

ADVANCED ALGORITHMS

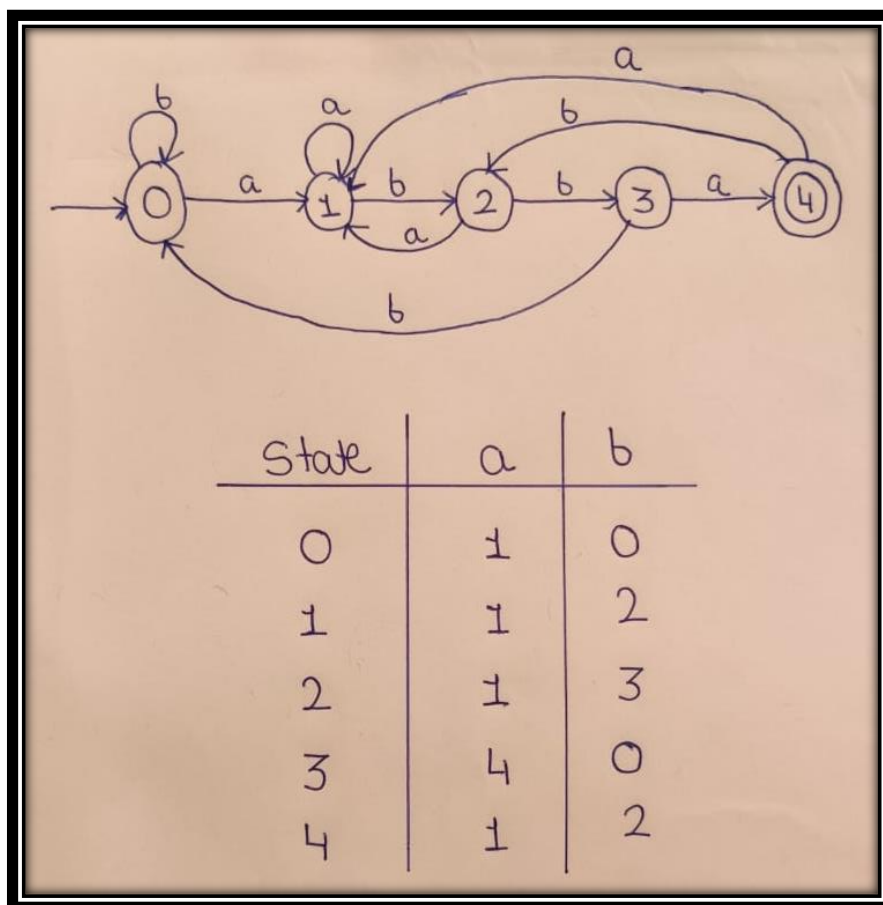
AIM: To construct program of string matching using finite automata.

ROLL NO: CE056

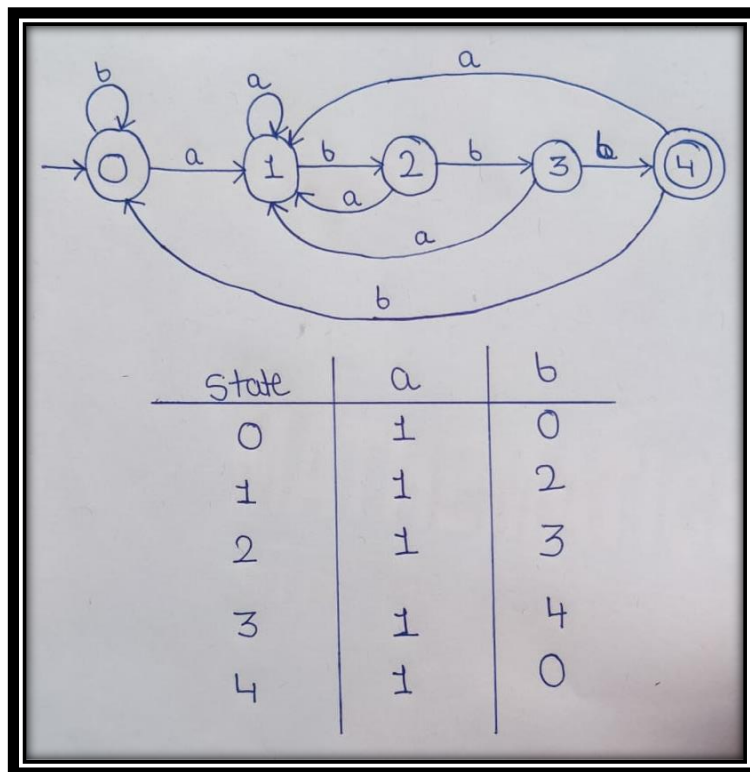
LAB NO: 02

➤ Finite Automata and Transition Table:

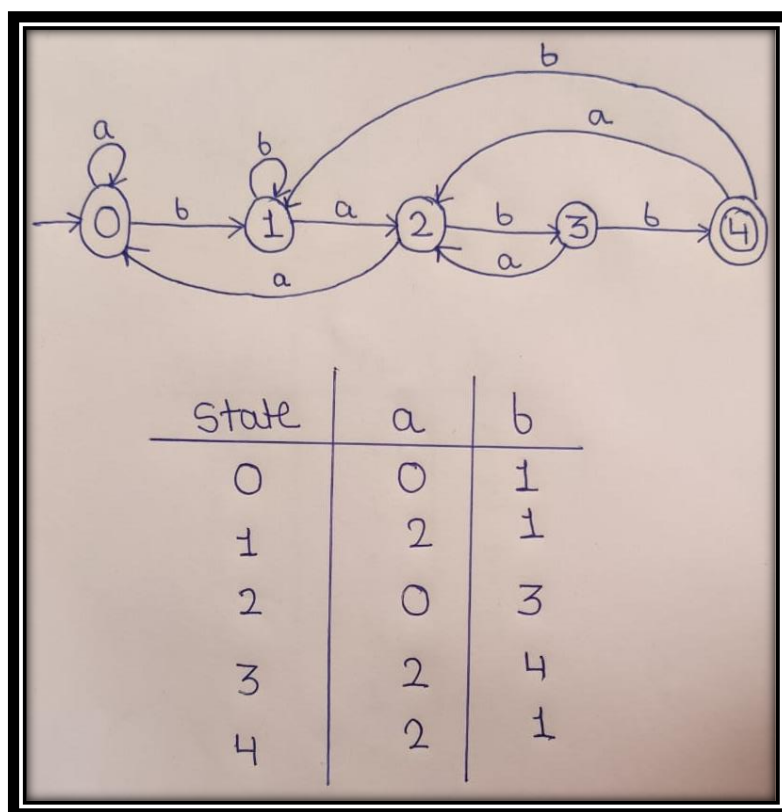
(1) abba



(2) abbb



(3) babb



➤ CODE:

```
#include <iostream>

#include <bits/stdc++.h>

using namespace std;

void FA_StringMatcher(int number_of_states,int final_state,int
initial_state[],int after_ip_a[],int after_ip_b[],string text)
{
    int length = text.length();
    int q = 0,shift;
    bool flag=false;
    for(int i=0;i < length;i++)
    {
        if(text[i] == 'a')
        {
            q = after_ip_a[q];
            if(q == final_state)
            {
                shift = i-final_state+1;
                cout << "Pattern is at index " << shift << endl;
                flag = true;
            }
        }
    }
}
```

```

        else if(text[i] == 'b')
        {
            q = after_ip_b[q];
            if(q == final_state)
            {
                shift = i-final_state+1;
                cout << "Pattern is at index " << shift << endl;
                flag = true;
            }
        }
    }
    if(flag == false)
        cout << "Pattern not found in text." << endl;
}

```

```

int main()
{
    int number_of_states,final_state;
    string text;
    cout << "Enter number of states: ";
    cin >> number_of_states;
    cout << "Enter final state: ";
    cin >> final_state;
    cout << "Enter text pattern: ";
}

```

```

    cin >> text;

    int
    initial_state[number_of_states],after_ip_a[number_of_states],after_
    ip_b[number_of_states];

    cout << "Enter the transition table in row wise sequence like 'initial
    state state after input a state after input b:" << endl;

    for(int i=0; i < number_of_states;i++)
    {

        cin >> initial_state[i] >> after_ip_a[i] >> after_ip_b[i];

    }

    FA_StringMatcher(number_of_states,final_state,initial_state,after_i
    p_a,after_ip_b,text);

    return 0;
}

```

FOR PATTERN: abba



The screenshot shows a terminal window titled 'aakarsh@aakarsh-VirtualBox: ~/Desktop'. The user has compiled a C++ program named 'AA_LAB02.cpp' and is running the executable 'a.out'. The program prompts for several inputs: number of states (5), final state (4), text pattern ('babbbabbabba'), and a transition table. The transition table is entered row by row for 5 states. The program then outputs the indices where the pattern 'abba' was found: index 5 and index 8.

```

aakarsh@aakarsh-VirtualBox: ~/Desktop$ g++ AA_LAB02.cpp
aakarsh@aakarsh-VirtualBox:~/Desktop$ ./a.out
Enter number of states: 5
Enter final state: 4
Enter text pattern: babbbabbabba
Enter the transition table in row wise sequence like 'initial state state after input a state after input b:
0 1 0
1 1 2
2 1 3
3 4 0
4 1 2
Pattern is at index 5
Pattern is at index 8
aakarsh@aakarsh-VirtualBox:~/Desktop$ 

```

FOR PATTERN: abbb

```
aakarsh@aakarsh-VirtualBox: ~/Desktop
aakarsh@aakarsh-VirtualBox:~/Desktop$ ./a.out
Enter number of states: 5
Enter final state: 4
Enter text pattern: aaabaab
Enter the transition table in row wise sequence like 'initial state state after input a state after input b':
0 1 0
1 1 2
2 1 3
3 1 4
4 1 0
Pattern not found in text.
aakarsh@aakarsh-VirtualBox:~/Desktop$
```

FOR PATTERN: babb

```
aakarsh@aakarsh-VirtualBox: ~/Desktop
aakarsh@aakarsh-VirtualBox:~/Desktop$ g++ AA_LAB02.cpp
aakarsh@aakarsh-VirtualBox:~/Desktop$ ./a.out
Enter number of states: 5
Enter final state: 4
Enter text pattern: abbabbbba
Enter the transition table in row wise sequence like 'initial state state after input a state after input b':
0 0 1
1 2 1
2 0 3
3 2 4
4 2 1
Pattern is at index 2
aakarsh@aakarsh-VirtualBox:~/Desktop$
```