THEORY OF COMPUTATION AND COMPILERS

Unit - I | Lecture- 03

INTRODUCTION TO FINITE AUTOMATA (FA)

Dr. R. Madana Mohana
Professor, Artificial Intelligence & Data Science | I/c-Head, Artificial Intelligence & Machine Learning
CHAITANYA BHARATHI INSTITUTE OF TECHNOLOGY
Hyderabad - 500 075, Telangana, INDIA
www.cbit.ac.in

Introduction to Finite Automata (FA)

- **Introduction to Finite Automata (FA)**
- Structural Representations
- Automata and Complexity

Introduction to Finite Automata (FA) / Finite State Machine (FSM)

Finite Automata (FA):

- The **Finite Automata (FA)** or **Finite State Machine (FSM)** represents a mathematical model of a system with certain inputs. Finally the model gives its corresponding outputs.
- **Finite Automata** was developed by Robin and Scott in 1950's as a model of computer with limited memory.

Finite Automata (FA):

- **`Finite Automata`** involves states and transitions, among states in response to inputs.
- **`Finite Automata`** are useful for building different kinds of softwares including the lexical analysis which is a component of a compiler and systems for verifying the correctness of circuits or protocols.

Finite Automata (FA):

- Example of **`Finite Automata`** is the control mechanism of an elevator. This mechanism only remembers the current floor number pressed, it does not remember all the previously pressed floor numbers.
- **`Finite Automata`** is a very good tool for the programs such as text editors and lexical analyzer. The lexical analyzer is a program, which scans the program character by character and recognizes these as tokens (Ex., Identifiers, keywords, constants, operators etc.)

Model or block diagram of Finite Automata:

- The model or block diagram of **`Finite Automata (FA)`** is shown in the following diagram.

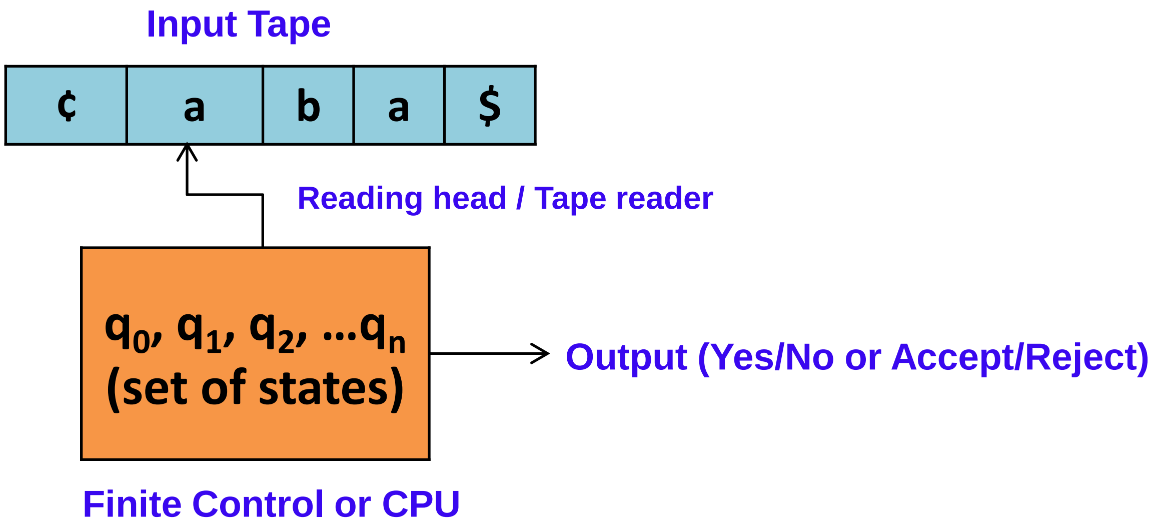


Figure: Model of FA

Model or block diagram of Finite Automata:

- This model is a mathematical model of a system with discrete inputs and discrete outputs.
- The computational result given by **FA** is either **Yes** (**Accepted**) or **No** (**Rejected** / **Not Accepted**).

Model or block diagram of Finite Automata:

Components of FA model:

FA has the following four components:

1. Input Tape
2. Reading Head or Tape Reader
3. Finite Control or CPU
4. Output

Model or block diagram of Finite Automata:

1. Input Tape:

- The **input tape** is divided into number of squares or cells, in which one symbol can store in each square or cell from the input alphabet **Σ**.
- **Input tape** is a linear tape having some number of cells.
- ⊄ and **$** are two end markers at the left end and right end of the **input tape** respectively.
- The absence of end markers indicates that the tape is of infinite length.

Model or block diagram of Finite Automata:

2. Reading Head or Tape Reader:

- A **Reading Head** or **Tape Reader** can read one symbol at a time from the input tape and moves ahead (i.e., either to the left side or to the right side).
- So **Reading Read** is a movable head.
- The movement of **reading head** is restricted only to the right side.

Model or block diagram of Finite Automata:

3. Finite Control or CPU:

- It acts like a **CPU** of a digital computer.
- A **finite control** which works within a finite set of states. At each step it changes its state depending on the current state and current input symbol read.
- The **tape reader** reads the cells one by one from left to right and at a time only one input symbol is read.

Model or block diagram of Finite Automata:

3. Finite Control or CPU: cont'd.

- **Finite control** change of state is specified by a transition function or moving function. It accepts the input, if at the end it is in a set of final or accepting sate.

4. Output:

- The **output** of a **FA** may be accept (yes) or reject (No).
- When end of input is encountered, the **control unit** may be in accept or reject state.

Formal or Mathematical Definition of Finite Automata:

A **Finite Automata (FA)** or **Finite State Machine (FSM)** can be defined as a Quin tuple or 5-tuple denoted by **M**.

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

Where

Q	: Finite or non-empty set of States or Internal Sates
Σ	: Input Alphabet
q_0	: Initial State or Start State and q_0 is in **Q**, i.e. $q_0 \in Q$ (In any Automata initial or start state is only one)
F	: Set of Final or Accepting States, $F \subseteq Q$

Formal or Mathematical Definition of Finite Automata:

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

δ : Transition function or Moving function or Mapping function.

Using this function, the next state can be determined.

Transition function is mapping from $Q \times \Sigma$ to Q i.e.,

$$\delta : Q X \sum \rightarrow Q$$

This mapping is usually represented by a transition table or transition diagram.

Example of Finite Automata (FA):

Let a **FA M = (Q, Σ, δ, q_0, F)**

Let $M = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_1\})$

Where

Q = {q_0, q_1}

Σ = {a, b}

q_0 is Initial or Start State

F = {q_1}

And **δ** is given below:

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

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

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

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

Transition Table:

- Transition Table is a tabular representation of **Finite Automata (FA)**.
- For transition table, the transition function **δ** is used.
- The table which represents the list of transition rules (functions) of a **Finite Automata** is called transition table or state table.

Transition Table:

- In transition table, each row indicates the **`states`** in **`finite automata`** and each column is a letter of **`input alphabet (Σ)`**.
- In the transition table, **`initial state`** or **`start state`** is represented by drawing a prefixed arrow to that state and **`final state`** is denoted by a single circle.

Example of Transition Table:

Let a **FA M = (Q, Σ, δ, q_0, F)**

Let $M = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_1\})$

Where

Q = {q_0, q_1}

Σ = {a, b}

q_0 is Initial or Start State

F = {q_1}

And **δ** is given below:

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

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

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

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

Example of Transition Table:

States (Q)/Input (Σ)	Input (Σ)	
	a	**b**
→ q_0	q_1	q_0
q_1	q_1	q_0

In this transition table, **q_0** is initial sate and **q_1** is final state.

