Foundations of Computer Science
Logic, models, and computations

Chapter: Introduction

Course CSC_41012_EP

of l'Ecole Polytechnique

Olivier Bournez

bournez@lix.polytechnique.fr

Version of August 20, 2024

Introduction

Objectives

This course is about algorithms and their efficiency.

More precisely, the objective of this course is to answer to the following questions: What are the limits of algorithms, and of today's computers?

Algorithm?

The word "algorithm" comes from the name of mathematician Al-Khwârizmî (Latinised at middle age as Algorithmi), who at 9th century wrote several books on the resolution of equations. We will discuss the notion of algorithm, and the notion of problem solvable by an algorithm, or of function computed by some algorithm.

We will first prove that there are problems that cannot be solved by an algorithm.

Out of the problems that admit a solution by an algorithm, we will then try to determine those which admit a solution with reasonable resources: We will discuss the resources (time, memory, etc) necessary to solve a problem.

Thanks The author of this document would like to thank strongly Stefan Mengel for many comments on previous versions of this document. I also would like to thanks warmly the students of the École Polytechnique Bachelor course CSE-304 for the year 2019-2020 for comments and feedback. Some special thanks to Louis de Benoist De Gentissart, Agathe De Vulpian, Guillaume Lainé and Skander Moalla for some detailed feedback, or bugs about previous versions of some of the chapters of this document, or about related slides.

Some parts of this document are very strongly inspired from a French version, that has been used for the course INF423, and then INF412 at École Polytechnique. The author of this document would like to thank strongly Johanne Cohen, Bruno Salvy and David Monniaux for their comments on preliminary versions of this latter document in French. I also thank the promotions 2011-2012, 2012-2013, 2013-2014, 2014-2015, 2015-2016, 2016-2017, 2017-2018, 2018-2019, 2019-2020, 2020-2021, 2021-2022, 2022-2023 of École Polytechnique for their feedback on INF423 and then INF412. Some special thanks to Louis Abraham, Sariah Al Saati, Olivier Bailleux, Juliette Buet, Ismaël Cahu, Carlo Ferrari, Léo Gaspard, Estienne Granet, Pierre-Jean Grenier, Roberto

Moura, Alexis Le Dantec, Denis Langevin, Emmanuel Lazard, Stéphane Lengrand, Arnaud Lenoir, Louis-François Rigano, Louis Rustenholz, Matthieu Vermeil, and Zigfrid Zvezdin, for some detailed feedback, with precise suggestions of improvement, or for having pointed out some problems about preliminary versions of previous French versions of some parts of this document. Thanks also to Romain Cosson and Rodrigue Lelotte for feedback on corrections of previous exams for INF423 and INF412.

This document is still in some non-perfect form.

All comments (even language, typographic, orthographic, etc) on this document are welcome and should be be sent to bournez@lix.polytechnique.fr.

On the exercises Some of the exercises are corrected. The solutions are found and the end of the document in a chapter devoted to the solutions. The exercises marked with a star require more thought.

1 Mathematical concepts

1.1 Sets, Functions

Let E be a set and e an element. We write $e \in E$ to mean that e is an element of set E. If A and B are two sets, we write $A \subset B$ to mean that every element of A is an element of B. We say in that case that A is a subset of B. When E is a set, the collection of all the subsets of E constitutes a set, called the *power set of* E, that we will denote by $\mathscr{P}(E)$. We will write $A \cup B$, $A \cap B$ for respectively the *union* and *intersection* of the sets A and B. When A is a subset of E, we will write A^c for the *complement* of A in E.

Exercise 1 *Let A, B be two subsets of E. Prove the Morgan laws: $(A \cup B)^c = A^c \cap B^c$ and $(A \cap B)^c = A^c \cup B^c$.*

Exercise 2 *Let A, B, C three subsets of E. Prove that $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$ and $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$.*

Exercise 3 *(solution on page 203) Let A, B, C three subsets of E. Prove that $A \cap B^c = A \cap C^c$ if and only if $A \cap B = A \cap C$.*

We call *Cartesian product* of two sets E and F, denoted by $E \times F$, the set of all the pairs made of an element of E and of an element of F:

$$E \times F = \{(x, y) | x \in E \text{ and } y \in F\}.$$

Given some integer $n \geq 1$, we write $E^n = E \times \cdots \times E$ for the Cartesian product of E by itself n times: E^n can also be defined[1] recursively by $E^1 = E$, and $E^{n+1} = E \times E^n$.

