DFA

A deterministic finite accepter or dfa is defined by the quintuple
$M = (Q, \Sigma, \delta, q_0, F)$,
where
Q is a finite set of **internal** states,
Σ is a finite set of symbols called the **input alphabet**,
$\delta : Q \times \Sigma \rightarrow Q$ is a total function called the **transition function**,
$q_0 \in Q$ is the **initial state**,
$F \subseteq Q$ is a set of **final states.**

A deterministic finite accepter operates in the following manner:
At the initial time, it is assumed to be in the initial state q_0 with its input mechanism on the leftmost symbol of the input string. During each move of the automaton, the input mechanism advances one position to the right, so each move consumes one input symbol. When the end of the string is reached, the string is accepted if the automaton is in one of its final states. Otherwise the string is rejected.
The input mechanism can move only from left to right and reads exactly one symbol on each step. The transitions from one internal state to another are governed by the transition function δ. For example, if

$$\delta (q_0, a) = q_1,$$

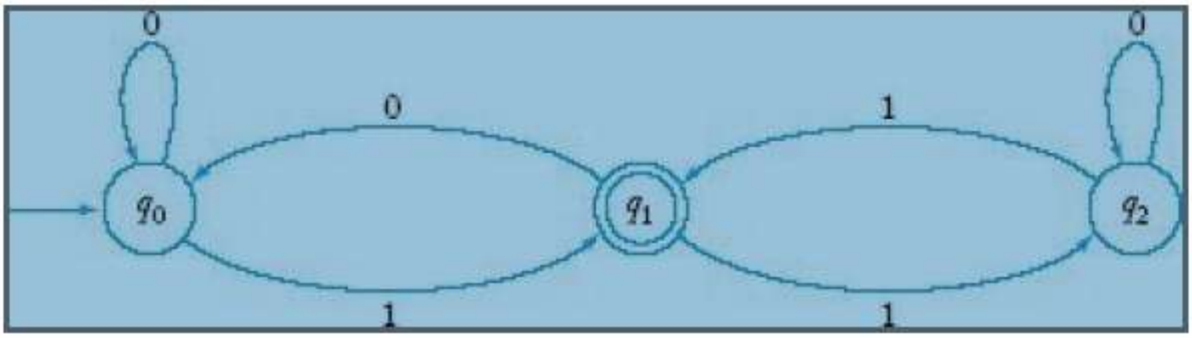
then if the dfa is in state q_0 and the current input symbol is a, the dfa will go into state q_1.
Transition Graph: We use transition graphs to visualize and represent finite automata. The vertices represent states and the edges represent transitions. The labels on the vertices are the names of the states, while the labels on the edges are the current values of the input symbol.
For example, if q_0 and q_1 are internal states of some dfa M, then the graph associated with M will have one vertex labeled q_0 and another labeled q_1. An edge (q_0, q_1) labeled a represents the transition $\delta(q_0, a) = q_1$.

The initial state will be identified by an incoming unlabeled arrow not originating at any vertex. Final states are drawn with a double circle.

Example: M =({q0,q1,q2},{0, 1},δ,q0,{ql}), where δ is given by

$$\begin{aligned} \delta(q_0,0) &= q_0, & \delta(q_0,1) &= q_1, \\ \delta(q_1,0) &= q_0, & \delta(q_1,1) &= q_2, \\ \delta(q_2,0) &= q_2, & \delta(q_2,1) &= q_1. \end{aligned}$$

The diagram of above DFA will be:



If

$$\delta(q0,a) = q1$$

and $$\delta(q1,b) = q2$$

then $$\delta^* (q0,ab) = q2.$$

we can define δ* recursively by

$$\begin{aligned} \delta^*(q,\lambda) &= q, \\ \delta^*(q,wa) &= \delta(\delta^*(q,w),a), \end{aligned}$$

$$\delta^*(q_0,ab) = \delta(\delta^*(q_0,a),b).$$

But

$$\begin{aligned} \delta^*(q_0,a) &= \delta(\delta^*(q_0,\lambda),a) \\ &= \delta(q_0,a) \\ &= q_1. \end{aligned}$$

So $\qquad \delta^*(q_0, ab) = \delta(q_1, b) = q_2,$

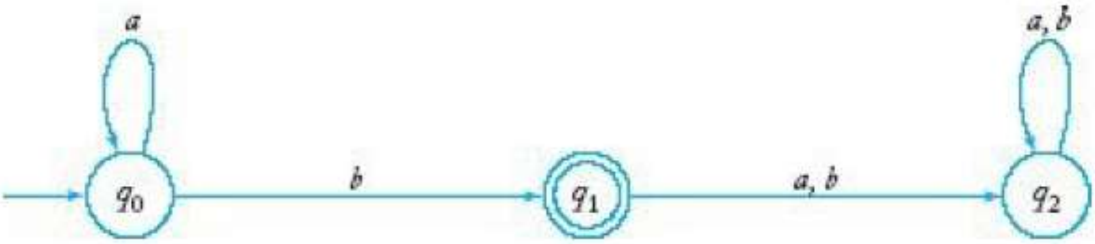
The language accepted by a dfa $M = (Q, \Sigma, \delta, q0, F)$ is the set of all strings on Σ accepted by *M*. In formal notation

$$L(M) = \{w \in \Sigma^* : \delta^*(q_0, w) \in F\}.$$

A dfa will process every string in Σ^* and either accept it or not accept it. Nonacceptance means that the dfa stops in a nonfinal state, so that

$$\overline{L(M)} = \{w \in \Sigma^* : \delta^*(q_0, w) \notin F\}.$$

Examples:



