Nondeterministic Finite Automata

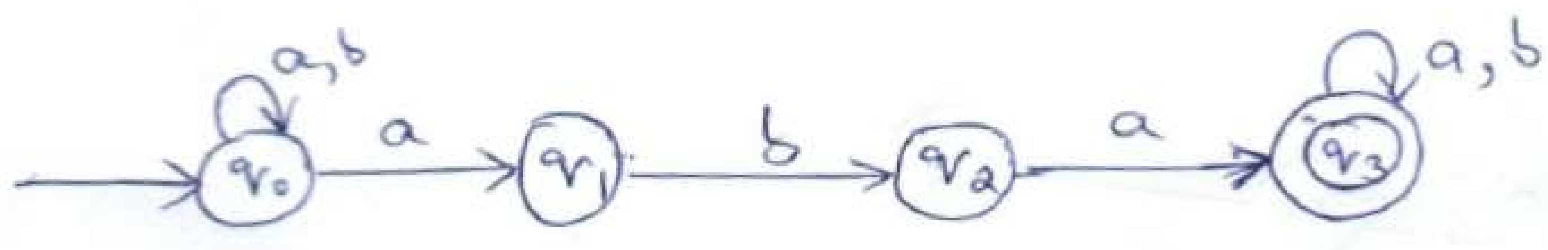
NON DETERMINISTIC FINITE AUTOMATA (NFA)

- It is an **finite automata where there exists 0, 1 or more transitions on a state for a given input symbol**.
- A nondeterministic finite automata (NFA) has the power to be in several states at once.
- This ability is often expressed as an ability to “guess” something about its input.
- **NFA is more useful when the automaton is used to search for all possible solutions.**

Non deterministic Finite Automata (NFA)

- When the automaton is used to search for certain sequences of characters (e.g., keywords) in a long text string, it is helpful to “guess” that we are at the beginning of one of those strings and use a sequence of states to do nothing but check that the string appears, character by character.

- **Example**: Design an NFA for the language of strings with 'aba' as substring.



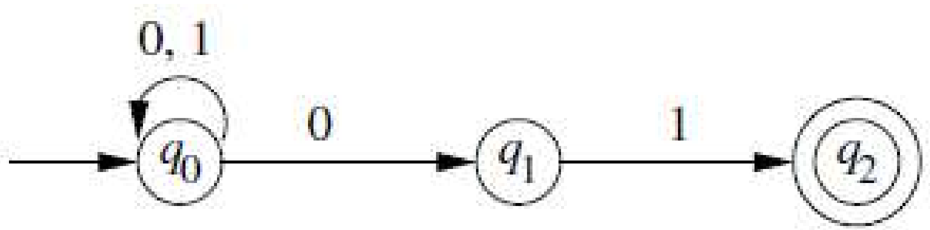
Non deterministic Finite Automata (NFA)

- **Definition of Nondeterministic Finite Automata**:
  A finite automata is a 5 tuple. It is the machine
  $M = (Q, \Sigma, \delta, q_0, F)$
  where

  - Q – Non empty set of finite number of states.
  - Σ – Non empty set of finite number of symbols or Finite input alphabet.
  - δ – State transition function, defined as
    $\delta: Q \times \Sigma \rightarrow 2^Q$ one or more states (power set of Q).
  - q_0 – It is initial or start state, $q_0 \in Q$.
  - F – $F \subseteq Q$, It is set of Final or Accepting states.

NON DETERMINISTIC FINITE AUTOMATA (NFA)

