Deterministic Finite Automata (DFA)

Represented as a quintuple

$$M = (Q, \Sigma, \delta, q_0, F)$$

where

$Q \Rightarrow$ non-empty finite set of states ($q_0, q_1, \ldots$)

$\Sigma \rightarrow$ non-empty set of input symbols passed to the FSM ($a, b, 0, 1 \ldots$)

$q_0 \rightarrow$ start state, (one of the states in Q)

$F \rightarrow$ non-empty set of final/accepting states (also belong to Q)
- represented by two concentric circles

$\delta \rightarrow$ transition function
- takes two arguments
  - a state and an input symbol.
- returns a single state

$$\delta(q, a) = q'$$

↑ output / state

Deterministic

↑ only one transition for every input symbol.

⇓ It is possible to exactly find out to which state the machine enters after consuming the input symbol

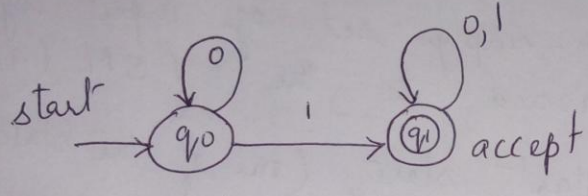
Transition function

$$\delta : Q \times \Sigma \text{ to } Q$$

i.e, δ is a transition function that maps $Q \times \Sigma$ to Q.

e.g; $\delta(q, a) = p$.

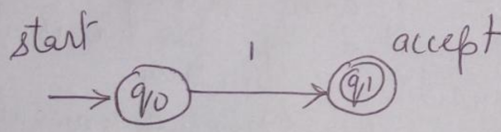
Examples



Transition diagram

δ	0	1
$\rightarrow q_0$	q_0	q_1
q_1^*	q_1	q_1

Transition table (TT)



Transition diagram

δ	1
$\rightarrow q_0$	q_1

TT

Configuration of a DFA

↓ snapshot that gives information about
(i) current state of DFA
(ii) string to be processed.

$(q_0, w) \leftarrow$ initial configuration

q_0 ← initial state

w ← string to be processed.

$(q, \varepsilon) \leftarrow$ final configuration

Yield denoted by symbol " $\vdash$ "

$\underline{q_1, cw}$

$\delta(q_1, c) = q_2$

or $(q_1, cw) \vdash (q_2, w)$

- Each of the states can be called as a configuration denoted by some symbol say $C_1, C_2 \ldots$

$C_1 \vdash C_2 \implies$ machine goes from C_1 to C_2 in a single move.

$C_1 \vdash^{*} C_2 \Rightarrow$ machine goes from C_1 to C_2 in more than <u>one</u> moves.

Computation

by a FA is a finite sequence of configurations.

$$\underset{\text{initial config}}{C_0} \vdash C_1 \vdash C_2 \vdash C_3 \cdots \vdash \underset{\text{final config.}}{C_n}$$

- FA accepts w iff (if and only if)

$$(q_0, w) \vdash (\underset{\text{final state}}{q_f}, \varepsilon)$$

- FA rejects w iff $(q_0, w) \vdash (\underset{\text{non-final state}}{q}, \varepsilon)$

- ~~FA halts after entering a final or halt state reject ... con~~

⇒ FA (finite automaton) halts whenever it enters either the <u>accepting</u> configuration or reject configuration.

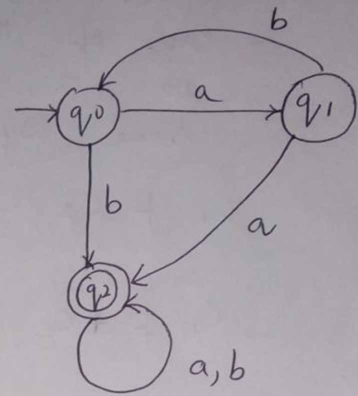
Transition Table to Transition Diagram

δ/Σ	a	b
initial state → q_0	q_1	q_2
q_1	q_2	q_0
final state → $*q_2$	q_2	q_2

(* designates a final state)

TT.

⇒



TD.

Language accepted by a DFA

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA.

