Chapter 1

Languages

While we may think that computers are just machines, there is an incredible work behind that aims to make some kind of language that computers can understand. It is well renewed that computers act depending on streams of binary code, but that same stream of binary code can be considered as a language, something that the computers "speak" and "think" with. Let's define more in detail this "computers' language".

DEFINITION

Alphabeth and Strings

An **alphabet** Σ is a **non-empty**, **finite set**, which contains elements called **symbols** (or **characters**). For instance, the following two sets are considered alphabets:

- $\Sigma = \{0, 1\}$
- $\Sigma = \{a, b, c, \ldots, x, y, z\}$

A **string** w over an alphabet Σ is a **sequence of symbols**, all belonging to Σ, which are all written one after the other and aren't separated by other symbols

Strings also have different properties, such as a **length**, a **reverse** and the possibility to include one or more **substrings**:

- **Length**: defined as $|w|$, it denotes the **number of symbols** contained within w. If a string has length 0, then such string is called **empty string**, and is denoted with ϵ;
- **Reverse**: defined as w^R, the reverse is a string which contains all the symbols of w in the reverse order;
- **Substring**: we say that a string z is a substring of w if z appears consecutively within w.

EXAMPLE

1.0.0

For instance, the string $w = 00101$ has a length of 5, its reverse is $w^R = 10100$, and has, as possible substrings the following: $001, 10, \epsilon$. Its alphabet is $\Sigma = \{0, 1\}$

When we have two strings x and y, we can **concatenate** them by appending y at the end of x. The result is denoted as xy. Clearly, the length of the concatenated string is equal to the sum of the length of the two strings ($|xy| = |x| + |y|$). A specific notation x^k denotes the concatenation of string x with itself for k times. So, if x was for instance "01", then we would have that

$$x^0 = \epsilon \qquad x^1 = 01 \qquad x^2 = 0101 \qquad x^3 = 010101 \quad \text{and so on...}$$

By combining the definitions of concatenation and substring, we can define properly what a **prefix** and **suffix** is:

- **Prefix**: we say that a string x is a prefix of a string z if there exists a string y such that $z = xy$. Moreover, we say that x is a **proper prefix** of z if, additionally, $x \neq z$, so if $y \neq \epsilon$;

- **Suffix**: we say that a string x is a suffix of a string z if there exists a string y such that $z = yx$. Moreover, we say that x is a **proper suffix** of z if, additionally, $x \neq z$, so if $y \neq \epsilon$.

Given these tools, we can now define what a language is:

DEFINITION

Language

A **language** is a **set of strings**. It is also defined as **prefix-free** if no member is a proper prefix of any other member

Languages can follow an order. There are different types of orders, one of which is the **lexicographic order**, which is defined by the order of the alphabet; we can think of it as the familiar dictionary order. Another order is the **shortlex order** (or **string order**), which is the same of the lexicographic order with the exception that shorter strings precede longer strings.

1.1 Automaton

We know that sometimes circuits can use the output from a given combination of inputs as input for the next output. In that case, we talk about circuits with states, or more specifically, we talk about **automata**.

DEFINITION

Automaton

We define a **finite automaton** as a tuple of 5 elements $(Q, \Sigma, \delta, q_0, F)$, where:

- Q is a finite set, denoting the **states** that the automaton can reach;
- Σ is a finite set, denoting the **alphabet** of the automaton;
- δ is a function, called **transition function**, which given Q and Σ returns the set of next states Q';
$$\delta : Q \times \Sigma \longmapsto Q'$$
- $q_0 \in Q$ is a state, and denotes the **starting state** of the automaton;
- $F \subseteq Q$ is a finite set, denoting the set of **accepted states** (or **final states**). An accepted state tells the automata if a string can be accepted or not.

Let's make an example:

EXAMPLE

1.1.1

Consider the following tuple:
$$(\{q_1, q_2, q_3\}, \{0, 1\}, \delta, q_1, q_2)$$
It denotes that we have 3 possible states $\{q_1, q_2, q_3\}$, the alphabet is made of 2 symbols, $\{0, 1\}$, the starting state is q_1, the final state is q_2 and the transition function is δ. Such transition function could be the following:

	0	1
q_1	q_1	q_2
q_2	q_3	q_2
q_3	q_2	q_2

We can represent it visually as follows: