



National College of Computer Studies

Paknajol, Kathmandu

Three vertical lines of varying lengths are positioned in the center of the page, above the title. The longest line is in the middle, with two shorter lines on either side.

Lab Report on Theory of Computation

Submitted by:

Atullya Maharjan

BSc. CSIT 4th Semester

Roll. No: 05

Submitted to:

Prashant Gautam

NCCS

Lab 2

1. WAP to design a DFA for the language of string over {0,1} in which each string end with 11

```
#include <iostream>
#include <string>
using namespace std;
int main(){
    string str;
    char state = 0;
    cout << "Enter the string: ";
    cin >> str;
    for (int i = 0; i < str.length(); i++){
        if (str[i] != '0' && str[i] != '1') {
            cout << "String not accepted.\nPlease enter a string over {0,1}" <<
endl;
            return 0;
        }
        if (state == 0 && str[i] == '0')
            state = 0;
        else if (state == 0 && str[i] == '1')
            state = 1;
        else if (state == 1 && str[i] == '0')
            state = 0;
        else if (state == 1 && str[i] == '1')
```

```
        state = 2;
    else if (state == 2 && str[i] == '0')
        state = 0;
    else if (state == 2 && str[i] == '1')
        state = 1;
}
if (state == 2)
    cout << "String accepted";
else
    cout << "String not accepted";
return 0;
}
```

```
Enter the string: 11011
String accepted
```

```
Enter the string: 00100
String not accepted
```

2. WAP to design a DFA accepting the string over {a,b} such that each string does not end with ab

```
#include <iostream>
#include <string>
using namespace std;

int main()
{
    string str;
    char state = 0; // initial state (q0)
    cout << "Enter the string: ";
    cin >> str;
    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] != 'a' && str[i] != 'b')
        {
            cout << "String not accepted.\nPlease enter a string over {a,b}"
<< endl;
            return 0;
        }

        if (state == 0 && str[i] == 'a')
            state = 1;
        else if (state == 0 && str[i] == 'b')
            state = 0;
        else if (state == 1 && str[i] == 'a')
            state = 1;
        else if (state == 1 && str[i] == 'b')
            state = 2;
        else if (state == 2 && str[i] == 'a')
            state = 1;
        else if (state == 2 && str[i] == 'b')
            state = 0;
    }
}
```

```
if (state == 0 || state == 1)
    cout << "String accepted";
else
    cout << "String not accepted";

return 0;
}
```

```
Enter the string: ababa
String accepted
-----
```

```
Enter the string: ababab
String not accepted
-----
```

3. WAP to design a DFA for the language of string over {a,b} such that each string contain “aba” as substring

```
#include <iostream>

#include <string>

using namespace std;

int main()
{
    string str;
    char state = 0; // initial state (q0)
    cout << "Enter the string: ";
    cin >> str;
    for (int i = 0; i < str.length(); i++)
    {
        if (str[i] != 'a' && str[i] != 'b')
        {
            cout << "String not accepted.\nPlease enter a string over {a,b}" <<
endl;
            return 0;
        }
        if (state == 0 && str[i] == 'a')
            state = 1;
        else if (state == 0 && str[i] == 'b')
            state = 0;
        else if (state == 1 && str[i] == 'a')
            state = 1;
```

```

        else if (state == 1 && str[i] == 'b')
            state = 2;
        else if (state == 2 && str[i] == 'a')
            state = 3;
        else if (state == 2 && str[i] == 'b')
            state = 0;
        else if (state == 3 && str[i] == 'a')
            state = 3;
        else if (state == 3 && str[i] == 'b')
            state = 3;
    }

    if (state == 3)
        cout << "String accepted";
    else
        cout << "String not accepted";

    return 0;
}

