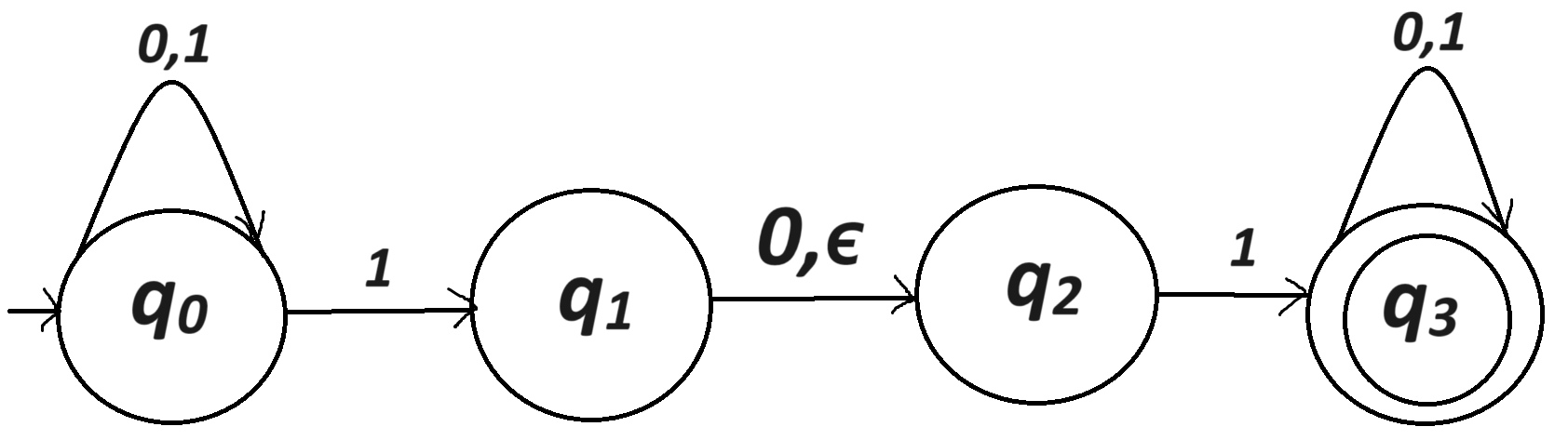
**Chapter 2: NON-DETERMINISTIC FINITE AUTOMATA**

**Topic – 1: General Information**

* The automaton can travel to **multiple states simultaneously**.
* There is transition from one state to another **without** reading any bit.
* This transition is denoted by **Є**.
* If we end up at **even one** accept state, then we accept the input.
* A state can transit to **multiple states**, even no state.

**Topic – 2: Example 1**



**Transition Explanation**

**{q0, 0}= {q0}**

**{q0, 1}= {q0, q1}**

**{q1, 0}= {q2}**

**{q1, 1}= { }**

**Etc.**

**Symbol For NFA Set**

**ΣЄ = Σ U {Є}**

**Finding Final State**

* It is with reference to the previous example, see the diagram above.

**Solve: 010110**

|  |  |
| --- | --- |
| **Active States** | **New Bit** |
| **{q0}** | **0** |
| **{q0}** | **1** |
| **{q0, q1, q2}** | **0** |
| **{q0, q2}** | **1** |
| **{q0, q1, q2,q3}** | **1** |
| **{q0, q1, q2, q3}** | **0** |
| **{q0, q2, q3}** | **-** |

**Topic – 3: Definition**

**Difference Between DFA & NFA**

|  |  |
| --- | --- |
| **DFA** | **NFA** |
| **(q,a) 🡪 Single state** | **(q,a) 🡪 Multiple states** |
| **Single computation** | **Multiple computation** |
| **No Є transition** | **Є transition** |
| **Input is accepted if computation end up at accept state.** | **Input is accepted if computation ends at any accept state.** |

**Definition**

* Set of elements contained in **N**.