Intuitively, a *application* f from a set E to a set V is an object which associates to every element e of a set E a unique element $f(e)$ in F. Formally, a function f (one also talks of *partial function*) from a set E to a set F is a subset Γ of $E \times F$, such that for all $x \in E$ there is at most one $y \in F$ with $(x, y) \in \Gamma$. Its *domain* is the set of the $x \in E$ such that $(x, y) \in \Gamma$ for a certain $y \in F$. Its *image* is the set of the $y \in F$ such that $(x, y) \in \Gamma$ for a certain $x \in E$. An *application* f (this is also called a *total function*) from a set E to a set F is a function whose domain is E.

A *family* $(x_i)_{i \in I}$ *of elements of a set* X is some application from a set I to X. I is called the *set of indices* and the image by its application of element $i \in I$ is denoted x_i.

The Cartesian product generalizes to a family of sets:

$$E_1 \times \cdots \times E_n = \{(x_1, \ldots, x_n) | x_1 \in E_1, \cdots, x_n \in E_n\}.$$

The union and intersection generalize to some arbitrary family of subsets of a set E. Let $(A_i)_{i \in I}$ be a family of subsets of E.

$$\bigcup_{i \in I} A_i = \{e \in E | \exists i \in I \; e \in A_i\};$$
$$\bigcap_{i \in I} A_i = \{e \in E | \forall i \in I \; e \in A_i\}.$$

Exercise 4 *Let A be a subset of E, and $(B_i)_{i \in I}$ a family of subsets of E. Prove the two following equalities:*

$$A \bigcup \left(\bigcap_{i \in I} B_i \right) = \bigcap_{i \in I} (A \cup B_i)$$

$$A \bigcap \left(\bigcup_{i \in I} B_i \right) = \bigcup_{i \in I} (A \cap B_i)$$

We will write $\mathbb{N}$ for the set of natural integers, $\mathbb{Z}$ for the set of (positive, null, or negative) integers, $\mathbb{R}$ for the set of reals, and $\mathbb{C}$ for the set of complex numbers. $\mathbb{Z}$ is a *ring*. $\mathbb{R}$ and $\mathbb{C}$ are *fields*. We will write $\mathbb{R}^{>0}$ for the set of non-negative reals.

1.2 Alphabets, Words, Languages

We now recall some basic definitions about *words* and *languages*. The terminology, borrowed from linguistics, remind that historically first works on the concepts of formal languages were on the modeling of natural language.

A finite set Σ is fixed: In this context, such a set is also called an *alphabet*. and the elements of Σ are called *letters* or *symbols*.

[1] There is a bijection between the objects defined by the two definitions

Example 1

- $\Sigma_{bin} = \{0,1\}$ *is the* binary alphabet.
- $\Sigma_{Latin} = \{A, B, C, D, \ldots, Z, a, b, c, d, \ldots, z\}$ *is the alphabet which consists of the letters of the Latin alphabet.*
- $\Sigma_{number} = \{0, 1, 2, \cdots, 9\}$ *is the alphabet which consists of digits in radix* 10.
- *The set of the printable ASCII*[a] *characters, or set of printed characters is an alphabet, that one can write* Σ_{ASCII}.
- $\Sigma_{exp} = \{0, 1, 2, \cdots, 9, +, -, *, /, (,)\}$ *is the alphabet of arithmetic expressions.*

[a] We will not go here to discussions about whether this is precisely what is called the ASCII characters in all generality. We assume Σ_{ASCII} is the set of symbols that can be printed with a keyboard of a computer. It contains symbols such as é, ö, etc. Actually the original 7-bit version of ASCII did it fact not contain accents and those were added later on and there were lots of different incompatible version for different languages, and we do not intend in this document to go to these discussions: For us, it contains symbols that can be printed with a keyboard of a computer.

A *word* w on alphabet Σ is a finite sequence $w_1 w_2 \cdots w_n$ of letters (i.e. elements) of the alphabet Σ. The integer n is called the *length* of word w. It will be denoted length(w).

