# DISCRETE MATHEMATICS

Mircea Olteanu

# Contents

# Foreword

This textbook contains basic topics in discrete mathematics: sets, logic, relations, graphs, finite automata and boolean algebras. The text addresses to the mathematicians, engineers, and students. It contains theoretical notions and results, as well as worked-out examples.
The references were used as follows:
for the first chapter, [1], [2], [4], [6];
for the second chapter, [3], [5], [8], [9];
for the third chapter, [1], [2];
for the fourth chapter, [1], [3], [5], [9];
for the fifth chapter, [7].

# Chapter 1

# Sets and Logic

## 1.1 Sets, sets of numbers

**1. Definitions**

Any collection of objects is called a **set**. The objects composing a set are called the **elements** of the set. Usually, the sets are denoted by capitals: $A, B, X, Y, ...$ and the elements by small letters: $a, b, x, y, ....$ The fact that $x$ is an element of the set $X$ is denoted $x \in X$; if $a$ is not an element of $X$ we put $a \notin X$. Two sets are said to be equal (we write $A = B$) if they have the same elements. The set which has no elements is called the **empty set** (or null set) and is denoted $\emptyset$.

Let $A$ and $B$ be two sets; we say that $B$ is a **subset** of $A$ (we write $B \subseteq A$ or $A \supseteq B$ ) if every element of $B$ is an element of $A$. If $B \subseteq A$ and $A \neq B$, then we say that $B$ is a **proper subset** of $A$ and we write $B \subset A$. Let us observe that $A \subseteq A$ and $\emptyset \subseteq A$ for every set $A$.

Two obvious properties are:
(i) $A = B$ if and only if $A \subseteq B$ and $B \subseteq A$;
(ii) if $A \subseteq B$ and $B \subseteq C$ then $A \subseteq C$; this property is called transitivity.

If $A$, $B$ are sets, then, their **intersection**, denoted $A \cap B$ is the set of elements which belong to both $A$ and $B$. The sets $A$ and $B$ are said to be **disjoints** if $A \cap B = \emptyset$. The **union** of the sets $A$ and $B$, denoted by $A \cup B$ is the set of elements which lie in $A$ or $B$ (the word "or" is used in the inclusive sense: if $x \in A \cup B$, it is possible that $x \in A \cap B$). The **difference** of $A$ from $B$, denoted by $B \setminus A$ is the set $\{x\,;\, x \in B \text{ and } x \notin A\}$. If $A \subseteq B$, the the difference $B \setminus A$ is called the **complement** of $A$ in $B$ and is denoted $\mathcal{C}_B A$.

If $X$ is a set, we denote by $\mathcal{P}(X)$ the **power set** of $X$ which is defined as the set of all subsets of $X$, i.e. $\mathcal{P}(X) = \{A\ ;\ A \subseteq X\}$.

Some elementary properties of union, intersection and difference are summarized below.

**2. Proposition**

For every sets $A, B, C$ we have:
(i) if $A \subseteq B$ then $A \cup B = B$
(ii) $A \cup B = B \cup A$
(iii) $A \cup (B \cup C) = (A \cup B) \cup C$
(iv) if $A \subseteq B$ then $A \cap B = A$
(v) $A \cap B = B \cap A$
(vi) $A \cap (B \cap C) = (A \cap B) \cap C$
(vii) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
(viii) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
(ix) if $A \subseteq B$ then $B \setminus (B \setminus A) = A$
(x) if $C \supseteq B \supseteq A$ then $C \setminus A \supseteq C \setminus B$
(xi) $C \setminus (A \cup B) = (C \setminus A) \cap (C \setminus B)$
(xii) $C \setminus (A \cap B) = (C \setminus A) \cup (C \setminus B)$.
The last two properties are called De Morgan's rules.

**Proof**

We illustrate the ideas by proving (iii) and (xi).
(iii) We first prove the inclusion $A \cup (B \cup C) \subseteq (A \cup B) \cup C$. If $x \in A \cup (B \cup C)$, then either $x \in A$ or $x \in B \cup C$; if $x \in A$ then $x \in A \cup B$ and so $x \in (A \cup B) \cup C$. If $x \in B \cup C$, then either $x \in B$ or $x \in C$; if $x \in B$, then $x \in A \cup B$ and so $x \in (A \cup B) \cup C$; if $x \in C$ then $x \in (A \cup B) \cup C$. Analogously, $(A \cup B) \cup C \subseteq A \cup (B \cup C)$.
(xi) The inclusion $C \setminus (A \cup B) \subseteq (C \setminus A) \cap (C \setminus B)$: if $x \in C \setminus (A \cup B)$, then $x \in C$ and $x \notin A \cup B$; it results that $x \notin A$ and $x \notin B$, hence $x \in C \setminus A$ and $x \in C \setminus B$; in conclusion $x \in (C \setminus A) \cap (C \setminus B)$.
The other inclusion: $(C \setminus A) \cap (C \setminus B) \subseteq C \setminus (A \cup B)$; if $x \in (C \setminus A) \cap (C \setminus B)$, then $x \in (C \setminus A)$ and $(C \setminus B)$, hence $x \in C$ and $x \notin A$ and $x \notin B$. It results that $x \notin A \cup B$, so $x \in C \setminus (A \cup B)$.

**3. Definition**

Let $A, B$ be two nonempty sets; if $a \in A$ and $b \in B$, we call $(a, b)$ an **ordered pair**. Two ordered pairs $(a, b)$ and $(c, d)$ are equal if and only if

$a = c$ and $b = d$. The **Cartesian product** of $A$ and $B$, denoted $A \times B$ is the set of all ordered pairs: $A \times B = \{(a, b)\ ;\ a \in A \text{ and } b \in B\}$. If $A = B$ we can write $A^2$ for $A \times A$.

A **relation** is a set of ordered pairs; more precisely, a relation is a subset $\mathcal{R} \subseteq A \times B$. The **domain** of the relation $\mathcal{R}$ is the set $\text{Dom}(\mathcal{R}) = \{a \in A\ ;\ \text{there is}\ \ b \in B \text{ such that } (a, b) \in \mathcal{R}\}$. The **range** of $\mathcal{R}$ is $\text{Ran}(\mathcal{R}) = \{b \in B\ ;\ \text{there is } a \in A \text{ such that } (a, b) \in \mathcal{R}\}$.

A **function** (map) defined on $A$ with values in $B$ (the usual notation is $f : A \mapsto B$) is a relation $f \subseteq A \times B$ such that for every $a \in A$ there exists a unique $b \in B$ for which $(a, b) \in f$; we denote $b = f(a)$ if $(a, b) \in f$. The set $A$ is the domain and $B$ is called the codomain of $f$.

The function $f : A \mapsto B$ is called **injective** (one to one) if from $x \neq y$ it results $f(x) \neq f(y)$.
The range (image) of $f$, denoted $f(A)$ or $\text{Ran}(f)$ is the set
$\{y \in B$ there is $\ x \in A$ such that $f(x) = y\}$.
The function $f : A \mapsto B$ is called **surjective** (onto) if $f(A) = B$.

A function is said to be **bijective** if it is injective and surjective. If $f : A \mapsto B$ is bijective map, the **inverse map** of $f$ is the map denoted $f^{-1}$, defined by $f^{-1} : B \mapsto A$, $f^{-1}(t) = s$ if $f(s) = t$. Obviously $f(f^{-1}(t)) = t$ for every $t \in B$ and $f^{-1}(f(s)) = s$ for every $s \in A$. The inverse map $f^{-1}$ is bijective, too.

**3. Equinumerous sets**

Let $A$ and $B$ be two sets. We say that $A$ is **equinumerous** to $B$ if there exists a bijective map $f : A \mapsto B$. This definition (Cantor) seems to be according to our intuition. However, as we shall see, it has some (at first sight) bizarre consequences; for example, a proper subset $B \subset A$ can be equinumerous to $A$.

A set $A$ is said to be **infinite** if there exists a proper subset $B \subset A$ which is equinumerous to $A$. A set is called **finite** if it is not infinite.

**4. The set of natural numbers**

The axiomatic definition (Peano axioms) of what we usually call the set of natural numbers (denoted by $\mathbf{N}$) is as follows.
N1. There is an element $0 \in \mathbf{N}$.
N2. There is a map $s : \mathbf{N} \mapsto \mathbf{N}$ such that $s : \mathbf{N} \mapsto \mathbf{N} \setminus \{0\}$ is bijective.
N3. If $S \subseteq \mathbf{N}$ is a subset such that $0 \in S$ and $s(n) \in S$ for every $n \in S$, then $S = \mathbf{N}$.

If $n \in \mathbf{N}$, the element $s(n)$ is called the **successor** of $n$; obviously, $s(0) \neq 0$. We denote $s(0) = 1$, $s(1) = 2$, etc. The set $\mathbf{N} \setminus \{0\}$ is usually denoted by $\mathbf{N}^\star$.
Axiom N2 implies that $\mathbf{N}$ is an infinite set.

The consequence of axiom N3 is the **principle of induction**:
Suppose $P$ to be a statement such that:
(i) $P(0)$ is true.
(ii) If $P(n)$ is true, then $P(s(n))$ is true.
Then $P(n)$ is true for every $n \in \mathbf{N}$.
To prove it, consider the set $S = \{n \in \mathbf{N}\ ;\ P(n) \text{ is true }\}$ and apply N3.

Starting from the axioms, one can define the usual operations on $\mathbf{N}$ (addition and multiplication) and prove their properties such as commutativity, associativity, distributivity, etc. For example, $n + 0 = n, n + 1 = s(n)$, etc. We suppose known all these.

The **natural order** on $\mathbf{N}$ is the relation defined by $n \leq m$ if and only if there exists $k \in \mathbf{N}$ such that $n + k = m$. It is simple to check that the following properties hold for every $n, m, k \in \mathbf{N}$:
(i) $n \leq n$ (reflexivity)
(ii) if $n \leq m$ and $m \leq n$ then $n = m$ (antisymmetry)
(iii) if $n \leq m$ and $m \leq k$ then $n \leq k$ (transitivity).
This is a **total ordering**, i.e. for every $n, m \in \mathbf{N}$, then $n \leq m$ or $m \leq n$.

We now prove that $\mathbf{N}$ is **well-ordered**:

**5. Proposition**
Every non empty subset $S \subseteq \mathbf{N}$ has a **least element**, i.e. there exists $n_0 \in S$ such that $n_0 \leq m$ for every $m \in S$.
To prove this, let $S$ be as above and let

$$T = \{n \in \mathbf{N}\ ;\ n \leq x, \text{ for every } x \in S\}.$$

Obviously, $0 \in T$ and $T \neq \mathbf{N}$, hence there exists $n_0 \in T$ such that $n_0 + 1 \notin T$ (by induction). It is simple to check that $n_0$ is the least element of $S$.

**6. The cardinal of finite sets**

Let $A$ be a set. If $A = \emptyset$, then, by definition its **cardinal** (number of elements) is 0. If $A \neq \emptyset$ and if there exists $n \in \mathbf{N}$, $n \neq 0$ such that $A$ is equinumerous to $\{1, 2, ..., n\}$, then by definition the cardinal of $A$ (denoted $\text{card}(A)$) is $n$.

As we already mentioned, $\mathbf{N}$ is an infinite set. By definition, the cardinal of $\mathbf{N}$ is $\aleph_0$ (read: $\text{aleph}_0$). A set which is equinumerous to $\mathbf{N}$ is called **countable** (denumerable). A set is termed **at most countable** if it is finite or countable.

**7. Observation**

A set $A$ is countable if and only if it can be written as a **sequence**, i.e. $A = \{a_0, a_1, a_2, ...\}$.

**Proof**

If $A = \{a_0, a_1, a_2, ...\}$, then the map $f : \mathbf{N} \mapsto A,\ f(n) = a_n$ is a bijective. Conversely, if $A$ is countable, there exists a bijective map $f : \mathbf{N} \mapsto A$; if we put $a_n = f(n)$, then the proof is over.

We now prove the following important result:

**8. Theorem**

Every infinite set contains a countable subset.

**Proof**

Let $A$ be an infinite set. Let $a_0 \in A$; then $A \setminus \{a_0\} \neq \emptyset$, hence there exists $a_1 \in A \setminus \{a_0\}$, so $a_1 \neq a_0$. At the step $n$, there exists $a_n \in A \setminus \{a_0, a_1, ..., a_{n-1}\}$, so $a_n$ is different from $a_0, a_2, ..., a_{n-1}$. Obviously, the set $\{a_0, a_1, ...\}$ is countable and it is a subset of $A$.

A consequence of the previous result is:

**9. Corollary**

If $A$ is an infinite set, then for every finite subset $F \subset A$, the sets $A$ and $A \setminus F$ are equinumerous.

**Proof**

Let $F = \{a_0, a_1, ..., a_{n-1}\}$; then $A \setminus F$ is infinite, hence it contains a countable subset: $\{a_n, a_{n+1}, ...\} \subseteq A \setminus F$. The sets $A$ and $A \setminus F$ are equinumerous

because the map $f : A \setminus F \mapsto A$,

$$f(x) = \begin{cases} x, & \text{if } x \in A \setminus \{a_0, a_1, ...a_{k-1}, a_k, a_{k+1}, ...\} \\ a_{n-1}, & \text{if } x = a_n,\ n \in N^\star \end{cases}$$

is a bijection.

**10. Proposition**

Let $X, Y$ be two non empty sets and let $f : X \mapsto Y$.

**i.** If $X$ and $Y$ are countable sets, then $X \cup Y$ is countable.

**ii.** If $f$ is injective and $Y$ is countable, then $X$ is countable.

**iii.** If $f$ is surjective and $X$ is countable, then $Y$ is countable.

**iv.** The Cartesian product $\mathbf{N} \times \mathbf{N}$ is countable.

**Proof**

The first two assertions are obvious (exercise); for the third one, consider the injective map $g : Y \mapsto X,\ g(y) = x$, where $x$ is an element such that $f(x) = y$.

**iv.** The function $f : \mathbf{N} \times \mathbf{N} \mapsto \mathbf{N},\ f(m, n) = 2^m 3^n$ is injective.

It can be proved (by using iv above) that an at most countable union of at most countable sets is at most countable.

**11. The set of integers**

For each natural number $n \in \mathbf{N}^\star$, we select a new symbol denoted by $-n$; the set of **integers** , denoted by $\mathbf{Z}$ is the union of $\mathbf{N}$ and all the symbols $-n$, i.e. $\mathbf{Z} = \{..., -3, -2, -1, 0, 1, 2, 3, ...\}$.

We suppose known the usual operations with integers; the natural order of $\mathbf{N}$ can be extended to $\mathbf{Z}$ by $-n \leq -m$ if and only in $m \leq n$ for every $m, n \in \mathbf{N}^\star$, etc. Obviously, $\mathbf{N} \subset \mathbf{Z}$; we now prove that in fact they are equinumerous, so $\text{card}(\mathbf{Z}) = \aleph_0$.

**12. Proposition**

The set of integers is countable.

**Proof** The set $\mathbf{Z}$ can be written as a sequence: $\mathbf{Z} = \{0, 1, -1, 2, -2, 3, ...\}$, or equivalently, the map $f : \mathbf{N} \mapsto \mathbf{Z}$,

$$f(n) = \begin{cases} -\frac{n}{2}, & \text{if } n \text{ is even} \\ \frac{n+1}{2}, & \text{if } n \text{ is odd} \end{cases}$$

is bijective.

**13. The set of rational numbers**

If $m \in \mathbf{Z}$ and $n \in \mathbf{N}^\star$ we call $\frac{m}{n}$ **rational number**. The usual notation of the set of all rational numbers is $\mathbf{Q}$. Two rational numbers $\frac{m}{n}$ and $\frac{k}{p}$ are equals if $mp = nk$; the usual properties of rational numbers, operations, the natural order, etc. are supposed to be known.

**14. Proposition**

The set of rational numbers is countable, hence $\text{card}(\mathbf{Q}) = \aleph_0$.

**Proof**

The set of positive rational numbers can be written as a sequence:

Alternative proof: the map $f : \mathbf{Z} \times \mathbf{N}^\star \mapsto \mathbf{Q},\ f(m,n) = \frac{m}{n}$ is surjective; now apply proposition 10(iii).

An example of a uncountable set is the following.

**15. Example: an uncountable set**

Let $X$ be a set with 2 elements, for example, $X = \{0,1\}$ and let $A = \{a_1a_2a_3.... ;\ a_i \in X\}$. Then the set $A$ is uncountable.

**Proof**

We suppose that the conclusion is false, hence the set $A$ can be written as a sequence: $A = \{x_1, x_2, x_3, ...\}$; by the definition of $A$, it results that

$$x_1 = a_1^1a_2^1a_3^1...., x_2 = a_1^2a_2^2a_3^2...., x_3 = a_1^3a_2^3a_3^3....,$$

with obvious notations. We consider the element $y = b_1b_2b_3....$ such that for every $i \in \mathbf{N}^\star$, $b_i \in X$ and $b_i \neq a_i^i$. It results that $y \in A$ but $y \neq x_n$, for every $n \in \mathbf{N}^\star$, which is a contradiction.

**16. Real numbers**

The definition of the set $\mathbf{R}$ of **real numbers** is not the goal of this work. Just let us remember that every real number $x$ can be represented in the **decimal form** as $x = k, a_1a_2a_3...., \text{ where } k \in \mathbf{Z} \text{ and } a_i \in \{0,1,2,...,9\}$.
By using a similar argument as in example 15, prove:

**17. Proposition**

The set of real numbers is uncountable.

By definition, its cardinal is $\aleph_1$.

## 1.2 Propositions, predicates, quantifiers

Logic is usually known as the science of reasoning. The symbolic techniques are required for computer logic for at least two reasons:

(i) at the hardware level, the use of symbolic logic simplifies the design of logic circuits to implement instructions;

(ii) at the software level, symbolic logic is helpful in the design of programs.

**1. Definition**

A **proposition** (sentence) is a statement (expression) which is either **true** or **false** (but not both). If a proposition is true, it has the **truth value** "true" (denoted T or 1) and if it is false it has the truth value "false" (denoted F or 0).

In this section we discuss about propositions, their truth or falsity and ways of connecting propositions to form new propositions. The new propositions which are obtained from the old ones by using **symbolic connectives** are called **compound propositions**.

**2. Definitions**

Let $p$ and $q$ be two propositions.

The **conjunction** of $p$ and $q$ is true if the two propositions $p$, $q$ are both true and it is false otherwise; the conjunction is denoted by $p \wedge q$ (read "and").

The **disjunction** of $p$ and $q$ is true if at least one of two propositions $p$, $q$ is true and it is false otherwise; the disjunction is denoted by $p \vee q$ (read "or").

We can summarize these definitions in the following **truth table**:

| $p$ | $q$ | $p \wedge q$ | $p \vee q$ |
|---|---|---|---|
| 1 | 1 | 1 | 1 |
| 1 | 0 | 0 | 1 |
| 0 | 1 | 0 | 1 |
| 0 | 0 | 0 | 0 |

The **negation** of the proposition $p$ is true if $p$ is false and it is false if $p$ is true. We denote the negation of $p$ by $\overline{p}$ (read "not" $p$). The truth table is:

| $p$ | $\overline{p}$ |
|---|---|
| 1 | 0 |
| 0 | 1 |

The **implication** $p \to q$ (read "if $p$ , then $q$") is false only when $p$ is true and $q$ is false. It is defined by the truth table:

| $p$ | $q$ | $p \to q$ |
|---|---|---|
| 1 | 0 | 0 |
| 1 | 1 | 1 |
| 0 | 1 | 1 |
| 0 | 0 | 1 |

The connective $\to$ is called "conditional". In $p \to q$, the proposition $p$ is called **hypothesis** (or assumption) and $q$ is called **conclusion**. It is convenient to note that if we draw a false conclusion from a true hypothesis, then our argument must be faulty. In any other case, our argument is valid.

The proposition $p \leftrightarrow q$ (read: "$p$ if and only if $q$") is true if the two sentences $p, q$ are both true or false and it is false otherwise; the truth table is:

| $p$ | $q$ | $p \leftrightarrow q$ |
|---|---|---|
| 1 | 1 | 1 |
| 1 | 0 | 0 |
| 0 | 1 | 0 |
| 0 | 0 | 1 |

The connector $\leftrightarrow$ is called "double conditional".

**3. Definition**
A **tautology** is a proposition which is true regardless of the truth values of the basic propositions which comprise it. The propositions $p$ and $q$ are called **logically equivalent** if the proposition $p \leftrightarrow q$ is a tautology.

**4. Proposition**

Let $p, q, r$ be propositions; the following propositions are tautologies.

(i) Commutative laws:

$p \wedge q \leftrightarrow q \wedge p$

$p \vee q \leftrightarrow q \vee p$

(ii) Associative laws:

$(p \wedge (q \wedge r)) \leftrightarrow ((p \wedge q) \wedge r)$

$(p \vee (q \vee r)) \leftrightarrow ((p \vee q) \vee r)$

(iii) Distributive laws:

$(p \wedge (q \vee r)) \leftrightarrow ((p \wedge q) \vee (p \wedge r))$

$(p \vee (q \wedge r)) \leftrightarrow ((p \vee q) \wedge (p \vee r))$

(iv) De Morgan laws

$\overline{(p \vee q)} \leftrightarrow (\overline{p} \vee \overline{q})$

$\overline{(p \vee q)} \leftrightarrow (\overline{p} \wedge \overline{q})$

(v) $p \vee \overline{p}$

(vi) $(p \rightarrow q) \leftrightarrow (\overline{q} \rightarrow \overline{p})$.

(vii) $(p \rightarrow q) \leftrightarrow (\overline{p} \vee q)$.

The propositions $p \rightarrow q$ and $\overline{q} \rightarrow \overline{p}$ are logically equivalent; the latter is called the **contrapositive** of the first. It can be proved that the propositions $p \rightarrow q$ and $q \rightarrow p$ are not logically equivalent; the latter is called the **converse** of the first. The proposition $\overline{p} \rightarrow \overline{q}$ is called the **inverse** of $p \rightarrow q$; it is logically equivalent with the converse.

**Proof** Use the truth tables.

**5. Inference laws**

The usual problem of logic is how the truth of some propositions is related with the truth of other propositions.

An **argument** is a set of two or more propositions related to each other in such a way that all but one of them (the **premises**) are supposed to provide support for the remaining one (the **conclusion**). The transition from premises to conclusion is the **inference** upon which the argument relies.

Let us suppose that the premises of an argument are all true; the conclusion may be either true or false. If the conclusion is true then the argument is **valid**; if the conclusion is false, then the argument is **invalid**.

To test the validity of an argument, one follows the steps:

(i) Identify the premises and the conclusion of the argument.

(ii) Compute the truth table of the premises and of the argument.

(iii) Find the rows in which all premises are true.

(iv) If in all rows of step (iii) the conclusion is true, then the argument is valid; otherwise the argument is invalid.