```

```

Enter the string: aaaaaaabbbbbbbba
String not accepted
-----

```

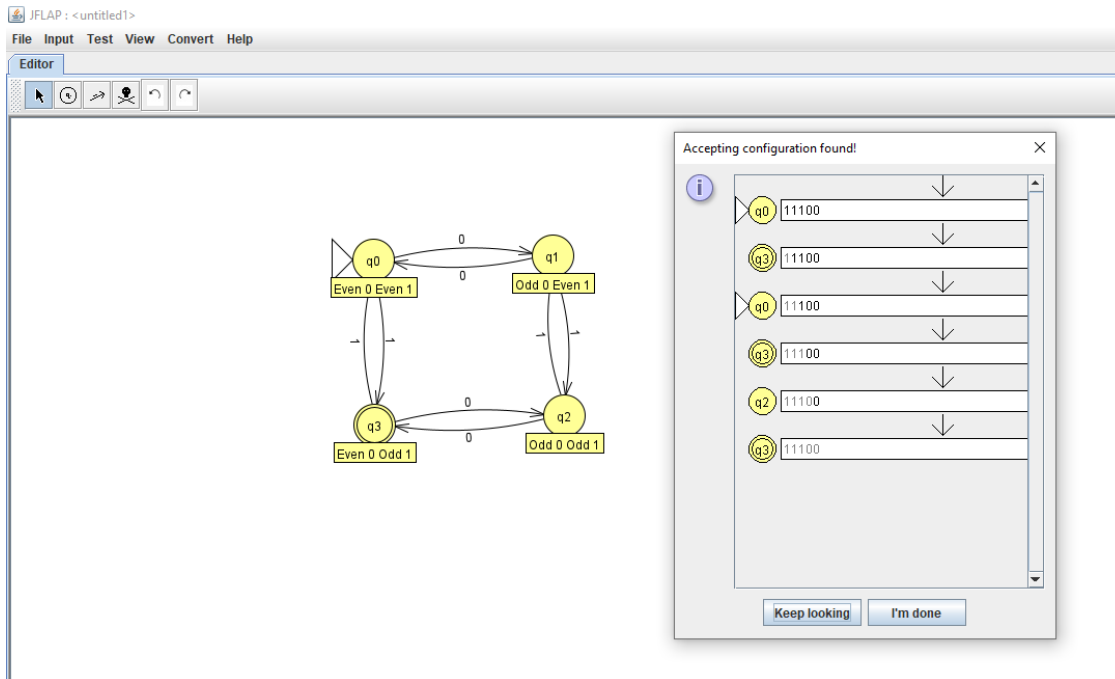
```

Enter the string: aabbabab
String accepted
-----

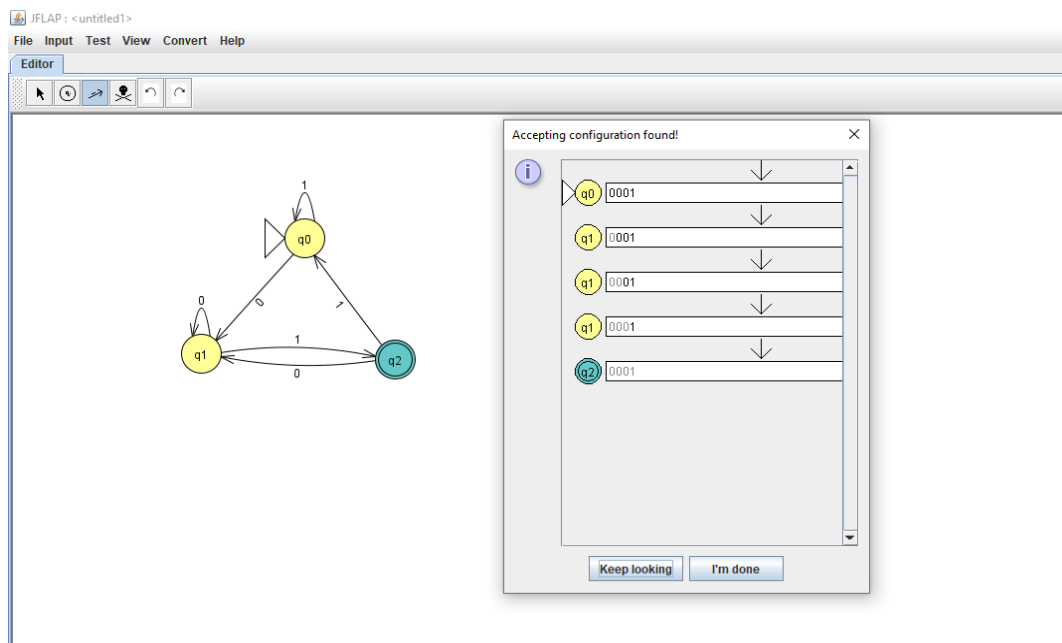
```

LAB 1

1 Construct a dfa over alphabet $\Sigma=\{0,1\}$ that accepts string with Even 0, Odd 1.



2. Construct a dfa over alphabet $\Sigma=\{0,1\}$ that accepts string ending with 01.



3. Construct a nfa over alphabet $\Sigma=\{0,1\}$ that accepts string ending with 01.