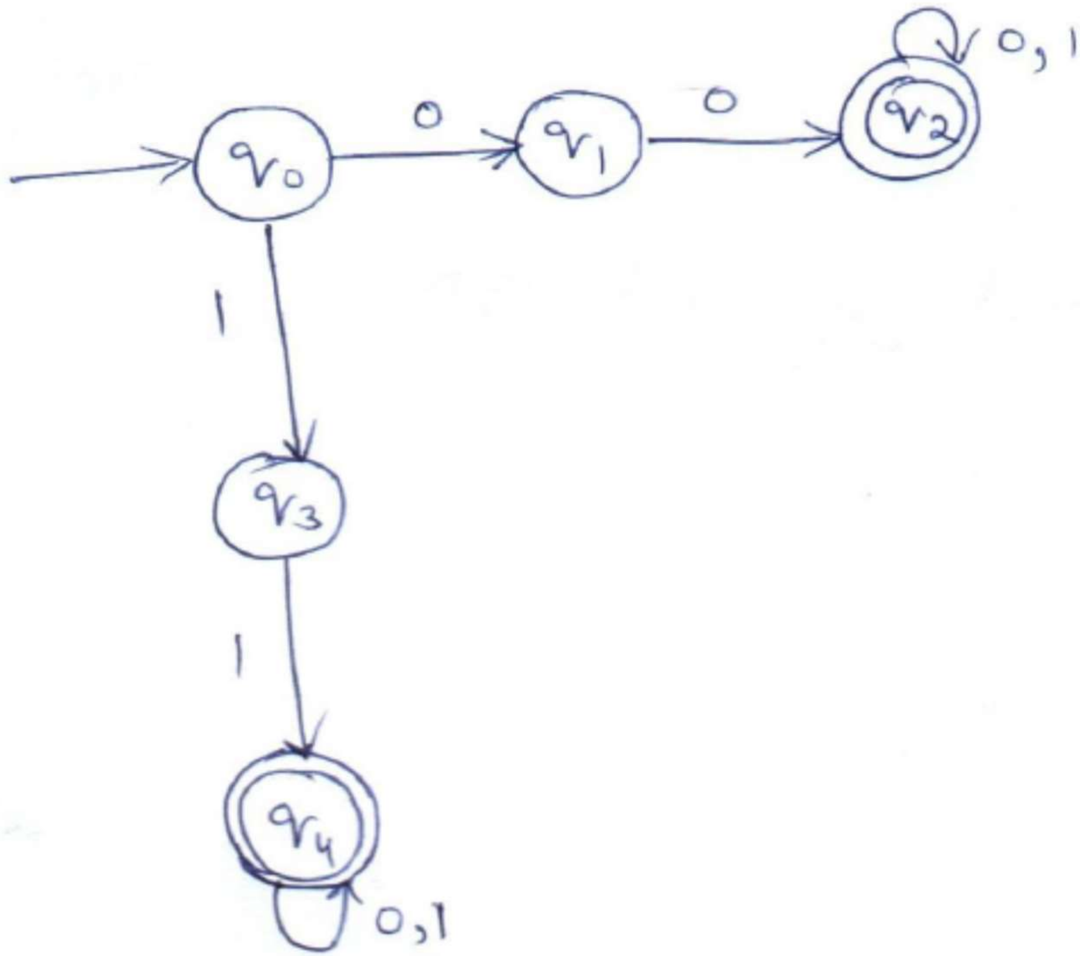
- **Example**: Let $\Sigma = \{0, 1\}$. Design an NFA for the language of strings end with '01'.



An NFA accepting all strings that end in 01

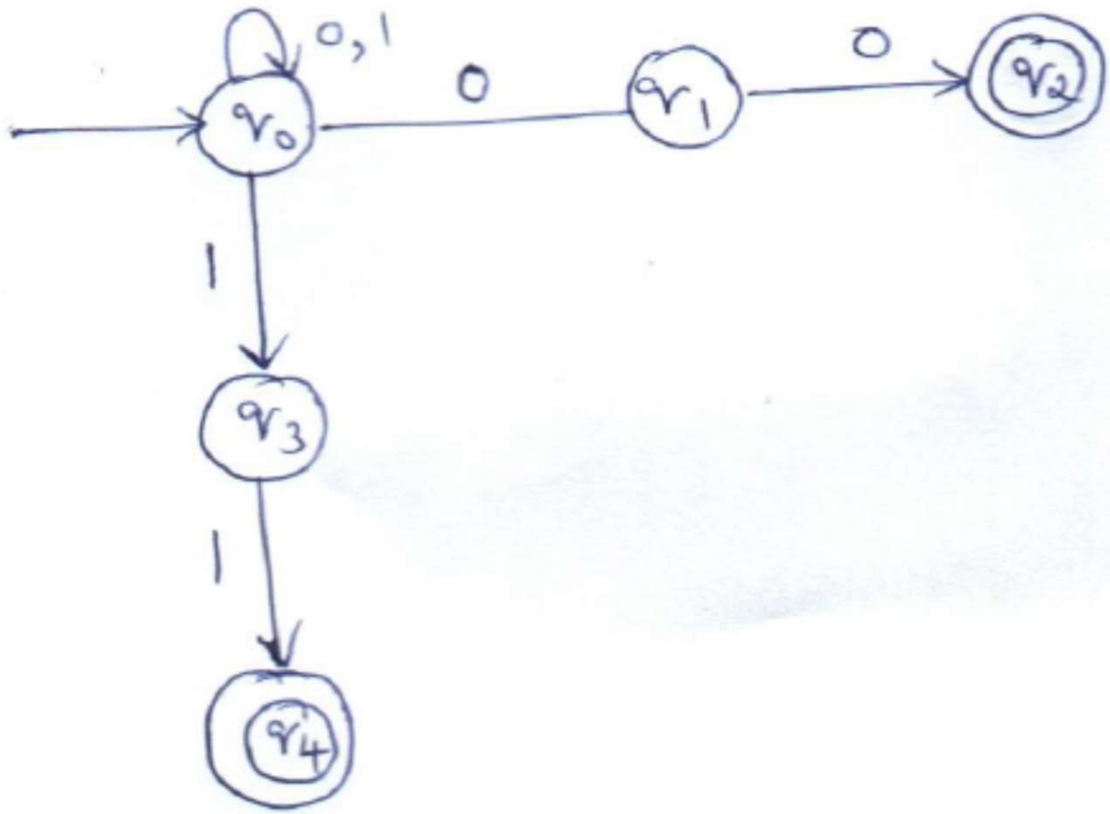
NON DETERMINISTIC FINITE AUTOMATA (NFA)

