CSE-233 : Week 1

Summer 2020

Introduction to Automata

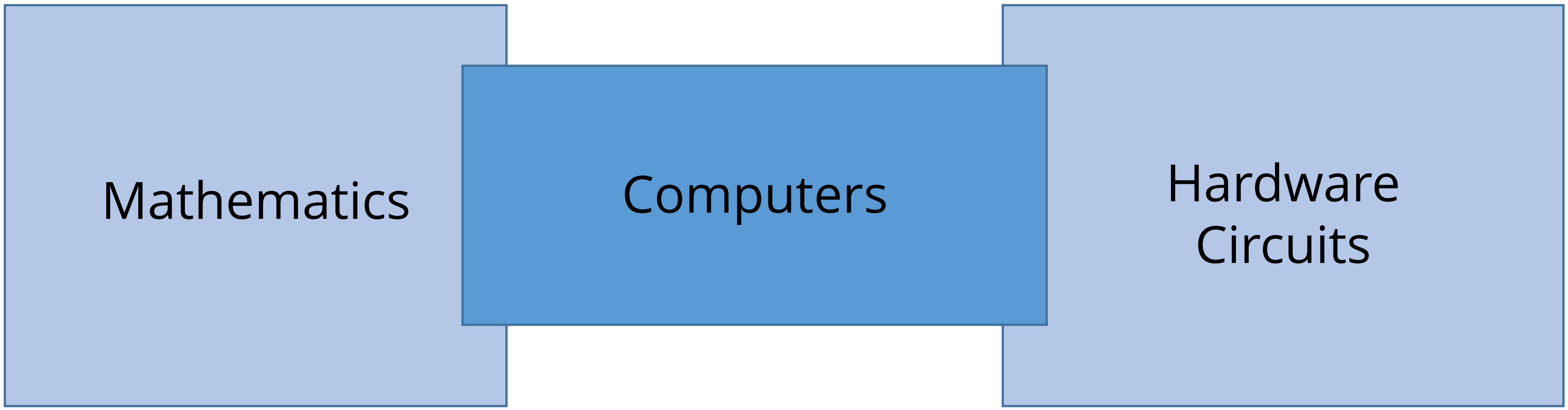
Reference:

Book Chapter 1.5

Md. Saidul Hoque Anik

anik@cse.uiu.ac.bd

Computers



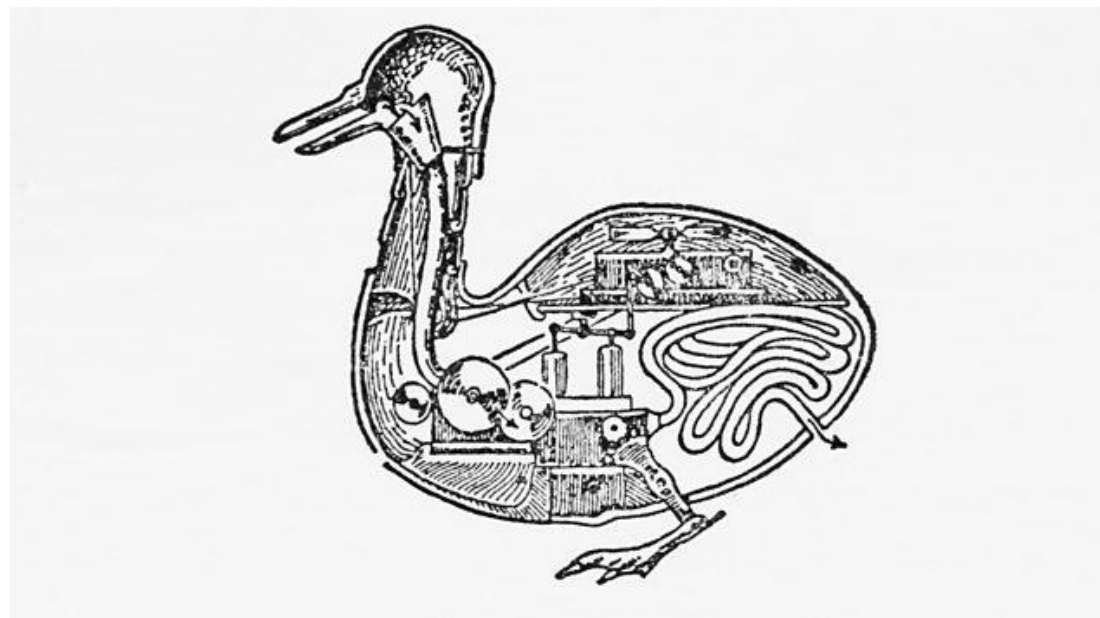
Theory of Computation

Key areas:

