Theory of Computation

Mohammed Fahad

July 11, 2025

1 Why TOC?

- It helps us to understand the limits of what computer can do and how to model computation using mathematics

Q1: What is the motivation for studying theory behind computation? OR Needs of TOC?
A:

- Understanding the capability of a computer
- To find steps to solve a problem
- Increase efficiency while doing a task

Q2: List the problems that cannot be solved by a computer.
A:

1. Ethical problems. Eg: Self-driving car deciding to save the driver/passenger or the pedestrian
2. Generating truly original art of emotion

Automaton (pl.: Automata)

A simplified mathematical model of a machine (digital computer). It

- Accepts input
- Produces output
- Nay have some temporary storage
- can make descisions in transforming the input into the output

Q1: Why study computability and theory?
A:

- It helps to answer: "Can this be solved by a computer?"

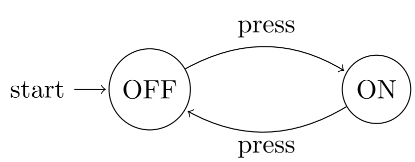
- Understand the principles behind algorithms and programs
- Explore the boundary between what is possible and impossible in computing

1.1 Need for mathematical modelling

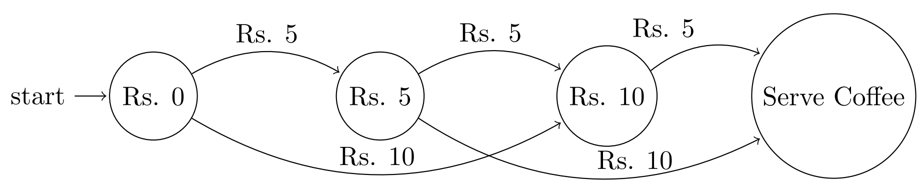
- Computers work on rules & logic
- We can represent computers using abstract models
- These models helps us study complex behaviour in a simplified way

2 Introduction to finite automata

1. ON/OFF Switch:



2. Coffee vending machine (Inputs: Rs. 5 & Rs. 10 — Rs. 15 for one coffee):



3 Formal Language

A formal language is an abstraction of the general characteristics of programming langauges.
A formal language consists of set of symbols and some rules of formation by which these symbols can be combined into entities called sentences.

4 Central concepts of automata theory

4.1 Alphabets

It a finite, non-empty set of symbols. Alphabets are represented by 'Σ'.

Binary alphabets can be represented as:

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

Set of lowercase letters:

$$\Sigma = \{a, b, c, ..., z\}$$

4.2 Strings

A string is a finite seeunce of symbols chosen from some alphabets.
Eg: Let $\Sigma = \{0, 1\}$ be the alphabet.

Examples of strings in Σ:

$$0,\ 1,\ 00,\ 01,\ 10,\ 11,\ 000,\ 010,\ \ldots$$

4.2.1 Length of a string

The number of occurances of symbols in the string.

$$\begin{aligned} \text{Length one:} \quad & 0,\ 1 \\ \text{Length two:} \quad & 00,\ 01,\ 10,\ 11 \end{aligned}$$

The std. notation for length of a string w is $\|w\|$

4.2.2 Empty string (ε)

A string with zero occurances OR string with length '0'

4.2.3 Σ^*

Set of all strings over an alphabet

4.2.4 Σ^+

Set of all strings excluding empty string over an alphabet

$$\Sigma^* = \Sigma^0 \cup \Sigma^+$$

4.3 Powers of an alphabet

If Σ is an alphabet, Σ^k is the set of strings with length 'k', each of whose symbols is in 'Σ'.

$$\begin{aligned} &\Sigma^3 = \{000,\ 001,\ 010,\ 011,\ 100,\ 101,\ 110,\ 111\} \\ &\Sigma^0 = \{\varepsilon\} \end{aligned}$$

NOTE: $\Sigma \neq \Sigma^1$, Their definitions differentiates them.

