OverView

Deterministic Finite Automata

In DFA, for each input symbol, one can determine the state to which the machine will move. Hence, it is called **Deterministic Automaton**. As it has a finite number of states, the machine is called **Deterministic Finite Machine** or **Deterministic Finite Automaton.**

Formal Definition

A DFA can be represented by a 5-tuple (Q, ∑, δ, q0, F) where −

* **Q** is a finite set of states.
* **∑** is a finite set of symbols called the alphabet.
* **δ** is the transition function where δ: Q × ∑ → Q
* **q0** is the initial state from where any input is processed (q0 ∈ Q).
* **F** is a set of final state/states of Q (F ⊆ Q).

Graphical Representation of a DFA

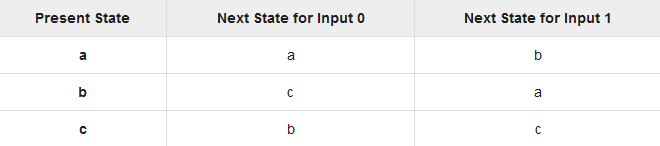
A DFA is represented by digraphs called **state diagram**.

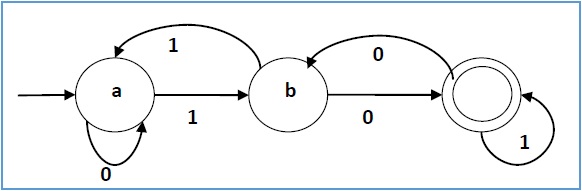
* The vertices represent the states.
* The arcs labeled with an input alphabet show the transitions.
* The initial state is denoted by an empty single incoming arc.
* The final state is indicated by double circles.

**Example**

Let a deterministic finite automaton P be P = ( Q, ∑, q0, F), where Q = {a, b, c}, ∑ = {0, 1}, q0 = {a}, F = {c}.

Transition function δ as shown by the following table

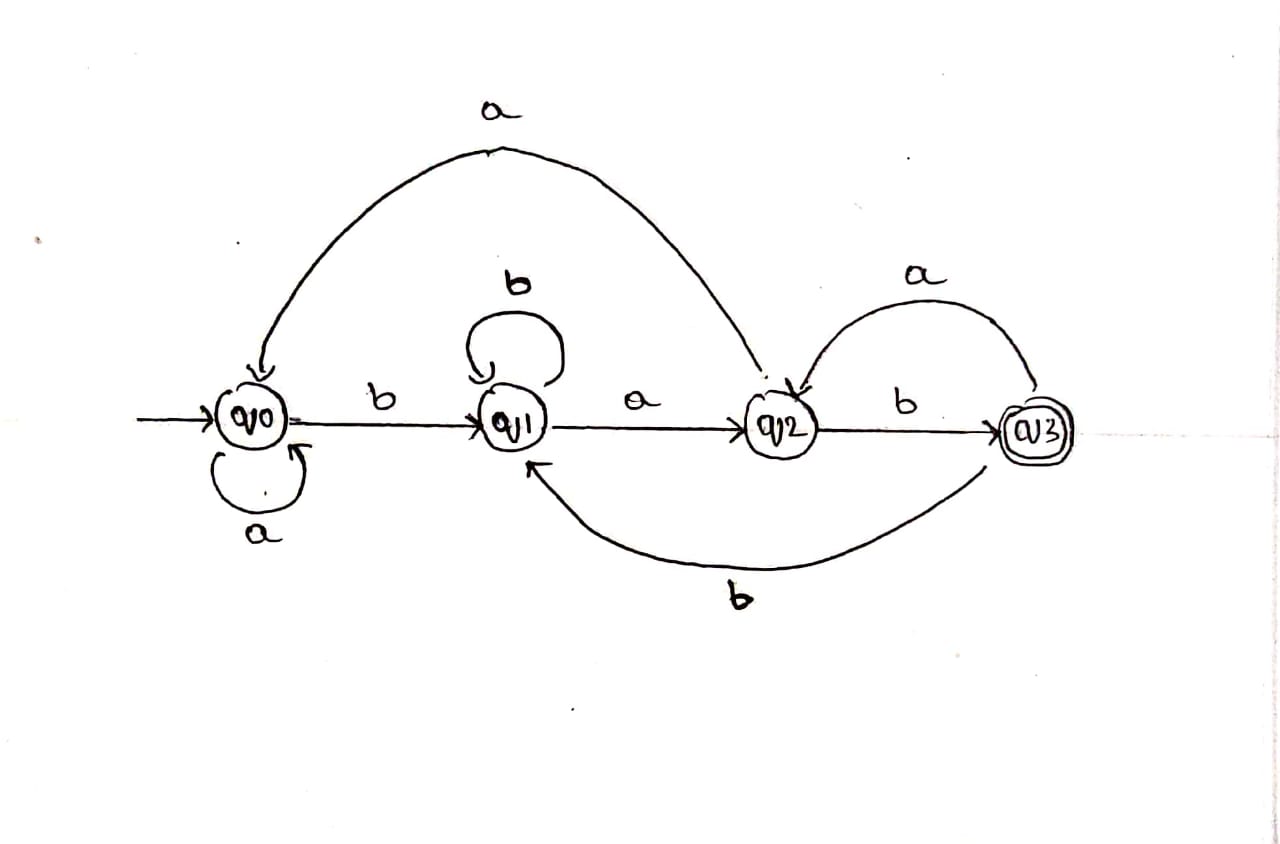


Its graphical representation would be as follows

Deterministic Finite Automaton To Accept Strings Ending With bab

**State diagram** for a Deterministic Finite Automaton designed to accept strings ending with bab is as follows:

|  |  |  |
| --- | --- | --- |
| **Present State** | **Input** | **Next State** |
| q0 | b | q1 |
| q0 | a | q0 |
| q1 | b | q1 |
| q1 | a | q2 |
| q2 | b | q3 |
| q2 | a | q0 |
| q3 | a | q2 |
| q3 | b | q1 |

**Transition Diagram**

Source Code

Source Code



Output



Output



References

<https://www.geeksforgeeks.org/program-to-construct-a-dfa-which-accept-the-language-l-anbm-n-mod-20-m1/>

Project Repository

<https://github.com/aksharsramesh/TOCproject>