THEORY OF COMPUTATION

What is TOC???

- In theoretical computer science, the theory of computation is the branch that deals with whether and how efficiently problems can be solved on a model of computation, using an algorithm.
- The field is divided into three major branches: automata theory, computability theory and computational complexity theory.

- These abstract machines are called automata,
  this automaton consists of
  - states (represented in the figure by circles),
  - and transitions (represented by arrows).
- As the automaton sees a symbol of input, it makes a transition (or jump) to another state, according to its transition function (which takes the current state and the recent symbol as its inputs).

Introduction to formal proof

Basic Symbols used :

U – Union

∩- Conjunction

ϵ - Empty String

Φ – NULL set

7- negation

‘ – compliment

= > implies

- Languages : The languages we consider for our discussion is an abstraction of natural languages. That is, our focus here is on formal languages that need precise and formal definitions. Programming languages belong to this category.
- Symbols : Symbols are indivisible objects or entity that cannot be defined. That is, symbols are the atoms of the world of languages. A symbol is any single object such as , a, 0, 1, #, begin, or do.
- Alphabets : An alphabet is a finite, nonempty set of symbols. The alphabet of a language is normally denoted by . When more than one alphabets are considered for discussion, then subscripts may be used (e.g. etc) or sometimes other symbol like G may also be introduced.

- Strings or Words over Alphabet : A string or word over an alphabet £ is a finite sequence of concatenated symbols of .
- Example : 0110, 11, 001 are three strings over the binary alphabet { 0, 1 } . aab, abcb, b, cc are four strings over the alphabet { a, b, c }.
- Length of a string : The number of symbols in a string w is called its length, denoted by |w|.
- Example : | 011 | = 4, |11| = 2, | b | = 1

