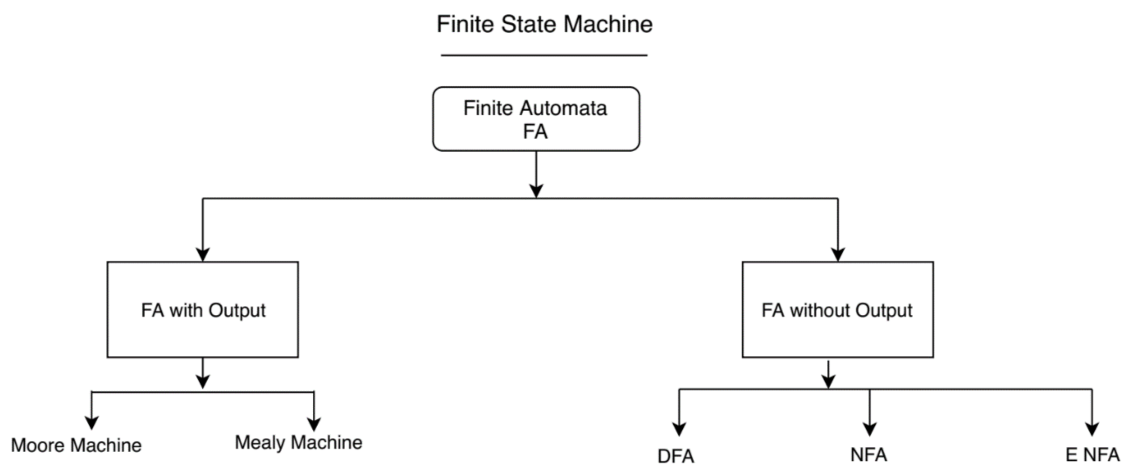
Finite State Machine

Name: Md. Rafat Rahman Tushar
ID: 1621450042
Course: CSE 425
Sec: 03
Faculty: Md. Shahriar Karim (MSK1)

Group Name: ‘INFINITY’

Finite State Machine (FSM):

The term finite state machine (FSM) is also known as finite state automation. FSM is a calculation model that can be executed with the help of hardware otherwise software. This is used for creating sequential logic as well as a few computer programs. FSMs are used to solve the problems in fields like mathematics, games, linguistics, and artificial intelligence. In a system where specific inputs can cause specific changes in state that can be signified with the help of FSMs.



In this paper we will be focusing only on **Deterministic Finite Automata (DFA).**

Definition: A deterministic finite automaton (DFA) consists of

- A finite set of states (Often denoted as S)
- A finite set of symbols (alphabets, I)
- Transition function that takes as argument a state and a symbol and returns a state (often denoted by δ or f)
- A start state often denoted s_0
- a set of final or accepting states (often denoted F)

We have $s_0 \in S$ and F ⊂ S.

So, DFA mathematically represented as 5 tuples (S, I, f, s_0, F).

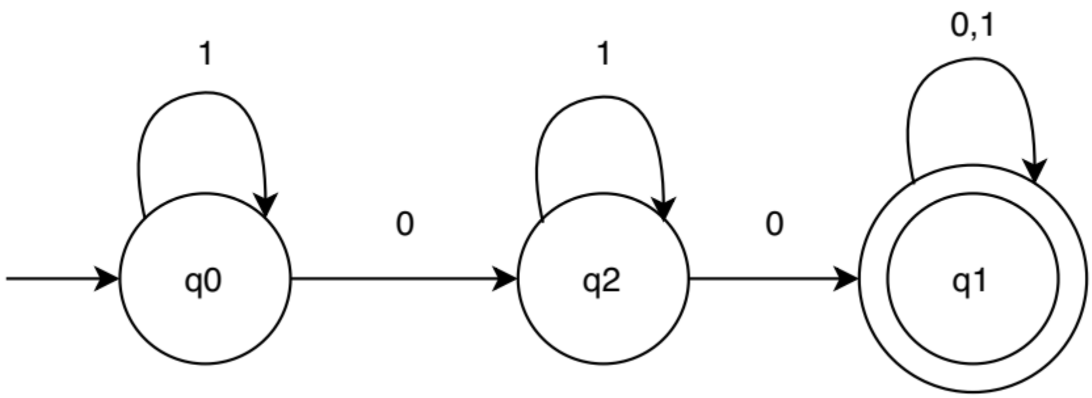
The transition function, f is a function in $S \times I \rightarrow S$. $S \times I$ is a set of two tuples (s, i) with $s \in S$ and $i \in I$.