The usual types of valid arguments are listed below:

**6. Proposition**

Let $p, q, r$ be propositions; the following are tautologies:
(i) **modus ponens** (or method of affirming) $(p \wedge (p \to q)) \to q$;
(ii) **modus tollens** (or method of denial) $((p \to q) \wedge \overline{q}) \to \overline{p}$;
(iii) **law of syllogism** $((p \to q) \wedge (q \to r)) \to (p \to r)$.

**Proof**

Use the truth tables.

**7. Examples**

Test if the following form an argument:

(i) Premises: The Earth is larger than the Sun; lemons are yellow. Conclusion: Politehnica University is in Bucharest.

This is not an argument because the truth or falsity of the conclusion is not a consequence of the truth values of the premises.

(ii) Premises: Peter is a student; all students study mathematics. Conclusion: Peter study mathematics.

This is an argument; moreover, this is a valid argument. Indeed, let $p$: "Peter is a student", $q$: "all students study mathematics" and $r$: "Peter study mathematics"; now use the truth tables.

**8. Exercises**

Let $p$ and $q$ be two propositions; test the validity of the following arguments.

(i) $\mathcal{P}$ (premises): $p \to q$, $q$; $\mathcal{C}$ (conclusion): $p$.

It is false (this is called **converse error**).

(ii)$\mathcal{P}$: $p \to q$, $\overline{p}$; $\mathcal{C}$: $\overline{q}$.

It is false (**inverse error**).

(iii) **Disjunctive syllogism**: $\mathcal{P}$: $p \vee q$, $\overline{q}$; $\mathcal{C}$: $p$.

(iv) **Rule of contradiction** If $c$ is a **contradiction** (i.e. a proposition always false), then the argument:

$\mathcal{P}$: $\overline{p} \to c$; $\mathcal{C} : p$
is valid.

**9. Predicates**

Statements such as "$x \leq 1$", "$x + y$ is even" (which contain one or more variables) are usually found in mathematical assertions and in computer programming. These statements are not propositions as long as the variables are not specified. We call such expression a **predicate** (or propositional function); it involves one or more variables which belong to a set (called domain). By substitution of the variables with values from the domain, one obtains propositions (true or false). The set of the variables for which the predicate is true is called **the truth set**.
We shall denote predicates with capitals letters: $P(x), Q(x, y)$, etc, x,y...are the **free variables** (variables).

**10. Example**

Let $P(x)$ be the predicate "$2x^2 + x - \frac{1}{2} = 0$" with the domain the set of natural numbers, $\mathbf{N}$. Then the truth set is $\emptyset$. If we consider the domain $\mathbf{Z}$, the truth set is $\{-1\}$ and if the domain is $\mathbf{Q}$, then the truth set is $\{-1, \frac{1}{2}\}$.

**11. Quantifiers**

Let $P(x)$ be a predicate (with one variable) with domain $D$.
We consider the following two propositions:
$p$ : "$P(x)$ is true for all values of $x \in D$".
$q$ : "$P(x)$ is true for at least one value $x \in D$".
The usual notations for the above propositions are:
$\forall x, P(x)$ (read "for all $x$ , $P(x)$ is true")
$\exists x, P(x)$ (read "there exists $x$, $P(x)$ is true"), respectively.
Sometimes (if the domain is not understood) one can write:
$\forall x \in D, P(x)$ and $\exists x \in D, P(x)$, respectively.
The symbols $\forall$ and $\exists$ are called the **universal quantifier** and **existential quantifier**, respectively.
Note that the proposition $\forall x \in D, P(x)$ is false if $P(x)$ is false for at least one value $x_0 \in D$; in this case $x_0$ is called a **counterexample**.
The notation $\exists!$ stands for "there exists a unique".

If $P(x)$ and $Q(x)$ are two predicates, then the proposition:

$$\forall x \in D,\ P(x) \to Q(x)$$

is called the **universal conditional proposition** ("for all $x \in D$, if $P(x)$, then $Q(x)$"). For example, $\forall x \in \mathbf{R}$, if $x \leq -3, \text{then } |x| \geq 3$ is a universal

conditional proposition.

**12. Exercise**

Write by using quantifiers the following propositions:
$p$: "Every square is a rectangle".
$q$: "If a real number is an integer then it is a rational number".
$r$: "Every natural number is the sum of the squares of four integers" (Lagrange theorem).
$s$: "Every even natural number greater than 2 is the sum of two primes" (this is known as "Goldbach conjecture" and it is not yet known if it true or false).

**13. Negation of quantifiers**

The following rules of negation hold:

$$\overline{\forall x \in D, P(x)} = \exists x \in D, \overline{P(x)}$$

$$\overline{\exists x \in D, P(x)} = \forall x \in D, \overline{P(x)}$$

As an example, the negation of the universal conditional proposition is:

$$\overline{\forall x \in D,\ P(x) \to Q(x)} = \exists x \in D, \overline{P(x) \to Q(x)} = \exists x \in D, P(x) \wedge \overline{Q(x)}$$

**14. Examples**

(i) The negation of the Goldbach conjecture (see exercise 12) is (we denote by $\mathcal{P}$ the set of prime numbers):

$$\overline{\forall n \in \mathbf{N} \setminus \{1\}, \exists p, q \in \mathcal{P}, 2n = p + q} = \exists n \in \mathbf{N} \setminus \{1\}, \forall p, q \in \mathcal{P}, 2n \neq p + q$$

(ii) The definition of the limit; $L = \lim_{x \to a} f(x)$ if and only if

$$\forall \epsilon > 0, \exists \delta > 0, |x - a| < \delta \to |f(x) - L| < \epsilon.$$

The negation is:

$$\exists \epsilon > 0, \forall \delta > 0, \overline{|x - a| < \delta \to |f(x) - L| < \epsilon} =$$

$$= \exists \epsilon > 0, \forall \delta > 0, |x - a| < \delta, |f(x) - L| \geq \epsilon.$$

(iii) The definition of the Riemann integral: $f$ is Riemann integrable on $[a, b]$ if, by definition:

$$\exists I \in \mathbf{R}, \forall \epsilon > 0, \exists \delta > 0, \forall \Delta = (x_i)_{0 \leq i \leq n} \text{ a division of } [a, b], \text{ with}$$

$$\| \Delta \| < \delta,\ \forall (\xi_i)_{1\leq i\leq n}, \xi_i \in [x_{i-1}, x_i],\ \text{then } \left| I - \sum_{i=1}^{n} f(\xi_i)(x_i - x_{i-1}) \right| < \epsilon.$$

The negation is:

$$\forall I \in \mathbf{R},\ \exists \epsilon > 0, \forall \delta > 0,\ \exists \Delta = (x_i)_{0\leq i\leq n} \text{ a division of } [a,b], \text{with}$$

$$\| \Delta \| < \delta, \exists (\xi_i)_{1\leq i\leq n}, \xi_i \in [x_{i-1}, x_i],\ \text{such that}$$

$$\left| I - \sum_{i=1}^{n} f(\xi_i)(x_i - x_{i-1}) \right| \geq \epsilon.$$

**15. Exercises**

(i) Find the contrapositive, the converse and the inverse of the universal conditional proposition.

(ii) Find the negations of the following propositions:
$\forall x, \exists y, P(x,y)$
$\exists x, \forall y, P(x,y)$
$\forall x, \exists y, \forall z, \forall w, P(x,y,z,w)$

(iii) Find the truth value of the following propositions:
$\exists! x \in \mathbf{R}, \forall y \in \mathbf{R}, xy = y.$
$\forall x \in \mathbf{Z}, \exists! y \in \mathbf{Z}, xy^2 = x.$

**16. Valid arguments with quantified premises**

Prove that the following arguments are valid ($\mathcal{P}$ is the hypothesis and $\mathcal{C}$ is the conclusion):

(i) **Universal instantiation**
$\mathcal{P} : \forall x \in D, P(x);\ a \in D.$
$\mathcal{C} : P(a).$

(ii) **Universal modus ponens**
$\mathcal{P} : \forall x \in D, P(x) \to Q(x);\ \exists a \in D, P(a).$
$\mathcal{C} : Q(a).$

(iii) **Universal modus tollens**
$\mathcal{P} : \forall x \in D, P(x) \to Q(x);\ \exists a \in D, \overline{Q(a)}.$
$\mathcal{C} : \overline{P(a)}.$

**17. Invalid arguments with quantified premises**

Prove that the following arguments are invalid.

(i) **Converse error**

$\mathcal{P} : \forall x \in D,\ P(x) \to Q(x); \exists a \in D, Q(a);$
$\mathcal{C} : P(a).$
(ii) **Inverse error**
$\forall x \in D, P(x) \to Q(x),\ \exists a \in D, \overline{P(a)};$
$\mathcal{C} : \overline{Q(a)}.$

**18. Exercise**
(i) Fill in the true conclusion in the following argument according to universal modus ponens:
$\forall k \in \mathbf{Z}$, if $\exists m \in \mathbf{Z}$ such that $k = 2 \cdot m$ then $k$ is even;
$6 = 2 \cdot 3$.
Conclusion: ?
(ii) Fill in the true conclusion in the following argument according to universal modus tollens:
All students in Politehnica study mathematics.
George does not study mathematics.
Conclusion: ?

**19. Exercise**
What kind of error does the following arguments exhibit:
(i) All students in Politehnica study mathematics.
Paul studies mathematics.
Hence Paul is student in Politehnica.
(ii) All students in Politehnica study mathematics.
Michael is not a student in Politehnica.
Hence Michael does not study mathematics.

**Digital Logic Design**
The use of digital systems is to manipulate discrete information, represented by physical quantities such as voltages and current. The smallest unit is one **bit** (binary digit). Every electronic switch has two physical states (high voltage and low voltage), so we associate the bit 1 to high voltage and 0 for low voltage. A **logic gate** is the smallest processing unit in a digital system; it has few bits as **input** and generates one bit as **output**. A **logic circuit** is composed of several logic gates connected by wires; it has a group of bits as input and generates one or more bits as output. The **input-output table** of a logic circuit is the truth table of the output for all truth values of the input.

**20. Basic logic gates**

There are five basic logic gates.

(I) **NOT** gate (or inverter). The input has one bit, $p$; if $p = 0$, then the output is 1; if $p = 1$, the output is 0. The symbol is $\overline{p}$.

(II) **AND** gate. The input has two bits, $p$ and $q$; The output is 1 if $p = q = 1$ and it is 0 otherwise. It is denoted by $p \wedge q$.

(III) **OR** gate. The input has two bits, $p$ and $q$; the output is 1 if either $p$ or $q$ is 1 and it is 0 otherwise. Its symbol is $p \vee q$.

(IV) **NAND** gate. The input has two bits, $p$ and $q$; the output is is 0 if $p = q = 1$ and it is 0 otherwise. Its symbol is $\overline{p \wedge q}$.

(V) **NOR** gate. The input has two bits, $p$ and $q$; the output is 0 if at least one of $p$ or $q$ is 1 and it is 1 otherwise. Its symbol is $\overline{p \vee q}$.

**21. Exercise**

Write the **input-output table** (or table of truth) of the basic logic gates.

**23. Example**

Find the logic circuit of the output: $\overline{p \wedge q} \vee \overline{r}$.

## 1.3 Methods of Proofs

This section contains few common methods of proofs in mathematics.

Loosely speaking, a **mathematical theory (system)** consists of **axioms, definitions, theorems**. An axiom is a statement assumed to be true. The new concepts are introduced by definitions, while the theorems (lemmas propositions, corollaries, etc) are statements that has been proved to be true. An argument used to establish the truth of a theorem is called a **proof**. Perhaps the most common example is the Euclidean geometry (for example: the axiom of the parallels, the definition of the equilateral triangle, Pythagoras' theorem, etc).

**1. Existence proofs**

The proof of a theorem of the form "$\exists x \in D$ such that $P(x)$" is called an **existence proof**. Such a proof is **constructive** if either it finds a particular $x \in D$ such that the proposition $P(x)$ be true or by exhibiting an algorithm for finding an $x$ for which $P(x)$ be true. The proof is **nonconstructive** if either it shows the existence of $x \in D$ by using a previous result (axiom,

theorem) or by proving that the assumption that there is no such $x \in D$ leads to a contradiction.

**2. Example**

(i) Prove that there are natural numbers which are the sum of squares of two natural numbers. It can be written as: $\exists n \in \mathbf{N}$, $\exists k, m \in \mathbf{N}$, $n^2 = k^2 + m^2$. One possible proof is $5^2 = 4^2 + 3^2$. Another proof: $13^2 = 12^2 + 5^2$.

(ii) As an example of algorithm, below we give the algorithm to convert an integer $n \in \mathbf{Z}$ from base 10 to base 2:

**First step**: Write $n = 2q_0 + r_0$, where $q_0$ is the quotient and $r_0$ is the remainder of the division of $n$ by 2.

**Second step**: If $q_0 = 0$, then $n$ is in fact written in base 2; if not, then divide $q_0$ by 2 to obtain $q_0 = 2q_1 + r_1$.

**Third step**: If $q_1 = 0$, then the form of $n$ in base 2 is $n_{(2)} = r_1 r_0$; if not, repeat the process.

**Fourth step**: After a finite number of divisions (say, $k + 1$), the quotient will be null: $q_k = 0$. In this case, $n_{(2)} = r_k r_{k-1} ... r_1 r_0$.

**3. Universal propositions**

Most theorems are of the form $\forall x \in D$, if $P(x)$, then $Q(x)$; here $P(x)$ is called **hypothesis** and $Q(x)$ is the **conclusion**.

Let us suppose that the statement has the particular form $\forall x \in D$, then $Q(x)$; if the set $D$ is finite, then one can check (eventually by using a computer) the truth value of $Q(x)$ for each $x \in D$. This is called the method of **exhaustion**.

For example, prove that "$\forall n \in \{1, 2, 3, ...10\}$, $n^2 - n + 11$ is a prime number" or prove that "23 is a prime number".

If the set $D$ is not finite, the most powerful method to prove a universal theorem "$\forall x \in D$, if $P(x)$, then $Q(x)$" is the **method of generalizing from a generic particular**. This consists of picking an arbitrary fixed element $x \in D$ (called a **generic element**), check that the hypothesis $P(x)$ is true and then (by using rules of inference, definitions, axioms, previous theorems) conclude that $Q(x)$ is true. A **direct proof** is a method that consists of showing that if $P(x)$ is true, then $Q(x)$ is true.

To prove that a universal proposition is false it is sufficient to find an element $x \in D$ such that $P(x)$ is true and $Q(x)$ is false. Such an element $x$ is called a **counterexample**.

**4. Example**

(i) Prove that the sum of two even integers is even.

Let $m, n$ be two even integers; then there exist $k_1, k_2 \in \mathbf{Z}$ such that $n = 2k_1$ and $m = 2k_2$. Then $m + n = 2k_1 + 2k_2 = 2(k_1 + k_2)$, hence $m + n$ is even.

(ii) Prove that $\forall x \in \mathbf{R}, x^2 + x + 1 > 0$.

Compute $\Delta = -3$, etc.

(iii) Prove that the proposition $\forall x \in \mathbf{R}$, if $x < 1$, then $x^2 < 1$ is false; a counterexample is $x = -2$.

**Proof by contradiction**

Suppose one wants to prove that the proposition $p$ is true. A **proof by contradiction** consists in supposing that $p$ is false (so $\overline{p}$ is true) and then derive a contradiction.

**5. Examples**

(i) If $n^2$ is an even integer, then $n$ is an even integer too.

(ii) Prove that $\sqrt{2}$ is irrational.

**Proof**

(i) Suppose the contrary, so $n$ is odd. It results that there exists $k \in \mathbf{Z}$ such that $n = 2k+1$. We get $n^2 = (2k+1)^2 = 4k^2+4k+1 = 2(2k^2+2k)+1$, so $n^2$ is odd; this is a contradiction with the hypothesis.

(ii) Suppose the contrary, i.e. $\sqrt{2}$ is not rational, hence it is rational; then there exist $m, n \in \mathbf{Z}$, with no common divisors such that $\sqrt{2} = \frac{m}{n}$. It results that $m^2 = 2n^2$, hence $m$ is even, i.e. there exists $k \in \mathbf{Z}$ such that $m = 2k$. We get that $2n^2 = m^2 = 4k^2$, hence $n^2 = 2k^2$; it finally results that $n$ is even (see (i) above) which contradicts the assumption that $m, n$ have no common divisors.

**6. Proof by contrapositive**

We know that the propositions $p \to q$ and $\overline{q} \to \overline{p}$ are equivalent; **proof by contrapositive** means to prove $\overline{q} \to \overline{p}$ instead of proving $p \to q$.

**7. Exercise**
Adapt the proofs of example 5 by using the contrapositive method.

**8. Principle of induction (strong form)**
We already stated in this chapter one form of the principle of induction. The so called strong form of this principle is as follows.

Let $P(n)$ be predicate depending on a natural free variable $n \in \mathbf{N}$. If it satisfies the conditions:
(i) $P(m_0)$ is true for a $m_0 \in \mathbf{N}$.
(ii) $P(n+1)$ is true whenever $P(k)$ is true for all $n_0 \leq k \leq n$,
then $P(n)$ is true for all naturals $n \geq m_0$.
**Proof**
We prove by contradiction; If the conclusion does not hold, then the set $S = \{n \in \mathbf{N} \ ; \ n \geq m_0 \ P(n) \text{ is false}\}$ is not empty. Let $n_0$ be the least element of $S$; if $n_0 = m_0$, this contradicts the assumption (i). If $n_0 > m_0$, then $P(m)$ is true for all $m \leq n_0 - 1$ but $P(n_0$ is false, contradicting the assumption (ii).

In the following we give few examples of applications of the principle of induction.

**9. Exercises**
Prove by induction the following identities $\forall n \in \mathbf{N}^\star$:

$$1 + 2 + ... + n = \frac{n(n+1)}{n}$$

$$1^2 + 2^2 + ...n^2 = \frac{n(n+1)(2n+1)}{6}.$$

$$1 + q + q^2 + ... + q^{n-1} = \frac{1-q^n}{1-q}, \ \forall q \in \mathbf{R} \setminus \{1\}$$

**10. Exercise**
Prove by induction the inequalities.
(i) $\forall n > 3, \ 3^n > n^3$.
(ii) $\forall n \geq 4, \ 2^n < n!$

(iii) Let $h > -1$; $\forall n \in \mathbf{N}:\ 1 + nh \leq (1+h)^n$.

**11. Exercise**
(i) Prove that $\forall n \in N^\star,\ 4^n - 1$ is divisible by 3.
(ii) Prove that $2^{3n} - 1$ is divisible by 7.

**12. De Moivre Formula**
Let $t \in \mathbf{R}$; then for all $n \in \mathbf{N}$ the following formula holds:

$$(\cos t + i \sin t)^n = \cos(nt) + i \sin(nt)$$

**13. The Binomial Formula**
For every $n, k \in \mathbf{N},\ k \leq n$, the binomial coefficient is defined by

$$\left( \begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$$

Then the Binomial Formula holds:

$$(a+b)^n = \sum_{k=0}^{n} \left( \begin{array}{c} n \\ k \end{array} \right) a^{n-k} b^k$$

**14. Fibonacci's Sequence**
In the XIII-th century, the Italian mathematician Leonardo Fibonacci proposed the study of the following sequence

$$F_0 = 0, F_1 = 1, F_{n+1} = F_n + F_{n-1}, \forall n \geq 1.$$

Such a definition is called a **recurrence**, or **definition by induction**. The numbers in Fibonacci's sequence are called **Fibonacci's numbers**.

There are many interesting relations with Fibonacci's numbers. Some of them are listed below:
(i) $F_0 + F_1 + ... + F_n = F_{n+2} - 1$.
(ii) $F_1 + F_3 + ... + F_{2n-1} = F_{2n}$.
(iii) $F_0 - F_1 + F_2 - F_3 + ... - F_{2n-1} + F_{2n} = F_{2n-1} - 1$.
(iv) $F_0^2 + F_1^2 + ... + F_n^2 = F_n F_{n+1}$.
(v) $F_{n-1} F_{n+1} - F_n^2 = (-1)^n$.

We prove by induction the first identity; the formula holds for $n = 0$. We now suppose that it is true for all $k \in \mathbf{N}, 0 \leq k \leq n$ and we prove it for $n+1$:

$$(F_0 + F_1 + ... + F_n) + F_{n+1} = F_{n+2} - 1 + F_{n+1} = (F_{n+2} + F_{n+1}) - 1 = F_{n+3} - 1$$

**15. The formula for the Fibonacci's numbers**

Prove by induction that for all $n \in \mathbf{N}$, the following formula holds:

$$F_n = \frac{1}{\sqrt{5}} \left( \left( \frac{1+\sqrt{5}}{2} \right)^n - \left( \frac{1-\sqrt{5}}{2} \right)^n \right)$$

The number $\frac{1+\sqrt{5}}{2} \approx 1,618$ is called the **golden ratio**, usually denoted by $\varphi$; it is the solution of dividing a segment in **extreme and mean ratio**. More precisely, if we consider two segments of length $a$ and $b$, respectively, then $\varphi = \frac{a+b}{a} = \frac{a}{b}$. It results that $\varphi$ is the positive solution of the quadratic equation $x^2 - x - 1 = 0$. It is also the ratio between the diagonal and the side of a regular pentagon.

**16. Counting subsets**

(i) **The power set**

Let $A$ be a set with n elements; prove by induction that the power set, $\mathcal{P}(A)$, has $2^n$ elements.

(ii) **The number of ordered $k$-subsets**

Let $A$ be a set with $n$ elements and let $k \in \mathbf{N},\ 0 \leq k \leq n$; by an **ordered $k$- subset** of $A$ we mean a $k$-tuple $(a_1, a_2, ..., a_k)$, with $a_i \in A$, $\forall i = 1, 2, ..., k$.

Prove that the number of ordered $k$-subsets is $\frac{n!}{(n-k)!}$.

(iii) **The number of $k$-subsets**

Prove by induction that the number of subsets of cardinal $k$ of a set with $n$ elements is $\left( \begin{array}{c} n \\ k \end{array} \right) = \frac{n!}{k!(n-k)!}$.

**17. Examples of wrong proofs**

(i) Let us consider the following (obviously false) proposition:

"All horses have the same color"

Find the mistake in the following "proof" by induction:

Let $H$ be the set of all horses; the induction will be with respect to the cardinal of $H$, say $n$. If $n = 1$, then the proposition is obviously true. We

now suppose that the property is true if $\text{card}(H) = k$, for all $0 \leq k \leq n$ and we prove it if $\text{card}(H) = n + 1$.
Let $H = \{h_1, h_2, ..., h_n, h_{n+1}\}$; according to the assumption, in every set containing maximum $n$ elements, all horses have the same color.

$$H = \{\underbrace{h_1, h_2, ..., h_n}, h_{n+1}\} = \{h_1, \underbrace{h_2, ..., h_n, h_{n+1}}\}$$

In the set $\{h_1, h_2, ..., h_n\}$ all the horses have the same color; by the same argument, all the horses in the set $\{h_2, h_3, .., h_{n+1}\}$ have the same color, so all horses $\{h_1, h_2, ...h_n, h_{n+1}\}$ have the same color.

