Unit - I | Lecture- 05

Nondeterministic Finite Automata(NFA)

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

Nondeterministic Finite Automata(NFA)

- Introduction to NFA
- Definition of NFA
- Extended Transition Function
- The Language of NFA

Types of Finite Automata (FA):

There are two types of FA

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

Nondeterministic Finite Automata (NFA):

- Computers are completely deterministic machines i.e., **DFA**.
- The state of the computer can be predicted from the input and the initial state.
- We cannot find a computer which is nondeterministic.
- All the disadvantages of **DFA** can be overcome using **Nondeterministic Finite Automata (NFA)**.

Advantages of NFA:

- NFA is very easy to construct.
- NFA has the ability to guess something about its input.
- NFA is more powerful than DFA.
- NFA has the power to be in several states at once.
- NFA is an efficient mechanism to describe some complicated languages concisely.

Model or block diagram of NFA:

The abstract model and operation of NFA is similar to FA except NFA shall permit several possible next states for a given combination of current state and input symbols.

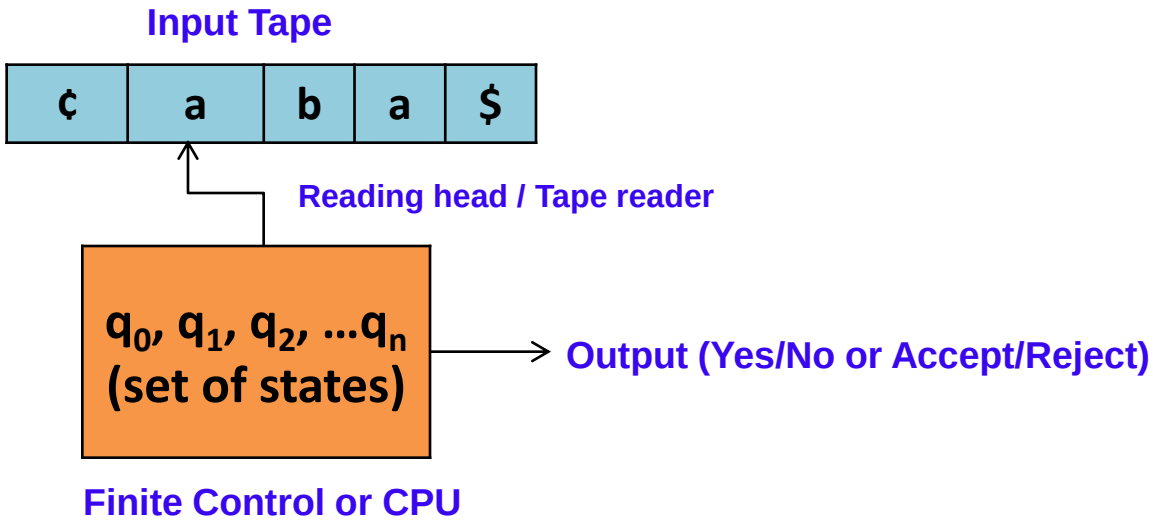
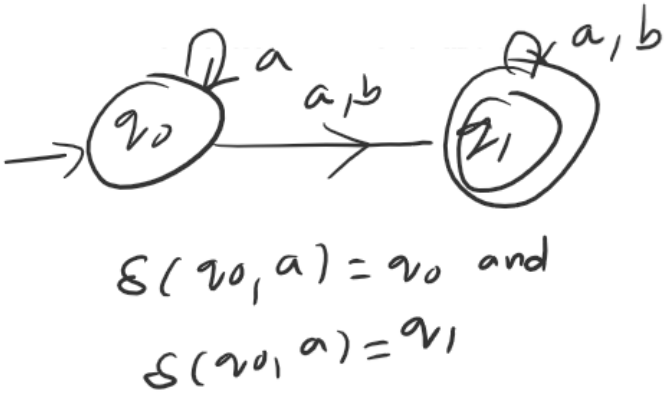


Figure: Model of NFA

Model or block diagram of NFA:

NFA allows zero, one or more transactions from a state on the same input symbol.



Explanation:

From the transition diagram, for the state **q_0** there exists two transitions for the same input symbol **'a'** as mentioned here, and these can be combined and written as:

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

Formal or Mathematical Definition of NFA:

A **NFA** is 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 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 **2^Q** i.e.,

$$\delta : QX \sum \rightarrow 2^Q$$

Where **2^Q** is power set of **Q,** the set of all states of **Q.**

Example of NFA:

Let a **NFA, 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}

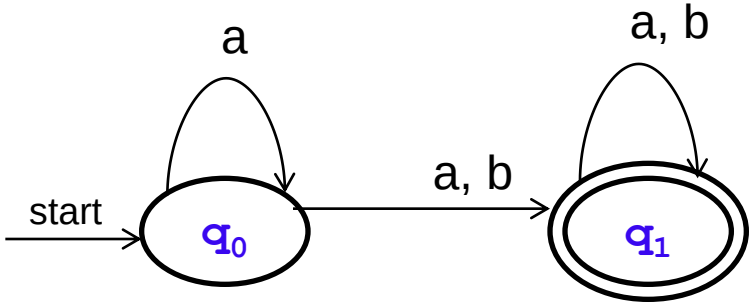
And **δ** is given below: $\delta(q_0, a) = \{q_0, q_1\}$