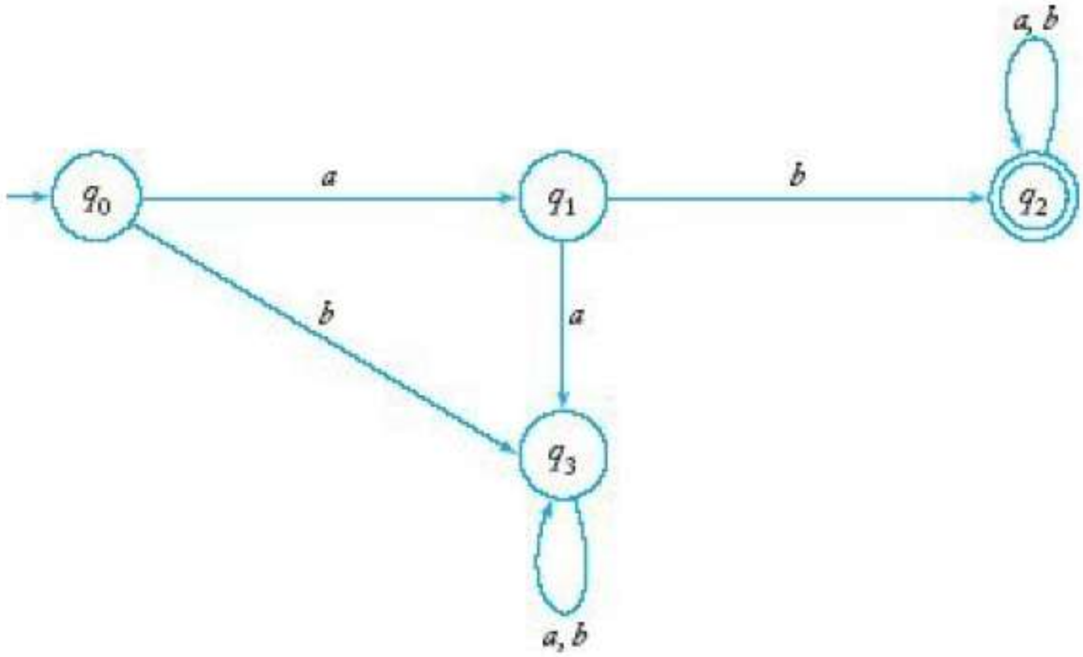
The automaton in the above Figure remains in its initial state *q*0 until the first *b* is encountered. If this is also the last symbol of the input, then the string is accepted. If not, the dfa goes into state *q*2, from which it can never escape. The state *q*2 is a trap state.
We see clearly from the graph that the automaton accepts all strings consisting of an arbitrary number of a's, followed by a single *b*. All other input strings are rejected. In set notation, the language accepted by the automaton is

$$L = \{a^n b : n \geq 0\}.$$

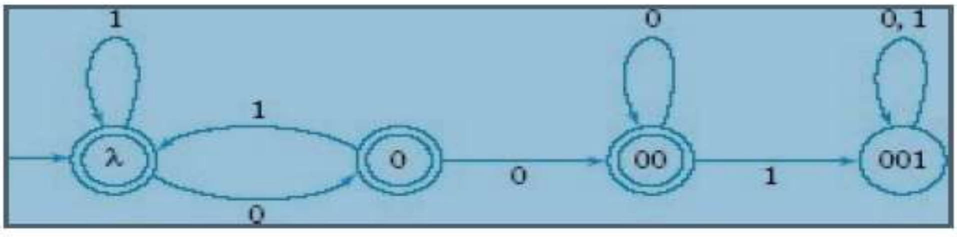
Transition Table: While graphs are convenient for visualizing automata, other representations are also useful. For example, we can represent the function δ as a table.
Following table is equivalent to the above Figure.

	a	b
q_0	q_0	q_1
q_1	q_2	q_2
q_2	q_2	q_2

Example: Find a deterministic finite accepter that recognizes the set of all strings on $\Sigma = \{a,b\}$ starting with the prefix *ab*.



Example: Find a dfa that accepts all the strings on {0,1}, except those containing the substring 001.



Q1. Set of all strings starting with b.
Q2. Construct a dfa which accepts set of all strings over $\Sigma = \{a, b\}$ of length 3.
Q3. Construct a dfa which accepts set of all strings over $\Sigma = \{a, b\}$ of length more than or equal to 2.
Q4. Construct a dfa which accepts set of all strings over $\Sigma = \{a, b\}$ of length less than or equal to 2.