**­**

|  |  |
| --- | --- |
| **Total Marks:** | **04** |
| **Obtained Marks:** |  |

**Finite Automata Theory and Formal Languages**

**Assignment # 01**

**Last date of Submission: 3rd October 2024**

**Submitted To: Muhammad Nadeem Khokhar**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Student Name: Ubaid-Bin-Waris**

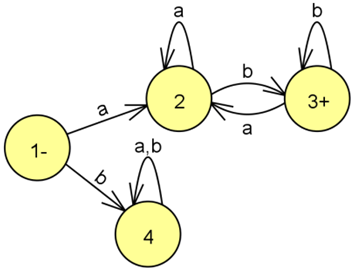
**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Reg Number: 2212416 \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

***Instructions****: Copied or shown assignments will be marked zero. Late submissions are not entertained in any case.*

**Question**

Consider the following transition diagram for a language defined over ∑={a,b} that accepts the strings starting with a and ending in b.



1. Write a C/C++ program that stays in an infinite loop, prompts the user for a string, terminates if the string is QUIT, and otherwise implements the DFA using the scheme that allows state to state function-call and recursion.
2. Give the source code and the runtime screen while testing the strings aabab, bbbaba, bba and abbb.

**Note:**

1. Change the filename to your ID, e.g. 2073105.doc
2. Upload the .doc on Google Classroom.
3. Do not use system calls.
4. Make sure that the output screen does not have colored/black background.
5. Poor indentation and wrong format will result in deduction of marks.

**Solution**

#include <iostream>

using namespace std;

string language;

int index;

int current\_state;

void state\_1()

{

if (index >= language.length())

{

cout << "\n\t" << language << " : String NOT Accpected ! ! ! \n\n";

return;

}

if (language[index] == 'a')

current\_state = 2;

else if (language[index] == 'b')

current\_state = 4;

else

current\_state = -1;

}

void state\_2()

{

if (index >= language.length())

{

cout << "\n\t" << language << " : String NOT Accpected ! ! ! \n\n";

return;

}

if (language[index] == 'a')

current\_state = 2;

else if (language[index] == 'b')

current\_state = 3;

else

current\_state = -1;

}

void state\_3()

{

if (index >= language.length())

{

cout << "\n\t" << language << " : String Accpected > > >\n\n";

return;

}

if (language[index] == 'a')

current\_state = 2;

else if (language[index] == 'b')

current\_state = 3;

else

current\_state = -1;

}

void state\_4()

{

if (index >= language.length())

{

cout << "\n\t" << language << " : String NOT Accpected ! ! ! \n\n";

return;

}

if (language[index] == 'a')

current\_state = 4;

else if (language[index] == 'b')

current\_state = 4;

else

current\_state = -1;

}

int main()

{

while (true)

{

index = 0;

current\_state = 1;

cout << "Enter 'QUIT' to stop the program. ";

cout << "Enter a string : ";

cin >> language;

if (language == "QUIT")

{

cout << "\nProgram is Quiting ! ! ! \n";

return 0;

}

while (index < language.length() + 1)

{

switch (current\_state)

{

case -1:

cout << "\n\t" << language << " : String Contains Invalid Characters ! ! ! \n\n";

index = language.length() + 1;

break;

case 1:

state\_1();

break;

case 2:

state\_2();

break;

case 3:

state\_3();

break;

case 4:

state\_4();

break;

default:

cout << "Code have some Errors and working ! ! !" << endl;

return 0;

}

index++;

}

}

}

**Output**