- **Example**: Let Σ = {0, 1}. Design an NFA for the language of strings begin with either '00' or '11'.



NON DETERMINISTIC FINITE AUTOMATA (NFA)

- **Example**: Let $\Sigma = \{0, 1\}$. Design an NFA for the language of strings end with either '00' or '11'.



NON DETERMINISTIC FINITE AUTOMATA (NFA)

- **Acceptance of a string by NFA**: A string is said to be accepted by NFA, if there exists at least one path that takes NFA from initial state to the final state.

CONVERSION OF NFA TO DFA

- **Step 1**: The initial state of DFA is same as the initial state of NFA.

- **Step 2**: Apply every input symbol on the initial state of DFA following the transitions in NFA. Use the following rules:
  $\delta(q_0, a) = \{q_1, q_2, , \ldots, q_i\}$ then
  $\delta'(q_0', a) = [q_1, q_2, , \ldots, q_i]$

- **Step 3**: Apply every input symbol on the new states obtained by step2 by using the following rule.
  $\delta'([q_1, q_2, \ldots, q_i], a) = [p_1, p_2, , \ldots, p_j]$
  if and only if
  $\delta(\{q_1, q_2, \ldots, q_i\}, a) = \{p_1, p_2, \ldots, p_j\}$
  $\delta(\{(q_1, a) \cup (q_2, a) \cup \ldots, (q_i, a)\} = \{p_1, p_2, \ldots, p_j\}$

Conversion of NFA to DFA

- **Step 4**: Repeat step3 until no more new states are generated.

- **Step 5**: The final states of DFA are going to be those states which contains at least one final state of NFA.

CONVERSION OF NFA TO DFA

- **Example 1**: Convert the following NFA to DFA.

- **Solution**: Transition table of NFA.

δ	0	1
$\rightarrow q_0$	$\{q_0, q_1\}$	$\{q_0\}$
q_1	$\{q_2\}$	ϕ
$* q_2$	$\{q_2\}$	$\{q_2\}$