Transition Diagram:

- Transition Diagram or Sate Diagram or Transition Graph is a directed graph.
- The pictorial representation of **`finite automata`** that gives us more of feel for the transitions. This is called as transition diagram or state diagram or transition graph.

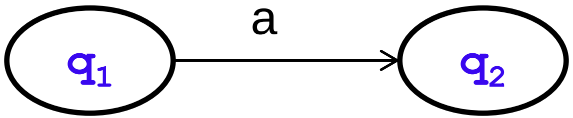
Transition Diagram:

A Transition Diagram can be defined as a collection of

- Finite set of States (**Q**)
- Finite set of input symbols (**Σ**)
- Initial or start state q_0 ($q_0 \in Q$)
- Set of final or accepting states (**F**), $F \subseteq Q$
- A transition function **δ**, $\delta : QX \sum \rightarrow Q$

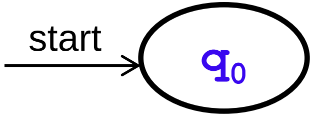
Construction of Transition Diagram:

1. The vertices or nodes in the transition diagram are the states of Finite Automata (**FA**).
2. The edges represents the transitions. If there is a transition from state q_1 to q_2 on an input symbol '**a**', then draw an edge from q_1 to q_2 labeled by '**a**'.

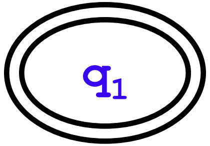


Construction of Transition Diagram:

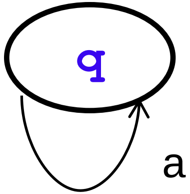
3. The initial state will be q_0 and the initial or start state is labeled with the phrase '**`start`**' or by an arrow symbol (→) or both together.



4. The final states are denoted by double circle i.e.,



5. If a certain symbol makes a state go back to itself, this is denoted by an arrow that returns to the same circle. This arrow is called a loop.



Example of Transition Diagram:

Let a **FA, M = (Q, Σ, δ, q_0, F)**

Let $M = (\{q_0, q_1\}, \{a, b\}, \delta, q_0, \{q_1\})$

Where

Q = {q_0, q_1}

Σ = {a, b}

q_0 is Initial or Start State

F = {q_1}

And **δ** is given below:

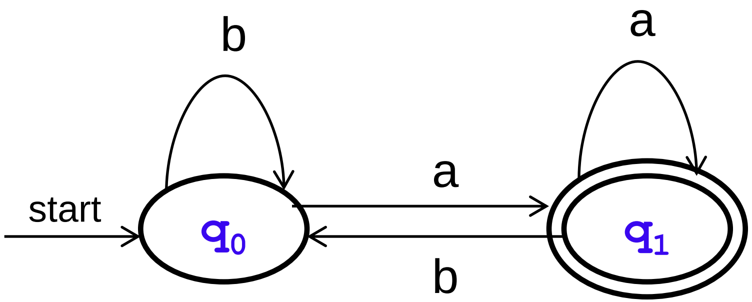
$$\delta(q_0, a) = q_1$$

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

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

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

Example of Transition Diagram:



States (Q)/Input (Σ)	Input (Σ)	
	a	b
→ q_0	q_1	q_0
q_1	q_1	q_0

Exercise Problem: FA

QUESTION:

For the given Finite Automata (FA) shown below, draw the transition table and transition diagram?

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

Where $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $F = \{q_0\}$ and δ is given below:

$\delta(q_0, 0) = q_0$

$\delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_0$

$\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_0$

$\delta(q_2, 1) = q_2$

Exercise Problem: FA

QUESTION:

For the given Finite Automata (FA) shown below, draw the transition table and transition diagram?