(ii) Same problem for the following proposition:

"Let $d_1, d_2, ...d_n$ be $n \geq 2$ distinct lines on the plane such that no two of which are parallel; then all these lines have a point in common."

"Proof"

For $n = 2$ the assertion is obviously true. We assume now that the assertion holds for $n$ and we prove it for $n+1$. Let $d_1, d_2, ...d_n, d_{n+1}$ be $n+1$ lines as in the statement. By the inductive hypothesis, $d_1, d_2, ...d_n$ have a point in common, say $X$. By the same reason, the lines $d_1, d_2, ...d_{n-1}, d_{n+1}$ have a point in common, say $Y$. The line $d_1$ is in both above sets, so it contains both points $X$ and $Y$; the same is true for $d_{n-1}$, so it contains also the points $X$ and $Y$. It results that $X = Y$ because the lines $d_1$ and $d_{n-1}$ have only one point in common, hence the "proof" is over.

## 1.4 Prime numbers

In this section we investigate some common properties of the integers. Number theory is a very old field of mathematics: its roots go back about 2500 years ago, at the beginning of Greek mathematics.

**Divisibility of integers**

Let $a, b \in \mathbf{Z}$, $a \neq 0$; we say that $a$ divides $b$ (we write $a|\ b$) if $\exists c \in \mathbf{Z}$ such that $b = ac$. In this case, $a$ is said to be a divisor of $b$ or $b$ is a multiple of $a$.

The following properties are obvious:

(i) $\forall a \in \mathbf{Z}^\star,\ a|\, a$ and $a|\, -a$.

(ii) $\forall a \in \mathbf{Z},\ 1|\, a$ and $-1|\, a$.

(iii) If $a|\, b$ and $b|\, c$, then $a|\, c$.

(iv) If $a|\, b$ and $a|\, c$ then $\forall m, n \in \mathbf{Z},\ a|\, (bm + cn)$.

**1. Theorem**
Let $a \in \mathbf{N}^\star$ and $b \in \mathbf{Z}$; then there exist $q, r \in \mathbf{Z}$ such that:

$$b = aq + r \text{ and } 0 \leq r < a$$

The number $r$ is called the **remainder** and $q$ is called the **quotient**.
**Proof**
We first show the existence of $q$ and $r$. Let

$$S = \{b - as \geq 0 \ ; \ s \in \mathbf{Z}\}$$

Obviously, $S \subseteq \mathbf{N}$ and $S \neq \emptyset$. By the well-ordering property it results that $S$ has a smallest element, say $r$. We choose $q \in \mathbf{Z}$ such that $b - aq = r$. We now prove by contradiction that $r < a$. If $r \geq a$, then:

$$b - a(q+1) = (b - aq) - a = r - a \geq 0,$$

hence $b - a(q+1) \in S$. But $b - a(q+1) < r$, so we have a contradiction with the fact that $r$ is the smallest element of $S$. We now prove the uniqueness. If

$$b = aq_1 + r_1 = aq_2 + r_2$$

with $0 \leq r_1 < a$ and $0 \leq r_2 < a$, we get:

$$a| \ q_1 - q_2 \ | \ = | \ r_1 - r_2 \ | \ < a$$

It results that $| \ q_1 - q_2 \ | < 1$ and $| \ q_1 - q_2 \ | \in \mathbf{N}$, so $| \ q_1 - q_2 \ | = 0$. Finally $q_1 = q_2$ and $r_1 = r_2$.

**2. Definition**
Let $a \in \mathbf{N}$, $a \geq 2$; the number $a$ is called **prime** if it has only two natural divisors, namely 1 and $a$; it is called **composite** if it is not prime. The number 1 is neither prime or composite.

**3. Proposition**
The set of primes is infinite.
**Proof**
Suppose the set of primes is not infinite, hence it is finite, so it can be written as $\{p_1, p_2, ..., p_n\}$. It results that there exists the largest prime number, say $p_n$. Consider the natural number $P = p_1 p_2 ... p_n + 1$ (the product of

all prime numbers plus 1); then $P$ is a prime number and it is larger than $p_n$, contradiction.

**4. Proposition**

(i) Let $a, b \in \mathbf{Z}$ and $p$ a prime number. If $p|\ ab$ then $p|\ a$ or $p|\ b$.

(ii) Generalization: let $a_1, a_2, ..., a_k \in \mathbf{Z}$ and $p \in \mathbf{N}$ be a prime. If $p \mid a_1 a_2 ... a_k$ then $\exists i \in \{1, 2, ..., k\}$ such that $p \mid a_i$.

**Proof**

(i) If $a = 0$ or $b = 0$, the result is obvious. Assume that $a, b \in \mathbf{N}^\star$; if $p \not| a$, then we consider the set

$$S = \{b \in \mathbf{N}\ ;\ p \mid ab \text{ and } p \not| b\}$$

We shall prove by contradiction that $S = \emptyset$; if $S \neq \emptyset$. From the well-ordering property we get that $S$ has a smallest element, say $c$. It results that $p|ac$ and $p \not| c$; since $p \not|a$, then necessarily $c > 1$. It can be proved (exercise) that $c < p$. Finally we've got $1 < c < p$; by applying theorem 1 there exist $q, r \in \mathbf{Z}$ such that $p = cq + r$ and $0 \leq r < c$. Since $p$ is a prime number, it results that $r \geq 1$, so $1 \leq r < c$. We have:

$$ar = a(p - cq),$$

hence $p \mid ar$. In conclusion: $p \mid ar$ and $p \not| r$ But $r < c$ and $r \in \mathbf{N}$, contradicting that $c$ is the smallest element of $S$.

**5. Fundamental Theorem of Arithmetic**

Every natural $n \geq 2$ can be written as a product of primes and this factorization is unique up to the order of the prime factors.

**Proof**

We first prove the existence by induction. If $n = 2$ the result is clear. Assume now that $n > 2$ and that every $m \in \mathbf{N}$, $2 \leq m \leq n - 1$ can be written as a product of primes. We shall prove that $n$ can be written as a product of primes. If $n$ is a prime, the result is obvious. If $n$ is not a prime, then there exist $n_1,\ n_2 \in \mathbf{N}$ such that:

$$n = n_1 n_2 \ \text{ and } \ 2 \leq n_1 \leq n - 1,\ \ 2 \leq n_2 \leq n - 1$$

By the hypothesis, both $n_1$ and $n_2$ can be written as products of primes, so $n$ can be written as a product of primes; this proves the existence. To prove

the uniqueness, let us suppose that:

$$n = p_1 p_2 ... p_r = p'_1 p'_2 ... p'_s,$$

where $p_1 \leq p_2 \leq ... \leq p_r$ and $p'_1 \leq p'_2 \leq ... \leq p'_s$ are all prime numbers. Since $p_1 | \, n = p'_1 p'_2 ... p'_s$, by applying proposition 3 (ii), we get that $\exists j \in \{1, 2, ..., s\}$ such that

$$p_1 | \, p'_j$$

Since $p_1$ and $p'_j$ are both primes, it results $p_1 = p'_j$. Analogously, $p'_1 | \, p_1 p_2 ... p_r$, so $\exists i \in \{1, 2, ..., r\}$ such that $p'_1 | \, p_i$, hence $p'_1 = p_i$. We have:

$$p_1 = p'_j \geq p'_1 = p_i \geq p_1,$$

so $p_1 = p'_1$. We now continue reasoning as above starting from the equality:

$$p_2 p_3 ... p_r = p'_2 p'_3 ... p'_s$$

We finally get $r = s$ and $p_i = p'_i$, $\forall i \in \{1, 2, 3, ..., r\}$.

Grouping together the equal primes in the above factorization, we get:

**6. Corollary**

Let $n \in \mathbf{N}$, $n \geq 2$. Then there exist primes $p_1 < p_2 < ... < p_r$ and $m_1, m_2, .., m_r \in \mathbf{N}$ such that

$$n = p_1^{m_1} p_2^{m_2} ... p_r^{m_r}$$

Moreover, this factorization is unique.

**7. Greatest common divisor**

Suppose that $a, b \in \mathbf{N}$. Then there exists a unique $d \in \mathbf{N}$ a unique $d \in \mathbf{N}$ such that:

(a) $d \, | \, a$ and $d \, | \, b$

(b) if $x \in \mathbf{N}$ and $x \, | \, b$, then $x \, | \, d$.

Before proving, we recall that the number $d$ is called **the greatest common divisor** (g.c.d.) of the numbers $a$ and $b$ and is denoted by $d = (a, b)$. Two numbers $a$ and $b$ are called **coprime** (or relatively prime) if $(a, b) = 1$.

**Proof**

If $a = 1$ or $b = 1$ then obviously $d = 1$. If $a > 1$ and $b > 1$, then we consider the factorizations (as in corollary 5):

$$a = p_1^{m_1} p_2^{m_2} ... p_r^{m_r},\ b = p_1^{n_1} p_2^{n_2} ... p_r^{n_r}$$

Mention that if $p_j$ is not a prime factor of $a$ (or $b$) then the corresponding exponent $m_j$ (or $n_j$) is 0. The greatest common divisor of $a$ and $b$ is:

$$d = \prod_{j=1}^{r} p_j^{\min\{m_j, n_j\}}$$

It's easy to check that $d \mid a$ and $d \mid b$; if $x \in \mathbf{N}$ and $x \mid a$ and $x \mid b$, then

$$x = p_1^{v_1} p_2^{v_2} ... p_r^{v_r}$$

with

$$0 \leq v_j \leq m_j \text{ and } 0 \leq v_j \leq m_j,\ \forall j = 1, 2, ..., r,$$

hence $x \mid d$. The uniqueness of $d$ results from the uniqueness of the factorizations of $a$ and $b$.

**8. The least common multiple**

Analogously one can prove that for every $a, b \in \mathbf{N}$ there exists a unique $m \in \mathbf{N}$ such that:

(a) $a \mid m$ and $b \mid m$

(b) if $x \in \mathbf{N}$ and $a \mid x$ and $b \mid x$ then $m \mid x$

The number $m$ is called **the least common multiple** (l.c.m)of $a$ and $b$ and is denoted by $[a, b]$.

**9. Proposition**

For every $a, b \in \mathbf{N}$, then there exist $x, y \in \mathbf{Z}$ such that

$$(a, b) = ax + by$$

**Proof**

The idea of the proof is to consider the set:

$$S = \{ax + by > 0\ ;\ x, y \in \mathbf{Z}\}$$

Then by the well-ordering property of $\mathbf{N}$ it results that $S$ has a least element, say $d$; it can be proved that $d$ is the g.c.d. of $a$ and $b$ (we leave the details of the proof to the reader).

The proof of the existence of the g.c.d. is not an easy method to compute it (at least for large numbers). One of the most famous (it goes back to ancient Greek mathematicians) algorithms in mathematics is

**10. Euclid's Algorithm**
Let $a, b \in \mathbf{N}$ such that $a > b$. Suppose that

$$q_1, q_2, ..., q_{n+1} \in \mathbf{Z}$$

are the quotients and

$$0 < r_n < r_{n-1} < ... < r_1 < b$$

are the remainders of the divisions:

$$a = bq_1 + r_1,$$

$$b = r_1q_2 + r_2,$$

$$r_1 = r_2q_3 + r_3,$$

.....

$$r_{n-2} = r_{n-1}q_n + r_n,$$

$$r_{n-1} = r_nq_{n+1}.$$

Then $(a, b) = r_n$.

**Proof**

The idea is to prove that if $q_1$ is the quotient and $r_1$ is the remainder of the division of $a$ at $b$, then the numbers $a, b$ and $b, r_1$ have the same greatest common divisor, i.e. $(a, b) = (b, r_1)$. Obviously

$$(a, b) \mid (a - bq_1) = r_1 \text{ and } (a, b) \mid b,$$

hence $(a, b) \mid (b, r_1)$. On the other hand,

$$(b, r_1) \mid (bq_1 + r_1) = a \text{ and } (b, r_1) \mid b,$$

so $(b, r_1) \mid (a, b)$. It results $(a, b) = (b, r_1)$. By the same method one can prove:

$$(b, r_1) = (r_1, r_2) = (r_2, r_3) = ... = (r_{n-1}, r_n)$$

The result follows from the equality:

$$(r_{n-1}, r_n) = (r_n q_{n+1}, r_n) = r_n$$

Based on the previous result, the algorithm for computing the g.c.d. of $a$ and $b$ has the following steps:

(i) If $a < b$ then we interchange $a$ and $b$.

(ii) If $a > 0$, divide $b$ by $a$ and get the remainder $r$; replace the number $b$ by $r$ and return to step (i).

(iii) Else (if $a = 0$), then $b$ is the g.c.d. and stop.

An important question (for all algorithms) is how long it takes? More precisely, how many steps it takes before it stops. Of course, the Euclidian algorithm depends on the magnitude of the numbers $a$ and $b$, but not only. In fact in can easily tested that the number of iterations is less than the sum $a + b$. In the following we give few examples.

**11. Examples**

(i) For two consecutive numbers, the Euclidian algorithm stops in two steps.

(ii) By using two consecutive Fibonacci numbers, show that the Euclidian algorithm can last arbitrarily many steps.

**Proof**

(i) $(a, a + 1) = (a, 1) = (0, 1) = 1$

(ii) Let $(F_k)_{k \in N}$ be Fibonacci's sequence; the remainder of $F_{k+1}$ divided by $F_k$ is $F_{k-1}$, since $F_{k+1} = F_k + F_{k-1}$. It results:

$$(F_{n+1}, F_n) = (F_n, F_{n-1}) = ... = (F_3, F_2) = 1$$

The number of steps is $n - 1$.

**12. Theorem (the length of the Euclidian algorithm)**
The number of steps of the Euclidian algorithm applied to $a, b \in N^\star$ is at most $\log_2 a + \log_2 b$.

**Proof**
The key idea of the proof is the following

**Lemma**
In the euclidian algorithm, in every iteration the product of the two current numbers is reduced by a factor of 2 (at least). We first prove the lemma. Let $a, b \in N$, $0 < b < a$ and let $r$ be the remainder of the division of $a$ at $b$. At every step, the pair $a, b$ is replaced by the pair $r, b$. We have:

$$a = bq + r \geq b + r > 2r,$$

since $b > r$. It results $br < \frac{1}{2}ab$ and the lemma is proved.

If we apply the euclidian algorithm to $a, b$, after $k$ steps, the product of the two current numbers is at most $2^{-k}ab$. This product is at least 1, so $2^k \leq ab$. By applying logarithms, we get:

$$k \leq \log_2(ab) = \log_2 a + \log_2 b$$

**13. Example**
The last estimation is much better than our original estimation of the number of iterations of the euclidian algorithm (the sum $a + b$ was replaced by the sum $\log_2 a + \log_2 b$). For example if we consider the problem of computing the g.c.d. of two integers of 100 digits, then theorem 12 gives for the number of steps the estimation $k \leq 2 \log_2 10^{100} = 200 \log_2 10 < 720$.

**14. Exercise**
Let $0 < b < a$ be two integers such that the euclidian algorithm applied to them takes $n$ steps. Prove that $a \geq F_{n+1}$ and $b \geq F_n$.

**Proof**
By induction; if $n = 1$, then $a \geq 2$ and $b \geq 1$ is obviously true. We suppose that the assertion is true for every $1 \leq k \leq n-1$ and we prove it for $n$. Suppose that the euclidian algorithm for computing $(a, b)$ takes $n$ steps. Let $a = bq + r$ ($q$ is the quotient and $r$ is the remainder); then $1 \leq r < b$. The euclidian algorithm for computing $(b, r)$ takes $n - 1$ steps, so (according

to the induction hypothesis) it results

$$b \geq F_n \text{ and } r \geq F_{n-1}$$

We get:

$$a = bq + r \geq b + r \geq F_n + F_{n-1} = F_{n+1},$$

and the proof is over.

We continue with few basic results on prime numbers. An elementary (but useful) property is:

**15. Fermat's (little) Theorem**
Lat $a \in \mathbf{Z}$ and let $p$ be a prime; then:

$$p \mid a^p - a$$

In other terms, the integers $a$ and $a^p$ give the same remainder when divided by $p$.

**Proof**

Let $a$ and $p$ be as in the statement; we first prove the following observation:

$$p \mid \left( \begin{array}{c} p \\ k \end{array} \right), \ \forall k \in \{2, 3, ..., p-1\}$$

According to the definition of $\left( \begin{array}{c} p \\ k \end{array} \right)$, we have:

$$\left( \begin{array}{c} p \\ k \end{array} \right) = \frac{p(p-1)....(p-k+1)}{k(k-1)....1}$$

The prime $p$ divides the numerator but not the denominator (all the factors are smaller than $p$ and $p$ is prime) so the observation is clear.

We now prove Fermat's little theorem by induction on $a \in \mathbf{N}$. If $a = 0$ it is clearly true. Let $a > 0$ and let $b \in \mathbf{N}$ such that $a = b + 1$. We have:

$$a^p - a = (b+1)^p - (b+1) =$$

$$= b^p + \left( \begin{array}{c} p \\ 1 \end{array} \right) b^{p-1} + ... + \left( \begin{array}{c} p \\ p-1 \end{array} \right) b + 1 - b - 1 =$$

$$= (b^p - b) + \begin{pmatrix} p \\ 1 \end{pmatrix} b^{p-1} + ... + \begin{pmatrix} p \\ p-1 \end{pmatrix} b$$

The number $(b^p - b)$ is smaller than $a$, so according to the induction hypothesis we have: $(b^p - b) \mid p$ ; all the other terms of the sum are divisible by $p$ by the above observation. This concludes the proof.

The previous theorem is usually called "little" because Fermat is famous for his "last theorem":

**Fermat's last theorem**

Let $n \in \mathbf{N}$, $n \geq 3$. Then the sum of the $n$-th powers of two positive integers is not the $n$-th power of a positive integer. The statement was formulated in the 17-th century by Fermat, while the proof was found in 1995 by Andrew Wiles.

A natural (but difficult) problem about prime numbers is how are they distributed. More precisely, one can ask, for example, how many primes are between two natural numbers. It is simple to observe that the "gaps" of primes between two naturals are larger as we consider larger naturals. A result in this direction is the following.

16. **Proposition**

For every natural number $n$ there exist $n-1$ consecutive composite naturals.

**Proof**

Let $n \in \mathbf{N}$; then the numbers:

$$n! + 2, n! + 3, n! + 4, ..., n! + n$$

are all composite. The first is divisible by 2, the second by 3, etc.

An important question about primes is how many primes are there up to a given number $n$ ? The usual notation for the number of primes up to $n$ is $\pi(n)$. It is accepted that an exact formula for $\pi(n)$ is impossible to get. The basic result about $\pi(n)$ is:

**17. The prime number theorem**

Let $n \in \mathbf{N}, n \geq 2$ and let $\pi(n)$ be the number of primes between 1 and $n$;

then:

$$\lim_{n\to\infty} \frac{\pi(n)}{\frac{n}{\ln n}} = 1$$

Loosely speaking, $\pi(n)$ is arbitrarily close to $\frac{n}{\ln n}$ if $n$ is sufficiently large.

The proof is extremely difficult (the result was conjectured in the 18-th century and the proof was given at the end of the 19-th century); instead, we illustrate it to answer the following question:

**18. Example**

How many primes (approximatively !) with 200 digits are there?

Obviously, the answer is $\pi(10^{200}) - \pi(10^{199})$. According to the prime number theorem we have:

$$\pi(10^{200}) - \pi(10^{199}) \approx \frac{10^{200}}{\ln(10^{200})} - \frac{10^{199}}{\ln(10^{199})} =$$

$$= \frac{10^{200}}{200 \ln 10} - \frac{10^{199}}{199 \ln 10} \approx 1.95 \cdot 10^{197}.$$

**Testing for prime numbers**

An important question in number theory (with crucial applications to modern cryptography) is how can one decide if a natural number is a prime? We end this section with some results in this direction.

If $n \in N$ then one can test if $n$ is prime by testing if it is divisible (or not) by any natural $k, 2 \leq k < n$. Of course, for large numbers (say, more than 20 digits) this procedure is totally inefficient. A small improvement is given by the following observation (the proof is left to the reader):

**19. Observation**

If $n \in \mathbf{N}$ is composite, then $n$ has a prime divisor less or equal than $\sqrt{n}$.

However, the above observation does not solve the problem: the method is still too slow for large numbers.

Better results (to prove that a number is not a prime) can be obtained by using Fermat's little theorem: if $p$ is a prime, then $p \mid a^p - a,\ \forall a \in N$.

For instance (for an odd $n$), by taking $a = 2$, if $n \not| 2^{n-1} - 1$, then $n$ is not a prime; below we have some examples:

(i) $9 \not| 2^8 - 1 = 255$

(ii) $15 \not| 2^{14} - 1 = 16383$

(iii) $21 \not| 2^{20} - 1 = 1048575$

(iv) $25 \not| 2^{24} - 1 = 16777215$

The major problems of this method are listed below.

(i) The computation of $2^{n-1}$ (for large $n$).

(ii) Testing if $n \not| 2^{n-1} - 1$ is complicated for large $n$.

(iii) If $n \mid 2^{n-1} - 1$ it results nothing about $n$ simply because the converse of Fermat's little theorem is not true (a counterexample is $341 = 11 \cdot 31$); moreover, there exist numbers $n \in \mathbf{N}$ such that $n \mid a^n - a$, $\forall a \in N$ but $n$ is not a prime. These numbers are called Carmichael numbers; an example of a Carmichael number is $561 = 3 \cdot 11 \cdot 17$ and $561 \mid a^{561} - a$, $\forall a \in \mathbf{Z}$.

In the next we discuss how the above problems (i), (ii) and (iii) can be partially solved.

**20. Observation**

For the computation of the powers of 2, one can repeat squaring starting with a small number, as in the following examples (we compute $2^{24}$ and $2^{29}$):

(a) for computing $2^{24}$:

$2^3 = 8$

$2^6 = (2^3)^2 = 64$

$2^{12} = (2^6)^2 = 4096$

$2^{24} = (2^{12})^2 = 16777216$

We reduced the number of operations from 23 to 5.

