Rajshahi University of Engineering & Technology

CSE 2206: Sessional Based on CSE 2205

Lab Report 01

Date: November 23, 2018

Submitted to

Sadia Zaman Mishu

Instructor, CSE 2205 & CSE 2206

Lecturer, Dept. of CSE

Submitted by

Fuad Al Abir

Roll: 1603021

Section: A

Dept. of CSE

Sessional – Cycle 4 – Problem A

Design a DFA that accepts all the binary strings having substring "011".

Code:

```
/*-----------------------------
    I N T R O D U C T I O N
-----------------------------
Author:    Fuad Al Abir
Date:      November 12, 2018
Name:      dfa-011.cpp
Objective: This program simulates a DFA design that accepts all the
           binary strings having substring "011".
*/
/*---------------------------
    H E A D E R   F I L E
---------------------------
Header: iostream
Reason: Input/Output stream
*/
#include <iostream>

using namespace std;

string input_string;
string current_state = "q0";

/*-----------------------------------------------
    U S E R   D E F I N E D   F U N C T I O N
-----------------------------------------------
Function:   char q0(char inp)/char q1(char inp)/char q2(char inp)/char q3(char inp)
Reason:     This function takes a character as input parameter and
            set current_state by its measure from it's very own
            state.
Function:   void checkString(string inpStr);
Reason:     This function is the main DFA function that takes the
            string as argumant and measures if it is an accepted
            string or not.
*/
char q0(char inp)
{
    if(inp == '0') current_state = "q1";
    else current_state = "q0";
    return current_state[1];
}
char q1(char inp)
{
    if(inp == '0') current_state = "q1";
    else current_state = "q2";
    return current_state[1];
}
char q2(char inp)
{
    if(inp == '0') current_state = "q1";
    else current_state = "q3";
```

```
    return current_state[1];
}
char q3(char inp)
{
    if(inp == '0') current_state = "q3";
    else current_state = "q3";
    return current_state[1];
}

void checkString(string inpStr)
{
    current_state = "q0";
    cout << "Initial State: " << current_state << endl;

    for(int i = 0; i < inpStr.size(); i++)
    {
        cout << "Input Char: " << inpStr[i] << " ";
        int state = current_state[1] - 48;
        cout << "\tCurrent State: " << current_state;
        switch(state)
        {
            case 0: q0(inpStr[i]); break;
            case 1: q1(inpStr[i]); break;
            case 2: q2(inpStr[i]); break;
            case 3: q3(inpStr[i]); break;
        }
        cout << "\tOutput State: " << current_state << endl;
    }
}
/*--------------------------------
    M A I N   F U N C T I O N
--------------------------------*/
int main()
{
    cout << "Input String: ";
    cin >> input_string;
    int flag = 0;

    /********************** Transition Table *************************/
    cout << "\n\t|\t0\t|\t1\n----------------------------------------
-\n";
    cout << "->q0\t|\tq"<< q0('0') <<"\t|\tq" << q0('1') << "\n
q1\t|\tq" << q1('0') << "\t|\tq" << q1('1') << "\n  q2\t|\tq" <<
q2('0') << "\t|\tq" << q2('1')
         <<"\n *q3\t|\tq" << q3('0') << "\t|\tq" << q3('1')
<<"\n\n";

    checkString(input_string);

    if(current_state == "q3") cout << "Accepting State: " <<
current_state << endl;
    else  cout << "Rejecting State: " << current_state << endl;
}
```

Input/Output:

```
Input String: 0010010

        |       0       |       1
----------------------------------------
->q0    |       q1      |       q0
  q1    |       q1      |       q2
  q2    |       q1      |       q3
 *q3    |       q3      |       q3

Initial State: q0
Input Char: 0   Current State: q0       Output State: q1
Input Char: 0   Current State: q1       Output State: q1
Input Char: 1   Current State: q1       Output State: q2
Input Char: 0   Current State: q2       Output State: q1
Input Char: 0   Current State: q1       Output State: q1
Input Char: 1   Current State: q1       Output State: q2
Input Char: 0   Current State: q2       Output State: q1
Rejecting State: q1
```

```
Input String: 1001101

        |       0       |       1
----------------------------------------
->q0    |       q1      |       q0
  q1    |       q1      |       q2
  q2    |       q1      |       q3
 *q3    |       q3      |       q3

Initial State: q0
Input Char: 1   Current State: q0       Output State: q0
Input Char: 0   Current State: q0       Output State: q1
Input Char: 0   Current State: q1       Output State: q1
Input Char: 1   Current State: q1       Output State: q2
Input Char: 1   Current State: q2       Output State: q3
Input Char: 0   Current State: q3       Output State: q3
Input Char: 1   Current State: q3       Output State: q3
Accepting State: q3
```