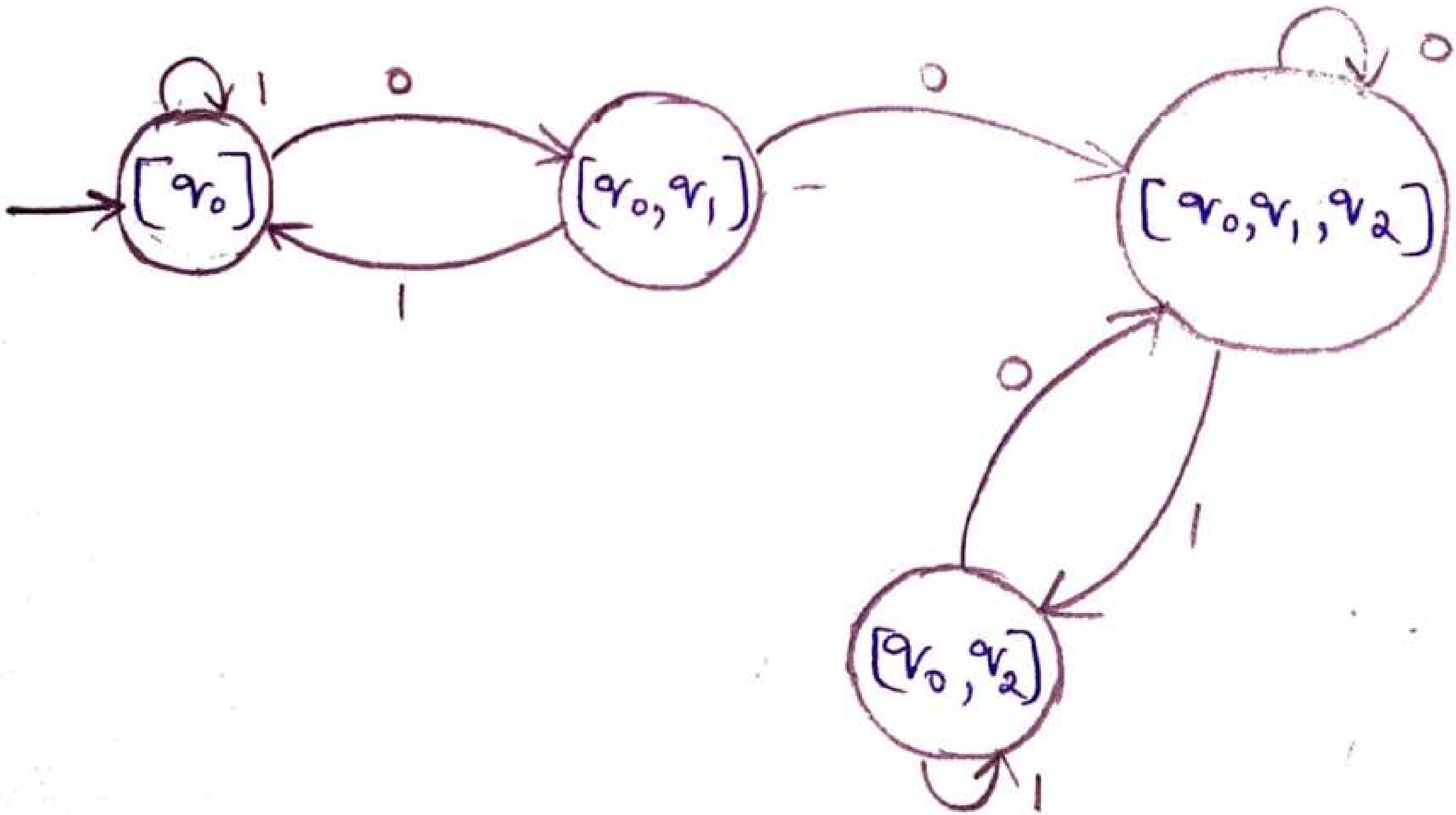
Conversion of NFA to DFA

- Transition table of DFA.

δ'	0	1
$\rightarrow [q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0]$
$[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_2]$
$[q_0, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_2]$

Conversion of NFA to DFA

- Transition diagram of DFA.