(b) the problem is that 29 is no more divisible by a (large) power of 2; however, we can reduce the number of the operations in a similar way:

$2^2 = 4$

$2^3 = 2^2 \cdot 2 = 8$

$2^6 = (2^3)^2 = 64$

$2^7 = 2^6 \cdot 2 = 128$

$2^{14} = (2^7)^2 = 16384$

$2^{28} = (2^{14})^2 = 268435456$

$2^{29} = 2^{28} \cdot 2 = 536870912$

The idea is to compute an odd power by multiplying the previous power by 2 and to compute an even power by squaring an appropriate smaller power. The general result is:

**21. Proposition**
Let $n \in \mathbf{N}$; if $n$ has $k$ digits in base 2, then $2^n$ can be computed by using at most $2k$ multiplications.
**Proof**
We prove it by induction on $k$; if $k = 1$, clear. If $n \geq 2$, has $k+1$ digits in base 2, then we write $n = 2q + r$, where $r$ is the remainder, so $r \in \{0, 1\}$. It results that $q$ has $k$ digits in base 2, so we can compute $2^q$ by using at most $2k$ multiplications; since $2^n = (2^q)^2 \cdot 2^r$, we finally get at most $2k + 1 + 1 = 2(k + 1)$ multiplications.

We now discuss the second problem: testing $n \mid 2^{n-1} - 1$; for large $n$, the number $2^{n-1} - 1$ is too large, so one cannot test the divisibility by $n$. However there exist a method to replace $2^{n-1} - 1$ with a smaller number (less than $n^2$). The idea is to replace a power $2^k$ larger than $n$ by the remainder of the division of $2^k$ by $n$. The reason why this method holds is the following elementary observation.

**22. Observation**
$n \mid 2^{n-1} - 1$ if and only if $n \mid 2r - 1$, where $r$ is the remainder of the division of $2^{n-2}$ by $n$.

**23. Example**
Let us test if $25 \mid 2^{24} - 1$; we need (as above) to compute $2^{24}$. We start to compute the powers of 2 and we test each time if the current power is less less than 25; if yes, we continue by computing the next power. If not, we divide it by 25 and we continue with the powers of the remainder.
$2^3 = 8 < 25$
$2^6 = 64 > 25$
$64 = 25 \cdot 2 + 14$
Now we need to compute $2^{12} = (2^6)^2$, but instead we compute $14^2 = 196 > 25$
$196 = 25 \cdot 7 + 21$
Finally we need to compute $2^{24} = (2^{12})^2$; instead, we compute
$21^2 = 441 > 25$.

$441 = 25 \cdot 17 + 16$

Now instead testing $25 \mid 2^{24} - 1$, we test if $25 \mid 16 - 1$; since this is not true, we get that 25 is not a prime.

We now discuss the third question: what if $n \mid a^n - 1, \forall a \in N$ ? We already have an example of a Carmichael number: 561. An improvement of the test based on Fermat's little theorem is the following:

**24. The Miller-Rabin test**

Let $n > 1$ be an odd natural number. Suppose we want to test if $n$ is a prime. Let $a \in \mathbf{N}$, $0 \leq a \leq n - 1$. If $n \mid a^n - a$, then we factor $a^n - a = a(a^{n-1} - 1) = ....$ as long as we can by using usual identities like $x^2 - 1 = (x-1)(x+1)$, $x^3 - 1 = (x-1)(x^2+x+1)$, etc. If $n$ is a prime, then it must divides at least one the factors (for all $a$); if this is not true, then $n$ is not a prime number. Of course, if the test fails ($n$ divides one factor), then nothing can be said. However, it can be proved that the test fails with a probability of 0.5. So, if we repeat the test 10 times (for 10 different values of $a$) and each time the test fails ($n$ divides one factor), then the probability for $n$ not to be a prime is about $2^{-10}$.

**25. Example**

We apply the Miller-Rabin test for 561. We factor $a^{561} - a$:

$$a^{561} - a = a(a^{560} - 1) =$$

$$= a(a^{280} - 1)(a^{280} + 1) =$$

$$= a(a^{140} - 1)(a^{140} + 1)(a^{280} + 1) =$$

$$= a(a^{70} - 1)(a^{70} + 1)(a^{140} + 1)(a^{280} + 1) =$$

$$= a(a^{35} - 1)(a^{35} + 1)(a^{70} + 1)(a^{140} + 1)(a^{280} + 1)$$

If 561 would be a prime number, then according to Fermat's little theorem, it should divide $a^{561} - a$, $\forall a \in \mathbf{N}$ (this is in fact true); it results (561 was supposed to be prime) that 561 must divide at least one of the factors; but for $a = 2$ none of the factors is a multiple of 561, so 561 is not a prime number.

# Chapter 2

# Relations and Functions

## 2.1 Introduction

**1.Definition**
Let $M$ be a non empty set and let $M \times M = \{(x, y) \mid x, y \in M\}$ be the cartesian product. A **relation** on the set $M$ is every subset $\xi \subseteq M \times M$. If $(x, y) \in \xi$, we denote $x \xi\, y$ and we say that "$x$ is in the relation $\xi$ with $y$". If $(x, y) \notin \xi$, then we denote $x \not\xi\, y$.

**2.Examples**
**i.** Let $M = Z$ be the set of integers; for every $m, k \in Z$ let $m \rho\, k \Leftrightarrow m \mid k$ , ("$m$ divide $k$", or "$k$ is a multiple of $m$"); by definition, we put $0 \rho\ 0$. For example, $2 \mid 4$ and $3 \nmid -4$.
**ii.** Let $R$ be the set of real numbers and let $x \rho\, y \Leftrightarrow x \leq y$ (the relation "less or equal").
**iii.** Let $n \in N^{\star}$ be a fixed natural number. The relation "congruence modulo $n$" is defined on $Z$ as $x \equiv_n y \Leftrightarrow n \mid x - y, \forall x, y \in Z$.
**iv.** The relation of equality can be defined on any nonempty set $M$ : $x \rho\, y \Leftrightarrow x = y$.
**v.** The universal relation can be defined on any non empty set $M$:
$x \rho\ y,\ \forall x, y \in M$.
**vi.** On the set of all students in the University, one can define the relation $x \rho\, y \Leftrightarrow$ the students $x$ and $y$ are in the same group.

**3.Observation**
Let $M \neq \emptyset$; it is possible to regard functions (mappings) from $M$ to $M$ as a special case of relations.
Let $(M, \rho)$ be a relation such that for every $x \in M$ there is exactly one element $y \in M$ such that $x \rho\, y$. Such a relation is in fact a function $f_\rho : M \mapsto M$ defined by

$$f_\rho(x) = y \Leftrightarrow x \rho\, y.$$

**4.Definitions**
Let $(M, \rho)$ be a relation.
**i.** $\rho$ is **reflexive** iff $x \rho\, x,\ \forall x \in M$.
**ii.** $\rho$ is **symmetric** iff $x \rho\, y \Rightarrow y \rho\, x,\ \forall x, y \in M$.
**iii.** $\rho$ is **antisymmetric** iff $x \rho\, y$ and $y \rho\, x\ \Rightarrow x = y,\ \forall x, y \in M$.
**iv.** $\rho$ is **transitive** iff $x \rho\, y$ and $y \rho\, z\ \Rightarrow x \rho\, z$.

**5. Exercise**
Check the above properties for the relations defined in Example 2.

**6. Operations with relations**
Let $M \neq \emptyset$ and let $\rho$ and $\phi$ be two relations on $M$.
The **union** of $\rho$ and $\phi$ is defined as

$$\rho \cup \phi = \{(x, y)\ ;\ (x, y) \in \rho \text{ or } (x, y) \in \phi\},$$

i.e., $x \rho \cup \phi\, y \Leftrightarrow$ at least one of the relations $x \rho\, y\ ,\ x \phi\, y$ holds.
Analogously, the **intersection** of $\rho$ and $\phi$ is defined by

$$x \rho \cap \phi\, y \ \Leftrightarrow\ \text{both}\ \ x \rho\, y\ \ \text{and}\ x \phi\, y \text{ hold.}$$

The **inverse** of the relation $\rho$, denoted by $\rho^{-1}$ is:

$$x \rho^{-1}\, y \ \Leftrightarrow\ y \rho\, x.$$

The **product** of the relations $\rho$ and $\phi$, denoted by $\rho\phi$ is defined by:

$$x \rho \phi\, y \Leftrightarrow\ \exists z \in M \text{ such that } x \rho\, z \text{ and } z \phi\, y \text{ hold.}$$

A special case is when $\rho = \phi$; the relation $\rho^2$ is:

$$x \rho^2\, y \ \Leftrightarrow\ \exists z \in M \text{ such that } x \rho\, z \text{ and } z \rho\, y.$$

By recurrence, one can define the $n$-th power of the relation $\rho$ by:

$$x\rho^n\, y \;\Leftrightarrow\; \exists z_0 = x, z_1, z_2, ..., z_n = y \text{ elements in } M \text{ such that :}$$

$$z_0\rho\, z_1,\;\; z_1\rho\, z_2, ..., z_{n-1}\rho z_n.$$

The **transitive closure** of the relation $\rho$ is denoted by $\overline{\rho}$ and is defined in the following way:

$$x\overline{\rho}\, y \;\Leftrightarrow\; \exists n \in N \text{ and elements } \; z_0 = x, z_1, z_2, ..., z_n = y \text{ such that :}$$

$$z_0\rho\, z_1,\;\; z_1\rho\, z_2, ..., z_{n-1}\rho z_n.$$

In fact, by using the powers of $\rho$ and the union, the transitive closure can be written in the form:

$$\overline{\rho} = \rho \cup\; \rho^2 \cup\; \rho^3 \cup\; ... \cup\; \rho^n \;\cup ....$$

**7. The matrix associated to a relation**

Let $M$ be a non empty **finite** set with $n$ elements, $M = \{x_1, x_2, ..., x_n\}$. If $\xi$ is a relation on $M$, the **matrix associated to** $\xi$, is, by definition the $n \times n$ matrix $A_\xi = (a_{ij})_{ij}$, with $a_{ij} = \begin{cases} 1 & \text{if} \quad x_i \xi\, x_j \\ 0 & \text{if} \quad x_i \not\xi\, x_j \end{cases}$. The matrices associated to relations consist only of zeros and ones. We define the following operations (so called Boolean) on the set $\{0, 1\}$:

$$0 \vee 0 = 0, 0 \vee 1 = 1, 1 \vee 0 = 1, 1 \vee 1 = 1$$

$$0 \cdot 0 = 0\ ,\ 0 \cdot 1 = 0\ ,\ 1 \cdot 0 = 0\ ,\ 1 \cdot 1 = 1.$$

We can now investigate the operations on matrices corresponding to the operations on relations. Let $\rho$ and $\phi$ be two relations on $M$ with the associated matrices $M_\rho = (a_{ij})$ and $M_\phi = (b_{ij})$.
Then the matrix of the union $\rho \cup \phi$ is $M_{\rho\cup\phi} = (a_{ij} \vee b_{ij})$ and the matrix of the intersection $\rho \cap \phi$ is $M_{\rho\cap\phi} = (a_{ij} \cdot b_{ij})$.
The matrix of the product $\rho\phi$ is $M_{\rho\phi} = (a_{i1} \cdot b_{1j} \vee a_{i2} \cdot b_{2j} \vee ... \vee a_{in} \cdot b_{nj})$.
The matrix of the inverse $\rho^{-1}$ is the transpose $M_\rho^T = (a_{ji})$.
The matrix of the transitive closure of $\rho$ is the matrix of the union of all the powers of of $\rho$.

**8. Exercise**
Let $M = \{1, 2, 3\}$ and the relations (on $M$):

$$x \rho\, y \Leftrightarrow x \text{ divide } y$$

$$x \phi\, y \Leftrightarrow x = y + 1.$$

Find the matrices of the relations $\rho$, $\phi$, $\rho \cup \phi$, $\rho \cap \phi$, $\rho \cdot \phi$, $\rho^n$, $\phi^n$, $\overline{\rho}$, $\overline{\phi}$.

## 2.2 Relations of equivalence

**9. Definition.**
A relation $(M, \sim)$ is a **relation of equivalence** on $M$ iff it is reflexive, symmetric and transitive.

**10. Examples**
**i.** The equality is a relation of equivalence.
**ii.** The universal relation is a relation of equivalence.
**iii.** The congruence mod $n$ is a relation of equivalence.
**iv.** On the set of integers $Z$ let $m \sim k$ iff $m \mid k$ and $k \mid m$; then $\sim$ is an equivalence on $Z$.
**v.** Let $f : M \mapsto R$ an arbitrary function and let $x \sim_f y \Leftrightarrow f(x) = f(y)$. Then $\sim_f$ is an equivalence on $M$. Obviously, $\sim_f$ is the equality iff $f$ is injective and $\sim_f$ is the universal relation iff $f$ is constant.

**11. Definition**
Let $(M, \sim)$ be a relation of equivalence; for every $x \in M$ we define **the class of equivalence of** $x$ as $\widehat{x} = \{y \in M \mid y \sim x\}$. The set $\widehat{M} = \{\widehat{x} \mid x \in M\}$ is called the set of classes of equivalence, or the factor set.

**12. Proposition**
Let $(M, \sim)$ be a relation of equivalence; then:
**a.** $\widehat{x} \neq \emptyset, \forall x \in M$.
**b.** Let $x, y \in M$; if $x \sim y$ then $\widehat{x} = \widehat{y}$ and if $x \not\sim y$ then $\widehat{x} \cap \widehat{y} = \emptyset$.
**c.** $\bigcup_{x \in M} \widehat{x} = M$.
**Proof a.** $x \in \widehat{x}, \forall x \in M$ (by using the reflexivity).
**b.** Let $x, y \in M$ such that $x \sim y$; we prove that $\widehat{x} = \widehat{y}$ by double inclusion; first, $\widehat{x} \subseteq \widehat{y}$. Let $z \in \widehat{x}$; then $z \sim x$ and $x \sim y$ implies $z \sim y$ (by the

transitivity), hence $z \in \widehat{y}$. Analogously, $\widehat{y} \subseteq \widehat{x}$.
Let us now suppose that $x \not\sim y$ and that $\widehat{x} \cap \widehat{y} \neq \emptyset$. Let $t \in \widehat{x} \cap \widehat{y}$; then $t \sim x$ and $t \sim y$, hence $x \sim y$, contradiction.
**c.** Obviously, because $x \in \widehat{x}, \forall x \in M$.

**13. Definition**
Let $M \neq \emptyset$; a **partition** on $M$ is a family $\mathcal{A} = (A_i)_{i \in J}$ such that:
**a.** $A_i \subseteq M, \forall i \in J$.
**b.** $A_i \cap A_j = \emptyset, \forall i \neq j$.
**c.** $\bigcup_{i \in J} A_i = M$.

**14. Theorem**
**a.** Let $(M, \sim)$ be a relation of equivalence. Then the family $\widehat{M} = (\widehat{x})_{x \in M}$ is a partition on $M$.
**b.** Let $\mathcal{A} = (A_i)_{i \in J}$ be a partition on a non empty set $M$. Then the relation $x \sim_{\mathcal{A}} y \Leftrightarrow \exists i \in J$ such that $x, y \in A_i$ is a relation of equivalence on $M$ and the associated set $\widehat{M}$ of classes of equivalence is the partition $\mathcal{A}$.
**Proof a.** The fact that the family of classes of equivalence is a partition was proved in Proposition 7.
**b.** The fact that $\sim_{\mathcal{A}}$ is a relation of equivalence is obvious. Let now $x \in M$; then there is $i \in J$ such that $x \in A_i$. We claim that $\widehat{x} = A_i$. If $y \in \widehat{x}$, then $y \sim_{\mathcal{A}} x$; by the definition of $\sim_{\mathcal{A}}$ it results $y \in A_i$. If $y \in A_i$, then $y \sim_{\mathcal{A}} x$, hence $y \in \widehat{x}$.

**15. Exercise**
Prove that the two constructions of the above Theorem are inverse one to each other, i.e. given a relation of equivalence $(M, \sim)$ we define (as in **a**) the partition $\widehat{M}$; we now associate to this partition the relation of equivalence $\sim_{\widehat{M}}$ (as in **b**). It results that $\sim = \sim_{\widehat{M}}$. Conversely, we start with a partition $\mathcal{A}$ and we define the relation of equivalence $\sim_{\mathcal{A}}$ (as in **b**). We now associate to $\sim_{\mathcal{A}}$ the partition $\widehat{M}$, defined by its classes of equivalence (as in **a**). It results that $\mathcal{A} = \widehat{M}$.

**16. Examples**
**i.** If the relation is the equality, then each class of equivalence contains one element: $\widehat{x} = \{x\}$.
**ii.** For the universal relation, there is only one class of equivalence.

**iii.** If $M = Z$ and $m \sim\ k \Leftrightarrow m|\ k$ and $k|\ m$, then $\widehat{0} = \{0\}$ (by definition) and $\widehat{m} = \{-m, m\}, \forall m \neq 0$.
**iv.** Let $n \in N^\star$ and let $(Z, \equiv_n)$ be the congruence (mod $n$). If $k \in Z$, then $\widehat{k} = \{m \in Z\ ;\ n \,|\, m - k\}$. There are $n$ classes of equivalence, $\widehat{0} = \{mn;\ m \in Z\}$, $\widehat{1} = \{mn + 1\ ;\ m \in Z\}$, $\widehat{2} = \{mn + 2\ ;\ m \in Z\}, ..., \widehat{n-1} = \{mn + (n-1)\ ;\ m \in Z\}$. The set of classes of equivalence is denoted (in this case) by $Z_n$.
**v.** Let $T = \{z \in C\ ;\ |z| = 1\}$ be the unit circle and let $n \in N$, $n \geq 2$ be a fixed natural number. Let $U_n = \{z \in T\ ;\ z^n = 1\}$. On $T$ we define the relation $z \sim_n w \Leftrightarrow zw^{-1} \in U_n$, $\forall z, w \in T$. Prove that $\sim_n$ is an equivalence on $T$ and the class of equivalence of an element $z \in T$ is $\widehat{z} = \{zt\ ;\ t \in T\}$.

**17. Exercise**
**(a)** Let $f : M \mapsto R$ an arbitrary function and let $x \sim_f\ y \Leftrightarrow f(x) = f(y)$. If $x \in M$, then $\widehat{x} = \{y \in M\ ;\ f(x) = f(y)\}$. Let $\widehat{M}$ be the factor set and let $g : \widehat{M} \mapsto R$, $g(\widehat{x}) = f(x)$. Then the map $g$ is well defined and injective.
**Solution** To prove that $g$ is well defined we have to prove that if $\widehat{x} = \widehat{y}$, then $g(\widehat{x}) = (\widehat{y})$, i.e. if $x \sim_f\ y$, then $f(x) = f(y)$; this is a direct consequence of the definition of $\sim_f$.
Let now suppose that $g(\widehat{x}) = g(\widehat{y})$. Then, by the definition of $g$ it results that $f(x) = f(y)$, hence $x \sim_f\ y$; it results that $\widehat{x} = \widehat{y}$, hence $g$ is injective.

## 2.3 Relations of order

**18. Definition**
A relation $(M, \rho)$ is called a **relation of order** iff it is reflexive, antisymmetric and transitive.
The relation of order is **total**, or the set $M$ is **totally ordered** by the relation $\rho$ iff $\forall x, y \in M\ \Rightarrow\ x\rho\ y$ or $y\rho\ x$.

**19. Examples**
**i.** The equality is a total order; the trivial relation is not antisymmetric.
**ii.** The usual sets of numbers $N, Z, Q, R$ are totally ordered by the usual relation $\leq$ ("less or equal").
**iii.** On the set of natural numbers, $N$, let $m\rho\ n \Leftrightarrow m|\ n$ and by definition $0\rho\ 0$. Then $\rho$ is a relation of order but it is not total.

**iv.** On the set of integers, $Z$, the above relation ("divide") is no more antisymmetric.
**iv.** Let $f : M \mapsto R$ and let $x\rho_f y \Leftrightarrow f(x) \leq f(y)$. The relation $\rho_f$ is reflexive and transitive; it is also antisymmetric iff $f$ is injective. If this is true, then $\rho_f$ is a total order on $M$.
**v.** Let $X \neq \emptyset$ and let $\mathcal{P}(X) = \{A \ ; \ A \subseteq X\}$. We consider on $\mathcal{P}(X)$ the usual relation of inclusion: $A \subseteq B$. Then the set $(\mathcal{P}(X), \subseteq)$ is an ordered set. It is totally order iff $X$ has only one element.

**20. Definitions**
Let $(M, \leq)$ be an ordered set and let $A \subseteq M$.
**a.** An element $x \in M$ is called an **upper bound** of $A$ iff $a \leq x, \forall a \in A$.
**b.** An element $y \in M$ is called a **lower bound** of $A$ iff $y \leq a, \forall a \in A$.
**c.** The subset $A$ is called **upper bounded** iff the set of its upper bounds is not empty and it is called **lower bounded** iff the set of its lower bounds is not empty. The subset $A$ is called **bounded** if it is both upper bounded and lower bounded.
**d.** Let $A$ be an upper bounded subset. An element $\psi \in M$ is called **the least upper bound of** $A$ (denoted $\sup(A)$) if:
**i.** $\psi$ is an upper bound of $A$, i.e. $a \leq \psi$, $\forall a \in A$, and
**ii.** if $x \in M$ is an upper bound of $A$, then $\psi \leq x$.
**e.** Let $A$ be a lower bounded subset. An element $\phi \in M$ is called **the greatest lower bound of** $A$ (denoted $\inf(A)$) if:
**i.** $\phi$ is a lower bound of $A$, i.e. $\phi \leq a$, $\forall a \in A$ and
**ii.** If $y \in M$ is a lower bound of $A$, then $y \leq \phi$.
If $\sup(A) \in A$, then it is called the greatest element of $A$ and if $\inf(A) \in A$, then it is called the least element of $A$.

**21. Example**
Let $(Q, \leq)$ be the set of rational numbers with the usual order; then the subset $A = \{x \in Q \ ; \ x^2 < 2\}$ is upper bounded, but it has not a least upper bound. In the set $(R, \leq)$ of real numbers with the same order, the same set $A$ has a least upper bound, namely $\sqrt{2}$. Cantor axiom asserts that in the set of real numbers every upper bounded subset has an upper bound.

**22. Example**
Let $(N, |)$ be the set of natural numbers ordered by the relation "divide". Let $A = \{6, 8, 12\}$; then $A$ is bounded and $\sup(A) = 2$, $\inf(A) = 24$.

Generalize the above example.

**23. Exercise**
**i.** Let $X \neq \emptyset$ and let $(\mathcal{P}(X), \subseteq)$. If $\mathcal{A} = \{H, K\} \subseteq \mathcal{P}(X)$, then $\sup(\mathcal{A}) = H \cup K$ and $\inf(\mathcal{A}) = H \cap K$.
Generalize to an arbitrary subset $\mathcal{A} \subseteq \mathcal{P}(X)$.
**ii.** Let $A, B \in \mathcal{P}(X)$. Prove that:
$A = B \Leftrightarrow \forall G \in \mathcal{P}(X)\ A \cap G = B \cap G$ and $A \cup G = B \cup G$.
**iii.** Let $A, B \in \mathcal{P}(X)$. Prove that:
$A = B \Leftrightarrow \exists G \in \mathcal{P}(X)$ such that $A \cap G = B \cap G$ and $A \cup G = B \cup G$.