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

Where $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1\}$, $F = \{q_0\}$ and δ is given below:

$\delta(q_0, 0) = q_0$

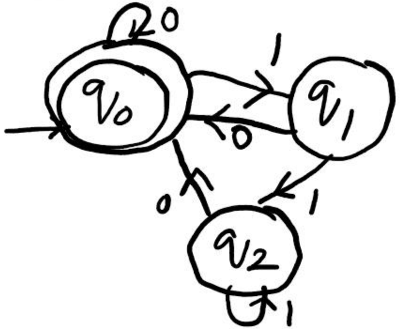
$\delta(q_0, 1) = q_1$

$\delta(q_1, 0) = q_0$

$\delta(q_1, 1) = q_2$

$\delta(q_2, 0) = q_0$

$\delta(q_2, 1) = q_2$



Transition Diagram

Q/Σ	0	1
$\rightarrow q_0$	q_0	q_1
q_1	q_0	q_2
q_2	q_0	q_2

Transition Table

Acceptance of Strings and Languages by FA

Properties of δ function of FA:

- Properties of **δ** function of FA or Extended **δ** definition of FA is denoted by $\bar{\delta}$ or $\delta^\wedge$
- The transition function **δ** takes two parameters i.e., **state (Q)** and **input symbol (Σ)**.
- $\delta^\wedge$ takes two parameters i.e., **state (Q)** and a **string (Σ*)**.

$$\therefore \delta^\wedge = QX \sum^* \rightarrow Q$$

Properties of δ function of FA :

Properties of $\delta^{\wedge}$

1. Property-1:

$$\delta^{\wedge}(q_0, \varepsilon) = q_0, q_0 \in Q$$

This means, the state of the system can be changed only by an input symbol.

2. Property-2:

$$\delta^{\wedge}(q_0, wa) = \delta(\delta^{\wedge}(q_0, w), a) = P$$

$$w \in \Sigma^*, q_0 \in Q, a \in \Sigma, P \in Q$$

Acceptance of Strings by FA:

A string w is accepted by a **Finite Automata (FA),**

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

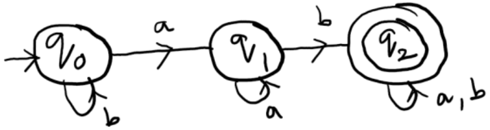
If $\delta^{\wedge}(q_0, w) = P$, for some **$P \epsilon F$**

This is basically the acceptability of a string by the final state.

Acceptance of Strings by FA:

Example Problem-1:

Consider the transition diagram as shown below:
Check whether the input string **aab** is accepted or not by the given Finite Automata ?



Here $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $F = \{q_2\}$

and δ is given below:

$\delta(q_0, a) = q_1$
$\delta(q_0, b) = q_0$
$\delta(q_1, a) = q_1$
$\delta(q_1, b) = q_2$
$\delta(q_2, a) = q_2$
$\delta(q_2, b) = q_2$

Acceptance of Strings by FA:

Example Problem-1: Solution

Given string aab, find $\hat{\delta}(q_0, aab)$?

$$
\begin{aligned}
\hat{\delta}(q_0, aab) &= \delta(\hat{\delta}(q_0, aa), b) \\
&= \delta(\delta(\hat{\delta}(q_0, a), a), b) \\
&= \delta(\delta(\delta(\underline{\hat{\delta}(q_0, \epsilon)}, a), a), b) \\
&= \delta(\delta(\underline{\delta(q_0, a)}, a), b) \quad [\hat{\delta}(q_0, \epsilon) = q_0] \\
&= \delta(\underline{\delta(q_1, a)}, b) \quad [\delta(q_0, a) = q_1] \\
&= \underline{\delta(q_1, b)} \quad [\delta(q_1, a) = q_1] \\
&= q_2, \; q_2 \in F
\end{aligned}
$$

∴ The string aab is accepted by given FA.

Acceptance of Strings by FA:

Example Problem-2:

Suppose an automata has states $Q=\{q_1, q_2\}$, $\Sigma=\{a, b\}$, $q_0=q_1$, $F=q_2$ and δ is defined as

$\delta(q_1,a) = q_1$
$\delta(q_1,b) = q_2$
$\delta(q_2,a) = q_1$
$\delta(q_2,b) = q_1$

Check whether the strings **abba** and **babbab** are accepted by the given FA or not?