**N = (Q, Σ, δ, q0, F)**

**δ = Q x ΣЄ 🡪 2Q**

**Formal Definition**

**r0 = q0 (initial condition)**

**ri Єδ(ri-1, bi) (transition condition)**

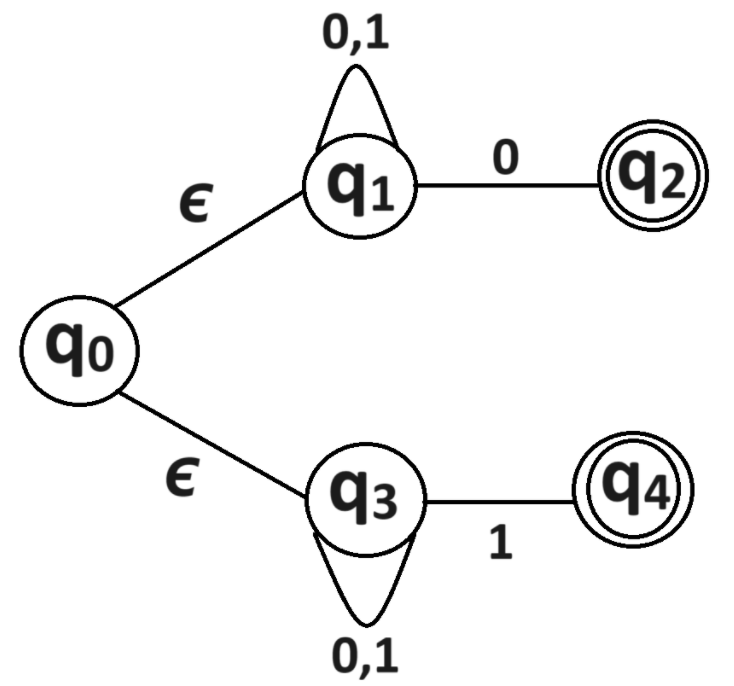
**rm Є F (acceptance condition)**

**NFA Language**

**L(N) = {w Є Σ\* | N accepts w}**

**Topic – 4: Example 2**

* Let’s consider making automaton which accepts **binary strings divisible by 2** or **binary strings which are** **odd number**.



**Topic – 5: Equivalence Of NFA & DFA**

**Introduction**

* Every **DFA** is also an **NFA**.
* Or simply, a **DFA** is a special type of **NFA** with one restriction.

**Theorem**

**Let N = (Q, Σ, δ, q0, F) be an NFA.**

**There exists a DFA where D = (Q', Σ, δ', q0', F') such that L(N) = L(D)**

**Construction Of D**

**Q' = 2Q [Set of strings in NFA are the powerset of set of strings in DFA]**

**Let A ⊆ Q, then δ'(A,a) = Ur Є A (δ(r,a))**

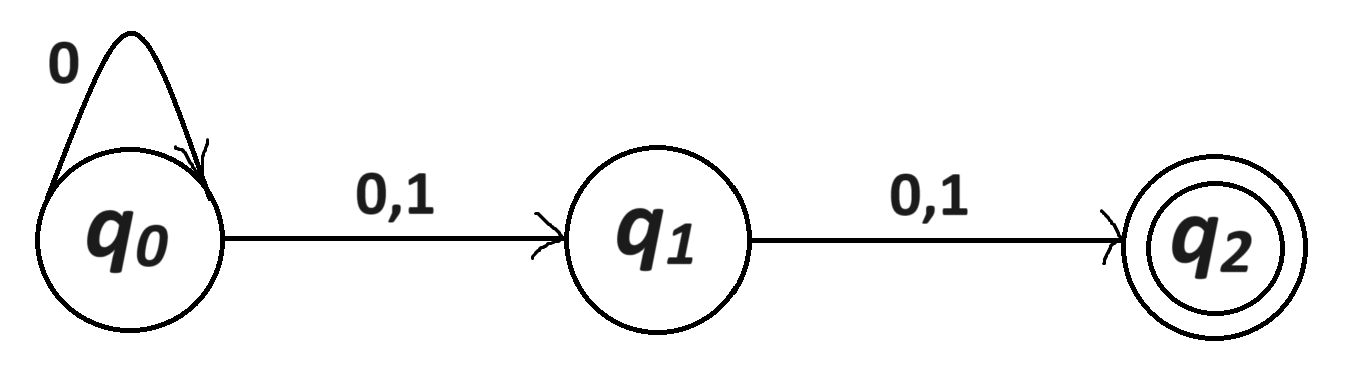
**q0' = {q0} [q0' equals to subset containing q0]**