**24. Exercise**
Let $M \neq \emptyset$ and let $\rho$ be a reflexive and transitive relation on $M$. On $M$ we define the relation:

$$x \sim_\rho y \Leftrightarrow x \rho\, y \text{ and } y \rho\, x.$$

**i.** Prove that $\sim_\rho$ is an equivalence on $M$.
**ii.** Let $\widehat{M}$ be the factor set associated to the relation $\sim_\rho$. On $\widehat{M}$ we define the relation $\widehat{x} \leq \widehat{y} \Leftrightarrow x \rho\, y$. Prove that $\leq$ is well defined and $(\widehat{M}, \leq)$ is an ordered set.
**iii.** Apply the above construction to the following case: $(M, \rho) = (Z, |)$.

# Chapter 3

# Graphs

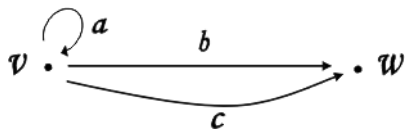
## 3.1 Directed Graphs

**1. Definitions**
A **directed graph (digraph)** is a 4-tuple $G = (V, A, i, t)$, where:
$V$ and $A$ are non empty sets; the elements of $V$ are called **vertices** and the elements of $A$ are called **arcs**.
$i$ and $t$ are two maps $i, t : A \mapsto V$; $i(a)$ is called **the initial** vertex of the arc $a$ and $t(a)$ is called **the terminal** vertex of the arc $a$.
Two vertices $v, w \in V$ are called **connected** (in $G$) if $\exists a \in A$ such that $i(a) = v$ and $t(a) = w$. We represent this property by the picture: $v \stackrel{a}{\longrightarrow} w$.
Two vertices are **adjacent** if $\exists a \in A$ such that $v \stackrel{a}{\longrightarrow} w$ or $w \stackrel{a}{\longrightarrow} v$.
Let $v \in V$. We denote by $\deg^+ v$ (input degree ) the number of arcs which "enter" in $v$ and by $\deg^- v$ (output degree ) the number of arcs which "go out" from $v$. The **degree** of $v$ is $\deg v = \deg^+ v + \deg^- v$.
A vertex $v \in V$ is called **initial** if $\deg^+ v = 0$ and it is called **terminal** if $\deg^- v = 0$.
A digraph is said to be with **simple links** iff for all vertices $v, w \in V$, there is at most one arc $a \in A$ such that $v \stackrel{a}{\longrightarrow} w$.
A digraph is **finite** iff it has a finite number of vertices and arcs.

**2. Remark**
In order to "visualize" a graph, we consider $V$ as a set of points and $A$ as a set of lines (arcs) connecting the vertices.

**3. Examples**

**i.** Let $V = \{v, w\}$, $A = \{a, b, c\}$ and the maps $i$ and $t$ defined by $i(a) = v$, $i(b) = v$, $i(c) = v, t(a) = v$, $t(b) = w$, $t(c) = w$.

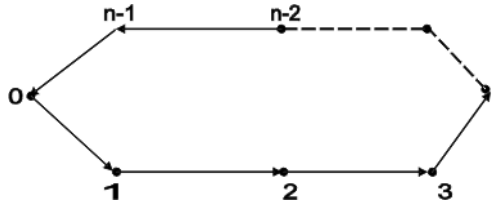



**ii.** Three typical graphs:

$G_\omega : \ 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow .....$

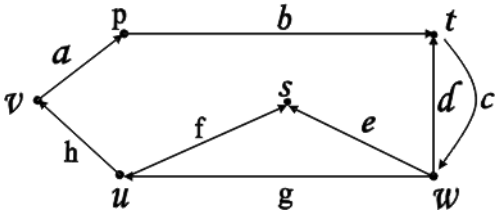
$P_n : \ 0 \longrightarrow 1 \longrightarrow 2 \longrightarrow ..... \longrightarrow (n-1) \longrightarrow n$

$Q_n :$




**4. Exercise**

**i.** On the following representation, find the digraph $G = (V, A, i, t)$.




**ii.** Represent the following digraph: $G = (V, A, i, t)$, where

$$V = \{x, y, z, u\}, A = \{a, b, c\}$$

$$i(a) = x, i(b) = y, i(c) = z, t(a) = x, t(b) = x, t(c) = u.$$

**iii.** The same question for $G = (V, A, i, t)$, where:

$$V = N, A = \{a_j\}_{j \in N},\ i(a_j) = j,\ t(a_j) = j + 2.$$

**5. Remark**
All type of networks (computers, water supply, telecommunications, transport, distribution) can be represented by digraphs.

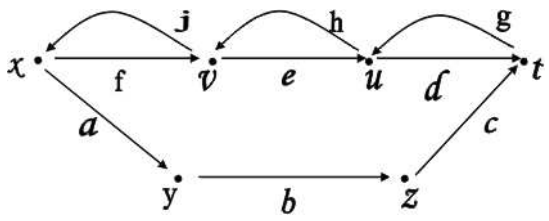
**6. Definition**
Two digraphs $G = (V, A, i, t)$ and $G' = (V', A', i', t')$ are said to be **isomorphic** if there are two bijective maps $h_A : A \mapsto A'$ and $h_V : V \mapsto V'$ such that: $h_V \circ i = i' \circ h_A$ and $h_V \circ t = t' \circ h_A$. The pair $h = (h_A, h_V)$ is called a graph isomorphism. If the maps $h_A$ and $h_V$ are no more bijective, the $h$ is simply a **morphism**. The idea of the definition of isomorphic digraphs is represented below:

$$v \xrightarrow{a} w \text{ in } G \Leftrightarrow h_V(v) \xrightarrow{h_A(a)} h_V(w) \text{ in } G'.$$

Let $G = (V, A, i, t)$ be a digraph and let $V' \subseteq V,\ A' \subseteq A$ such that $i(A') \subseteq V'$ and $t(A') \subseteq V'$. Let $i'$ and $t'$ be the restrictions of $i$ and $t$ to $A'$. Then the digraph $G' = (V', A', i', t')$ is called a **subgraph** of $G$.

**7. Example**
Let $G = (V, A, i, t)$ be a digraph and let $v \in V$ a fixed vertex and let:
$A' = \{a \in A\ ;\ i(a) = v \text{ or } t(a) = v\}$ and
$V' = \{w \in V\ ;\ \exists a \in A' \text{ such that } i(a) = w \text{ or } t(a) = w\}$.
Then $G' = (V', A', i', t')$ is the **subgraph generated by** $v$.

**8. Exercise**
In the following representation, find the subgraphs generated by every vertex.

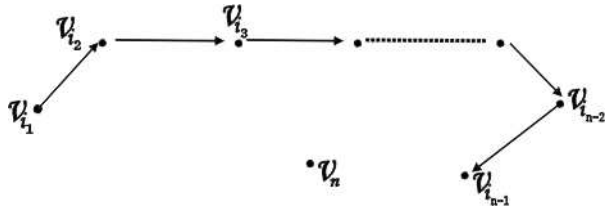


**9. Definition**
Let $G = (V, A, i, t)$ be a digraph. A **path** in $G$ is defined by two vertices, $v, w \in V$ and a finite number of arcs $a_1, a_2, ...a_n \in A$ such that there are $v_1, v_2, ..., v_{n-1} \in V$ with the properties: $i(a_1) = v$, $t(a_1) = v_1$, $i(a_2) = v_1$, $t(a_2) = v_2, ..., i(a_n) = v_{n-1}$, $t(a_n) = w$. The representation is:

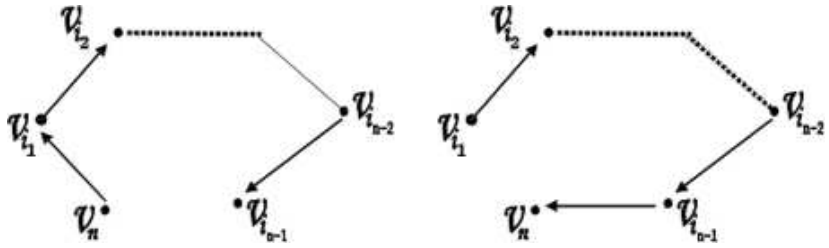
$$v \xrightarrow{a_1} v_1 \xrightarrow{a_2} v_2..... \longrightarrow v_{n-1} \xrightarrow{a_n} w$$

The natural number $n$ is called **the length** of the path. The vertices $v$ and $w$ are the **initial vertex** and **terminal vertex**, respectively, of the path. We can denote a path by its vertices: $vv_1v_2...v_{n-1}w$. Such a path **connects** the vertices $v$ to $w$.
If $v = w$, the path is called **closed (circuit)**. A closed path of length 1 is called a **curl**. Obviously, every path is a subgraph. In fact, to define a path in $G$ is equivalent to define a morphism from $P_n$ (or $Q_n$, if the path is closed), to $G$ (cf. Example 3).
The **null path** is defined by $n = 0$, i.e, it connects every vertex to itself and the length is 0.

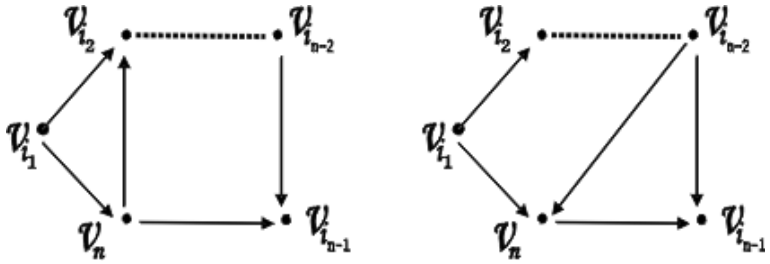
**10. Theorem**
Let $G = (V, A, i, t)$ be a finite digraph with simple links such that $\forall v, w \in V$. $\exists a \in A$ such that $v \xrightarrow{a} w$ or $w \xrightarrow{a} v$. Then there are paths which pass through all vertices one time only (such a path is called a Hamiltonian path).
**Proof**
Let $n \in N$ and let $V = \{v_1, v_2, ..., v_n\}$; we proceed by induction. If $n \in \{1, 2\}$, obviously. We suppose that the property is true for all $k \leq n - 1$ and we prove it for $n$. There is a Hamiltonian path connecting $n - 1$ vertices:




There are 2 possibilities:
**i.** $\exists v_n \longrightarrow v_{i_1}$ or $\exists v_{i_{n-1}} \longrightarrow v_n$. In any case we get a Hamiltonian path: $v_n \longrightarrow v_{i_1} \longrightarrow v_{i_2} \longrightarrow .... \longrightarrow v_{i_{n-1}}$ or $v_{i_1} \longrightarrow v_{i_2} \longrightarrow ... \longrightarrow v_{i_{n-1}} \longrightarrow v_{i_n}$, as in the picture:



**ii.** $\exists v_{i_1} \longrightarrow v_n$ and $\exists v_n \longrightarrow v_{i_{n-1}}$.
Again, there are 2 possibilities:
**i'.** $\exists v_n \longrightarrow v_{i_2}$ or $\exists v_{i_{n-2}} \longrightarrow v_n$. In any case we get a Hamiltonian path as follows:



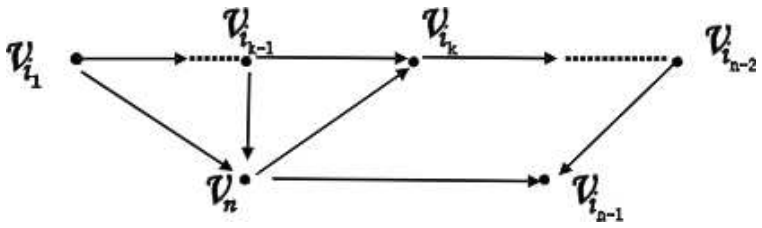
**ii'.** $\exists v_{i_2} \longrightarrow v_n$ and $v_n \longrightarrow v_{i_{n-2}}$; again, we have 2 possibilities....
Finally, we get that there are 2 vertices $v_{i_{k-1}}$ and $v_{i_k}$ such that:

$$v_{i_{k-1}} \longrightarrow v_n \text{ and } v_n \longrightarrow v_{i_k}.$$

The Hamiltonian path is:

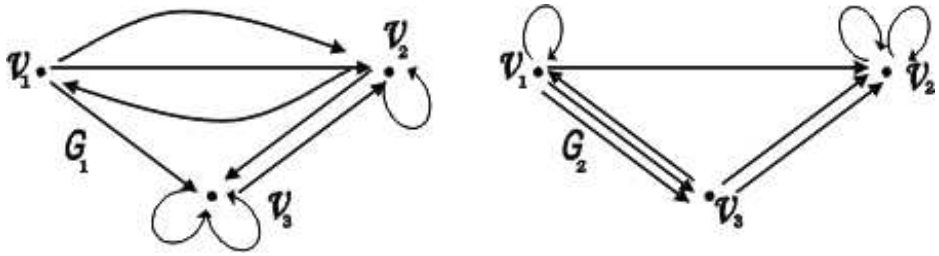
$$v_{i_1} \longrightarrow v_{i_2} \longrightarrow ... \longrightarrow v_{i_{k-1}} \longrightarrow v_n \longrightarrow v_{i_k} \longrightarrow v_{i_{k+1}} \longrightarrow ... \longrightarrow v_{i_{n-1}},$$

as in the picture:



**11. Definition**
Let $G = (V, A, i, t)$ be a finite digraph, $V = \{v_1, v_2, ..., v_n\}$ and let $b_{ij} = \text{card}\{a \in A \ ; \ v_i \longrightarrow v_j\}$. The **adjacent matrix associated to the digraph** $G$ is $M_G = (b_{ij})_{1 \leq i,j \leq n}$.

**12. Example**
Let $G_1$ and $G_2$ be as in the picture:




The associated adjacent matrices are:

$$M_{G_1} = \begin{pmatrix} 0 & 2 & 1 \\ 1 & 1 & 1 \\ 0 & 1 & 2 \end{pmatrix} M_{G_2} = \begin{pmatrix} 1 & 1 & 2 \\ 2 & 2 & 0 \\ 1 & 2 & 0 \end{pmatrix}.$$

By using the adjacent matrix we can get an isomorphism test for digraphs with simple links.

**13. Theorem (test isomorphism)**
Let $G_1$ and $G_2$ be two finite graphs with simple links and let $M_{G_1} = (x_{ij})_{1 \leq i,j \leq n}$ and $M_{G_2} = (y_{ij})_{1 \leq i,j \leq m}$ be their adjacent matrices. The following assertions are equivalent:
**a.** $G_1$ and $G_2$ are isomorphic.
**b.** $m = n$ and there is a permutation $\sigma$ on $\{1, 2, ..., n\}$ such that $x_{ij} = y_{\sigma(i)\sigma(j)}, \forall \ 1 \leq i, j \leq n$.
**Proof a⇒ b** Let $G_1$ and $G_2$ be isomorphic and let $G_1 = (V_1, A_1, i_1, t_1)$, $G_2 = (V_2, A_2, i_2, t_2), V_1 = \{v_1, v_2, ..., v_n\}$, $V_2 = \{w_1, w_2, ..., w_n\}$. Let $h = (h_V, h_A)$ be the isomorphism, i.e. $h_V : V_1 \mapsto V_2$, $h_A : A_1 \mapsto A_2$ bijections as in Definition 6. Let $i \in \{1, 2, ..., n\}$ be fixed and let $w_k = h_V(v_i)$; the map

$\sigma : \{1, 2, ..., n\} \mapsto \{1, 2, ..., n\}$, $, \sigma(i) = k$ is a permutation and, moreover, $h_V(v_i) = w_{\sigma(i)}$. The graphs are isomorphic, hence:

$$x_{ij} = 1 \Leftrightarrow v_i \longrightarrow v_j \text{ in } G_1 \Leftrightarrow h_V(v_i) \longrightarrow h_V(v_j) \text{ in } G_2 \Leftrightarrow$$

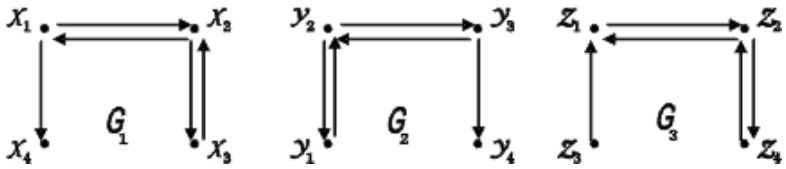
$$\Leftrightarrow w_{\sigma(i)} \longrightarrow w_{\sigma(j)} \Leftrightarrow y_{\sigma(i)\sigma(j)} = 1.$$

Of course, the same proof holds if $x_{ij} = 0$.
**b⇒ a** Let $m = n$ and let $\sigma$ be a permutation of $\{1, 2, ..., n\}$ such that $x_{ij} = y_{\sigma(i)\sigma(j)}, \forall\, i, j \in \{1, 2, ..., n\}$. Let $h_V : V_1 \mapsto V_2$, $h_V(v_i) = w_{\sigma(i)}$; let $v_i, v_j \in V_1$. Then:

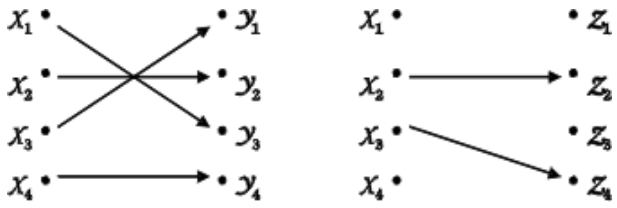
$$v_i \longrightarrow v_j \Leftrightarrow x_{ij} = 1 \Leftrightarrow y_{\sigma(i)\sigma(j)} = 1 \Leftrightarrow w_{\sigma(i)} \longrightarrow w_{\sigma(j)}.$$

The same proof holds if $v_i$ and $v_j$ are not connected (i.e. $x_{ij} = 0$), hence the graphs are isomorphic.

**14. An algorithm to test the isomorphism of digraphs**
Another way to check if two graphs are isomorphic is by using the semidegrees of the vertices. The idea of the algorithm: if $G_1$ and $G_2$ are two finite digraphs with simple links (denoted as in the previous theorem), we define a graph $\Gamma$ (usually denoted $(G_1 \longrightarrow G_2)$) as follows: the set of vertices is $V_1 \cup V_2$ and if $v \in V_1$ and $w \in V_2$, then $v \longrightarrow w$ in $\Gamma \Leftrightarrow$ $v$ and $w$ have the same semidegrees . If in this way we get a bijection between the sets $V_1, V_2$, then the graphs are isomorphic.
We describe the previous idea on the following example:
The graphs $(G_1 \longrightarrow G_2)$ and $(G_1 \longrightarrow G_3)$ are :




It results that $G_1$ and $G_2$ are isomorphic, but not $G_1$ and $G_3$.

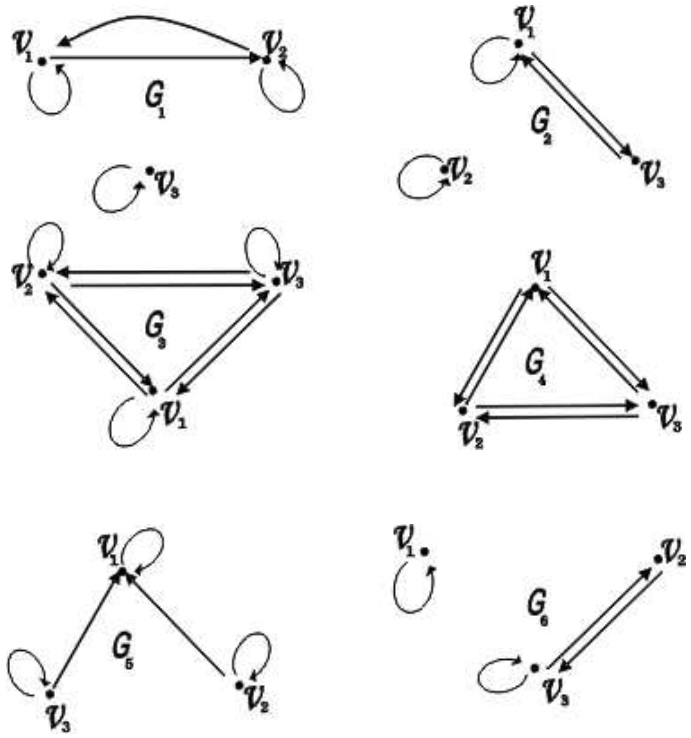


We now formalize the previous ideas.
Let $G_1$ and $G_2$ as above. The digraph $\Gamma = (G_1 \longrightarrow G_2)$ is defined as follows: let $U_1 \subseteq V_1$ be the maximal subset such that $\forall\, u \in U_1,\ \nexists\ v \in V_1,\ v \neq u$ such that $\deg^-(u) = \deg^-(v),\ \deg^+(u) = \deg^+(v)$; let $U_2 \subseteq V_2$ be the similar subset in $G_2$. An arc in $\Gamma$ is obtained by connecting a vertex of $U_1$ to a vertex in $U_2$ if and only if they have the same semidegrees. If this map between $U_1$ and $U_2$ is not a bijection, then the digraphs are not isomorphic. If it is a bijection, then we try to extend it to all $V_1$: for every $v \in V_1 \setminus U_1$, we consider all the vertices $u \in U_1$ such that $v \longrightarrow u$ or $u \longrightarrow v$; we do the same in $G_2$. We extend the previous map by connecting $v \in V_1 \setminus U_1$ to $w \in V_2 \setminus U_2$ if and only if they have the same semidegrees. If in this way we get a bijective map between $V_1$ and $V_2$, then the digraphs are isomorphic.
**The Algorithm**
**1)** Check if $\text{card}(V_1) = \text{card}(V_2)$; if yes, go to step 2; if not, go to step 7.
**2)** Compute the semidegrees of the vertices in $G_1$ and $G_2$ and find the sets $U_1$ and $U_2$. If $U_1 = V_1$ and $U_2 = V_2$, then go to step 4; if not, then go to step 3.
**3)** For every vertex in $V_1 \setminus U_1$ find the adjacent vertices in $U_1$; do the same in $G_2$. Define the digraph $\Gamma$. Go to step 5.
**4)** Define the digraph $\Gamma$. Go to step 5.
**5)** Check if the condition of isomorphism is fulfilled; if yes, go to step 6; if not, go to step 7.
**6)** $G_1$ and $G_2$ are isomorphic.
**7)** $G_1$ and $G_2$ are not isomorphic.

