Deterministic Finite Automata (DFA)

- The *finite automaton (FA)* is a mathematical model of a system, with discrete inputs and outputs.
- A finite automaton consists of a finite set of *states*, and a set of *transitions* from states to states that occur in response to external "*inputs*" chosen from an alphabet Σ.
- The purpose of a state is to remember the relevant portion of the system's history. Since there are only a finite number of states, the entire history generally cannot be remembered. So, the system must be designed carefully.
- A state of the system summarizes the information concerning past inputs that is needed to determine the behavior of the system on subsequent inputs.
- Example: On/Off switch, the control mechanism of an elevator etc.
- Deterministic vs Nondeterministic: The control is "*deterministic*", meaning that the automaton cannot be in more than one state at any one time or "*nondeterministic*", meaning that it may be in several states at once.

Formal Definition of DFA: A DFA is a quintuple (5-tuple), that is, a system which consists of 5 elements. We write,

$$A = (Q, \Sigma, \delta, q_0, F), \text{ where}$$

- **Q**: finite nonempty set of states;
- **Σ (capital sigma):** finite nonempty set of input symbols, input alphabet;
- **δ (small delta):** transition function that takes as arguments a state and an input symbol and returns a state, $\delta: Q \times \Sigma \rightarrow Q$;
- **q_0**: initial/start state, $q_0 \in Q$;
- **F**: set of final or accepting states, $F \subseteq Q$.

Two notations for describing automata:

- *Transition Diagram*, which is a graph.
- *Transition Table*, which is a tabular listing of the δ function.

Example: Specify a DFA that accepts all and only the strings of 0's and 1's that have the sequence 01 somewhere in the string.

So, L: $\{w \mid w$ is of the form $x01y$ for some string x and y consisting of 0's and 1's only$\}$

$\Sigma = \{0, 1\}$, Say initial state q_0.

This automaton has to remember the important facts about what inputs it has seen so far. To decide whether 01 is a substring of the input, the automaton needs to remember:

1. Has it already seen 01? If so, then it accepts every sequence of further inputs. (Say, state q_2)
2. Has it never seen 01, but its most recent input was 0, so if it now sees a 1, it will have seen 01 and can accept everything it sees from here on? (Say, state q_1)
3. Has it never seen 01, but its last input was either nonexistent (it just started) or it last saw a 1? (Say, state q_0)

So, $Q = \{q_0, q_1, q_2\}$ and $F = \{q_2\}$

Transition diagram:

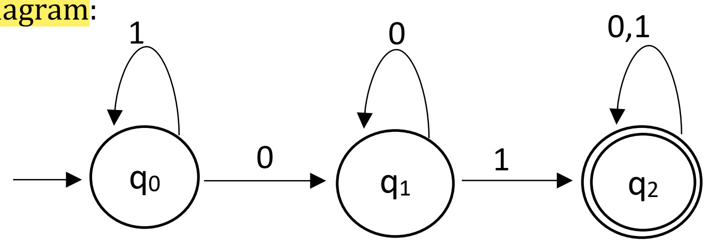

**Diagram Courtesy: Ashraful Haq Ove (CSE 38th Batch)*

Transition Table:

	0	1
→ q_0	q_1	q_0
q_1	q_1	q_2
*q_2	q_2	q_2

Here,

$Q = \{q_0, q_1, q_2\}$,

$\Sigma = \{0, 1\}$,

$\delta = \{((q_0, 0), q_1), ((q_0, 1), q_0), ((q_1, 0), q_1), ((q_1, 1), q_2), ((q_2, 0), q_2), ((q_2, 1), q_2)\}$,

q_0 and $F = \{q_2\}$.