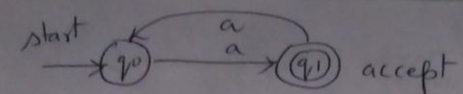
A string "w" is accepted by the DFA if it takes the machine from an initial configuration (q_0, w) to a final configuration (q_f, ε).

$$\delta\ (q_0, w) \vdash (q_f, \varepsilon)$$

Language accepted by DFA is $L(M)$ and is

$$L(M) = \{ w \mid w \in \Sigma \text{ and } (q_0, w) \vdash (q_f, \varepsilon) \}$$

Ex



Let w = 'aaaa' find if it accepted by DFA given above.

$$(q_0, aaaa) \vdash (q_1, aaa)$$
$$\vdash (q_0, aa)$$
$$\vdash (q_1, a)$$
$$\vdash (q_0, \varepsilon) \leftarrow \text{final configuration}$$

but not final state

$\therefore$ 'aaaa' not accepted lang.

Let $w = aaa$

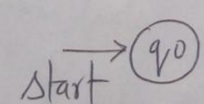
$$(q_0, aaa) \vdash (q_1, aa)$$
$$\vdash (q_0, a)$$
$$\vdash (q_1, \varepsilon) \leftarrow \text{final configuration}$$

$\uparrow$ final state

$L(M) = \{ w \mid w = aaa$ is lang accepted by DFA above $\}$

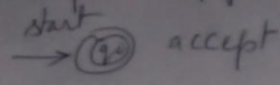
DFA's for Given Languages

① $L = \{\phi\}$ DFA to accept empty language

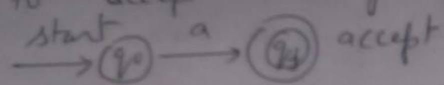


Examples contd

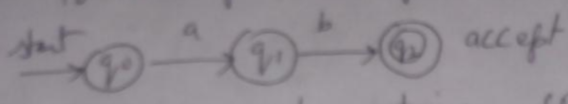
(2) DFA to accept empty string $L = \{\epsilon\}$



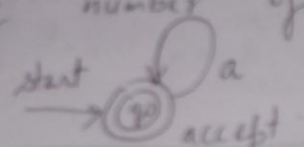
(3) DFA to accept exactly one a



(4) DFA to accept string ab

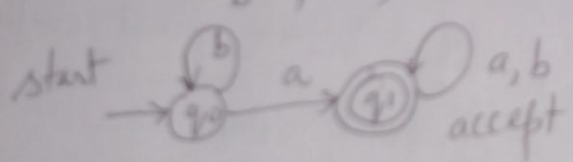


(5) DFA to accept a string consisting of any number of a's (a^*)



(6) DFA to accept strings of a's and b's with atleast one 'a'

$b^*, a, (a+b)^*$



$M = \{Q, \Sigma, \delta, q_0, F\}$

δ is as under

$Q = \{q_0, q_1\}$

$\Sigma = \{a, b\}$

q_0 = start state

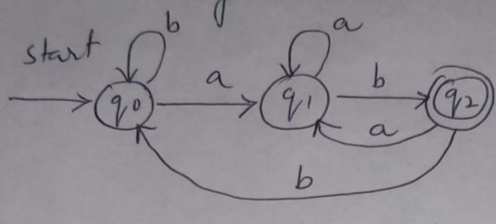
$F = \{q_1\}$

δ	a	b
$\rightarrow q_0$	q_1	q_0
$* q_1$	q_1	q_1

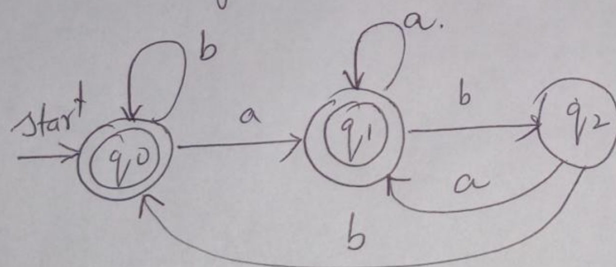
TT

Ex Obtain a DFA to accept strings of a's and b's which do not end with the string ab.

Sol:- Let's first construct a DFA to, accept strings of a's and b's ending with ab.



To get the solution for strings <u>not</u> ending with <u>ab</u>: (i) convert final-state/s to non-final state/s, and (ii) convert non-final states to final states



$M = \{Q, \Sigma, \delta, q_0, F\}$

$Q = \{q_0, q_1, q_2\}$

$\Sigma = \{a, b\}$

$q_0 \leftarrow$ start state

$F = q_0, q_1$

<u>δ:</u>

δ	a	b
$\rightarrow * q_0$	q_1	q_0
q_1	q_1	q_2
q_2	q_1	q_0