**15. Theorem**
Let $G = (V, A, i, t)$ be a finite digraph with simple links, $V = \{v_1, v_2, ..., v_n\}$ and let $M_G = (b_{ij})_{ij}$ be its adjacent matrix. Let $M_G^r = \left(b_{ij}^{(r)}\right)_{ij}$ be the $r$-th power of $M_G$.
**i.** $b_{ij}^{(r)}$ is the number of paths of length $r$ connecting $v_i$ to $v_j$.
**ii.** The graph $G$ has no circuits if and only if the matrix $M_G$ is nilpotent.
**iii.** Let $X = M_G + M_G^2 + ... + M_G^r$. If $X = (x_{ij})_{ij}$, then $x_{ij}$ is the number of paths of length less or equal than $r$ connecting $v_i$ to $v_j$.
**Proof i.** Induction on $r$. If $r = 1$, the conclusion is obvious. We suppose that the conclusion is true for $1 \leq m \leq r$; we now prove it for $r + 1$. By definition $M_G^{r+1} = M_G^r \cdot M_G$, i.e. $b_{ij}^{(r+1)} = \sum_{m=1}^{n} b_{ik}^{(r)} b_{kj}$. It results that $b_{ij}^{(r+1)}$ is

exactly the sum of those $b_{ik}^{(r)}$ such that $b_{kj} = 1$, i.e. $v_k \longrightarrow v_j$. According to the hypothesis, it results that $b_{ij}^{(r+1)}$ is the number of paths of length $r$ connecting $v_i$ to $v_j$.
**ii.** If $G$ has no circuits (closed paths) it results that there are no paths of length greater or equal to $n+1$; by applying **i**, we get $M_G^{r+1} = O$.
Conversely, if there is $m \in N$ such that $M_G^m = O$, then there are no paths of length greater than $m$; if the graph would have circuits, then there would be paths of any length, contradiction.
**iii.** Obviously, by applying **i**.

**16. Exercises**
Let $n \in N$. Find the number of paths of length less or equal than $n$ connecting $v_1$ to $v_2$ in the following digraphs:



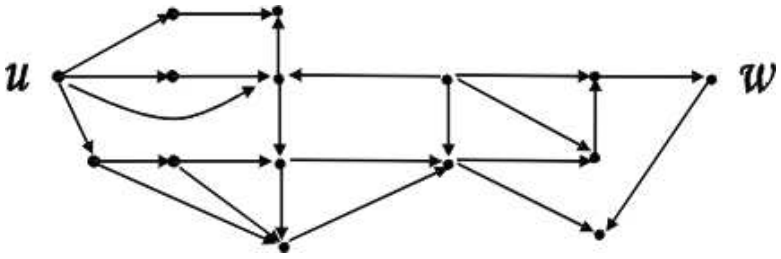

**17. Algorithm: the minimal path**
Let $G = (V, A, i, t)$ be a finite digraph. If $u$ and $w$ are two fixed vertices in $V$, then the problem is to find a path of minimum length connecting $u$ to

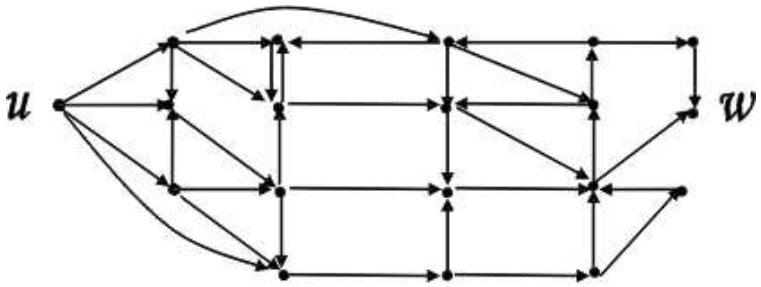
$w$; such a path is called **minimal path**. Obviously, it is not unique. In the following we give an algorithm to solve this problem.
**1)** Mark by 0 the vertex $u$.
**2)** Mark by 1 all the vertices $v \in V \setminus \{u\}$ such that $u \longrightarrow v$.
**3)** Let $V_p$ be the set of all vertices marked by $p$ ; mark by $p+1$ all the vertices $x$ such that

$$x \notin V_k, \forall\, k \leq p \text{ and } \exists\, y \in V_p \text{ such that } y \longrightarrow x$$

.**4)** Marking is over when the vertex $w$ is marked; let $w \in V_m$.
**5)** A minimal path $u \longrightarrow v_{i_{m-1}} \longrightarrow v_{i_{m-2}} \longrightarrow ... \longrightarrow v_{i_1} \longrightarrow w$ is obtained if we choose (in this order) $v_{i_1} \in V_{m-1},\ v_{i_2} \in V_{m-2}, ..., v_{i_{m-1}} \in V_1$.

**18. Exercises**
In the following digraphs, find a minimal path between the vertices $u$ and $w$.




## 3.2 Nondirected Graphs

**19. Definitions**
Let $V$ and $E$ be nonempty sets and let $V^{(2)}$ be the set of all nonoriented pairs

of $V$, i.e. $V^{(2)} = \{(v, w)\ ;\ v, w \in V$ with the convention $(v, w) = (w, v)\}$. A **nondirected graph** is a triple $H = (V, E, \varphi))$, where $\varphi : E \mapsto V^{(2)}$ is an arbitrary map.
The elements of $V$ are called vertices (nodes) and the elements of $E$ are called edges (arcs). The difference with the directed graphs is that an arc (edge) has no more an initial and a terminal vertex; they are both the ends of the arc (edge). We still have a geometrical representation:

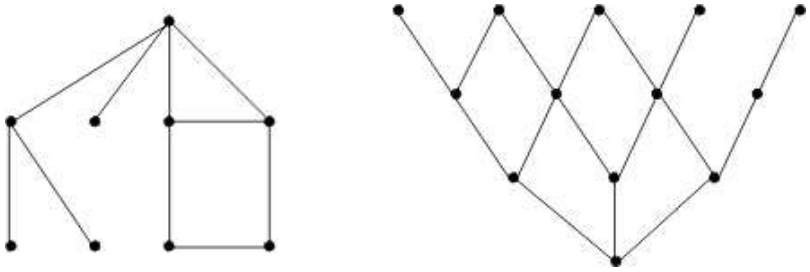
$$\varphi(a) = (u, w) \Leftrightarrow\ u \overset{a}{---} w.$$

Two vertices $u, w$ are connected by an edge if $\exists\, a \in E$ such that $\varphi(a) = (u, w)$.
A **chain** (or path) $d$ of length $r$ is a finite sequence of $r$ edges, $a_1, a_2, ..., a_r$ and $r + 1$ vertices $v_1, v_2, ..., v_{r+1}$ such that:

$$v_1 \overset{a_1}{---} v_2 \overset{a_2}{---} ..... \overset{a_r}{---} v_{r+1}$$

If this is the case, the chain $d$ connects the vertices $v_1$ and $v_{r+1}$. A closed chain is called a **cycle**.
The graph is finite if $V$ and $E$ are finite sets. The graph is with simple links if for every two vertices $v, w \in V$ there is at most one edge $a \in E$ such that $v \overset{a}{---} w$.

**20. Examples**



**21. Definition**
Let $H = (V, E, \varphi)$ be a nondirected graph with simple links. For every vertex $v \in V$, let:

$$C_v = \{v\} \cup \{w \in V\ ;\ \text{ there is a chain connecting } v \text{ and } w\}.$$

The set $C_v$ is called the **connected component** associated to $v$.
The graph is said to be **connected** if every two different vertices can be connected by a chain. On the set of vertices we define the relation:

$$u \sim w \Leftrightarrow \text{ there is a chain connecting } u \text{ and } w$$

**22. Proposition**
The above relation is an equivalence on $V$. For each vertex $v \in V$, the associated class of equivalence is the connected component of $v$. The graph is connected if there is exactly one connected component (or, equivalently, the factor set $\widehat{V}$ has one element).
**Proof** Obviously.

**23. Notation**
Let $H = (V, E, \varphi)$ be a finite nondirected graph. We denote by $n(H) = \text{card}(V)$ the number of vertices, $m(H) = \text{card}(E)$ the number of edges and with $p(H)$ the number of the connected components.
The **cyclomatic number** of $H$ is, by definition,

$$\mu(H) = m(H) - n(H) + p(H).$$

**24. Theorem**
Let $H = (V, E, \varphi)$ be a finite nondirected graph with simple links such that $n(H) \geq 2$. Then:
**a)** $\mu(H) \geq 0$.
**b)** If $H_1, H_2, ..., H_{p(H)}$ are the connected components of $H$, then

$$\mu(H) = \mu(H_1) + \mu(H_2) + ... + \mu(H_{p(H)})$$

**c)** $H$ has no cycles if and only if $\mu(H) = 0$.
**Proof** Let $H^{(1)}$ be the graph obtained by eliminating one edge in $H$. Then:

$$n(H^{(1)}) = n(H),\ m(H^{(1)}) = m(H) - 1 \text{ and}$$

$$p(H^{(1)}) = p(H) \text{ or } p(H^{(1)}) = p(H) + 1.$$

It results:

$$\mu(H^{(1)}) = m(H^{(1)}) - n(H^{(1)}) + p(H^{(1)}) =$$

$$= m(H) - 1 - n(H) + p(H^{(1)}) \leq \mu(H).$$

We continue to eliminate all the edges of $H$ one by one. We finally get the graphs $H^{(1)}, H^{(2)}, H^{(3)}, ..., K$, where $K$ has no more edges. Obviously, $\mu(H) \geq \mu(H^{(1)}) \geq \mu(H^{(2)}) \geq ... \geq \mu(K) = 0 - n(H) + p(H) = 0$.
**b)** The cyclomatic number of an arbitrary connected component is $\mu(H_i) = m(H_i) - n(H_i) + 1$, hence

$$\sum_{i=1}^{p(H)} \mu(H_i) = \sum_{i=1}^{p(H)} m(H_i) - \sum_{i=1}^{p(H)} n(H_i) + p(H) =$$

$$= m(H) - n(H) + p(H) = \mu(H).$$

**c)** The property is obvious for every connected component, i.e. $H_i$ has no cycles if and only if $\mu(H_i) = 0$, simply because $m(H_i) = n(H_i) - 1$. Obviously, the graph $H$ has no cycles if and only if each $H_i$ has no cycles. The proof is completed by applying b.

# Chapter 4

# Finite Automata

## 4.1 Alphabets and Languages

**1. Definitions**

An **alphabet** is a finite set. Its elements are usually called symbols (or letters). For example, the binary alphabet, $\{0, 1\}$ contains two symbols and the Roman alphabet, $\{a, b, c, ..., z\}$ has 27 letters. By definition, the empty set is the alphabet without any symbol.

A **string** (or **word**) over a fixed alphabet is a finite sequence of symbols of that alphabet. For example $aa, , t, dgez, mathematics$ are strings over the Roman alphabet, while $00, 101, 0, 0101$ are strings over the binary alphabet. The **empty string** is, by definition the string without any symbol. It is usually denoted by $e$. Generally, we shall denote strings by Greek letters: $\alpha, \beta, ...$ etc.

If $\mathcal{A}$ is an arbitrary alphabet, we denote by $\mathcal{A}^\star$ the set of all strings over $\mathcal{A}$. Of course, $\mathcal{A}^\star$ contains the empty string and the symbols of $\mathcal{A}$. The **length** of a string is the number of its symbols. For example, the string $aabzw$ over the Roman alphabet has the length 5. We shall denote $|\alpha|$ the length of the string $\alpha$. Generally, a string (but not the empty string), $\alpha \in \mathcal{A}^\star$ can be also defined as function $\alpha : \{1, 2, ...., |\alpha|\} \mapsto \mathcal{A},\ \alpha(j) =$ the symbol in the j-th position. For example, if $\alpha = start$, then $\alpha(1) = s, \alpha(2) = t, \alpha(3) = a, \alpha(4) = r, \alpha(5) = t$. It follows that if the alphabet $\mathcal{A}$ has $n$ symbols, then there are $n^k$ strings of length $k \in N$ (the number of functions from $\{1, 2, ..., k\}$ to $\mathcal{A}$).

**2. Proposition**

For every alphabet $\mathcal{A}$, the set of all strings, $\mathcal{A}^\star$ is countable.

**Proof** To define a bijective map from $N$ to $\mathcal{A}^\star$ we first fix some ordering of the alphabet $\mathcal{A} = \{x_1, x_2, ..., x_n\}$. The strings of $\mathcal{A}^\star$ can be enumerated as follows:

1. For each $k \in N$, all strings of length $k$ are enumerated before all the strings of length $k+1$.

2. The $n^k$ strings of length $k$ are enumerated lexicographically, i.e. the string $x_{i_1}x_{i_2}...x_{i_k}$ precedes the string $x_{j_1}x_{j_2}...x_{j_k}$ if $\exists\, m \in \{0, 1, ..., k-1\}$ such that $i_p = j_p, \forall\, p = 1, 2, ..., m$ and $i_{m+1} < j_{m+1}$.

For example, if $\mathcal{A} = \{a\}$, then $\mathcal{A}^\star = \{e, a, aa, aaa, ...\}$ and if $\mathcal{A} = \{a, b\}$, then $\mathcal{A}^\star = \{e, a, b, aa, ab, ba, bb, aaa, aab, aba, baa, abb, bab, bba, bbb, ...\}$.

**3. Definitions**

Let $\mathcal{A}$ be an alphabet and let $\alpha$, $\beta \in \mathcal{A}^\star$. The **concatenation** of $\alpha$ and $\beta$ is the string $\alpha\beta$ defined as follows: $|\alpha\beta| = |\alpha| + |\beta|$, $\alpha\beta(j) = \alpha(j)$ if $1 \leq j \leq |\alpha|$ and $\alpha\beta(|\alpha| + j) = \beta(j)$ if $1 \leq j \leq |\beta|$. In fact the string $\alpha\beta$ is the string $\alpha$ followed by $\beta$. For example, if $\alpha = ab, \beta = ba, \gamma = sent$, then $\alpha\beta = abba, \beta\alpha = baab, \alpha\gamma = absent$.

The concatenation is associative: $(\alpha)\beta\gamma = \alpha(\beta\gamma)$ and the empty string is the unit element: $\alpha e = e\alpha = \alpha$.

A string $\alpha$ is a **substring** of the string $\beta$ if there are strings $\gamma$ and $\delta$ such that $\beta = \gamma\alpha\delta$. The empty string is a substring in every string.

The **power** of a string is defined by induction: $\alpha^0 = e$, $\alpha^{i+1} = \alpha^i\alpha, \forall i \in N$.

The **reversal** of a string $\alpha$ , denoted by $\alpha^R$ is the string "spelled backwards". For example, if $\alpha = abc$, then $\alpha^R = cba$.

**4. Exercise**

Prove that for all strings $\alpha_1, \alpha_2, ..., \alpha_m$, the following equality holds: $(\alpha_1\alpha_2...\alpha_m)^R = \alpha_m^R\alpha_{m-1}^R...\alpha_1^R$.

**5. Definitions**

Let $\mathcal{A}$ be an alphabet. A **language** over $\mathcal{A}$ is any subset $\mathcal{L}$ of $\mathcal{A}^\star$.

Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two languages over the same alphabet, $\mathcal{A}$. The **concatenation** of $\mathcal{L}_1$ and $\mathcal{L}_2$ is denoted by $\mathcal{L}_1\mathcal{L}_2$ and, by definition, it is

$$\mathcal{L}_1\mathcal{L}_2 = \{\alpha\ ;\ \alpha = \beta_1\beta_2,\ \beta_1 \in \mathcal{L}_1 \text{ and } \beta_2 \in \mathcal{L}_2\}.$$

For example, if $\mathcal{L}_1 = \{\alpha \ ; \ \alpha \text{ has an odd number of } 0\text{'s}\}$ and
$\mathcal{L}_1 = \{\beta \ ; \ \beta \text{ starts with a 0 followed by an arbitrary number of 1's, then}$
$\mathcal{L}_1\mathcal{L}_2 = \{\gamma \ ; \ \gamma \text{ has an even number of } 0\text{'s}\}$.
The **union** of two languages $\mathcal{L}_1$ and $\mathcal{L}_2$ is their set union: $\mathcal{L}_1 \cup \mathcal{L}_2$.
Another way to obtain new languages is the **closure** (or **Kleene star**) of a single language. If $\mathcal{L}$ is a language over $\mathcal{A}$, then its closure is, by definition:

$$\mathcal{L} = \{\alpha \in \mathcal{A}^\star \ ; \ \exists\, k \in N \text{ and } \alpha_1, \alpha_2, ..., \alpha_k \in \mathcal{L} \text{ such that } \alpha = \alpha_1\alpha_2...\alpha_k\}.$$

By definition, the closure of the empty alphabet contains just the empty string: $\emptyset^\star = \{e\}$.
It results that the empty string and the strings of $\mathcal{L}$ are members of $\mathcal{L}^\star$.
For example, if $\mathcal{L} = \{001, 10, 1000\}$, then $100011000001 \in \mathcal{L}^\star$ because it is the concatenation of $10, 001, 1000, 001$.
The following properties are obvious:
$\mathcal{A}^\star$ is the closure of $\mathcal{A}$.
If $\mathcal{L}_1 \subseteq \mathcal{L}_2$, then $\mathcal{L}_1^\star \subseteq \mathcal{L}_2^\star$.
If $\mathcal{A} \subseteq \mathcal{L}$, then $\mathcal{L}^\star = \mathcal{A}^\star$.
$\mathcal{L}\mathcal{L}^\star = \{\alpha \ ; \ \exists k \in N \setminus \{0\} \text{ and } \alpha_1, \alpha_2, ..., \alpha_k \in \mathcal{L}_1 \text{ such that } \alpha = \alpha_1\alpha_2...\alpha_k\}$.
Obviously, the empty string is not necessary in $\mathcal{L}\mathcal{L}^\star$, unless $e \in \mathcal{L}$.

**6. Examples**

**a.** $\{a\}^\star = \{a^k \ ; \ k \in N\}$.
**b.** If $\mathcal{L}_1 = \{a\}$ , $\mathcal{L}_2 = \{b\}$, then $\mathcal{L}_1\mathcal{L}_2 = \{ab\}$, $(\mathcal{L}_1\mathcal{L}_2)^\star = \{(ab)^k \ ; \ k \in N\}$.
$\mathcal{L}_1 \cup \mathcal{L}_2 = \{a, b\}$ and $\mathcal{L}_1 (\mathcal{L}_1 \cup \mathcal{L}_2) = \{aa, ab\}$.
$\mathcal{L}_1^\star\mathcal{L}_2 = \{a^n b \ ; \ n \in N\}$, $\mathcal{L}_1^\star (\mathcal{L}_1 \cup \mathcal{L}_2) = \{a^{n+2}, a^{n+1}b \ ; \ n \in N\}$.
$\mathcal{L}_1 (\mathcal{L}_1 \cup \mathcal{L}_2)^\star = \{a\alpha \ ; \ \alpha \in \{a, b\}^\star\}$; in fact $\mathcal{L}_1 (\mathcal{L}_1 \cup \mathcal{L}_2)^\star$ consists of all strings over the alphabet $\{a, b\}$ starting with an $a$.

**7. Definitions**

We start with an example. Let

$$\mathcal{L} = \{\alpha \in \{0, 1\}^\star \ ; \ \alpha \text{ has at most two of } 1^s, \text{ which are not consecutive}\}.$$

The description of this language in terms of the previous operations is $\mathcal{L} = \{0\}^\star\{1\}\{0\}^\star (\emptyset^\star \cup \{0\}\{1\}\{0\}^\star)$. Of course, we can write without braces $\{, \}$, i. e. $\mathcal{L} = 0^\star 10^\star (\emptyset^\star \cup 010^\star)$.
The idea is that, sometimes, we can give finite representations of an infinite number of strings. Such a representation involves an alphabet and a finite

number of operations with languages.
An important issue is what strings (over an alphabet) can be represented by using a finite number of times the concatenation, the union and the closure. This leads to the following definition.
Let $\mathcal{A}$ be an alphabet. The **regular expressions** over $\mathcal{A}$ are obtained by following the rules:
**i.** $\emptyset$ and the symbols of $\mathcal{A}$ are regular expressions.
**ii.** If $\mu$ and $\nu$ are regular expressions, then $(\mu\nu)$ is a regular expression.
**iii.** If $\mu$ and $\nu$ are regular expressions, then $(\mu \cup \nu)$ is a regular expression.
**iv.** If $\mu$ is a regular expression, then $\mu^\star$ is a regular expression.
**v.** Nothing is a regular expression unless it follows from (i) to (iv).
Formally, a regular expression is a string over the alphabet $\mathcal{A} \cup \{(, ), \emptyset, \cup, ^\star\}$.
In fact, every regular expression represents a language (according to what the symbols $\cup, \cdot, ^\star$ mean).
The exact relation between a regular expression and the associated language is given by a map from strings to languages, i.e

$$L : \mathcal{A}^\star \mapsto \{\mathcal{L} \ ; \ \mathcal{L} \text{ language}\},$$

such that:
**i.** $L(\emptyset) = \emptyset$ and $L(a) = \{a\}$, $\forall a \in \mathcal{A}$.
**ii.** $L((\mu\nu)) = L(\mu)L(\nu)$, $\forall \mu, \nu$ regular expressions.
**iii.** $L((\mu \cup \nu)) = L(\mu) \cup L(\nu)$, $\forall \mu, \nu$ regular expressions.
**iv.** $L(\mu^\star) = L(\mu)^\star$, $\forall \mu$ regular expression.
A language $\mathcal{L}$ is said to be a **regular language** if it can be represented by a regular expression, i.e., formally, there is a regular expression $\mu$ such that $L(\mu) = \mathcal{L}$. In fact, the class of regular languages is the minimal set of languages containing $\emptyset, \{a\}, \forall a \in \mathcal{A}$ and is closed under the union, concatenation and closure.
A **language recognition device** is a device (algorithm) which can recognize (in a finite number of steps) if a string belongs (or not) to a given language.

**8. Example**

Let $\mathcal{A} = \{a, b\}$ and let us compute $L(((a \cup b)^\star a))$. It is not difficult to "guess" that the answer is "all strings over $\mathcal{A}$ which end with $a$". The formal proof according to the previous definition is:

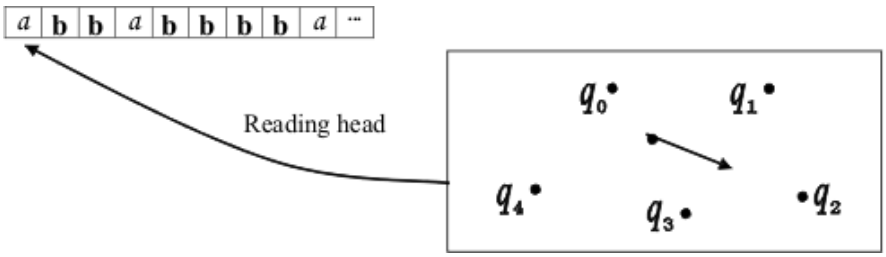
$$L(((a \cup b)^\star a)) = L((a \cup b)^\star)L(a) = L((a \cup b)^\star)\{a\} =$$
$$= L((a \cup b))^\star\{a\} = (L(a) \cup L(b))^\star\{a\} = (\{a\} \cup \{b\})^\star\{a\} = \{a, b\}^\star\{a\}.$$

## 4.2 Deterministic and nondeterministic Finite Automata

A finite automaton is a language recognition device. The input is a string over an alphabet and the output (answer) is an indication if the input belongs (or not) to the language. The answer must come after a finite number of steps (operations).