Example 2

- 10011 *is a word on alphabet* Σ_{bin} *of length* 5.
- 9120 *is a word on alphabet* Σ_{number}*,but not a word on the alphabet* Σ_{bin}.
- $Bonjour$ *is a word of length* 6 *on alphabet* Σ_{Latin}*;* $azrddfb$ *is also a word of length* 7 *on the same alphabet.* $;-)$ *is not a word on this alphabet, since the symbol* $;$ *is not in the alphabet* Σ_{Latin} *defined above.*
- $Student$*,* $Elephant$ *and* $££z'$!!!" *are words on the alphabet* Σ_{ASCII}.
- $243 + (5 * (1 + 6))$ *is a word on alphabet* Σ_{exp}.
- $24 * ((((5/+)//+$ *is a word on alphabet* Σ_{exp}.

A *language* on alphabet Σ is a set of words on alphabet Σ. The set of all the words on alphabet Σ is denoted by Σ^*. The empty word ϵ is the unique word of length 0. The empty word is a particular word: Similarly to what happens for any other word, It is possible that a language contains the empty word (which is a particular word), or that a language doesn't contain the empty word. Σ^* contains by definition the empty word.

Example 3

- $\{0,1\}^*$ *denotes the set of words over alphabet* $\Sigma_{bin} = \{0,1\}$*. For example,* $00001101 \in \{0,1\}^*$*. We have also* $\epsilon \in \{0,1\}^*$.
- $\{hello, goodbye\}$ *is a language on* Σ_{Latin}*. This language contains two words.*

- *The set of words of English dictionary is a language on the alphabet* Σ_{Latin}.
- *The set of words of French dictionary is a language on the alphabet* Σ_{ASCII}, *since a word such as élève can be written using accentuated letters.*
- *The set of the phrases of this document is a language on the alphabet of ASCII characters. Note that the character " ", that is to say the blank (space) character, used to separate the words in a sentence is a particular character of ASCII alphabet.*
- ${\Sigma_{exp}}^*$ *contains words such as* $24 * ((((5/+)//+$ *which is not a valid arithmetic expression. The set of words which are valid arithmetic expressions, such as* $5 + (2 * (1 - 3) * 3)$, *is a particular language on alphabet* Σ_{exp}.

One then defines an operation of *concatenation* on words: The concatenation of word $u = u_1 u_2 \cdots u_n$ and of word $v = v_1 v_2 \cdots v_m$ is the word denoted by $u.v$ defined by

$$u_1 u_2 \cdots u_n v_1 v_2 \cdots v_m,$$

that is to say the words whose letters are obtained by appending the letters of v after those of u. The operation of concatenation denoted by . is associative, but not commutative. The empty word is a right and left neutral element for this operation. Σ^* is also called the free *monoid* on alphabet Σ (since the operation of concatenation provides a structure of monoid.

We will also write uv for the concatenation $u.v$. Actually, every word $w_1 w_2 \cdots w_n$ can be seen as $w_1.w_2 \cdots .w_n$, where w_i represents the word of length 1 consisting only of the letter w_i. This interpretation of letters as words of length 1 is often very useful.

Example 4 *If* Σ *is the set* $\{a, b\}$, *then* $aaab$ *is the word of length* 4 *whose first three letters are* a, *and the last is* b. *It is also the concatenation of four words of length one:* a, a, a *and* b.

When i is some integer, and w is a word, we write w^i for the word obtained by concatenating the word w i times: If you prefer, w^0 is the empty word ϵ, and $w^{i+1} = w^i w = w w^i$ for every integer i.

Example 5 *By interpreting letters as words of length* 1, $aaabbc$ *can also be written* $a^3 b^2 c$.

A word u is some *prefix* of a word w, if there exists a word z such that $w = u.z$. This is a *proper!prefix* if $u \neq w$. A word u is a *suffix* of a word w if there exists some word z such that $w = z.u$. This is a *proper!suffix* if $u \neq w$.

1.3 Change of alphabet

It is often useful to rewrite a word on a given alphabet into a word on some other alphabet. For example, in computer science one often needs to code in binary, that is to say with alphabet $\Sigma = \{0, 1\}$.

One way to change the alphabet is to proceed one letter after the other.