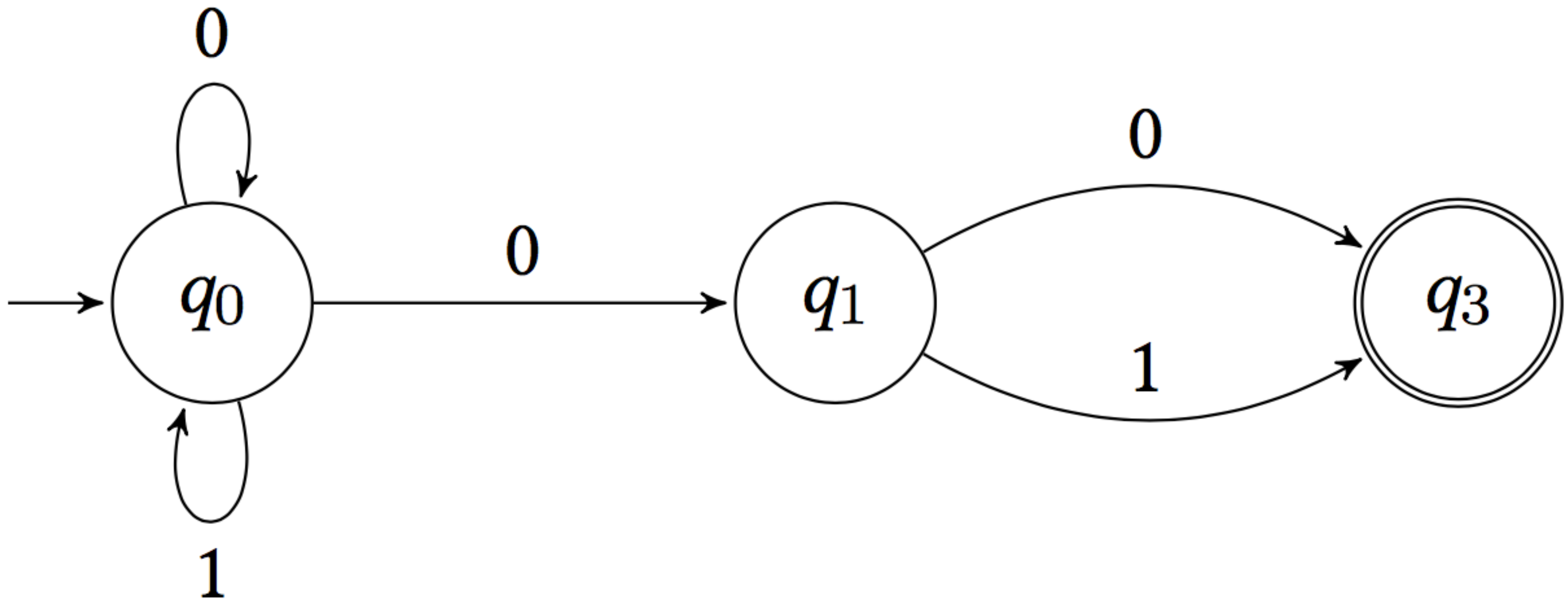
Automata

- An automata is an abstract computing device (or machine). There are different verities of such abstract machines (also called models of computation) which can be defined mathematically.

Finite Automata

- Automata (singular : automation) are a particularly simple, but useful, model of computation. They were initially proposed as a simple model for the behavior of neurons.
- States, Transitions and Finite-State Transition System :
- Informally, a state of a system is an instantaneous description of that system which gives all relevant information necessary to determine how the system can evolve from that point on.
- Transitions are changes of states that can occur spontaneously or in response to inputs to the states. Though transitions usually take time, we assume that state transitions are instantaneous (which is an abstraction).
- Some examples of state transition systems are: digital systems, vending machines, etc. A system containing only a finite number of states and transitions among them is called a finite-state transition system.
- Finite-state transition systems can be modeled abstractly by a mathematical model called finite automation

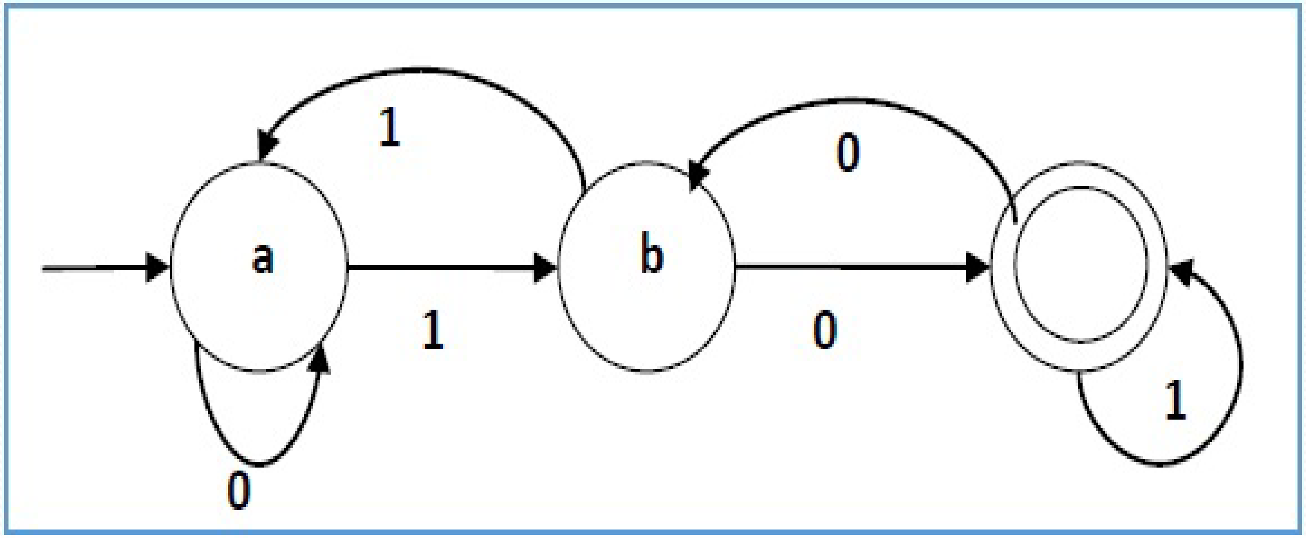
EXAMPLE



Deterministic Finite Automata

- In **DFA**, for each input symbol, one can determine the state to which the machine will move. Hence, it is called Deterministic Automaton.
- As it has a finite number of states, the machine is called Deterministic Finite Machine or Deterministic Finite Automaton.