5 Concatenation of strings

Let x and y be strings, then xy means combining both x and y.

$$\text{i.e., if } x = 01010 \text{ and } y = 110,$$
$$xy = 01010110$$

$$|x| = m \text{ and } |y| = n,$$
$$|xy| = m + n$$

For any string w, then the equation,

$$\varepsilon w = w\varepsilon = w$$

ie, ε is the identity of Concatenation

6 Languages

A set of strings all of which are chosen from Σ^*. If Σ is an alphabet, then $L \subset \Sigma^*$. Eg:

1. Set of all strings consisting of n 0s followed by n 1s, $n \geq 0$

$$L = \{\varepsilon, 01, 0011, 000111, ...\}$$

2. Set of all strings having equal number of 0s and 1s,

$$L = \{\varepsilon, 01, 0011, 0101, ...\}$$

NB: Σ^* is a language for any alphabet, Σ

$$L = \{\} \rightarrow \text{Empty language, } \varnothing$$
$$L = \{\varepsilon\} \rightarrow \text{Language containing the empty string}$$

There are 2 types of Languages:

1. **Infinte languages**. Eg: $\{0, 01, 001, 0001, 00001, ...\}$
2. **Finte langauges**. Eg: $\{\varepsilon, a, b, ab, ba\}$

7 Set-formers as a way to define language

$$\{w|w \text{ consists of equal number of 0s and 1st}\}$$

8 Problems

Problem is the question of deciding whether a given string is a member of some particular language.

9 Automata

An automaton is an abstract model of a digital computer.

9.1 Key components of automata

1. Input file
2. Control Ubit
3. storage
4. Output

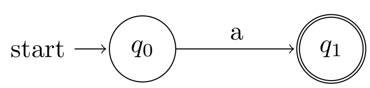
9.2 Types of automata

1. Deterministic (DFA): One move per configuration (Predictable)
2. Nondeterministic (NFA): Multiple possible moves

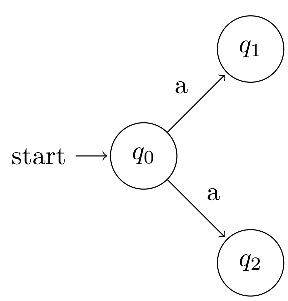
Then there is automata like:

- **Acceptor:** Says “yes” or “no” for an input.

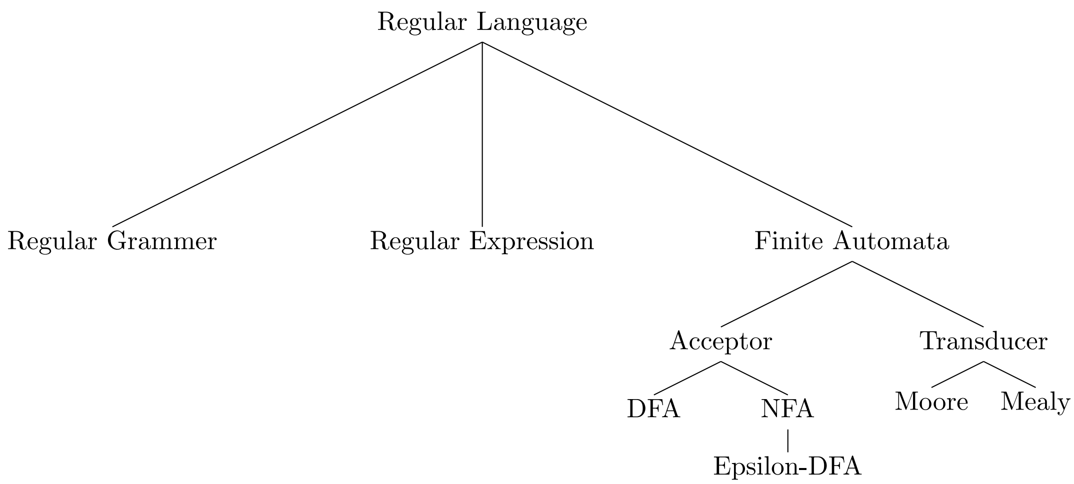
  This simple automaton accepts the string ‘a‘:



- **Transducer:** Produces an output string based on input.



9.3 Automata Overview



The mathematical representation of Regular Language (RL) is callled Finite Automata (FA)

10 Deterministic Finite Automata (DFA)

DFA is defined by a quintuple:

$$M = (Q, \Sigma, \delta, q_0, F)$$

Q - Finite set of internal states
Σ - Input alphabets
δ - Transition function, $\delta : Q\ \Sigma \rightarrow Q$
q_0 - initial state [$q_0 \varepsilon Q$]
F - Final states [$F \subset Q$]