Example 6 *If Σ is the alphabet $\{a,b,c\}$, and $\Gamma = \{0,1\}$, then one can encode Σ^* in Γ^* by the function h such hat $h(a) = 01$, $h(b) = 10$, $h(c) = 11$. The word $abab$ is then encoded by $h(abab) = 01100110$, that is to say by the word encoded by coding letter after letter.*

Very formally, given two alphabets Σ and Γ, an *homomorphism* is an application from Σ^* into Γ^* such that

- $h(\epsilon) = \epsilon$
- $h(u.v) = h(u).h(v)$ for every words u and v.

Obviously, every homomorphism is perfectly determined by its image on the letters of Σ. It then extends to words of Σ^* by

$$h(w_1 w_2 \cdots w_n) = h(w_1).h(w_2).\ldots.h(w_n)$$

for every word $w = w_1 w_2 \cdots w_n$.

1.4 Graphs

A *graph* $G = (V,E)$ consists of a set V, whose elements are called *vertices* and a subset of $E \subset V \times V$, whose elements are called *arcs*. In some books, vertices are called *nodes*.

If the arcs are undirected, that is say, if one assumes that every time that there is the arc (u,v) there is also the arc (v,u), one says that the graph G is *undirected* and the elements of E are called *edges*.

By default, all considered graphs will all be undirected. An edge will then be denoted uv or $\{uv\}$.

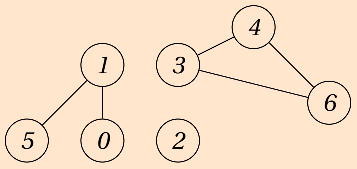
When there is an edge between u and v, that is to say when $\{u,v\} \in E$, one says that u and v are neighbours. The *degree of a vertex u* is the number of its neighbours.

A *path* from s to t is a sequence $(s = s_0, \ldots, s_n = t)$ of vertices such that, for all $1 \le i \le n$, (s_{i-1}, s_i) is an arc. A *simple path* is a path that does not go twice through the same vertex, i.e. $s_i \neq s_j$ for $i \neq j$. Its origin is the vertex $s = s_0$. Its end is the vertex $s_n = t$. A *circuit* is a path of non-null length whose origin coincides with its end, i.e. $s_0 = s_n$.

Example 7 *The (undirected) graph $G = (V,E)$ with*

- $V = \{0,1,\ldots,6\}$
- $E = \{(0,1),(3,4),(5,1),(6,3),(6,4)\}$.

is represented as below.

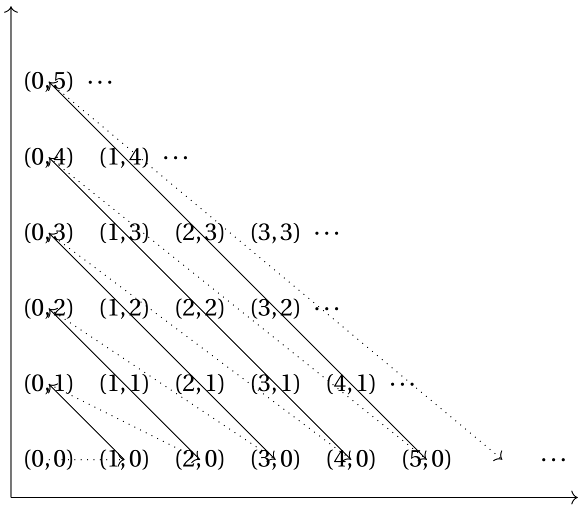


A graph is said to be *connected* if any two vertices are connected by a path.

Example 8 *The graph of Example 7 is not connected since there is no path between vertices* 1 *and* 6.