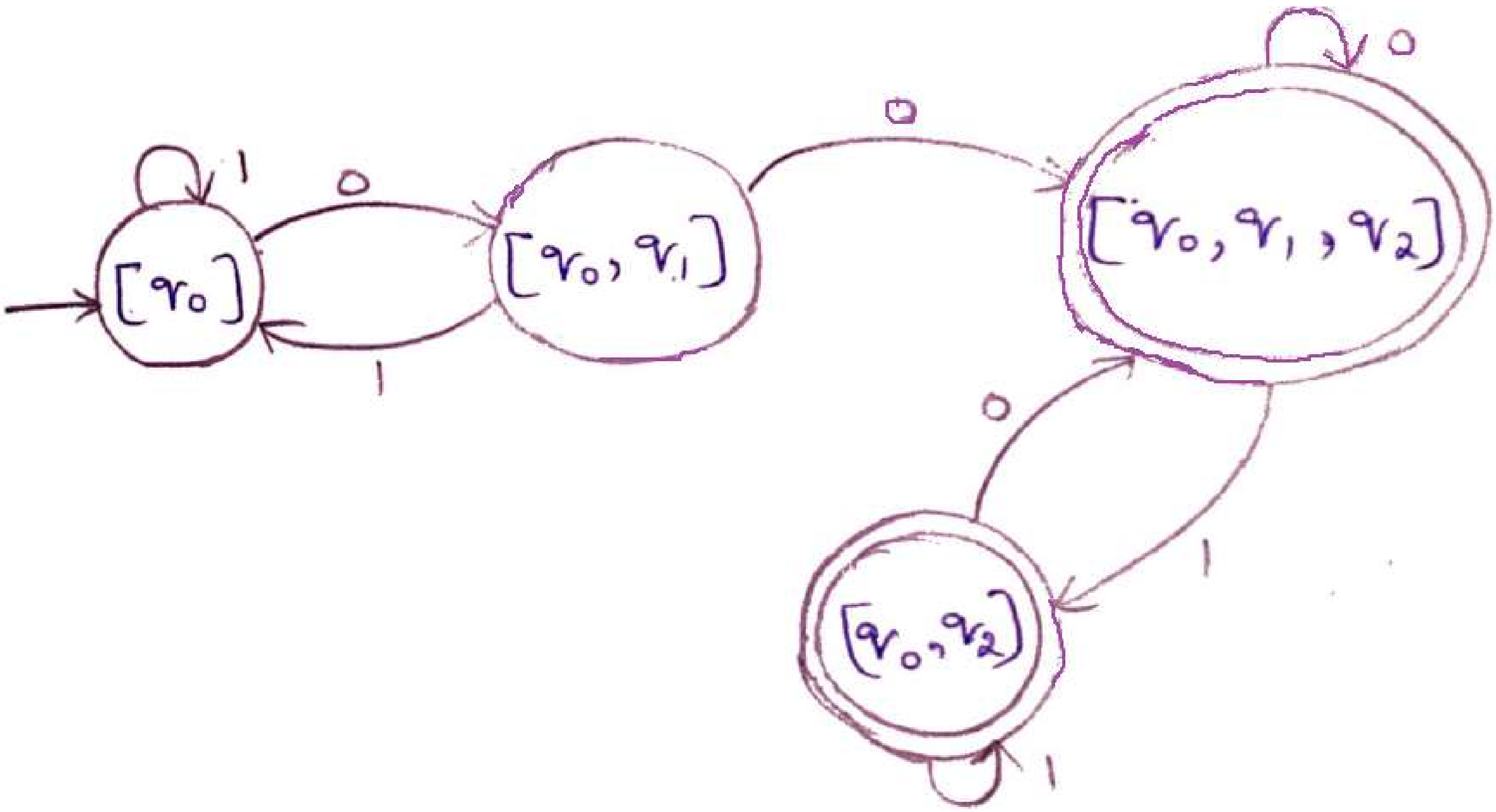
CONVERSION OF NFA TO DFA

- Transition table of DFA with final states.

δ'	0	1
$\rightarrow [q_0]$	$[q_0, q_1]$	$[q_0]$
$[q_0, q_1]$	$[q_0, q_1, q_2]$	$[q_0]$
$*[q_0, q_1, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_2]$
$*[q_0, q_2]$	$[q_0, q_1, q_2]$	$[q_0, q_2]$

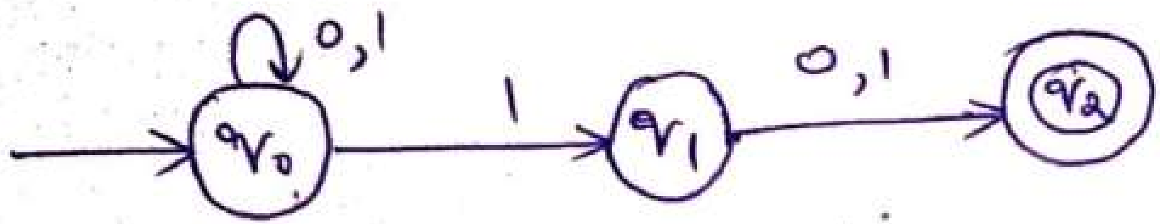
Conversion of NFA to DFA

- Transition diagram of DFA with final states.



Conversion of NFA to DFA

- **Example 2**: Convert the following NFA to DFA.



- **Solution**: Transition table of NFA.

