Finite Automata

by

Dr. Pratik Roy

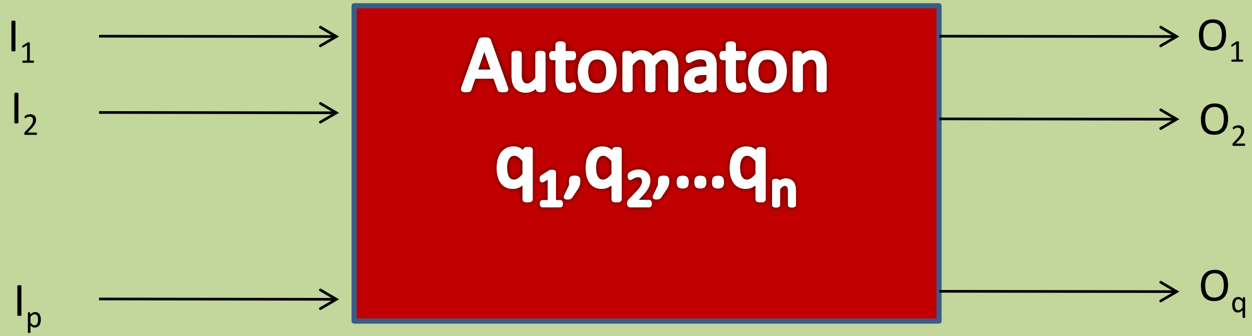
Automata

- An automaton is defined as a system where energy, materials and information are transformed, transmitted and used for performing some functions without direct participation of man.

- An automaton is an abstract model of a digital computer.

- Automaton = an abstract computing device which process discrete information

Model of Discrete Automaton



Input: $I_1, I_2, \ldots\ldots I_p$
Output: $O_1, O_2, \ldots O_q$
States: $q_1, q_2, \ldots q_n$

Why do we need abstract models?

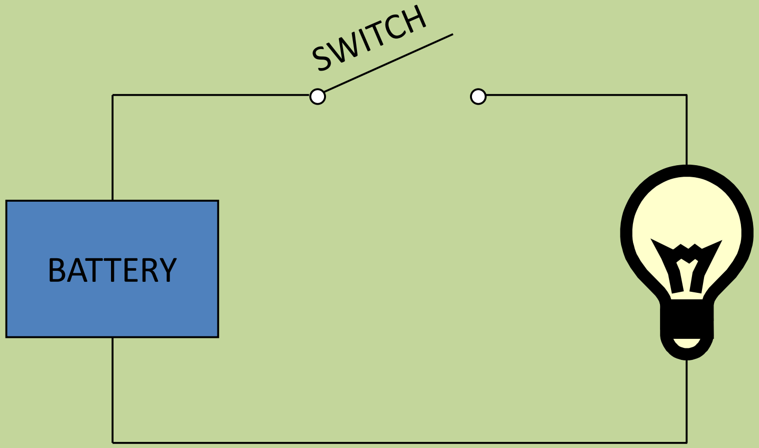
- Abstract model is free from "Programming Language".

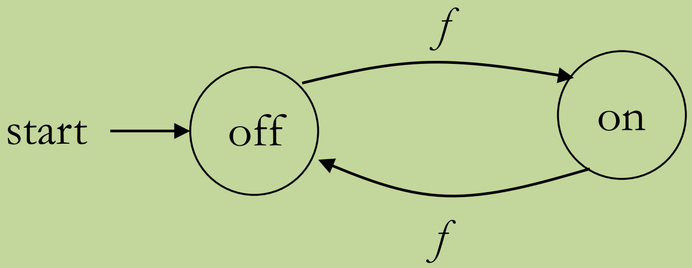
- It's easy to manipulate these theoretical machines mathematically to prove things about their capabilities.

Why Automata Theory?

- To study abstract computing devices or "machines" which are closely related to today's computers.
- Some Applications
  - Software for designing and checking the behavior of digital circuits.
  - "Lexical analyzer" of a typical compiler, i.e., the compiler component that breaks the input text into logical units, such as identifiers, keywords, etc.
  - Software for scanning large bodies of text to find occurrences of words, phrases, or other pattern.