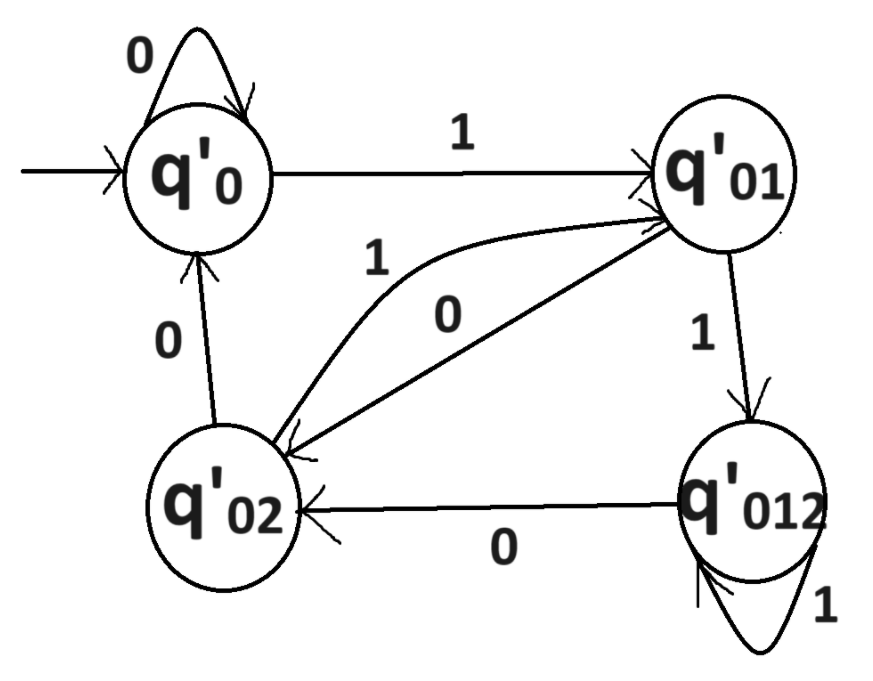
**F' = {A ⊆ Q | A ∩ F ≠ φ} [All subsets that contain some final state]**

