# System Software (CS306)

## Assignment - 4

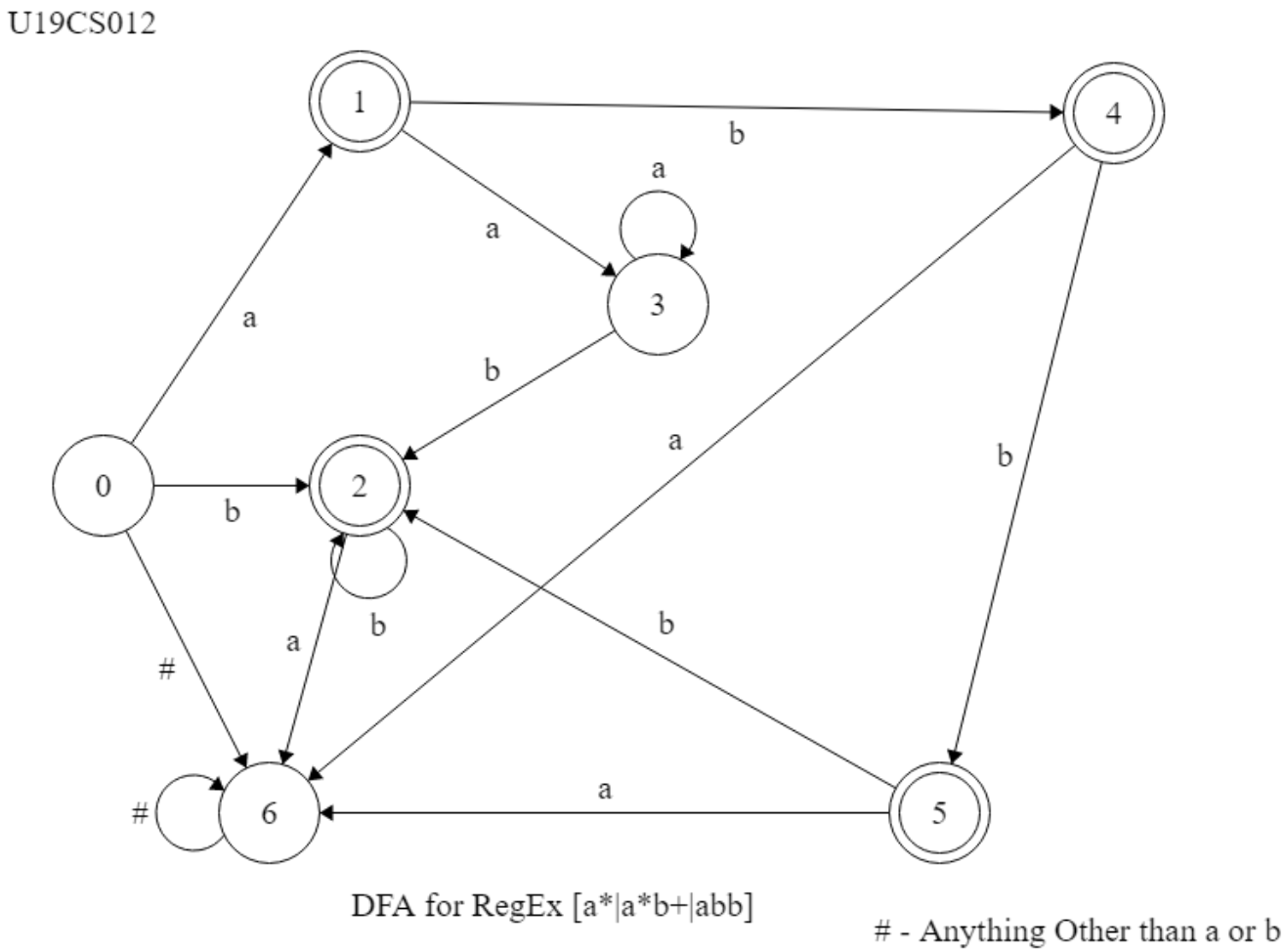
## U19CS012

1) Write a C program to recognize strings under 'a*', 'a*b+', 'abb'.

Valid Strings

| Pattern | Examples |
|---|---|
| a* | ε, a, aa, aaa, aaaa, aaaaa, ... |
| a*b+ | b, ab, bb, abb, bbb, abbb, bbbb, abbbb, bbbbb, abbbbb, ... |
| abb | abb |

DFA for Regular Expression - [a*|a*b+|abb]