Example





input: switch

output: light bulb

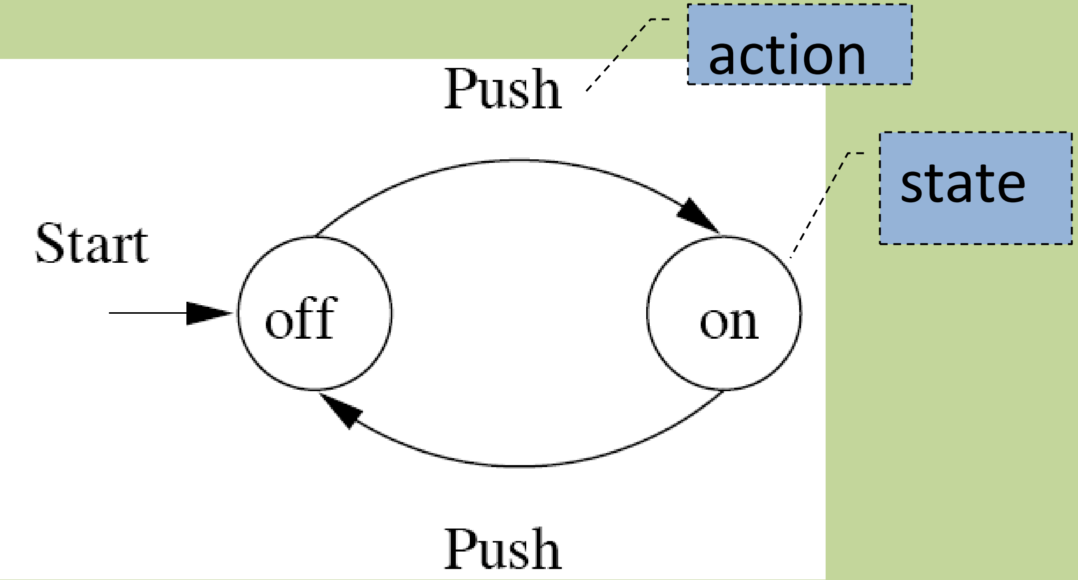
actions: f for "flip switch"

states: on, off

bulb is on if and only if there was an odd number of flips

Example of Finite Automata

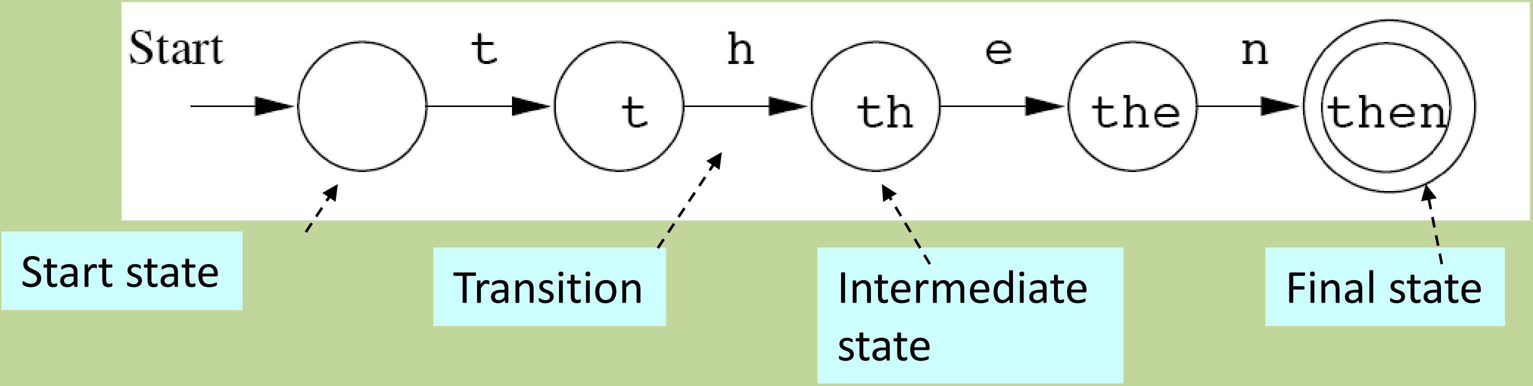
A finite automaton modeling an On/Off switch