Remember that, only the strings that reach to the final states are allowed and accepted, all other are rejected. So, we can easily define a FDA with specified requirements of string.

DFA with transition table:

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

The → indicates start state: here q_0
The * indicates final state: here q_1
This defines the following transition diagram,

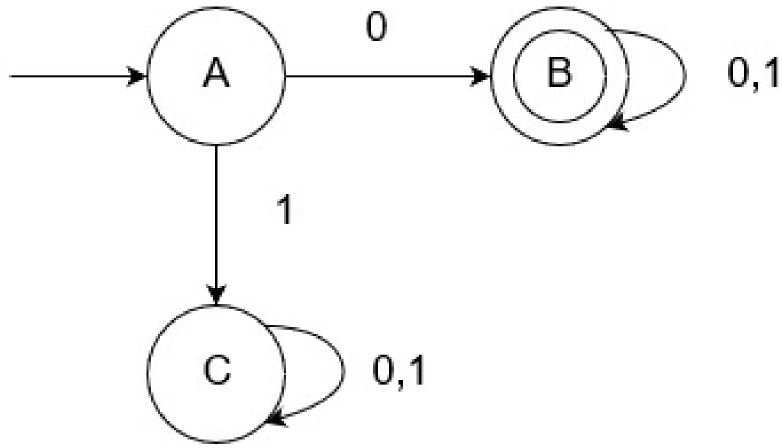


<u>Let it be clear with some simple examples.</u>

Example:
Design a DFA that accepts set of all strings that start with ‘0’ with alphabets (0, 1).

Solution: Set of all strings that start with ‘0’ are, L= {0, 00, 01, 000, 001, 010, 011, 0000, …….}

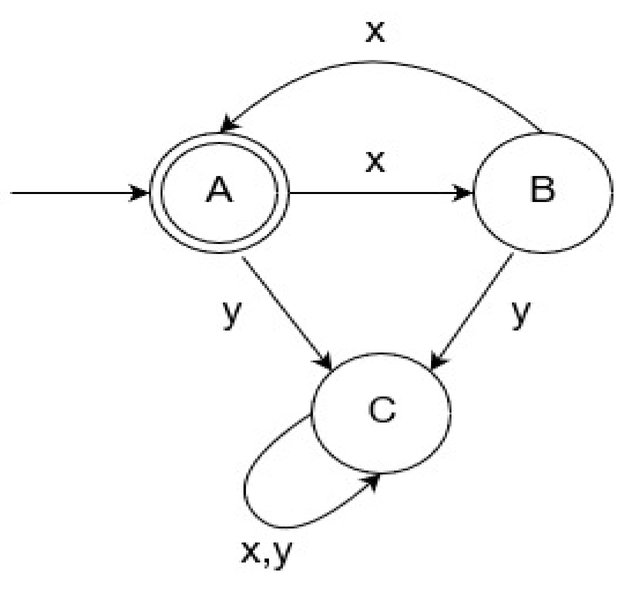
For a string from L (001) we can justify the answer. For 001, at first, start from the start state A. Then follow the alphabet’0’ which leads to the state B (string: 0). Then follow the alphabet ‘0’ in B, which leads to the state B again (string: 00). Then follow the alphabet ‘1’ which again leads to the final state B (string: 001). Since we get the deserved string (001) and since B is the final state, we can conclude our findings with $A \to B \to B$.



Here,
S = {A, B, C}
I = {0, 1}
f : S × I → S
s_0 = Start State ≡ A
F: {B}
Also, here C is the dead state or trap state, since it cannot go to the final state B.

Example:
For an alphabet set, I = {x, y}, draw DFA that recognize the language formed by all strings of 2n (where n = 0, 1, 2, 3, 4…..) numbers of x's. Take an example string from the aforementioned language and recognize it using the DFA you design.

Solution: Set of all strings with given requirements are, L= {Φ, xx, xxxx, xxxxxx……..}



Here,
S = {A, B, C}
I = {x, y}
f : $S \times I \rightarrow S$
s_0 = Start State ≡ A
F: {A}
Also, here C is the dead state or trap state, since it cannot go to the final state B.
From a string from L (xxxx), we can justify the answer. First, start from the starting state A. Then go to the B followed by alphabet x. Then go to the state A followed by alphabet x. Again, go to the state B followed by alphabet x, and then go to string A followed by alphabet x. A is also the finale state.
So the states are $A \rightarrow B \rightarrow A \rightarrow B \rightarrow A$.