    else
      state = "q3";
    path.push_back(state);
  }

  if(state == "q3")
    cout << "Accepted" << endl;
  else
    cout << "Not accepted" << endl;

  cout << "Path->";
    for(path_iterator=path.begin();       path_iterator!=path.end();
path_iterator++)
    cout << "->" <<*path_iterator;
  cout << endl;
}
```

Sample Input Output:

```
DFA_011
Enter string: 01010101010100101001
Transition table:
       || 0  | 1
  ===============
  ->q0 || q1 | q0
  _______________
    q1 || q1 | q2
  _______________
    q2 || q1 | q3
  _______________
   *q3 || q3 | q3
Not accepted
Path->->q0->q1->q2->q1->q2->q1->q2->q1->q2->q1->q2->q1->q2->q1->q2->q1->q1->q2->q1->q2->q1->q1->q2

Press any key to continue . . .
```

```
DFA_011
Enter string: 0010010011101101
Transition table:
       || 0  | 1
  ===============
  ->q0 || q1 | q0
  _______________
    q1 || q1 | q2
  _______________
    q2 || q1 | q3
  _______________
   *q3 || q3 | q3
Accepted
Path->->q0->q1->q1->q2->q1->q1->q2->q1->q1->q2->q3->q3->q3->q3->q3->q3->q3

Press any key to continue . . .
```