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

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

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

Example of Transition Diagram and Transition Table: NFA



Q/Σ	Input	
	a	b
→ q_0	$\{q_0, q_1\}$	q_1
(q_1)	q_1	q_1

Properties of δ function of NFA (Extended Transition Function)

- The intension is that **δ(q,a)** is the set of all states **p** such that there is a transition labeled '**a**' from **q** to **p**.
- The extended **δ** function is defined as

$$\therefore \delta^{\wedge} = QX\sum^{*} \rightarrow 2^{Q}$$

- $\delta^{\wedge}$ takes two parameters i.e., **states** and **strings**.

Properties of δ function of NFA :

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:

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

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

Properties of δ function of NFA :

Properties of $\delta^{\wedge}$

3. Property-3:

$$\delta(P, w) = \bigcup_{q_{in} P} \delta(q, w)$$

For each set of states $P \subseteq Q$

Ex.

P = {q_0, q_1} , w = a

δ({q_0, q_1}, a) = δ(q_0, a) U δ(q_1, a)

How a NFA Processes Strings (Acceptance of Strings)

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

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

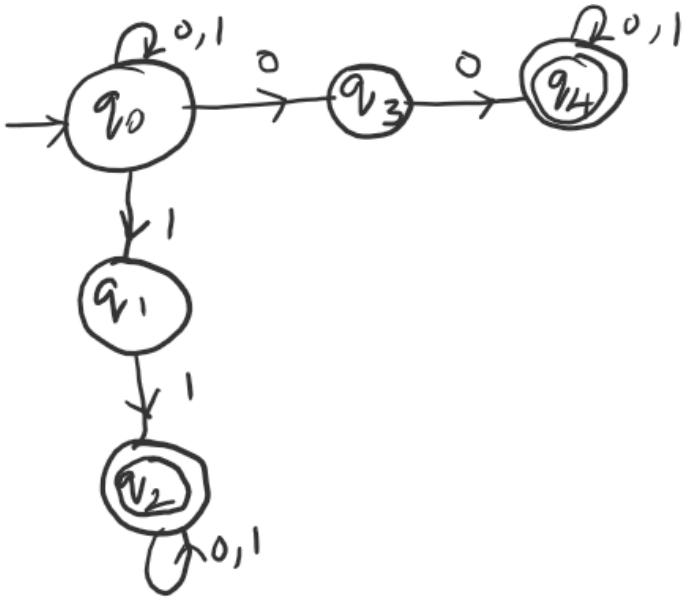
If $\delta^{\wedge}(q_0, w) = P$, for some **$P \in F$**

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

Acceptance of Strings by NFA:

Example Problem-1:

Consider the NFA given below:
Check whether the input string `01001` is accepted or not by the given NFA ?



Acceptance of Strings by NFA:

Example Problem-1: Solution

$\hat{\delta}(q_0, 01001) = ?$

1. $\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \epsilon), 0)$
$= \delta(q_0, 0)$
$= \{q_0, q_3\}$

2. $\hat{\delta}(q_0, 01) = \delta(\underline{\hat{\delta}(q_0, 0)}, 1)$
$= \delta(\{q_0, q_3\}, 1)$
$= \delta(q_0, 1) \cup \delta(q_3, 1)$
$= \{q_0, q_1\} \cup \phi$
$= \{q_0, q_1\}$

Acceptance of Strings by NFA:

Example Problem-1: Solution

3. $\hat{\delta}(q_0, 010) = \delta(\hat{\delta}(q_0, 01), 0)$

$= \delta(\{q_0, q_1\}, 0)$

$= \delta(q_0, 0) \cup \delta(q_1, 0)$

$= \{q_0, q_3\} \cup \phi$

$= \{q_0, q_3\}$

4. $\hat{\delta}(q_0, 0100) = \delta(\hat{\delta}(q_0, 010), 0)$

$= \delta(\{q_0, q_3\}, 0)$

$= \delta(q_0, 0) \cup \delta(q_3, 0)$

$= \{q_0, q_3\} \cup \{q_4\}$

$= \{q_0, q_3, q_4\}$

Acceptance of Strings by NFA:

Example Problem-1: Solution

5. $\hat{\delta}(q_0, 01001) = \delta(\hat{\delta}(q_0, 0100), 1)$

$= \delta(\{q_0, q_3, q_4\}, 1)$

$= \delta(q_0, 1) \cup \delta(q_3, 1) \cup \delta(q_4, 1)$

$= \{q_0, q_1\} \cup \phi \cup \{q_4\}$

$= \{q_0, q_1, q_4\}$

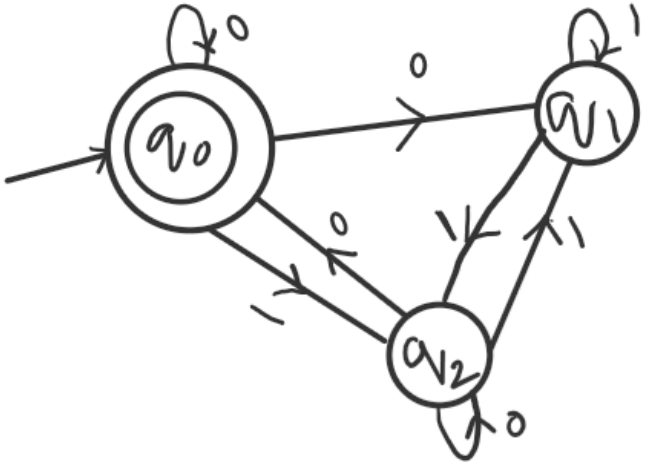
here $q_4 \in F$

$\therefore$ The string 01001 is accepted by NFA

Acceptance of Strings by NFA:

Example Problem-2:

For the NFA shown below:



Check whether the string **0100** is accepted by the given NFA or not?

Acceptance of Strings by NFA:

Example Problem-2: Solution

$$\hat{\delta}(q_0, 0100) = ?$$

1.

$$\hat{\delta}(q_0, 0) = \delta(\hat{\delta}(q_0, \varepsilon), 0)$$

$$= \delta(q_0, 0)$$

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

2.

$$\hat{\delta}(q_0, 01) = \delta(\hat{\delta}(q_0, 0), 1)$$

$$= \delta(\{q_0, q_1\}, 1)$$

$$= \delta(q_0, 1) \cup \delta(q_1, 1)$$

$$= \{q_2\} \cup \{q_1, q_2\}$$

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

Acceptance of Strings by NFA:

Example Problem-2: Solution cont'd.

3.

$$\delta^{\wedge}(q_0, 010) = \delta(\delta^{\wedge}(q_0, 01), 0)$$
$$= \delta(\{q_1, q_2\}, 0)$$
$$= \delta(q_1, 0) \cup \delta(q_2, 0)$$
$$= \phi \cup \{q_0, q_2\}$$
$$= \{q_0, q_2\}$$

4.

$$\delta^{\wedge}(q_0, 0100) = \delta(\delta^{\wedge}(q_0, 010), 0)$$
$$= \delta(\{q_0, q_2\}, 0)$$
$$= \delta(q_0, 0) \cup \delta(q_2, 0)$$
$$= \{q_0, q_1\} \cup \{q_0, q_2\}$$
$$= \{q_0, q_1, q_2\}, q_0 \in F$$

Conclusion:
The input string **0100** is accepted by the given NFA.

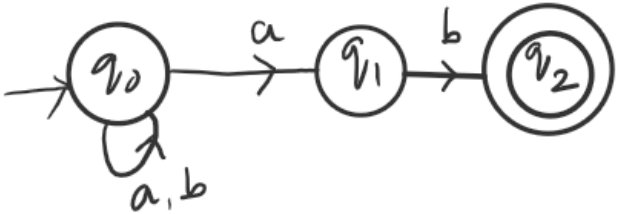
The Language of a NFA (Acceptance of Languages by NFA):

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

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

i.e., $L(M) = \{x \mid \hat{\delta}(q_0, x)$ is in some $F\}$, x is a string and $x \in L$

Moves made by of NFA (or Proliferation of Moves of NFA):



For the strings abaab and abb

1) abaab

$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{a} q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0$, $q_0 \notin F$ (Reject)

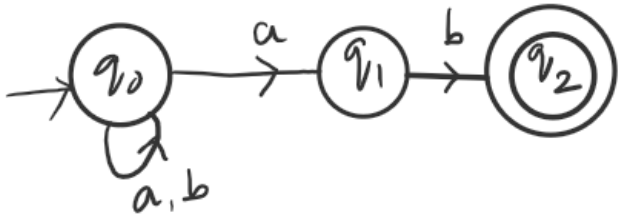
$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2$ (Rejected) ∵ Complete string not included

$q_0 \xrightarrow{a} q_1$ (Rejected) ∵ Complete string not included

$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2$, $q_2 \in F$ (ACCEPTED)

The string abaab is processed here.

Moves made by of NFA (or Proliferation of Moves of NFA):



For the strings abaab and abb

2) abb

$q_0 \xrightarrow{a} q_0 \xrightarrow{b} q_0 \xrightarrow{b} q_0$, $q_0 \notin F$ (Rejected)

$q_0 \xrightarrow{a} q_1 \xrightarrow{b} q_2 \xrightarrow{b} \times$ (Rejected)

complete string not processed

∴ Both cases the string is not accepted.

Differences between NFA and DFA

NFA	DFA
NFA is non-deterministic.	DFA is deterministic.
NFA is very easy to construct.	DFA is very difficult to construct.
A NFA has the ability to guess about its input.	DFA cannot guess its inputs.
NFA is more powerful.	DFA is not very powerful.
NFA has the power to be in several states at once.	DFA does not have the power to be in several states at once.

Lecture- 05
Nondeterministic Finite Automata (NFA)
Summary

- Introduction to NFA
- Definition of NFA
- Extended Transition Function
- The Language of NFA