δ	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
$*q_2$	ϕ	ϕ

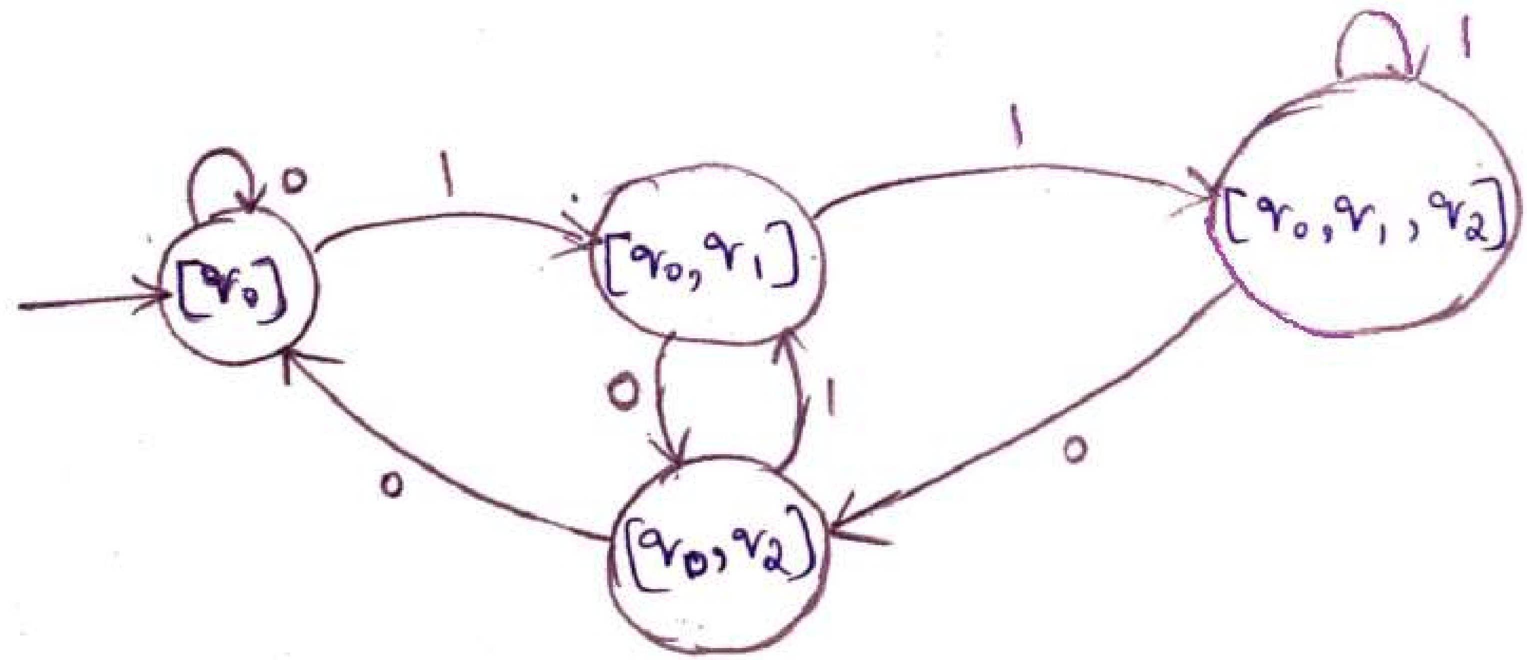
Conversion of NFA to DFA

- Transition table of DFA.

δ'	0	1
$\rightarrow [q_0]$	$[q_0]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_2]$	$[q_0, q_1, q_2]$
$[q_0, q_2]$	$[q_0]$	$[q_0, q_1]$
$[q_0, q_1, q_2]$	$[q_0, q_2]$	$[q_0, q_1, q_2]$

Conversion of NFA to DFA

- Transition diagram of DFA.



