Unit - I | Lecture- 04

Deterministic Finite Automata(DFA)

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

Deterministic Finite Automata(DFA)

- Definition of DFA
- How a DFA Processes Strings
- Simpler Notations for DFA's
- Extending the Transition Function to Strings
- The Language of DFA

Types of Finite Automata (FA):

There are two types of FA

1. Deterministic Finite Automata (DFA)
2. Nondeterministic Finite Automata (NFA or NDFA)

Deterministic Finite Automata (DFA):

- For every input symbol '**a**', from a given state there is exactly one transition (there can be no transitions from a state also) and we can determine to which state the machine enters. So the machine is called **deterministic machine**, since it has finite number of states. Thus the machine is called **Deterministic Finite Machine** (**Automaton**) i.e., **Deterministic Finite Automata (DFA)**.

Deterministic Finite Automata (DFA):

- The automaton reads on symbol form the input tape and then enters a new state that depends only on the current state and the symbol just read i.e., **deterministic** in their operations. So this is called **Deterministic Finite Automata (DFA)**.

Rules /Restrictions in DFA:

- In **DFA**, from a state with same input symbol, there is no more than one transition exist.

Ex.

- In **DFA**, there is no transition with **ε** (**epsilon**).

Ex.

Model or block diagram of DFA:

- The model or block diagram of **Deterministic Finite Automata (DFA)** is same as **FA** as shown in the following diagram.

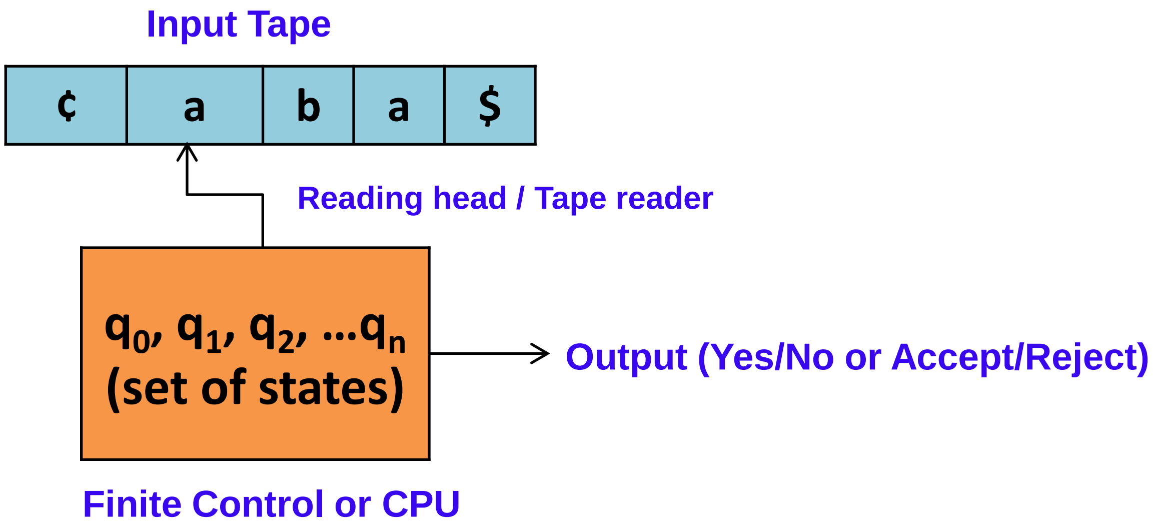


Figure: Model of DFA

Formal or Mathematical Definition of DFA:

A **DFA** 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 <u>initial or start state is only one</u>)

F : Set of Final or Accepting States, $F \subseteq Q$

Formal or Mathematical Definition of DFA:

M = (Q, Σ, δ, 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 X Σ** to **Q** i.e.,

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

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

Example of DFA:

Let a **DFA, 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:

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

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

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

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

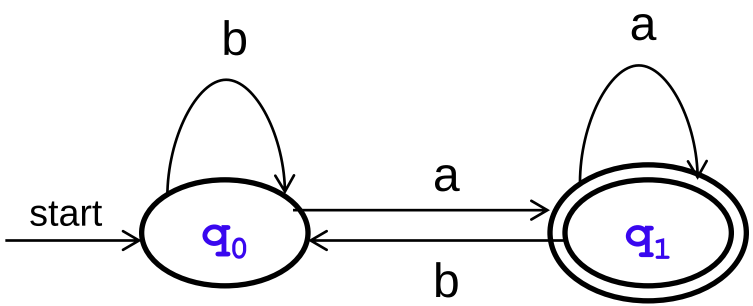
Simpler Notations for DFA's:

Specifying a **DFA** as a five-tuple with a detailed description of the **δ** transition function is both tedious and hard to read.

There are two preferred notations for describing automata:

1. A **transition diagram**, which is a graph such as the ones we saw in Lecture-03.
2. A **transition table** , which is a tabular listing of the δ function, which by implication tells us the set of states and the input alphabet, we saw in Lecture-03.

Example of Transition Diagram and Transition Table:



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

Properties of δ function of DFA (Extending the Transition Function to Strings)

- Properties of **δ** function of **DFA** or Extended **δ** definition of **DFA** 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*)**.

$$\therefore \delta^{\wedge} = Q X \sum^{*} \rightarrow Q$$

Properties of δ function of DFA :

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:

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

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

How a DFA Processes Strings (Acceptance of Strings)

A string w is accepted by a **DFA,**

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

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

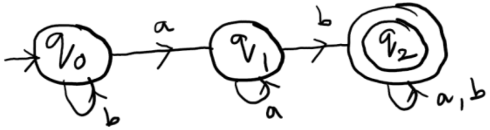
This is basically the acceptability of a string by the final state.

Acceptance of Strings by DFA:

Example Problem-1:

Consider the transition diagram as shown below:

Check whether the input string **aab** is accepted or not by the given DFA ?



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 DFA:

Example Problem-1: Solution

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

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

Conclusion:

Therefore, the string **aab** is **accepted** by DFA M.

Acceptance of Strings by DFA:

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 DFA or not?

Acceptance of Strings by DFA:

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 **not accepted** by DFA M.

Acceptance of Strings by DFA:

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 DFA M.

The Language of a DFA (Acceptance of Languages by DFA):

A Language L is accepted by a **DFA**,

$M = (Q, \Sigma, \delta, q_0, F)$ is denoted by L(M) and is the set of all strings accepted by M.

i.e., $L(M) = \{x \mid \delta(q0, x)$ is in some $F\}$, x is a string and $x \in L$

Acceptance of Languages by DFA :

Example Problem:

Let M = (Q, Σ, δ, q0, F)

Where $Q=\{q_0, q_1\}$, $\Sigma=\{a, b\}$, $F= \{q_0\}$ 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 DFA :

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 DFA M.

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

Disadvantages of DFA:

Various disadvantages of DFA are

1. Constructing a DFA is difficult.
2. The DFA cannot guess its inputs.
3. The DFA is not very powerful.
4. At any point of time, the DFA is only in one state, so a DFA does not have the power to be in several states at once.

Lecture- 04
Deterministic Finite Automata (DFA)
Summary

- Definition of DFA
- How a DFA Processes Strings
- Simpler Notations for DFA's
- Extending the Transition Function to Strings
- The Language of DFA

-- Happy Learning