**9. Definitions**

A **deterministic finite automaton** (d.f.a.) is a quintuple $\mathcal{M} = (K, \mathcal{A}, \delta, s, F)$, where:
$K$ is a nonempty finite set; its elements are called **states**.
$\mathcal{A}$ is an alphabet.
$s \in K$ is called the **initial state**.
$F \subseteq K$; its elements are called **final states**.
$\delta : K \times \mathcal{A} \mapsto K$ is called the **transition map**.
The input string is delivered to the automaton on an **input tape**; at any specified moment the automaton is in one state and by reading one symbol it passes to another state according to the transition function. The input tape moves to the right and the process is repeated until the last symbol of the input string is read. The automaton stops in a state, which is the answer of the device. We can "represent" this process as follows:




A **configuration** of a deterministic finite automaton is any element of $(q, \alpha) \in K \times \mathcal{A}^\star$. In the above representation the configuration is $(q_2, abbabbbba)$. A configuration of type $(q, e)$ means that the automaton has consumed all its input and it stops.
On the set of configurations we define the relation **"pass (yields) in one step"** as follows:

$$(q, \alpha) \mapsto (p, \beta) \Leftrightarrow \exists\, a \in \mathcal{A} \text{ such that } \alpha = a\beta \text{ and } \delta(q, a) = p.$$

The relation is not reflexive or transitive. We now define the relation **"pass (yields) in several (possibly zero) steps"**:

$$(q,\alpha) \overset{\star}{\mapsto} (p,\beta) \Leftrightarrow \exists \alpha_1, \alpha_2, ..., \alpha_k \in \mathcal{A}^\star \text{ and } q_1, q_2, ..., q_k \in K \text{ such that}$$

$$(q,\alpha) \mapsto (q_1,\alpha_1) \mapsto (q_2,\alpha_2) \mapsto ..... \mapsto (q_k,\alpha_k) \mapsto (p,\beta).$$

Obviously, the relation $\overset{\star}{\mapsto}$ is reflexive and transitive.
A string $\alpha \in \mathcal{A}^\star$ is called **accepted** if, by definition, there is a final state $q \in F$ such that $(s,\alpha) \overset{\star}{\mapsto} (q,e)$.
The **accepted language** by the automaton $\mathcal{M}$ is denoted by $L(\mathcal{M})$ and is defined as: $L(\mathcal{M}) = \{\alpha \in \mathcal{A}^\star \ ; \ \alpha \text{ is accepted by } \mathcal{M}\}$.

**10. Example**

Let $\mathcal{M} = (K, \mathcal{A}, \delta, s, F)$, such that:
$K = \{q_0, q_1\}$, $\mathcal{A} = \{a, b\}$, $s = q_0$, $F = \{q_0\}$
and the function $\delta$ is:
$\delta(q_0, a) = q_0$, $\delta(q_0, b) = q_1$,
$\delta(q_1, a) = q_1$, $\delta(q_1, b) = q_0$.
Let us compute the evolution from the state $q_0$ by reading the string $aabba$:

$$(q_0, aabba) \mapsto (q_0, abba) \mapsto (q_0, bba) \mapsto (q_1, ba) \mapsto (q_0, a) \mapsto (q_0, e).$$

Since $q_0$ is a final state it results that the string $aabba$ is accepted by $\mathcal{M}$.
Let us now compute:

$$(q_0, abbba) \mapsto (q_0, bbba) \mapsto (q_1, bba) \mapsto (q_0, ba) \mapsto (q_1, a) \mapsto (q_1, e).$$

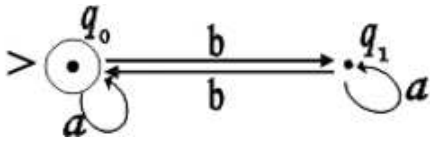
It results that $abbba$ is not an accepted string by $\mathcal{M}$.
It is not difficult to prove that the accepted language is
$L(\mathcal{M}) = \{\alpha \in \mathcal{A}^\star \ ; \ \alpha \text{ has an even number of } b\text{'s}\}$.

**11. Definition**

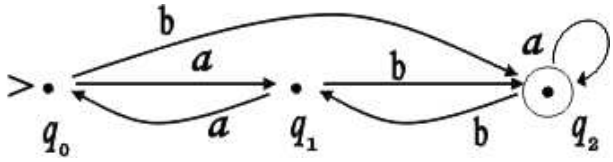
Let $\mathcal{M} = (K, \mathcal{A}, \delta, s, F)$ be a deterministic finite automaton. We can associate to $\mathcal{M}$ a digraph $G$ as follows. The vertices are the states, the arcs are the symbols and $q \overset{a}{\mapsto} p$ in $G \Leftrightarrow \delta(q, a) = p$. The initial state is marked by $>$ and the final states by a circle.

**12. Examples**

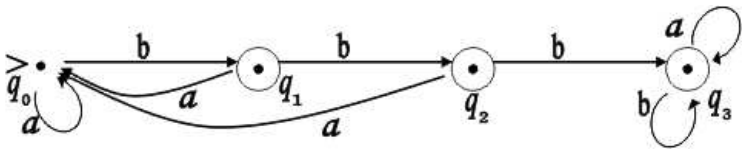
**i.** The digraph associated to the automaton from example 2 is:




**ii.** Find the automaton defined by the digraph:



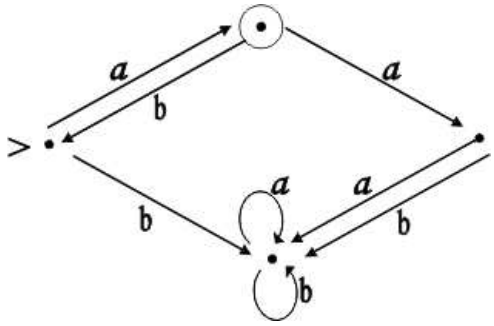

**iii.** The digraph defined by the automaton $\mathcal{M}$:
$K = \{q_0, q_1, q_2, q_3\}$, $\mathcal{A} = \{a, b\}$, $s = q_0$, $F = \{q_0, q_1, q_2\}$, and
$\delta(q_0, a) = q_0$, $\delta(q_0, b) = q_1$,
$\delta(q_1, a) = q_0$, $\delta(q_1, b) = q_2$,
$\delta(q_2, a) = q_0$, $\delta(q_2, b) = q_3$,
$\delta(q_3, a) = q_3$, $\delta(q_3, b) = q_3$
is



**13. Examples**

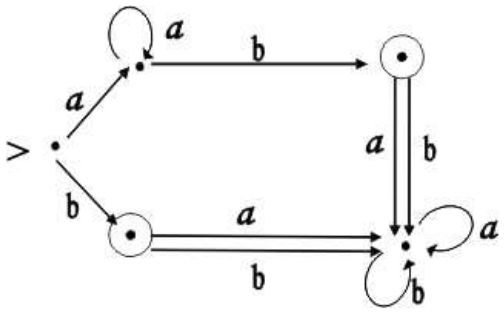
Find the languages accepted by the automata defined by the following digraphs:

**a.**




The strings $a, aba, ababa, ...$ are accepted; the language is $a(ba)^\star$.

**b.**




The strings $ab, a^2b, a^3b, ...$ are accepted; any string $a^nb\alpha, n \geq 1, \alpha \neq e$ is not accepted. The string $b$ is accepted, but any string $b\alpha$, $\alpha \neq e$ is not accepted. The language is $(a)^\star b$.

**14. Definition**

The "deterministic" behavior of a deterministic finite automaton refers to the fact that the next state is determined by the current state and by the input symbol. This property allows to predict the future evolution of the device if the present state and the input string are known.

In a nondeterministic finite automaton, the states are changing in a way

which is only partially determined by the current state and the input symbol. From the present state, by reading one input symbol, the device can pass, arbitrarily, in several possible states. Moreover, the automaton can read, at one moment, strings, not only symbols. Generally, it is not a simple problem to find a deterministic finite automaton with a given accepted language. The same problem for nondeterministic automata is much simpler due to the greater flexibility of these devices.

Formally, the definition is the following:
A **nondeterministic finite automaton** (n.d.f.a) is a quintuple
$\mathcal{N} = (K, \mathcal{A}, \Delta, s, F)$, where:
$K$ is a not empty finite set of **states**;
$\mathcal{A}$ is an **alphabet**;
$s \in K$ is the **initial state**;
$F \subseteq K$ is the set of **final states**;
$\Delta$ is the **transition relation**, i.e. $\Delta \subseteq K \times \mathcal{A}^\star \times K$.
A **configuration** of $\mathcal{N}$ is any element of $(p, \alpha) \in K \times \mathcal{A}^\star$. The relation **pass (yields) in one step** is defined on the set of configurations as follows:
$(q, \alpha) \mapsto (p, \beta) \Leftrightarrow \exists\, \gamma \in \mathcal{A}^\star$ such that $\alpha = \gamma\beta$ and $(q, \gamma, p) \in \Delta$.
The relation **pass (yields) in several steps** is:

$$(q, \alpha) \overset{\star}{\mapsto} (p, \beta) \Leftrightarrow \exists \alpha_1, \alpha_2, ..., \alpha_k \in \mathcal{A}^\star \text{ and } q_1, q_2, ..., q_k \in K \text{ such that}$$

$$(q, \alpha) \mapsto (q_1, \alpha_1) \mapsto (q_2, \alpha_2) \mapsto ..... \mapsto (q_k, \alpha_k) \mapsto (p, \beta).$$

A string $\alpha \in \mathcal{A}^\star$ is **accepted** by $\mathcal{N}$ if there is a state $q \in F$ such that $(s, \alpha) \overset{\star}{\mapsto} (q, e)$. The set of all accepted strings is the **language accepted** by $\mathcal{N}$.

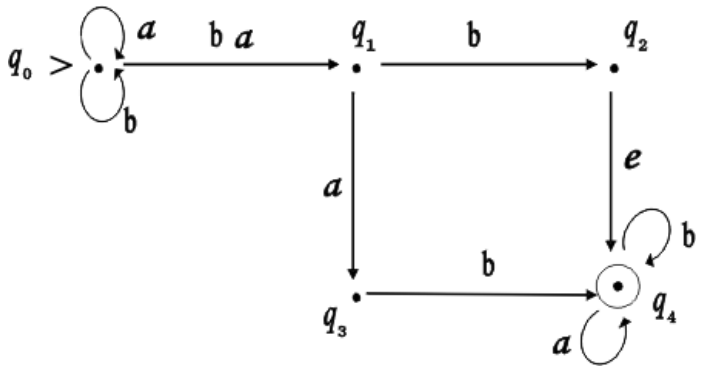
**15. Definition**

Let us consider the nondeterministic finite automaton defined as follows: $K = \{q_0, q_1, q_2, q_3, q_4\}$, $\mathcal{A} = \{a, b\}$, $s = q_0, F = \{q_4\}$ and the transition relation:

$$\Delta = \{(q_0, a, q_0), (q_0, b, q_0), (q_0, ba, q_1), (q_1, b, q_2),$$

$$(q_1, a, q_3), (q_2, e, q_4), (q_3, b, q_4), (q_4, a, q_4), (q_4, b, q_4)\}.$$

The associated digraph is:




We illustrate the nondeterministic behavior of the device by computing two possible evolutions starting from the configuration $(q_0, baababaab)$:

$$(q_0, baababaab) \mapsto (q_0, aababaab) \mapsto (q_0, ababaab) \mapsto (q_0, babaab) \mapsto$$

$$\mapsto (q_0, abaab) \mapsto (q_0, baab) \mapsto (q_0, aab) \mapsto (q_0, ab) \mapsto (q_0, b, ) \mapsto (q_0, e).$$

In this case the automaton stops in a non final state.

$$(q_0, baababaab) \mapsto (q_1, ababaab) \mapsto (q_3, babaab) \mapsto (q_4, abaab) \mapsto$$

$$\mapsto (q_4, baab) \mapsto (q_4, aab) \mapsto (q_4, ab) \mapsto (q_4, b) \mapsto (q_4, e).$$

This time the automaton stops in a final state. It results that the string $baababaab$ is accepted.
The different evolutions were obtained by the two different behaviors of the automaton at the state $q_0$: it can read the symbol $a$ or the string $ba$.

**16. Exercice**

In the previous nondeterministic automaton compute all possible evolutions starting from the configuration $(q_0, abbabb)$.

**17. Observation**

We have to notice that a deterministic finite automaton is a particular case of a nondeterministic one: the transition relation $\Delta \subseteq K \times \mathcal{A}^\star \times K$ is a function $\delta : K \times \mathcal{A} \mapsto K$. Consequently, a nondeterministic finite automaton is a deterministic one if the following conditions are fulfilled:
**i.** for every $(q, \alpha, p) \in \Delta$ it results that $\alpha \in \mathcal{A}$;
**ii.** for every $q \in K$ and $a \in \mathcal{A}$, there is only one $p \in K$ such that $(q, a, p) \in \Delta$.

**18. Exercises**

**i.** Design the graph associated to the automaton:
$K = \{q_0, q_1, q_2\},\ \mathcal{A} = \{a, b\},\ s = q_0,\ F = \{q_2\}$ and

$$\Delta = \{(q_0, e, q_1), (q_0, a, q_0), (q_0, b, q_2), (q_1, a, q_2),$$

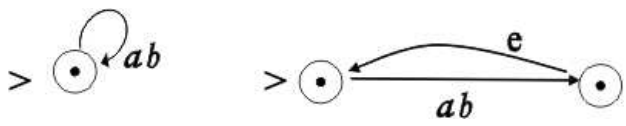
$$(q_1, ab, q_0), (q_2, b, q_2), (q_2, a, q_0), (q_2, aba, q_1)\}.$$

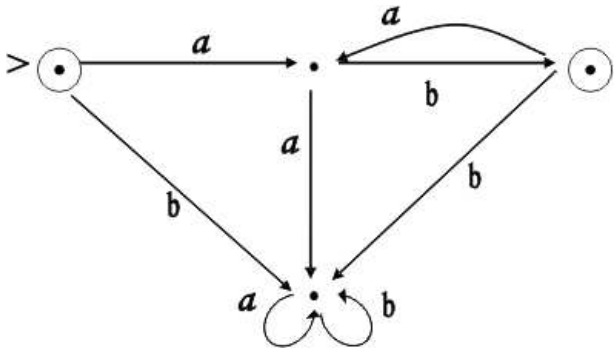
Compute two different evolutions starting from the configuration $(q_0, ababa)$.
Is $ababa$ an accepted string ?
The same questions for the configuration $(q_0, baba)$.
**ii.** Design the associated digraph of the automaton:
$K = \{q_0, q_1\}, \mathcal{A} = \{a, b\}, s = q_0, F = \{q_1\}$ and
$\Delta = \{(q_0, a, q_1), (q_0, ab, q_1), (q_0, b, q_0), (q_1, e, q_0), (q_1, bb, q_1), (q_1, a, q_0)\}$.
Compute three different transitions starting from the configuration $(q_0, aabbbba)$; is this an accepted string ?

## 4.3 The equivalence between deterministic and nondeterministic finite automata

**19. Definition**

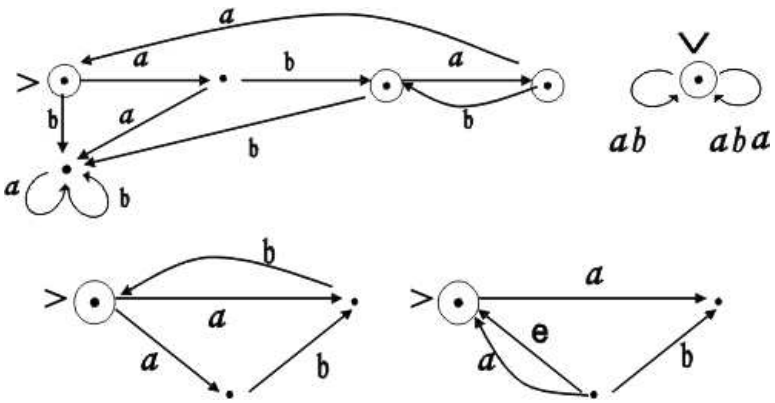
Two finite automata $\mathcal{M}_1$ and $\mathcal{M}_2$ are said to be **equivalent** if they have the same accepted language. We denote this fact by $\mathcal{M}_1 \sim \mathcal{M}_2$. Obviously, " $\sim$ " is a relation of equivalence on the set of finite automata.
For example, the automata defined by the following three digraphs they all accept the language $(ab)^\star$:






The first two are nondeterministic, while the third is deterministic.

**20. Example**
The following four automata accept the same language, $(ab \cup aba)^\star$.




**21. Theorem**
Let $\mathcal{N}$ be a nondeterministic finite automaton; then there is a deterministic finite automaton $\mathcal{M}$ such that $\mathcal{M} \sim \mathcal{N}$.
We don't prove the theorem; instead, we present the algorithm which associates to every nondeterministic automaton $\mathcal{N}$ a deterministic one, $\mathcal{M}$, which is equivalent to $\mathcal{N}$.

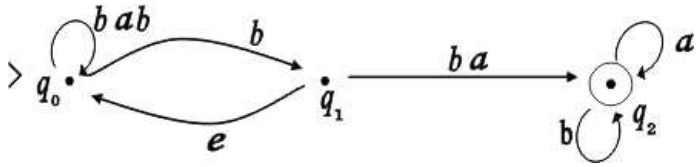
**22. Algorithm**
Let $\mathcal{N} = \{K, \mathcal{A}, \Delta, s, F)$ be a nondeterministic finite automaton.
**First step** We first eliminate the "multiple transitions", i.e. transitions

$(q, \alpha, p) \in \Delta$ with $\alpha$ of length strictly greater than 1.
For every transition $(q, \alpha, p) \in \Delta$ with $\alpha = a_1 a_2 ... a_k$ we introduce new (non final) states $p_1, p_2, ..., p_{k-1}$ and we replace the multiple transition

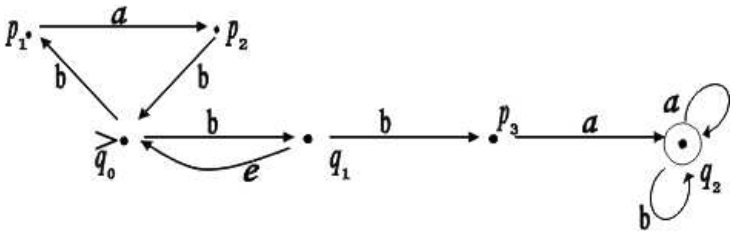
$$q \overset{\alpha}{\mapsto} p \text{ by } q \overset{a_1}{\mapsto} p_1 \overset{a_2}{\mapsto} p_2 \overset{a_3}{\mapsto} .... \overset{a_{k-1}}{\mapsto} p_{k-1} \overset{a_k}{\mapsto} p.$$

Let $\mathcal{N}' = (K', \mathcal{A}, \Delta', s, F)$ be the nondeterministic finite automaton obtained by adding to $K$ the new states and to $\Delta$ the new transitions (and by eliminating the multiple transitions). It is a simple observation that $\mathcal{N}' \sim \mathcal{N}$.

For example, the automaton $\mathcal{N}$




is replaced by $\mathcal{N}'$:




**Second step** For each state $q \in K'$ we compute the set:

$$E(q) = \{q\} \cup \{p \in K' \,;\, (q, e) \overset{\star}{\mapsto} (p, e)\}.$$

For example, for the automaton $\mathcal{N}'$ considered in the first step we get:

$$E(q_0) = \{q_0\},\ E(q_1) = \{q_0,\ q_1\}, E(q_2) = \{q_2\},$$

$$E(p_1) = \{p_1\},\ E(p_2) = \{p_2\},\ E(p_3) = \{p_3\}.$$

**Third step** We now construct a deterministic finite automaton $\mathcal{M}$ equivalent to $\mathcal{N}'$, hence to $\mathcal{N}$. The basic idea is to consider the states in $\mathcal{M}$ as subsets of $K'$. If, for example, the automaton $\mathcal{N}'$ is in a state $q \in K'$ and by reading a certain symbol it could pass in one of the states $p_1$ or $p_2$, then in the

automaton $\mathcal{M}$ the subset $\{p_1, p_2\}$ will be a state. In this way, we eliminate the nondeterministic behavior of $\mathcal{N}$.

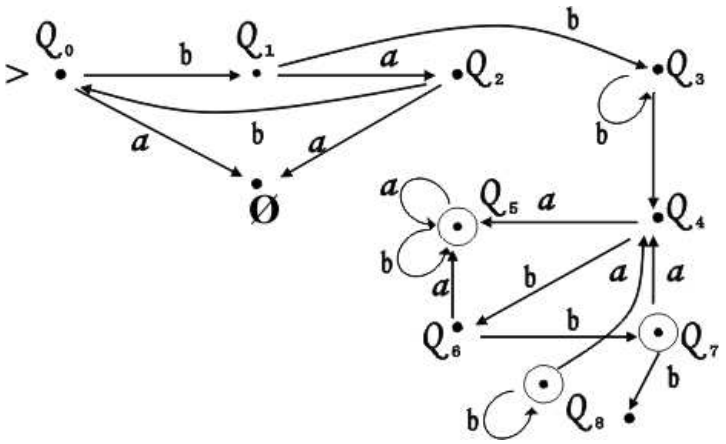
Formally, the definition of $\mathcal{M}$ is:
$\mathcal{M} = (K'', \mathcal{A}, \delta, s'', F'')$, where:
$K'' \subseteq \mathcal{P}(K')$;
$s'' = E(s)$;
$F'' = \{Q \subseteq K' \ ; \ Q \cap F \neq \emptyset\}$.
The transition map $\delta$ is:

$$\delta(Q, a) = \bigcup_{q \in K'} \{E(q) \ ; \ \exists \, p \in Q \text{ such that } (p, a, q) \in \Delta'\}.$$

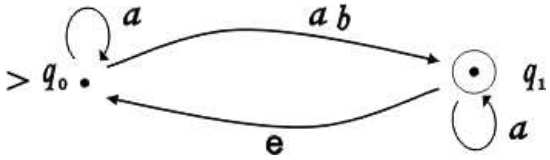
If $\Delta'$ does not contain elements of the type $(p, a, q)$ with $p \in Q$ and an arbitrary $q$, then, by definition, $\delta(Q, a) = \emptyset \in \mathcal{P}(K')$; if this is the case, then $\emptyset$ is a state in $\mathcal{M}$. The transitions starting from the empty set end to the empty set.

