AUTOMATA THEORY AND COMPUTABILITY

Automata theory is a branch of computer Science/Mathematics which deals with:

* Logic of computation
* Abstract machines.

Dr. Thippeswamy G
Professor & Dean - Academics
BMS Institute of Technology & Management
Avalahalli, Yelahanka, Bengaluru - 560 064.

Automata is the plural version of automation.

Basic terminologies associated with ATC:

- Symbol: Symbol is the smallest building block, which can be any letter or picture.

- Alphabet: Alphabet is finite set of symbols.

  Example: (i) $\Sigma = \{A, B, C, \ldots Z\}$ alphabet of English language.

(ii) $\Sigma = \{0, 1\}$ alphabet of Machine Language

(iii) $\Sigma = \{0, 1, 2, \ldots 9\}$ alphabet of decimal digits

(iv) $\Sigma = \{$अ, आ, … ह$\}$ alphabet of our National Language, Hindi.

* **String:** String is a finite sequence of symbols from some alphabet.
- Usually string is denoted by w.
- Length of the string is denoted by $|w|$.
- Empty string is a string with no symbols or zero occurrence of symbols.

Empty string is denoted by ϵ [epsilon]

Example:-

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

To generate strings of length 2 from above Σ.

$L = \{aa, ab, ba, bb\}$

Number of string is 4.

Note: For alphabet $\Sigma = \{a, b\}$ with length n, number of strings can be generated is 2^n

* **Language:** A language is a set of strings obtained from some Σ^*.

Ex: ① $\Sigma_1 = \{0, 1\}$ $L_1 = \{00, 01, 10, 11\}$ with length 2.
FINITE

② $\Sigma = \{a\}$, $L_2 = \{$Set of all strings starting with a$\}$ INFINITE
$L_2 = \{a, aa, aaa \ldots\}$

Kleene closure / Star closure:

Consider the following Σ.

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

Σ^0 = Set of all the strings over Σ with length 0, $\{\epsilon\}$

Σ^1 = — " — with length 1, $\{a, b\}$

Σ^2 = — " — with length 2 $\{aa, ab, ba, bb\}$

⋮

$\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \ldots\ldots$

Σ^* is called Kleene closure / Star closure

it is defined as

" Set of all the strings over Σ of any - - length including ϵ"

Positive closure:

$\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \ldots$

"Set of all the strings over Σ of any length excluding ϵ"

Note:

$$\left.\begin{aligned} \Sigma^* &= \Sigma^+ \cup \epsilon \\ \Sigma^+ &= \Sigma^* - \epsilon \end{aligned}\right\}$$

AUTOMATA: Automata is an abstract model of digital computer which follows a predetermined sequence of operations automatically.

Finite Automata (FA) / Finite State Machine is an automation with finite number of states.

Types of Finite State Machines / Finite Automata

There are two types

- Deterministic Finite State Machine - [DFSM]
  [Also Called DFA - Deterministic Finite Automata - DFA.]
- Non-Deterministic Finite State Machine - NFSM
  [Also Called non-deterministic finite automata] → NFA

DFA / DFSM:

Formal definition: A DFA is 5-tuple

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

where, Q is finite set of states

Σ is finite set of symbols, called alphabet.

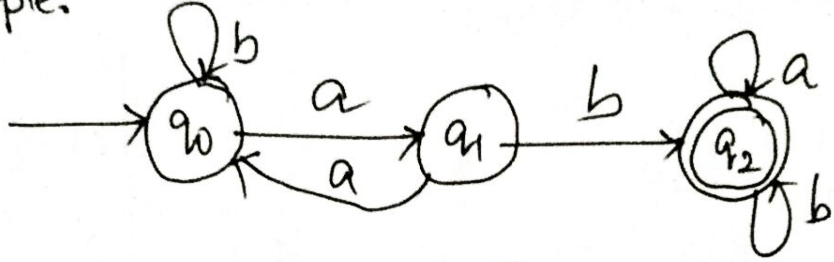
δ is transition function

q_0 is the start state. ($q_0 \in Q$)

F is a set of final states ($F \subseteq Q$).

Note: In DFA, from each internal State over each input symbol (alphabet), there will be exactly one transition.

Example:



The above DFA has exactly one transition from each State over each input symbol.

q_0: $\delta(q_0, a) = q_1$, $\delta(q_0, b) = q_0$ — From the State q_0 with a and b.

q_1: $\delta(q_1, a) = q_0$, $\delta(q_1, b) = q_2$ — From the State q_1 with a and b

q_2: $\delta(q_2, a) = q_2$, $\delta(q_2, b) = q_2$ — From the State q_2 with a and b.