**E(R) = U {t | t is reachable from r using 0 or more Є transition}**

**Example**

**L = {w | 2nd last symbol in w is 1}**



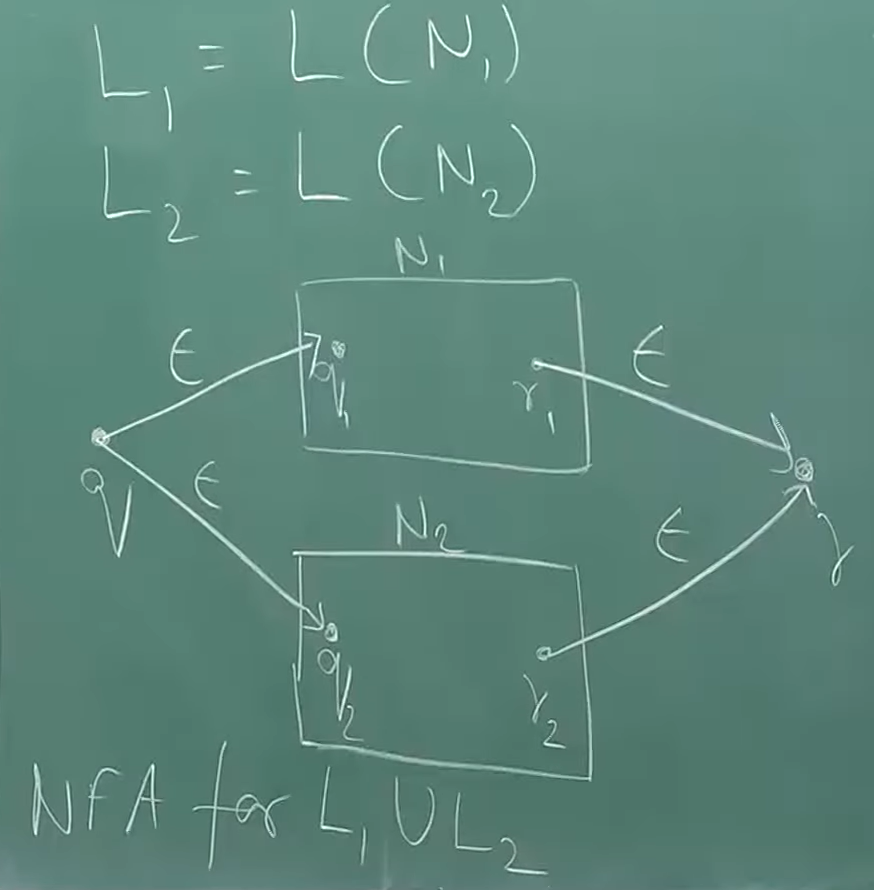


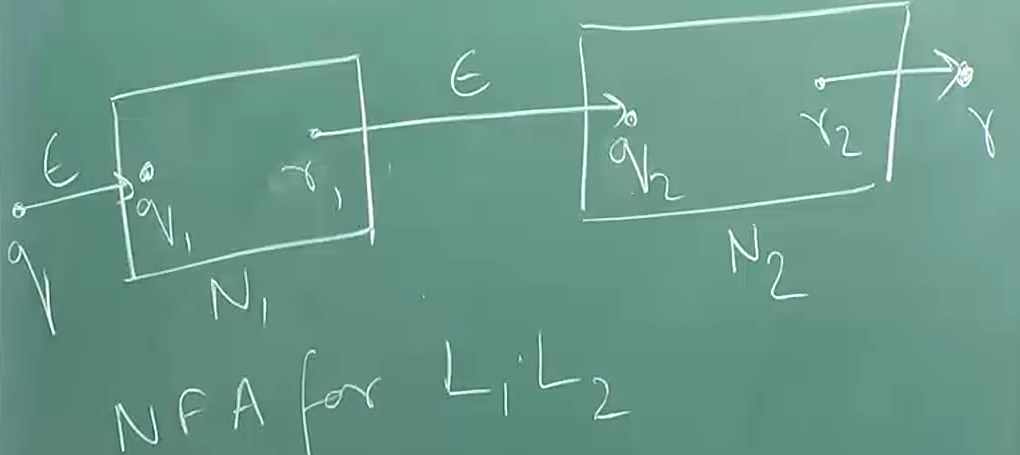
**Theorem – II**

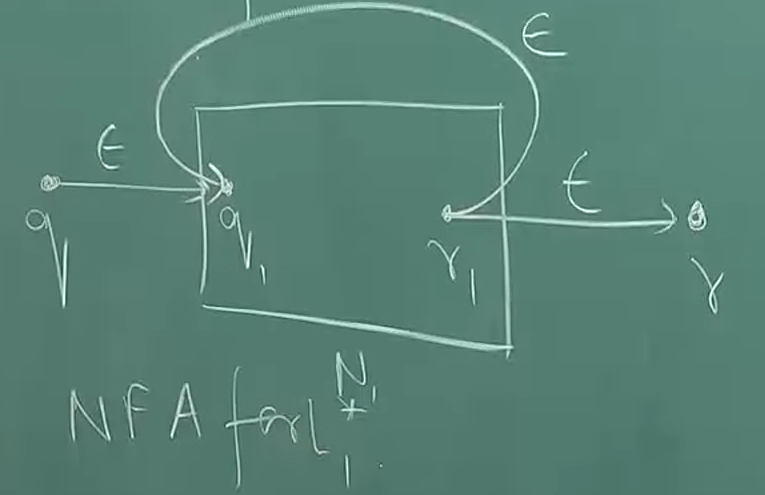
**🡪 If L1 and L2 are regular, then L1 U L2, L1.L2, L\* are also regular.**

**🡪 For a regular language we can assume that there is NFA accepting it with a unique accept state.**

* **r1**, **r2** etc below are corresponding **accept states**.







* These all somehow prove that the **closure properties** above are all **regular**.