EXAMPLE



Deterministic Finite Automata

- Q is the finite set of states,
- Σ is the alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ is the state transition function,
- qo is the initial state, and
- $F \subset Q$ is the set of accepting states.

NFA (Non-Deterministic finite automata)

- **NFA** stands for non-deterministic finite automata.
- It is easy to construct an **NFA** than DFA for a given regular language.
- The finite automata are called **NFA** when there exist many paths for specific input from the current state to the next state.

NFA (Non-Deterministic finite automata)

- Q is the finite set of states,
- Σ is the alphabet,
- $\delta : Q \times \Sigma \rightarrow 2^Q$ is the state transition function,
- qo is the initial state, and
- $F \subset Q$ is the set of accepting states.

EXAMPLE

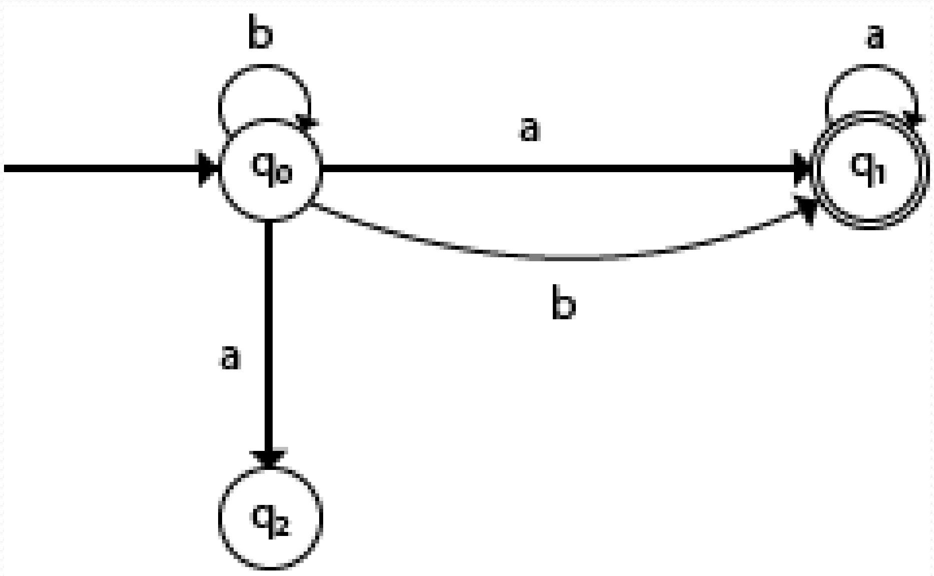


Fig:- NDFA

DIFFERENCE BETWEEN DFA AND NFA

- It stands for deterministic finite automata.
- The transition takes place from a state to a single particular state for each input symbol.
- There are no empty string transitions in DFA.
- DFA requires more space.
- There no such randomness as the transition can be determined in DFA.
- **Ex: $\delta: Q \times \Sigma \rightarrow Q$**
- **Ex: $\delta: Q \times \Sigma \rightarrow 2^Q$**

NDFA

- It stands for non-deterministic finite automata.
- For each input symbol, the transition can be to multiple next states.
- Empty string transitions are also permitted.
- NDFA requires less space.
- There is much more randomness as the transition from one state to other cannot be determined.
- **Ex: $\delta: Q \times \Sigma \rightarrow 2^Q$**