Acceptance of Strings by FA:

Example Problem-2: Solution

i) abba

$$\delta^{\wedge}(q_1, abba) = \delta(\delta^{\wedge}(q_1, abb), a)$$

$$= \delta(\delta(\delta^{\wedge}(q_1, ab), b), a)$$

$$= \delta(\delta(\delta(\delta^{\wedge}(q_1, a), b), b), a)$$

$$= \delta(\delta(\delta(\delta(\delta^{\wedge}(q_1, \varepsilon), a), b), b), a)$$

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

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

$$= \delta(\delta(q_2, b), a)$$

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

$$= q_1, q_0 \notin F$$

Conclusion:
Therefore, the string **abba** is **<u>not accepted</u>** by FA M.

Acceptance of Strings by FA:

Example Problem-2: Solution

ii) babbab

$$\delta^{\wedge}(q_1, babbab) = \delta(\delta^{\wedge}(q_1, babba), b)$$

$$= \delta(\delta(\delta^{\wedge}(q_1, babb), a), b)$$

$$= \delta(\delta(\delta(\delta^{\wedge}(q_1, bab), b), a), b)$$

$$= \delta(\delta(\delta(\delta(\delta^{\wedge}(q_1, ba), b), b), a), b)$$

$$= \delta(\delta(\delta(\delta(\delta(\delta^{\wedge}(q_1, b), a), b), b), a), b)$$

$$= \delta(\delta(\delta(\delta(\delta(\delta(\delta^{\wedge}(q_1, \varepsilon), b), a), b), b), a), b)$$

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

$$= \delta(\delta(\delta(\delta(\delta(q_2, a), b), b), a), b)$$

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

$$= \delta(\delta(\delta(q_2, b), a), b)$$

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

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

$$= q_2, q_2 \in F$$

Conclusion:

Therefore, the string **babbab** is **accepted** by FA M.

Acceptance of Languages by FA:

A Language L is accepted by a **Finite Automata (FA),** $M = (Q, \Sigma, \delta, q_0, F)$ is denoted by L(M) and is the set of all strings accepted by M.

i.e., **L(M) = {x|δ(q0,x)** is in some **F}** , **x** is a string and **x∈L**

Acceptance of Languages by FA :

Example Problem:

Let M = (Q, Σ, δ, q0, F)

Where Q={q_0, q_1}, Σ={a, b}, F= {q_0} and δ is given by

States (Q)/ Input (Σ)	Input (Σ)	
	a	b
→ q_0	q_0	q_1
q_1	q_1	q_0

L(M) is set of all strings in $\{a, b\}^*$ that have even number of b's.

Acceptance of Languages by FA :

Example Problem: Solution

Let the string **aabba** which consisting of even number of b's.

$$\delta^{\wedge}(q_0, aabba) = \delta(\delta^{\wedge}(q_0, aabb), a)$$

$$= \delta(\delta(\delta^{\wedge}(q_0, aab), b), a)$$

$$= \delta(\delta(\delta(\delta^{\wedge}(q_0, aa), b), b), a)$$

$$= \delta(\delta(\delta(\delta(\delta^{\wedge}(q_0, a), a), b), b), a)$$

$$= \delta(\delta(\delta(\delta(\delta(\delta^{\wedge}(q_0, \in), a), a), b), b), a)$$

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

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

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

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

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

$$= q_0, q_0 \in F$$

Conclusion:

Therefore, the string **aabba** is **accepted** by FA M.

Therefore, the language L(M) is **accepted** by given FA M.

Lecture- 03

Introduction to Finite Automata (FA)

Summary

Introduction to Finite Automata:

- Introduction
- Model or block diagram
- Formal Definition
- Examples
- Transition table and transition diagram
- Acceptance of Strings and Languages by FA

-- Happy Learning