**Compiler Design**

* **Practical-1: Implementation of Finite Automata and String Validation**
* **Design a deterministic finite automaton (DFA) for accepting the language.**
* **Regular expression for language L is,**

**L = (aa)\* (b)+**

* There are 3 steps involve which results in acceptance of string:

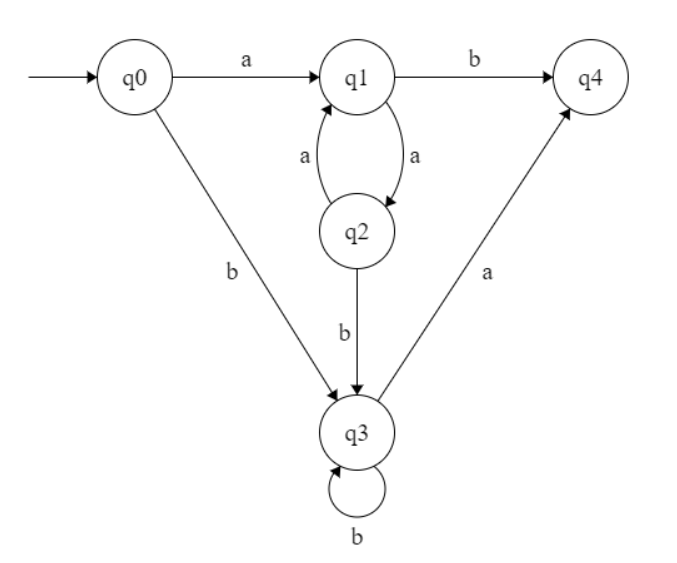
Step-1) Construct FA for (aa)\* means having even number of a’s.

Step-2) Construct FA for (b)+ means having any number of b’s greater than one.

Step-3) Concatenate the two FA and make single DFA.

Any other combination result is the rejection of the input string.

**DFA Transition Diagram**



**CODE:**

**def start(c):**

**if (c == 'a'):**

**dfa = 1**

**elif (c == 'b'):**

**dfa = 3**

**else:**

**dfa = -1**

**return dfa**

**def s1(c):**

**if (c == 'a'):**

**dfa = 2**

**elif (c == 'b'):**

**dfa = 4**

**else:**

**dfa = -1**

**return dfa**

**def s2(c):**

**if (c == 'b'):**

**dfa = 3**

**elif (c == 'a'):**

**dfa = 1**

**else:**

**dfa = -1**

**return dfa**

**def s3(c):**

**if (c == 'b'):**

**dfa = 3**

**elif (c == 'a'):**

**dfa = 4**

**else:**

**dfa = -1**

**return dfa**

**def s4(c):**

**dfa = -1**

**return dfa**

**def isAccepted(String):**

**l = len(String)**

**dfa = 0**

**for i in range(l):**

**if (dfa == 0):**

**dfa = start(String[i])**

**elif (dfa == 1):**

**dfa = s1(String[i])**

**elif (dfa == 2) :**

**dfa = s2(String[i])**

**elif (dfa == 3) :**

**dfa = s3(String[i])**

**elif (dfa == 4) :**

**dfa = s4(String[i])**

**else:**

**return 0**

**if(dfa == 3) :**

**return 1**

**else:**

**return 0**

**String = input("Input String: ")**

**#String = "aaaaaabbbb"**

**if (isAccepted(String)) :**

**print("ACCEPTED")**

**else:**

**print("NOT ACCEPTED")**

Output:

