**Chapter 2: FINITE AUTOMATA**

**Topic – 1: Basic Concepts**

**Purpose**

* ***Finite automata*** are used to **recognize patterns**.

**Working**

* Automaton takes a **string input** & **changes its state** as per symbols in it.
* Two types of states are ***accept state*** & ***reject state***.

**Formal Definition**

**FA = {Q, Σ, δ, q0, F}**

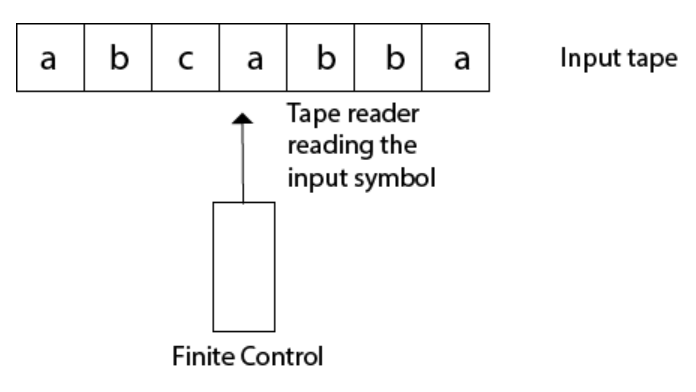
**Q = Set of states**

**δ = Transition function**

**F = Final state**

**Topic – 2: Finite Automata Model**

* It can be understood using **input tape** & **finite control**.
* **Finite control:** Reads each **cell** one by one from the tape.



**Topic – 3: Types Of Finite Automata**

**Deterministic Finite Automata**

* **DFA:** Moves to only **one state** for a given input character.
* It **doesn’t** accept **null** character.
* Used in **lexical analysis** in **compiler design**.

**Non-deterministic Finite Automata**

* **NFA:** Can move to **multiple states** for a given input character.
* It accepts a **null** character.
* More of a theoretical concept **without** much actual use.

**Note!**

**🡪 Every DFA is NFA, but not all NFA are DFA.**

**🡪 Both can have multiple final states.**

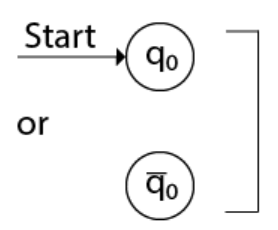
**Topic – 4: Transition Diagram**

**Transition Equation**

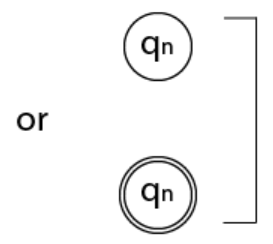
**δ(q,a) = p**

**Means transition from state q to state p after getting input character a.**

**Start State Symbols**



**Final State Symbols**



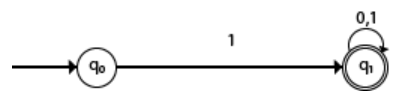
**Computing DFA**

* The state reached after computing whole input is represented by **r**.
* **r** should belong to the accept state **F** to be accepted (**r Є F**).

**Example**

**Given: Construct an NFA where strings start with 1 (Σ = {0, 1}).**

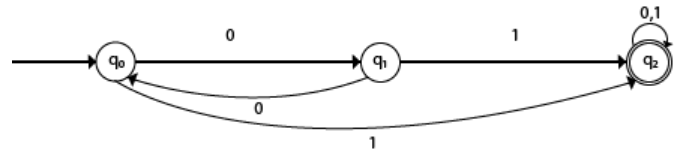
**Ans:**

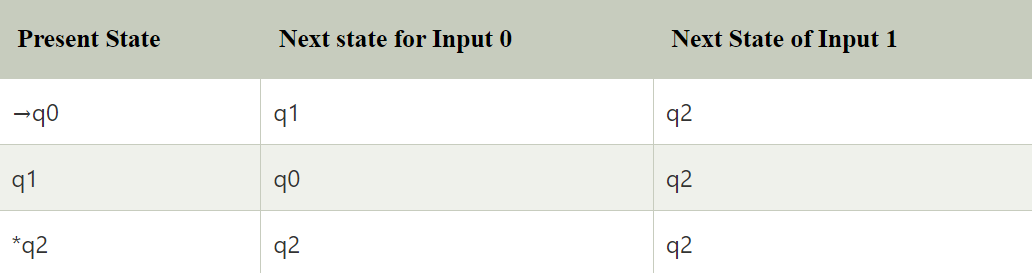
****

**There is no δ(q0, 0) transition because NFA is constructed in a way to reach the accept state only, discarding other possibilities.**

**Topic – 5: Transition Table**

**DFA Example**





**About NFA Example**

* In **NFA**, there might be transition **multiple states**.

**Topic – 6: Deterministic Finite Automata (DFA)**

**Definition**

* A finite automaton is **DFA** if it reads **one symbol** at a time from a given string.

**Nature**

* There is just **one path** from one state to another.

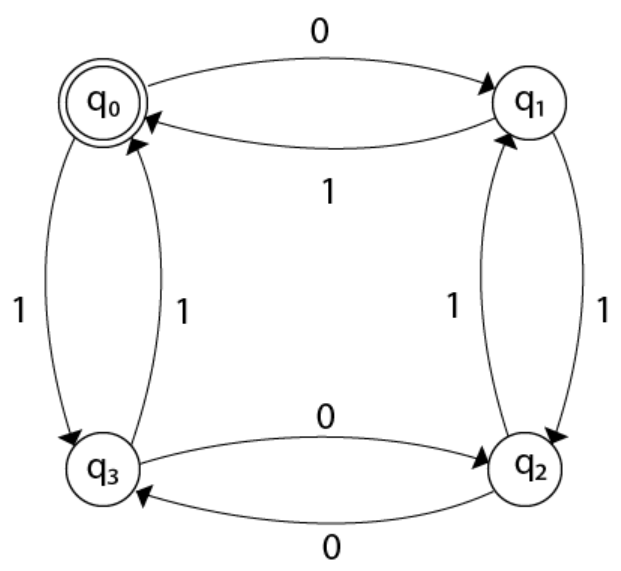
**Transition Function**

**δ: Q × Σ 🡪 Q**

* Means that cartesian product of **set of states** over **alphabet** gives another state from the same set.

**Example**

**Ques: Make FA which accepts even number of 0s & even number of 1s (Σ = {0, 1}).**

****

**Topic – 7: Non-Deterministic Finite Automata (NFA)**

**Note!**

**🡪 NFA contains Ԑ transition.**

**🡪 The state in last row of transition table is assumed to be the accept state.**

**Transition Function**

**δ: Q × Σ 🡪 2Q**

* Means the cartesian product of **set of states** over **alphabet** gives a **subset** from the same set.