1. Complexity Theory
2. Computability Theory
3. Automata Theory

Automata

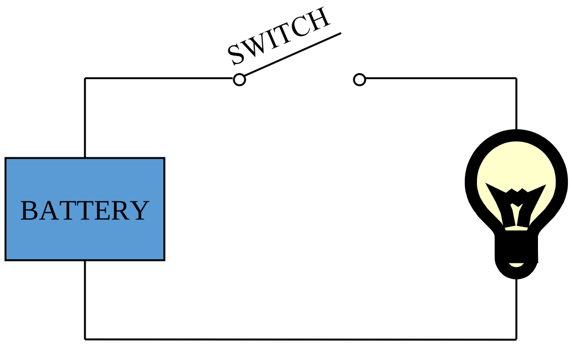
Automata



What is automata theory

- Automata theory is the study of abstract computational devices.
- Abstract devices are (simplified) models of real computational devices like computer.
- Computations happen everywhere: On your laptop, on your cell phone, …

A simple computer



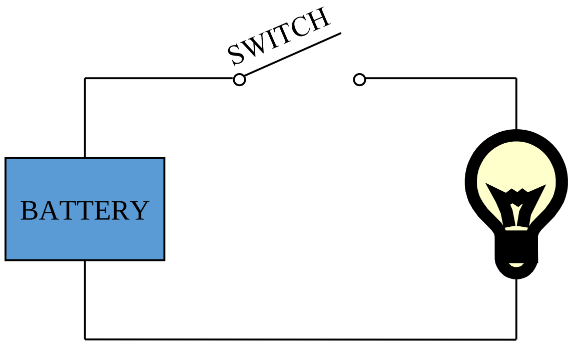
input: switch

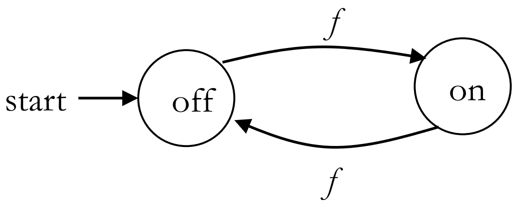
output: light bulb

actions: flip switch

states: on, off

Another simple “computer”





input: switch

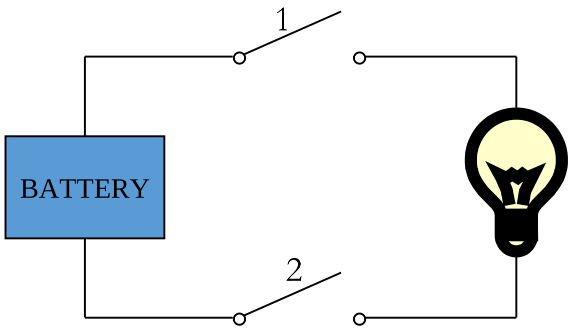
output: light bulb

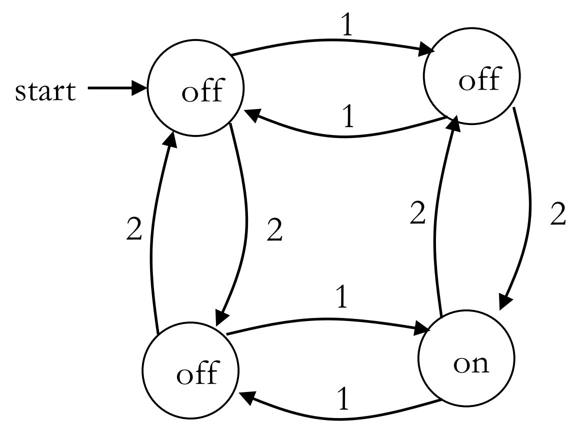
actions: f for “flip switch”

states: on, off

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

Another "computer"





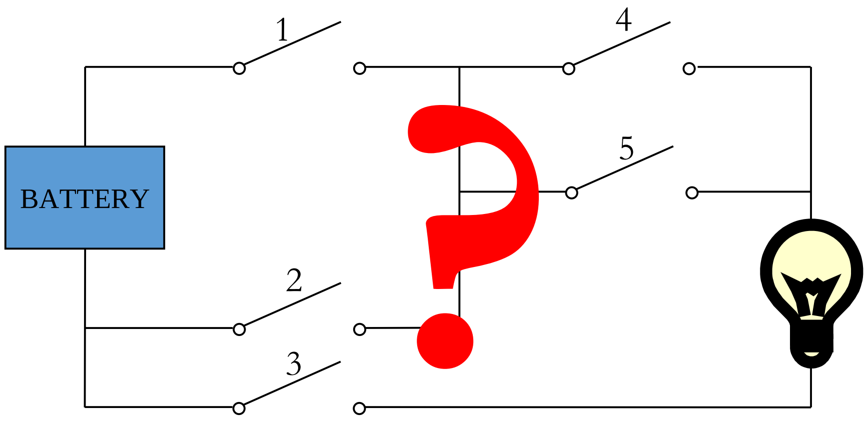
inputs: switches 1 and 2

actions: 1 for "flip switch 1"
2 for "flip switch 2"

states: on, off

bulb is on if and only if both switches were flipped an odd number of times

A design problem



Can you design a circuit where the light is on if and only if all the switches were flipped exactly the same number of times?