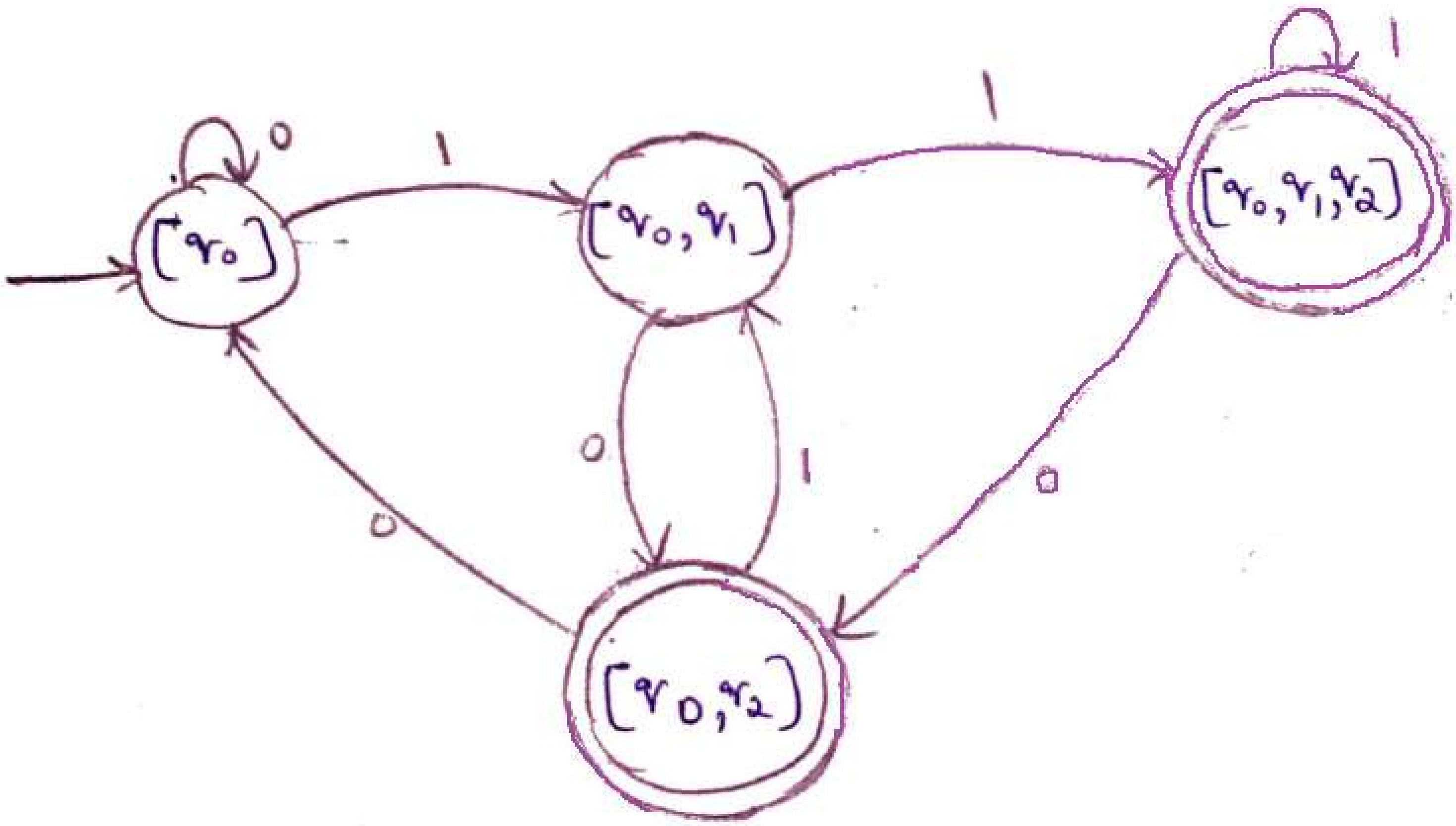
Conversion of NFA to DFA

- Transition table of DFA with final states.

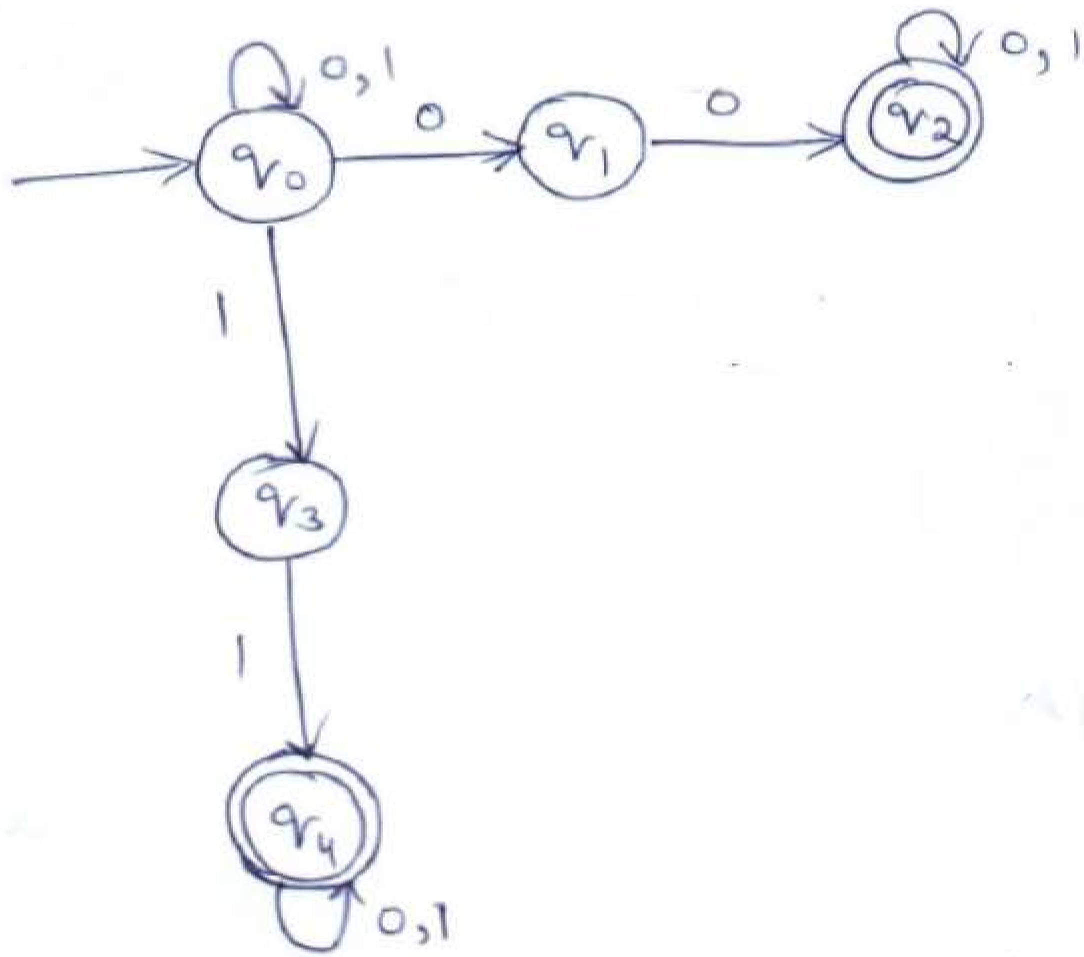
δ'	0	1
$\rightarrow [q_0]$	$[q_0]$	$[q_0, q_1]$
$[q_0, q_1]$	$[q_0, q_2]$	$[q_0, q_1, q_2]$
$*[q_0, q_2]$	$[q_0]$	$[q_0, q_1]$
$*[q_0, q_1, q_2]$	$[q_0, q_2]$	$[q_0, q_1, q_2]$

CONVERSION OF NFA TO DFA

- Transition diagram of DFA with final states.