2 The diagonalisation method

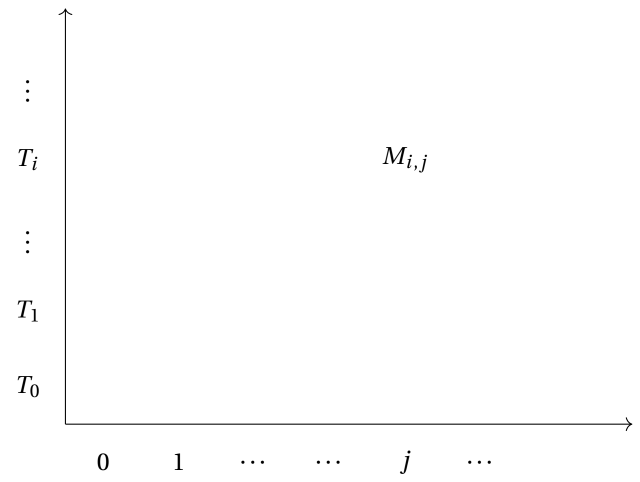
Remember that $\mathbb{N}^2 = \mathbb{N} \times \mathbb{N}$ is *countable*: It is possible to build a bijection between $\mathbb{N}$ and $\mathbb{N}^2$. Below, we illustrate one way of running through all the pairs of integers, in order to realize a bijection between $\mathbb{N}$ and $\mathbb{N}^2$.



Exercise 5 *(solution on page 203) Prove formally that* $\mathbb{N} \times \mathbb{N}$ *is countable by giving the bijection* $f : \mathbb{N}^2 \to \mathbb{N}$ *of the above figure.*

By contrast, the set of subsets of $\mathbb{N}$ is not countable: This can be shown with the *diagonalisation method* due to Cantor.

The reasoning is as follows: Suppose for contradiction that we can enumerate the subsets of $\mathbb{N}$. Then write these subsets as T_1, T_2, $\ldots T_n \ldots$. Every subset T_i of $\mathbb{N}$ can be seen as the row i of an (infinite) matrix $M = (M_{i,j})_{i,j}$ whose entries are in $\{0,1\}$ and whose element $M_{i,j}$ is 1 if and only if element j is in the ith subset of $\mathbb{N}$.



We consider then the subset T^* obtained by "*inverting the diagonal of M*". Formally, one considers $T^* = \{j | M_{j,j} = 0\}$. This subset of $\mathbb{N}$ is not among the enumeration, since otherwise it would have some index j_0: if $j_0 \in T_{j_0} = T^*$, then we should have $M_{j_0,j_0} = 1$ by definition of M, and $M_{j_0,j_0} = 0$ by definition of T^*, which is impossible. If $j_0 \not\in T^*$, then we should have $M_{j_0,j_0} = 0$ by definition of M, and $M_{j_0,j_0} = 1$ by definition of T^*, which is again impossible.

This argument is at the basis of various reasoning in computability theory, as we will see.

Exercise 6 *Prove that the set of sequences $(u_n)_{n\in\mathbb{N}}$ with values in $\{0,1\}$ is not countable.*

Exercise 7 *Prove that the set $\mathbb{R}$ of real numbers is not countable.*

Index

(V, E), 8
., 7
A^c, 4
Σ, 5
Σ^*, 6, 7
Σ_{ASCII}, 6
Σ_{ASCII}, 6
Σ_{exp}, 6
Σ_{latin}, 6
Σ_{number}, 6
$\cap$, 4
$\cup$, 4
ϵ, 6
$\mathscr{P}(E)$, 4
$\subset$, 4
$\times$, 4, 5
w^i, 7
Σ_{bin}, 6

alphabet, 5
    binary, 6
    Latin, 6
application, 5
arcs, 8

binary
    alphabet, 6

$\mathbb{C}$, 5
Cartesian product, 4
    of a family of sets, 5
circuit, 8
complement, 4
concatenation, 7
connected, 9
countable, 9

degree of a vertex, 8
diagonalisation method, 9
domain
    of an application, 5

edges, 8
empty
    word, 6

family of elements of a set, 5
field, 5
function
    total, *see* total function

graph, 8
    undirected, *see* undirected graph

homomorphism
    between languages, 8

image
    of an application, 5
intersection, 4

language, 5, 6
Latin alphabet, 6
length, 6
letters, 5

monoid, 7

$\mathbb{N}$, 5
nodes, 8
    of a graph
        synonym: vertex, *see* vertex

partial function, 5
path, 8
power set of E, 4

prefix, 7
proper
    prefix, 7
    suffix, 7

$\mathbb{R}$, 5
$\mathbb{R}^{>0}$, 5
ring, 5

set
    of words over an alphabet, 6
        notation, *see* Σ^*
simple path, 8
suffix, 7
symbols, 5

total function, 5

undirected graph, 8
union, 4

vertices, 8

word, 5, 6
    empty, *see* empty word

$\mathbb{Z}$, 5

Bibliography