FA is characterized into two types:

1. **Deterministic Finite Automata (DFA):**

DFA consists of 5 tuples {Q, Σ, q, F, δ}.

Q: a set of all states.

Σ: a set of input symbols. (Symbols that which machine takes as input)

q: Initial state. (Starting state of a machine)

F: the set of the final state.

δ: Transition Function, defined as δ: Q X Σ --> Q.

In a DFA, for a particular input character, the machine goes to one state only. A transition function is defined on every state for every input symbol. Also in DFA null (or ε or λ) move is not allowed.

For ***example***, below DFA with Σ = {0, 1} accepts all strings ending with 0.

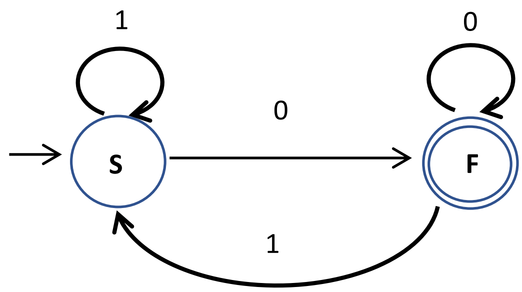


Figure: DFA with Σ = {0, 1}

2. **Nondeterministic Finite Automata(NFA):** NFA is similar to DFA except following additional features:
   - Null (or ε or λ) move is allowed i.e., it can move forward without reading symbols.
   - Ability to transmit to any number of states for a particular input.

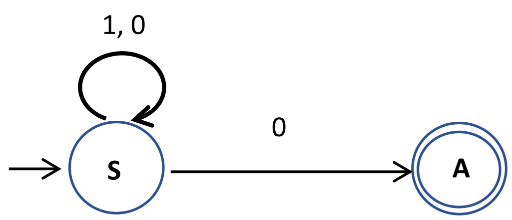


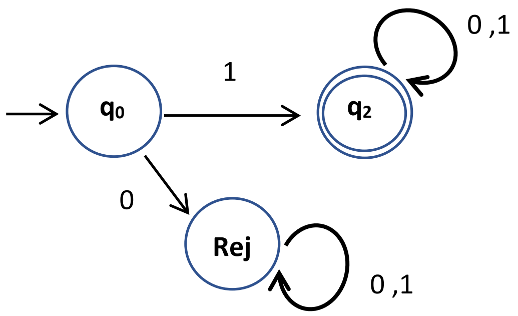
Figure: NFA with Σ = {0, 1}

Example /

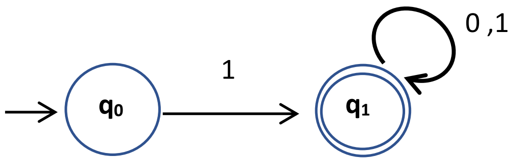
L= {1x|x ϵ {0,1}*} in DFA, NFA machines.

Answer /

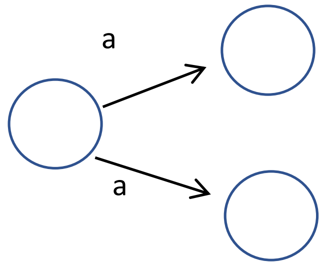
DFA:



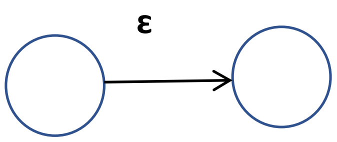
NFA:



❖ A nondeterministic finite automata (NFA) allows transitions on a symbol from one state to possibly more than one other state.



❖ Allow ε-transitions from one state to another whereby we can move from the first state to the second without inputting the next character.



❖ In an NFA a state may have zero, one, or more exiting arrows for each symbol of alphabet.

$\sum=\{a,b,c\}$