Note: Finite automata is represented by using directed graph (digraph).

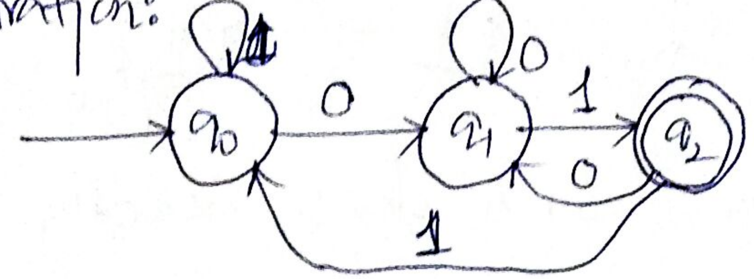
Transition: Means, change of State.

Transition function δ of DFA / DFSM.

$$Q \times \Sigma \rightarrow Q$$

[Each transition is from a given State (Q) with an input (Σ) to other or Same State (Q)]

Illustration:



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

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

$\Sigma = \{0, 1\}$

$\delta : Q \times \Sigma \rightarrow Q$

q_0 is Start State [$q_0 \in Q$]

F is Set of final States [$F \subseteq Q$]

$$\delta : Q \times \Sigma \rightarrow Q$$

$\delta(q_0, 0) \rightarrow q_1$

$\delta(q_0, 1) \rightarrow q_0$

$\delta(q_1, 0) \rightarrow q_1$

$\delta(q_1, 1) \rightarrow q_2$

$\delta(q_2, 0) \rightarrow q_1$

$\delta(q_2, 1) \rightarrow q_0$

Acceptance of Strings by finite automata [DFA / NFA].

A String is said to be accepted by DFA if and only if the DFA Starting at the Start State (q_0) with sequence of symbols, ends in final State (States).

Formally, the String w is accepted by DFA.

iff $\delta(q_0, w) = F$

where q_0 is Start State and F is final State.

For every input Symbol of String it takes one transition. It Starts with Start State q_0 and reaches one of the final State F.

Problems on DFA:

Q. Design DFA to accept the strings over $\Sigma = \{a, b\}$ with atleast one 'a'.

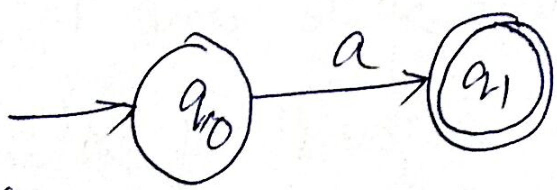
Solution:

Step-1: To write set of acceptable strings.

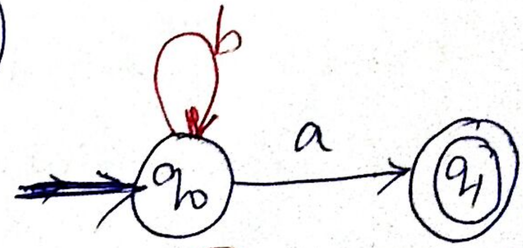
$$L = \{ a, baa, b^n a a^n \ldots\ldots \}$$

Step-2: To write transition graph.

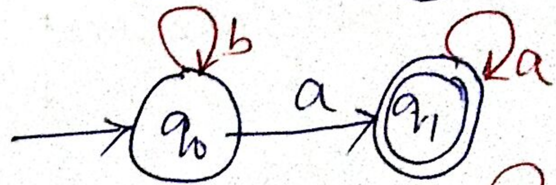
To accept a



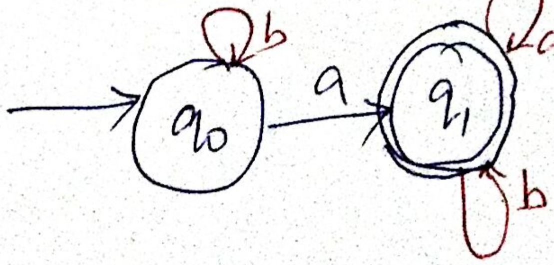
To accept $b^n a$



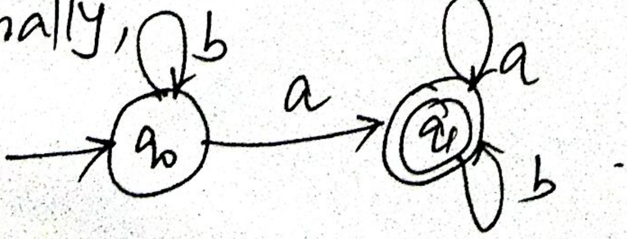
To accept $b^n a a^n$



To accept $b^n a a a b^n$



Finally,



Contd... Page-9

Step-3: To write transition table:
[Tabular representation of DFA]

δ_D	a	b
$\rightarrow q_0$	q_1	q_0
q_1^*	q_1	q_1

$Q \times \Sigma \rightarrow Q$.