Tool Used - https://www.cs.unc.edu/~otternes/comp455/fsm_designer/

## Code:

```cpp
#include <iostream>
#include <string>
// U19CS012 [BHAGYA VINOD RANA]

using namespace std;

int main()
{
    string s;
    char c;
    int state = 0, i = 0;

    cout << "Enter String : ";
    getline(cin, s);

    int n = s.length();

    // Corner Case [For Empty String] && "aa" string acc to DFA
    if (n == 0)
    {
        cout << "Ans : " << s << " is Accepted under RegEx [a*]\n\n";
        return 0;
    }

    // Iterate till the End of the String
    while (i < n)
    {
        // Check the State
        switch (state)
        {
        case 0:
            c = s[i++];
            // If 'a' occurs, then Transition from State '0' to '1'
            if (c == 'a')
                state = 1;
            // If 'b' occurs, then Transition from State '0' to '2'
            else if (c == 'b')
                state = 2;
            else
                state = 6;
            break;
        case 1:
            c = s[i++];
            // If 'a' occurs, then Transition from State '1' to '3'
            if (c == 'a')
                state = 3;
            // If 'b' occurs, then Transition from State '1' to '4'
            else if (c == 'b')
```

```
            state = 4;
        else
            state = 6;
        break;
    case 2:
        c = s[i++];
        // If 'a' occurs, then Transition from State '2' to '6'
        if (c == 'a')
            state = 6;
        // If 'b' occurs, then Transition from State '2' to '2'
        else if (c == 'b')
            state = 2;
        else
            state = 6;
        break;
    case 3:
        c = s[i++];
        // If 'a' occurs, then Transition from State '3' to '3'
        if (c == 'a')
            state = 3;
        // If 'b' occurs, then Transition from State '3' to '2'
        else if (c == 'b')
            state = 2;
        else
            state = 6;
        break;
    case 4:
        c = s[i++];
        // If 'a' occurs, then Transition from State '4' to '6'
        if (c == 'a')
            state = 6;
        // If 'b' occurs, then Transition from State '4' to '5'
        else if (c == 'b')
            state = 5;
        else
            state = 6;
        break;
    case 5:
        c = s[i++];
        // If 'a' occurs, then Transition from State '5' to '6'
        if (c == 'a')
            state = 6;
        // If 'b' occurs, then Transition from State '5' to '2'
        else if (c == 'b')
            state = 2;
        else
            state = 6;
        break;
    case 6:
        // Dead State
```

```
            c = s[i++];
            // If 'a' occurs, then Transition from State '6' to '6'
            if (c == 'a')
                state = 6;
            // If 'b' occurs, then Transition from State '6' to '6'
            else if (c == 'b')
                state = 6;
            else
                state = 6;
            break;
            // cout << "Invalid String Entered! [String Contains Other than letters 'a' and
'b'] \n\n";
        }
    }

    if (state == 1 || state == 3)
        cout << "Ans : " << s << " is Accepted under RegEx [a*]\n\n";
    else if ((state == 2) || (state == 4))
        cout << "Ans : " << s << " is Accepted under RegEx [a*b+]\n\n";
    else if (state == 5)
        cout << "Ans : " << s << " is Accepted under RegEx [abb]\n\n";
    else
        cout << "Ans : " << s << " is Not Accepted under Any RegEx [a*|a*b+|abb]\n\n";

    return 0;
}
```

## Output:

| TestCase | Output | TestCase | Output |
|---|---|---|---|
| **a** | ✓ | bbb | ✓ |
| **aa** | ✓ | abbb | ✓ |
| **ε** | ✓ | bba | ✗ |
| **b** | ✓ | bab | ✗ |
| **ab** | ✓ | baab | ✗ |
| **bb** | ✓ | abab | ✗ |
| **abb** | ✓ | aaaabbbba | ✗ |

```
PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Deskt
Enter String : a
Ans : a is Accepted under RegEx [a*]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Deskt
Enter String : aa
Ans : aa is Accepted under RegEx [a*]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Deskt
Enter String :
Ans :  is Accepted under RegEx [a*]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Deskt
Enter String : b
Ans : b is Accepted under RegEx [a*b+]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Deskt
Enter String : ab
Ans : ab is Accepted under RegEx [a*b+]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Deskt
Enter String : bb
Ans : bb is Accepted under RegEx [a*b+]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Deskt
Enter String : abb
Ans : abb is Accepted under RegEx [abb]
```

Empty String

```
PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Desktop\SS_A4\"
Enter String : bbb
Ans : bbb is Accepted under RegEx [a*b+]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Desktop\SS_A4\"
Enter String : bba
Ans : bba is Not Accepted under Any RegEx [a*|a*b+|abb]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Desktop\SS_A4\"
Enter String : bab
Ans : bab is Not Accepted under Any RegEx [a*|a*b+|abb]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Desktop\SS_A4\"
Enter String : baab
Ans : baab is Not Accepted under Any RegEx [a*|a*b+|abb]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Desktop\SS_A4\"
Enter String : abab
Ans : abab is Not Accepted under Any RegEx [a*|a*b+|abb]

PS C:\Users\Admin\Desktop\SS_A4> cd "c:\Users\Admin\Desktop\SS_A4\"
Enter String : aaaaabbbba
Ans : aaaaabbbba is Not Accepted under Any RegEx [a*|a*b+|abb]
```

**SUBMITTED BY**: U19CS012

BHAGYA VINOD RANA