Conversion of NFA to DFA



Equivalence of NFA and DFA

- **Theorem**: Let 'L' be a set accepted by a non deterministic finite automata then there exists a DFA which accepts 'L'.

- **Proof**:
- Let $M = (Q, \Sigma, \delta, q_0, F)$ be the NFA which accepts 'L'.

- Let us define a DFA $M' = (Q', \Sigma, \delta', q_0', F')$ be an DFA which accepts 'L'.
- The initial state of DFA is same as initial state of NFA.

$$q_0' = [q_0]$$
$$Q' = 2^Q$$

The states of the DFA are subsets of the set of states of NFA.

Equivalence of NFA and DFA

- $[q_1, q_2, , \ldots, q_i]$ is an individual state in DFA, where $q_1, q_2, , \ldots, q_i$ are separate states in NFA.

- The final states of DFA are those states which contains at least one final state of NFA.

- The transition function δ' is defined as
  $\delta'([q_1, q_2, \ldots, q_i], a) = [p_1, p_2, , \ldots, p_j]$
  if and only if
  $\delta(\{q_1, q_2, \ldots, q_i\}, a) = \{p_1, p_2, \ldots, p_j\}$.

- Now let us **prove** that **the language accepted by the NFA and DFA are equal**.
- For this we need to **prove** that **every string accepted by NFA is accepted by DFA**.

Equivalence of NFA and DFA

- Let us **apply** the **Mathematical Induction on the length of string 'x'**, to show that there are similar type of transitions in both NFA and DFA after processing string 'x'.

- That is, we need to prove that
  $\delta'(q_0', x) = [q_1, q_2, , \ldots, q_i]$
  iff
  $\delta(q_0, x) = \{q_1, q_2, , \ldots, q_i\}$.

- **Basis step**: Consider a string of length 0
  $\delta(q_0, \epsilon) = q_0$,
  $\delta'(q_0', \epsilon) = q_0'$,
  we know that $q_0' = [q_0]$.

Equivalence of NFA and DFA

- This means that on a string of length 0, both NFA and DFA are having similar type of transitions. Hence Basis step is proved.

- **Inductive Hypothesis**: Let us assume that for strings of length 'k' both NFA and DFA are having similar type of transitions.
  $\delta'(q_0', x) = [p_1, p_2, , \ldots, p_j]$
  iff
  $\delta(q_0, x) = \{p_1, p_2, , \ldots, p_j\}$.

- **Inductive Step**: We need to prove that for a string of length (k+1) also, NFA and DFA will be having similar type of transitions.

EQUIVALENCE OF NFA AND DFA

- i.e., we required to prove that
  $\delta'(q_0', xa) = [r_1, r_2, , \ldots, r_k]$
  iff
  $\delta(q_0, xa) = \{r_1, r_2, , \ldots, r_k\}$.

- Let us consider, $\delta'(q_0', xa) = \delta'(\delta'(q_0', x), a)$
  $= \delta'([p_1, p_2, , \ldots, p_j], a)$
- From Inductive Hypothesis,
  $\delta'(q_0', x) = [p_1, p_2, , \ldots, p_j]$
  iff $\delta(q_0, x) = \{p_1, p_2, , \ldots, p_j\}$.

  $\delta'([p_1, p_2, , \ldots, p_j] , a) = [r_1, r_2, , \ldots, r_k]$ ($\because$ from the definition of DFA)
  iff $\delta(\{p_1, p_2, , \ldots, p_j\}, a) = \{r_1, r_2, , \ldots, r_k\}$.

Equivalence of NFA and DFA

- From the definition
  $\delta(\delta(q_0, x), a) = \{r_1, r_2, , \ldots, r_k\}$
  $\delta(q_0, xa) = \{r_1, r_2, , \ldots, r_k\}$.

  Hence proved.

- Now we need to prove that every string 'x' accepted by NFA is also accepted by DFA.

- If x is accepted by NFA
  $\delta(q_0, x) \in F$,

- i.e., one of the states $q_1, q_2, , \ldots, q_i$ must be accepted by NFA.

Equivalence of NFA and DFA

- Form the definition, if one of the states of $\delta(q_0, x) = \{q_1, q_2, , \ldots, q_i\}$ is a final state then $[q_1, q_2, , \ldots, q_i]$ is a final state in DFA which implies that x is accepted by DFA.

- Hence language accepted by NFA and DFA are equal.