Note: Start State is denoted by $\rightarrow q_0$
Final State is denoted by q_f

Step-4: To write complete DFA.

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

$Q = \{q_0, q_1\}$

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

$\delta : Q \times \Sigma \rightarrow Q$

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

$\delta(q_0, b) \rightarrow q_0$

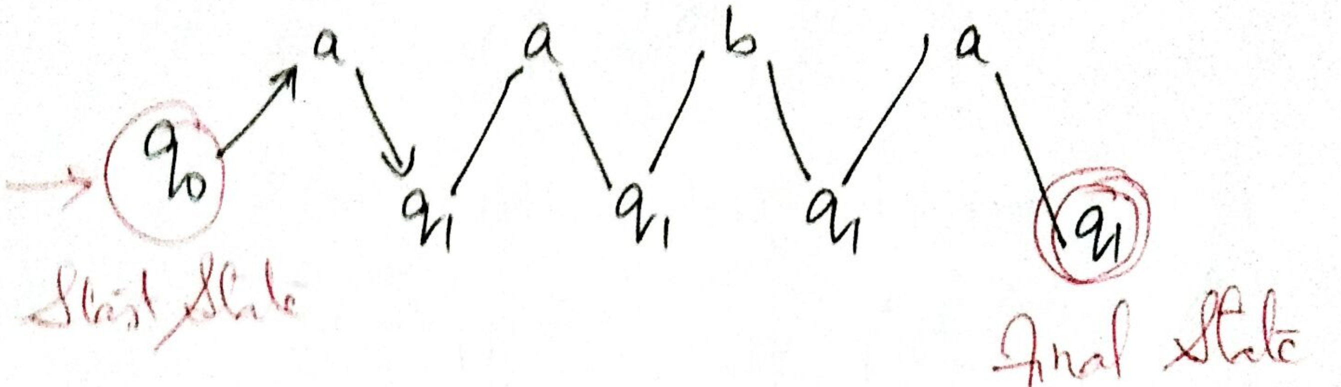
$\delta(q_1, a) \rightarrow q_1$

$\delta(q_1, b) \rightarrow q_1$

q_0 is Start State $q_0 \in Q$.

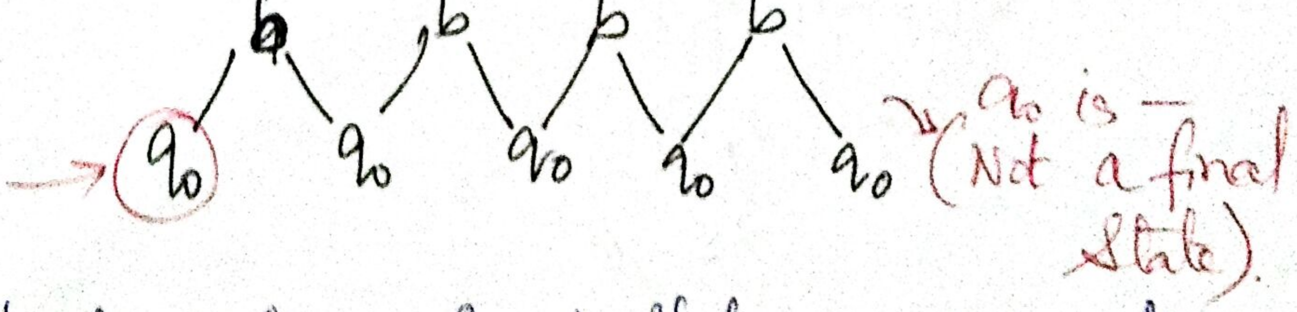
$F = \{q_1\}$ is set of final states.

Step-6: To show that the String aaba is accepted.



Starting from the start state q_0, with the sequence of symbols of String (aaba), reached final state q_1, hence the String aaba is accepted.

Step-6: To show that bbbb is rejected.



Starting from Start State q_0, with the sequence of symbols of String (bbbb) not possible to reach final State, hence the String is rejected.