A design problem

- Such devices are difficult to reason about, because they can be designed in an infinite number of ways.
- By representing them as abstract computational devices, or automata, we will learn how to answer such questions

These devices can model many things

- They can describe the operation of any "small computer", like the control component of an alarm clock or a microwave.
- They are also used in lexical analyzers to recognize well formed expressions in programming languages:

`ab1` is a legal name of a variable in C
`5u=` is not

Some devices we will see

finite automata	Devices with a finite amount of memory. Used to model "small" computers.
push-down automata	Devices with infinite memory that can be accessed in a restricted way. Used to model parsers, etc.
Turing Machines	Devices with infinite memory. Used to model any computer.
time-bounded Turing Machines	Infinite memory, but bounded running time. Used to model any computer program that runs in a "reasonable" amount of time.

Some highlights of the course

- Finite automata
  - We will understand what kinds of things a device with finite memory can do, and what it cannot do
  - Introduce simulation: the ability of one device to "imitate" another device
  - Introduce nondeterminism: the ability of a device to make arbitrary choices
- Types
  - Deterministic Finite Automata(DFA).
  - Non-Deterministic Finite Automata(NFA).

Preliminaries of automata theory

- How do we formalize the question.
- First, we need a formal way of describing the problems that we are interested in solving.

Problems

- Examples of problems we will consider
  - Given a word s, does it contain the subword “food”?
  - Given a number n, is it divisible by 7?
  - Given a pair of words s and t, are they the same?
  - Given an expression with brackets, e.g. (( ) ( )), does every left bracket match with a subsequent right bracket?
- All of these have “yes/no” answers.
- There are other types of problems, that ask “Find this” or “How many of that” but we won’t look at those.

Alphabets and strings

- A common way to talk about words, number, pairs of words, etc. is by representing them as strings
- To define strings, we start with an alphabet

An alphabet is a finite set of symbols.

- Examples

$\Sigma_1 = \{a, b, c, d, \ldots, z\}$: the set of letters in English

$\Sigma_2 = \{0, 1, \ldots, 9\}$: the set of (base 10) digits

$\Sigma_3 = \{a, b, \ldots, z, \#\}$: the set of letters plus the special symbol $\#$

$\Sigma_4 = \{\ (,\ )\ \}$: the set of open and closed brackets

Strings

A string over alphabet Σ is a finite sequence of symbols in Σ.

- Σ^* denotes this set of strings.
- The empty string will be denoted by ε
- Examples

abfbz is a string over $\Sigma_1 = \{a, b, c, d, \ldots, z\}$

9021 is a string over $\Sigma_2 = \{0, 1, \ldots, 9\}$

ab#bc is a string over $\Sigma_3 = \{a, b, \ldots, z, \#\}$

))()(() is a string over $\Sigma_4 = \{ (,) \}$

Strings

- $\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, 001, . . . \}$
- Length of a string
  - Number of symbols in a string
  - $|s|$
  - $|0110| = 4$
  - $| \varepsilon | = 0$

Powers of an Alphabet

- $\Sigma = \{0,1\}$
- Σ^k = the set of strings of length k, each of whose symbols is in Σ.

Powers of an Alphabet

If Σ is an alphabet, we can express the set of all strings of a certain length from that alphabet by using the exponential notation:

- Σ^k: the set of strings of length k, each of whose is in Σ
- Examples:
  - Σ^0 : $\{\epsilon\}$, regardless of what alphabet Σ is. That is ϵ is the only string of length 0
  - If $\Sigma = \{0, 1\}$, then:

    1. $\Sigma^1 = \{0, 1\}$
    2. $\Sigma^2 = \{00, 01, 10, 11\}$
    3. $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

    Note: confusion between Σ and Σ^1:

    1. Σ is an alphabet; its members 0 and 1 are symbols
    2. Σ^1 is a set of strings; its members are strings (each one of length 1)

Kleene Star

- Σ^*: The set of all strings over an alphabet Σ
  - $\{0,1\}^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \ldots\}$
  - $\Sigma^* = \Sigma^0 \cup \Sigma^1 \cup \Sigma^2 \cup \ldots$
- The symbol $*$ is called **Kleene star** and is named after the mathematician and logician Stephen Cole Kleene.
- $\Sigma^+ = \Sigma^1 \cup \Sigma^2 \cup \ldots$

  Thus: $\Sigma^* = \Sigma^+ \cup \{\epsilon\}$

Concatenation

- Define the binary operation . called **concatenation** on Σ^* as follows:
  If $a_1a_2a_3\dots a_n$ and $b_1b_2\dots b_m$ are in Σ^*, then

$$a_1a_2a_3\dots a_n.b_1b_2\dots b_m = a_1a_2a_3\dots a_nb_1b_2\dots b_m$$

- Thus, strings can be concatenated yielding another string:
  If x are y be strings then $x.y$ denotes the concatenation of x and y, that is, the string formed by making a copy of x and following it by a copy of y
- Examples:
  1. $x = 01101$ and $y = 110$
     Then $xy = 01101110$ and $yx = 11001101$
  2. For any string w, the equations $\epsilon w = w\epsilon = w$ hold.
     That is, ϵ is the **identity for concatenation** (when concatenated with any string it yields the other string as a result)
- If S and T are subsets of Σ^*, then

$$S.T = \{s.t \mid s \in S, t \in T\}$$