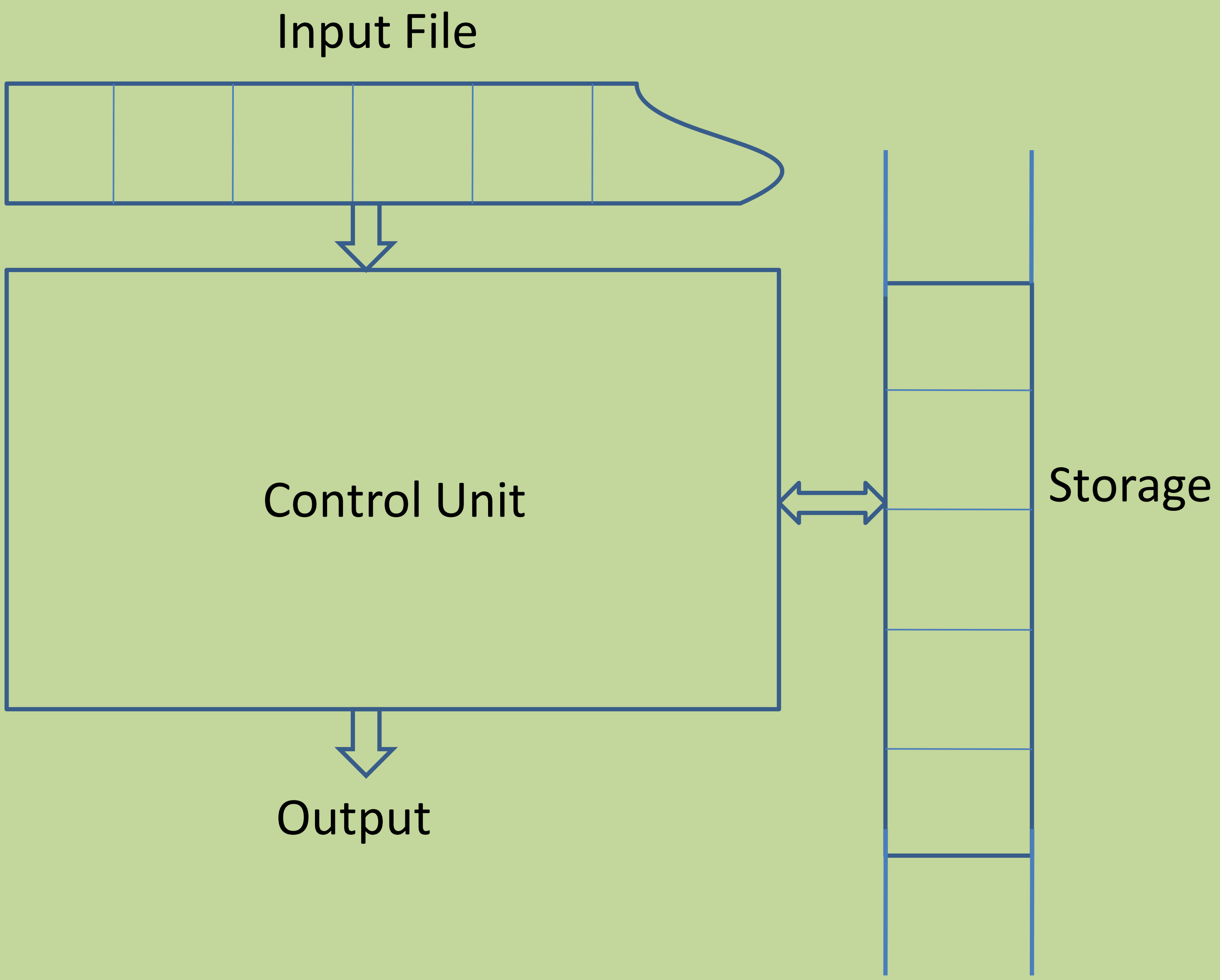
- Simplest finite automaton is an on/off switch.
- The device remembers whether it is in “on” state or “off” state, and allows user to press a button whose effect is different, depending on the state of the switch.
- If the switch is in off state, then pressing the button changes it to the on state, and if the switch is in on state, then pressing the same button turns it to the off state.

Example of Finite Automata

- Modeling recognition of the word "*then*"



Model of a general automaton



Model of a general automaton

- Essential features of an automaton:
  - Every automaton has a mechanism for reading input.
  - Input is a string over a given alphabet, written on an **input file**, which the automaton can read but not change.
  - Input file is divided into cells, each of which can hold one symbol.
  - Input mechanism can read the input file from left to right, one symbol at a time and can also detect the end of the input string.
  - The automaton can produce **output** of some form.

Model of a general automaton

- The automaton may have a temporary **storage** device, consisting of an unlimited number of cells, each capable of holding a single symbol from an alphabet (not necessarily the same one as the input alphabet).
- The automaton can read and change the contents of the storage cells.
- The automaton has a **control unit**, which can be in any one of a finite number of internal **states** at any instant of time, and which can change state in some defined manner.
- An automaton is assumed to operate in a discrete timeframe (t_1, t_2,…, t_n).
- At any given time, the control unit is in some internal state, and the input mechanism is scanning a particular symbol on the input file.

Model of a general automaton

- Internal state of the control unit at the next time step is determined by a **transition function**.
- Transition function gives the next state in terms of the current state, the current input symbol, and the information currently in the temporary storage.
- During the transition from one time interval to the next, output may be produced or the information in the temporary storage changed.
- The term **configuration** is used to refer to a particular state of the control unit, input file, and temporary storage.
- Transition of the automaton from one configuration to the next is called a **move**.

Finite Automata

- Finite automata involves states and transitions among states in response to inputs.
- A finite automaton has a set of states, and its “control” moves from state to state in response to external “inputs”.
- One of the crucial distinctions among classes of finite automata is whether the control is “deterministic”, meaning that the automaton can’t be in more than one state at any one time, or “nondeterministic”, meaning that it may be in several states at once.
- A **deterministic automaton** is one in which each move is uniquely determined by the current configuration. If we know the internal state, the input, and the contents of the temporary storage, we can predict the future behavior of the automaton exactly.

Finite Automata

- In a **nondeterministic automaton**, this is not so.
- At each point, a nondeterministic automaton may have several possible moves, so we can only predict a set of possible actions.
- An automaton whose output response is limited to a simple "yes" or "no" is called an **accepter**. Presented with an input string, an accepter either accepts the string or rejects it.
- A more general automaton, capable of producing strings of symbols as output, is called a **transducer**.