For example, for the automaton $\mathcal{N}'$ considered in the second step, we get:
$s'' = E(q_0) = Q_0$.
We now compute the transitions starting from $Q_0$:
$\delta(Q_0, a) = \emptyset$,
$\delta(Q_0, b) = E(q_1) \cup E(p_1) = \{q_0, q_1, p_1\} = Q_1$.
We have got two new states: $\emptyset$ and $Q_1$. We now compute the transitions starting from them.
$\delta(\emptyset, a) = \emptyset$,
$\delta(\emptyset, b) = \emptyset$.
$\delta(Q_1, a) = E(p_2) = \{p_2\} = Q_2$,
$\delta(Q_1, b) = E(q_1) \cup E(p_1) \cup E(p_3) = \{q_0, q_1, p_1, p_3\} = Q_3$.
We have got two new states: $Q_2$ and $Q_3$; the transitions are:
$\delta(Q_2, a) = \emptyset$,
$\delta(Q_2, b) = E(q_0) = Q_0$.
$\delta(Q_3, a) = E(p_2) \cup E(q_2) = \{p_2, q_2\} = Q_4$,
$\delta(Q_3, b) = E(q_1) \cup E(p_1) \cup E(p_3) = \{q_0, q_1, p_1, p_3\} = Q_3$.
We have got the new state $Q_4$.
$\delta(Q_4, a) = E(q_2) = \{q_2\} = Q_5$.
$\delta(Q_4, b) = E(q_0) \cup E(q_2) = \{q_0, q_2\} = Q_6$.
We have got the new states $Q_5$ and $Q_6$; the transitions are:
$\delta(Q_5, a) = E(q_2) = \{q_2\} = Q_5$.
$\delta(Q_5, b) = E(q_2) = Q_5$.

$\delta(Q_6, a) = Q_5$,
$\delta(Q_6, b) = E(q_1) \cup E(p_1) \cup E(q_2) = \{q_0, q_1, p_1, q_2\} = Q_7$.
We have a new state, $Q_7$; the transitions are:
$\delta(Q_7, a) = E(p_2, q_2) = \{p_2, q_2\} = Q_4$.
$\delta(Q_7, b) = E(q_1) \cup E(p_1) \cup E(p_3) \cup E(q_2) = \{q_0, q_1, p_1, p_3, q_2\} = Q_8$.
We have the new state $Q_8$; the transitions are:
$\delta(Q_8, a) = E(p_2) \cup E(q_2) = \{p_2, q_2\} = Q_4$.
$\delta(Q_8, b) = E(q_1) \cup E(p_1) \cup E(p_3) \cup E(q_2) = \{q_0, q_1, p_1, p_3, q_2\} = Q_8$.
It results that the deterministic finite automaton $\mathcal{M}$ equivalent to $\mathcal{N}$ is $\mathcal{M} = (K'', \mathcal{A}, \delta, s'', F'')$, where:
$K'' = \{Q_0, \emptyset, Q_1, Q_2, Q_3, Q_4, Q_5, Q_6, Q_7, Q_8\}$,
$\mathcal{A} = \{a, b\}$, $s'' = Q_0$,
$F'' = \{Q_4, Q_5, Q_6, Q_7, Q_8\}$
and the transition map $\delta$ defined above.
The digraph of $\mathcal{M}$ is:

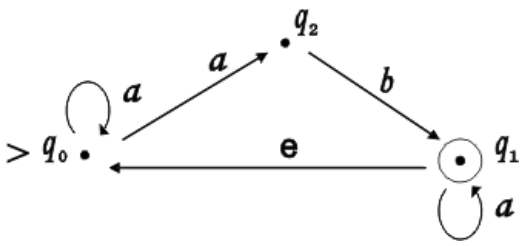



**23. Example**
Let $\mathcal{N}$ be the nondeterministic finite automaton defined by the digraph:

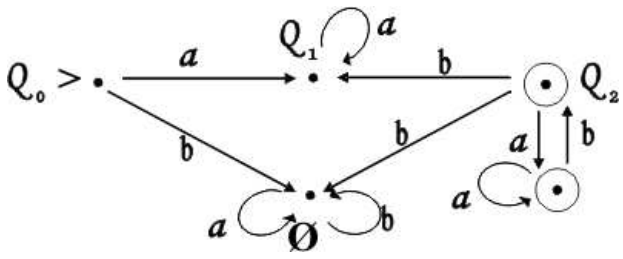



It is easy to observe that $L(\mathcal{N}) = ((a)^\star ab(a)^\star)^\star$.
The problem is to find a deterministic finite automaton equivalent to $\mathcal{N}$. We

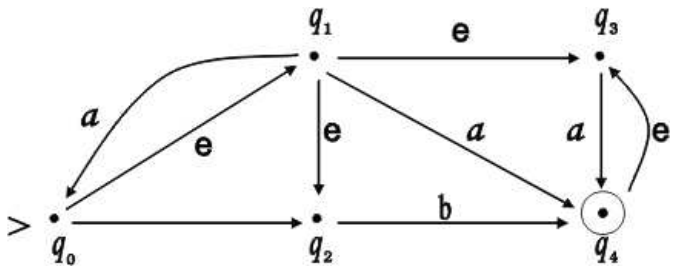
apply the previous algorithm.
First step; we introduce a new state $q_2$ and two new transitions and we get the automaton $\mathcal{N}'$:



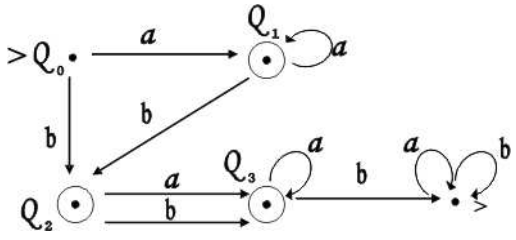

Second step; we now compute:
$E(q_0) = \{q_0\},\ E(q_1) = \{q_0, q_1\},\ E(q_2) = \{q_2\}$.
Third step; we define the automaton $\mathcal{M} \sim \mathcal{N}' \sim \mathcal{N}$.
$s'' = E(q_0) = \{q_0\} = Q_0$.
$\delta(Q_0, a) = E(q_0) \cup E(q_2) = \{q_0, q_2\} = Q_1$; $\delta(Q_0, b) = \emptyset$.
$\delta(Q_1, a) = E(q_0) \cup E(q_2 = Q_1$ ; $\delta(Q_1, b) = E(q_1) = \{q_0, q_1\} = Q_2$.
$\delta(Q_2, a) = E(q_0) \cup E(q_2) \cup E(q_1) = \{q_0, q_1, q_2\} = Q_3$ ; $\delta(Q_2, b) = \emptyset$.
$\delta(Q_3, a) = E(q_0) \cup E(q_2) \cup E(q_1) = \{q_0, q_1, q_2\} = Q_3$ ; $\delta(Q_3, b) = E(q_1) = Q_2$.
The deterministic finite automaton $\mathcal{M}$ has five states: $Q_0, Q_1, Q_2, Q_3, \emptyset$, the initial state $s'' = Q_0$ and the final states are $F'' = \{Q_2, Q_3\}$. The digraph of $\mathcal{M}$ is:



**24. Example**
Let $\mathcal{N}$ be defined by the digraph:

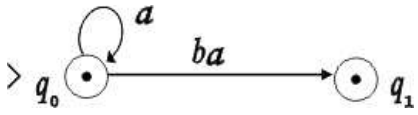



Find a deterministic finite automaton $\mathcal{M}$ equivalent to $\mathcal{N}$.
**Solution** The automaton $\mathcal{N}$ does not contain multiple transitions, hence we pass directly to the second step.
$E(q_0) = \{q_0, q_1, q_2, q_3\},\ E(q_1) = \{q_1, q_2, q_3\},$
$E(q_2) = \{q_2\},\ E(q_3) = \{q_3\}$ and $E(q_4) = \{q_3, q_4\}.$
Third step; we compute the new states and the transition map: $s'' = E(q_0) = \{q_0, q_1, q_2, q_3\} = Q_0.$
$\delta(Q_0, a) = E(q_0) \cup E(q_4) = \{q_0, q_1, q_2, q_3, q_4\} = Q_1.$
$\delta(Q_0, b) = E(q_2) \cup E(q_4) = \{q_2, q_3, q_4\} = Q_2.$
$\delta(Q_1, a) = Q_1, ;\ \delta(Q_1, b) = Q_2.$
$\delta(Q_2, a) = E(q_4) = \{q_3, q_4\} = Q_3.$
$\delta(Q_2, b) = E(q_4) = Q_3.$
$\delta(Q_3, a) = Q_3,\ \delta(Q_3, b) = \emptyset.$
The digraph of $\mathcal{M}$ is:

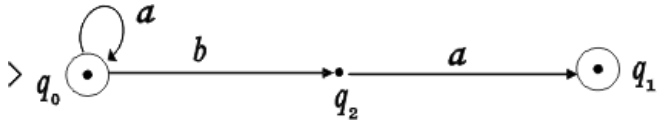


**25. Example**

Design a deterministic finite automaton which accepts the language $(a)^\star ba$.
**Solution** We first design a nondeterministic finite automaton $\mathcal{N}$ with the language $(a)^\star ba$; then, by applying the algorithm we get a deterministic finite automaton $\mathcal{M} \sim \mathcal{N}$.
A nondeterministic finite automaton $\mathcal{N}$ such that $L(\mathcal{N}) = (a)^\star ba$ is given by the digraph:



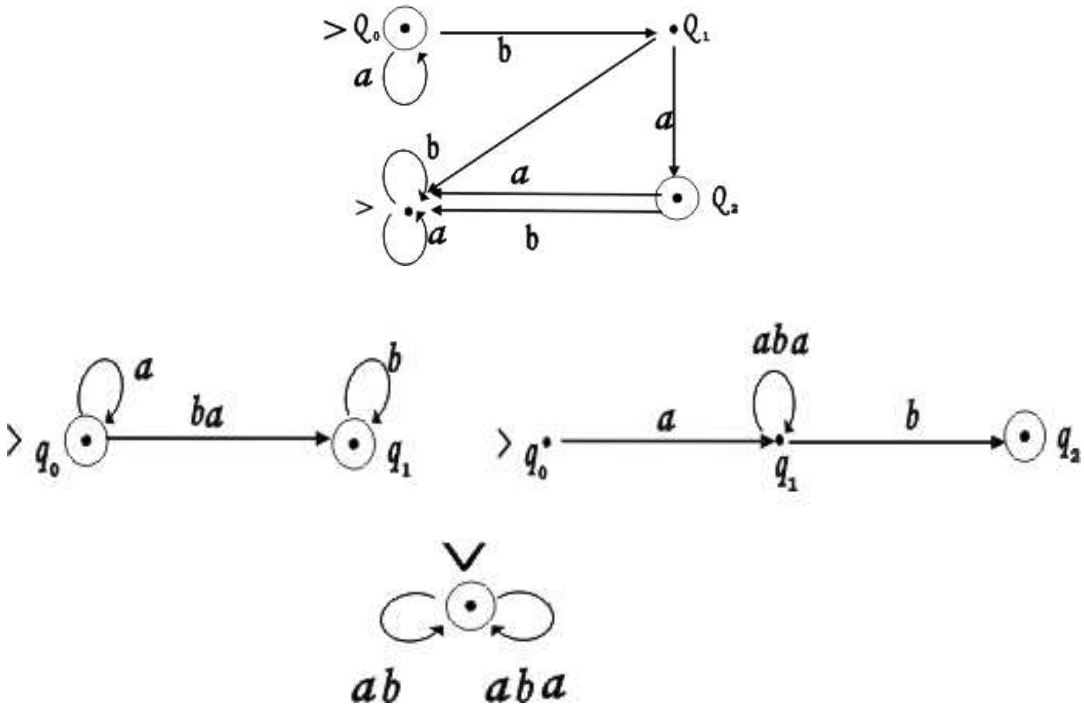

We now apply the algorithm; we introduce a new state $q_2$ and we get the automaton $\mathcal{N}'$:




Second step:
$E(q_0) = \{q_0\},\ E(q_1) = \{q_1\},\ E(q_2) = \{q_2\}$.
Third step:
$s'' = E(q_0) = \{q_0\} = Q_0$.
$\delta(Q_0, a) = E(q_0) = Q_0\ ;\ E(Q_0, b) = E(q_2) = \{q_2\} = Q_1$.
$\delta(Q_1, a) = E(q_1) = \{q_1\} = Q_2\ ;\ \delta(Q_1, b) = \emptyset$.
$\delta(Q_2, a) = \emptyset\ ;\ \delta(Q_2, b) = \emptyset$.
The initial state of $\mathcal{M}$ is $Q_0$ and the final states are $Q_0, Q_2$. The digraph of $\mathcal{M}$ is:

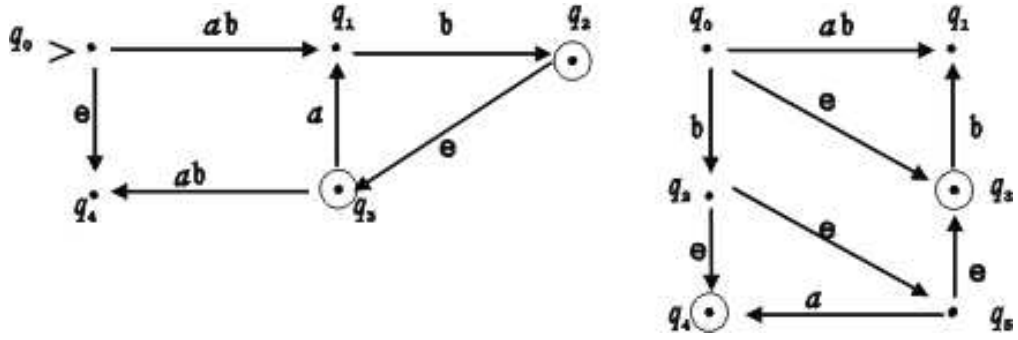
**26. Example**

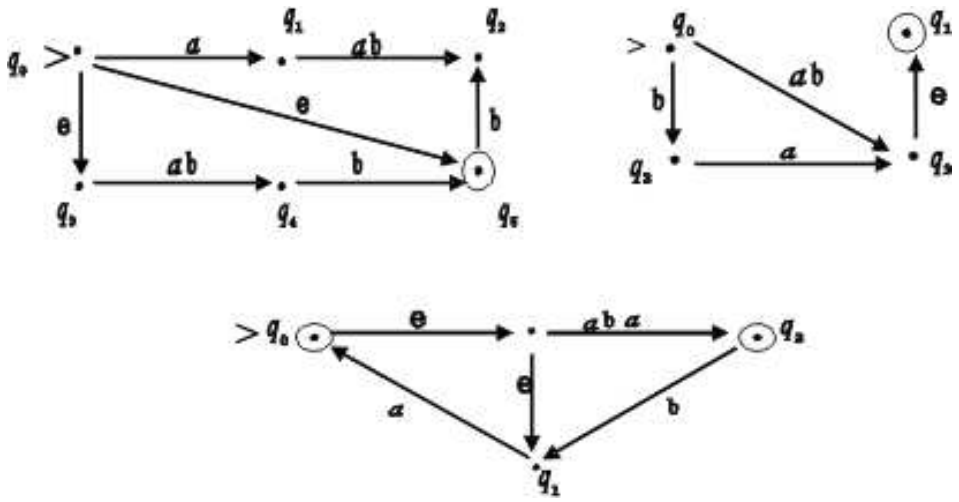
Design deterministic finite automata $\mathcal{M}_1$ and $\mathcal{M}_2$ such that
$L(\mathcal{M}_1) = (a)^\star ba(b)^\star$ and $L(\mathcal{M}_2) = a(aba)^\star b$
**Solution**
Start with the nondeterministic finite automata defined by the digraphs:




**27. Exercises**

Find the deterministic finite automata equivalent to those defined by the following digraphs:






**28. Exercises**

Design deterministic finite automata which accept the following languages: $(ab)^\star a$, $(a)^\star ba(b)^\star$, $(ab)^\star a \cup (ab)^\star b$, $(a)^\star b \cup (a)^\star a$.

## 4.4 Turing Machines

In this section we present few elementary facts about Turing machines (for more information on this subject, see [1], [3], [5]). In some sense, these are generalized versions of finite automata. Turing machines are basic abstract symbol-manipulating devices which, despite their simplicity, can be adapted to simulate the logic of any computer algorithm. They are not intended as a practical computing technology, but a thought (ideal) experiment about the limits of mechanical computation. Studying their abstract properties yields many insights into computer science and complexity theory.

**29. Definition**
Before the formal definition, we first give a "description" of a Turing machine. Let $\mathcal{A}$ be an alphabet (containing a special **blank symbol**) and $K$ be a finite set of states; the Turing machine consists of an **infinitely long tape** divided into boxes (cells), each marked with a symbol (including the blank symbol) of the alphabet $\mathcal{A}$. It also has a **head** which can read and write symbols in the boxes of the tape; it can move the tape left and right one and only one cell at a time. The behavior of a Turing machine is defined by a **transition function** (or **action table**); depending on the symbol of A in the current cell (read by the head), on the present state and according to the transition function, the machine performs the following actions:
i) it either leaves the current symbol unchanged, or it replaces it with a new one;
ii) the machine then moves to one of the two neighbouring cells (left or right);
iii) the machine either remains in the same state, or changes to another state;
the machine has a **state register** that stores the state; there is one special **start state** with which the state register is initialized.

The formal definition of a Turing machine is the following.
A Turing machine is a 7-th tuple $(K, \mathcal{A}, b, \Sigma, \delta, q_0, F)$, where:
$K$ is the (finite) set of states;
$\mathcal{A}$ is an alphabet;
$b \in \mathcal{A}$ is the blank symbol (the only symbol allowed to occur on the tape infinitely often at any step during the computation);
$\Sigma \subseteq \mathcal{A} \setminus \{b\}$ is the set of input symbols;
$\delta : K \times \mathcal{A} \mapsto K \times \mathcal{A} \times \{L, R\}$ is a relation (partial function) called the transition map; $L$ and $R$ denote the left and right shift, respectively. An alternative definition allows another symbol, $N$ which means "no shift";
$q_0$ is the initial state;
$F$ is the set of final (accepted) states.
The machine may not halt or it may halt by sending it in a **non existent state**.

**30. Example**
Let $\mathcal{A} = \{0, 1\}$ and let $K = \{A, B, C, D\}$. The action table is

| Table | 0 | 1 |
|---|---|---|
| A | 1 R A | 1 R E |
| B | 0 R D | 0 R A |
| C | 1 R D | 1 R B |
| D | 0 L B | 1 L C |

The state $E$ is the nonexistent state.
The action table is interpreted as follows; if the machine is in state $A$ and it is positioned at the cell with entry 0, then the machine will perform the following actions:
1) it replaces the symbol 0 by the symbol 1
2) it moves (one position) to the right (R)
3) it changes to the state A.
Usually, the fact that the machine is in state $A$ and it is positioned at the cell with entry 0 is represented as:

$$A$$

$$\downarrow$$

$$...0...$$

Of course, at the left and at the right of the symbol 0 are other symbols, too.
The action table may be given by several 5-tuples with the configuration:

(**current state**, **scanned symbol**, **print symbol**, **move tape**, **next state**)

For example the above action table may be written as:

$$\{(A, 0, 1, R, A), (A, 1, 1, R, E), (B, 0, 0, R, D), (B, 1, 0, R, A),$$

$$(C, 0, 1, R, D), (C, 1, 1, R, B), (D, 0, 0L, B), (D, 1, 1, L, C)\}$$

Below we give an example of the actions performed by the above Turing machine.

$$\begin{array}{c} D \\ \downarrow \\ \ldots\ldots\ 0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 1\ldots \end{array}$$

$$\begin{array}{c} C \\ \downarrow \\ \ldots\ldots\ 0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 1\ldots \end{array}$$

$$\begin{array}{c} B \\ \downarrow \\ \ldots\ldots\ 0\ \ 1\ \ 1\ \ 0\ \ 1\ \ 1\ldots \end{array}$$

$$\begin{array}{c} A \\ \downarrow \\ \ldots\ldots\ 0\ \ 1\ \ 0\ \ 0\ \ 1\ \ 1\ldots \end{array}$$

$$\begin{array}{c} A \\ \downarrow \\ \ldots\ldots\ 0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 1\ldots \end{array}$$

$$\begin{array}{c} E \\ \downarrow \\ \ldots\ldots\ 0\ \ 1\ \ 0\ \ 1\ \ 1\ \ 1\ldots \end{array}$$

The Turing machine halts in this last situation ($E$ is the non existent state).
For the same Turing machine, if we start from another situation, the machine may not halt; for example:

$$\begin{array}{c} B \\ \downarrow \\ \ldots\ldots\ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1\ldots \end{array}$$

$$\begin{array}{c} D \\ \downarrow \\ \ldots\ldots\ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1\ldots \end{array}$$

$$\begin{array}{c} B \\ \downarrow \\ \ldots\ldots\ 0\ \ 0\ \ 1\ \ 0\ \ 1\ \ 1\ldots \end{array}$$

Now the Turing machine will repeat the same actions and will never halt.

**31. Notations**

We consider the following notations and conventions:

1) The states of a Turing machine with $k$ states will be denoted by the natural numbers $0, 1, 2, 3, ..., k-1$ and the non existent state (halting state) will be denoted by $k$.

2) The alphabet will be $\mathcal{A} = \{0, 1\}$. The natural numbers will be represented in the 1-ary system, i.e. a string of $n$ 1's represents the natural number $n$. The numbers are interspaced by single 0's. The natural number 0 is the empty string (or by a tape consisting entirely of 0's). As an example, the sequence of natural numbers:

$$...7, 12, 1, 0, 9, 2, 5, ...$$

is represented on the tape of a Turing machine as:

$$...01111111011111111111101001111111110110111110...$$

The two successive 0's delimitate the number 0 (the empty string).

3) If the tape doesn't consist entirely of 0's, then the Turing machine starts at the left most 1.

4) The starting state is 0.

**Design of Turing Machines**

**32. Example**

Let $f : \mathbf{N} \mapsto \mathbf{N},\ f(n) = 1$. The problem is to design a Turing machine to compute $f$. This means that if we have the situation:

$$\begin{array}{c} \text{current state} \\ \downarrow \\ 0\underbrace{11...1}_{n}0 \end{array}$$

then the Turing machine must turn it to the situation:

$$\begin{array}{c} \text{halting state} \\ \downarrow \\ 010 \end{array}$$

The action table is:

| Table | 0 | 1 |
|---|---|---|
| 0 | 1L1 | 0R0 |
| 1 | 0R2 | |

The blank in the above table denotes a combination never reached. Indeed, the Turing machine defined by the above action table acts as follows:

it changes 1's to 0's as it moves to the right;

it stops at the symbol 0 and it replaces it by the symbol 1;

it moves to the halting state (denoted by 2).

**33. Example**

Let us design a Turing machine to compute the successor function, i.e.

$$f : \mathbf{N} \mapsto \mathbf{N},\ f(n) = n + 1.$$

So the Turing machine must turn the situation:

$$\begin{array}{c} \text{current state} \\ \downarrow \\ 0\underbrace{11...1}_{n}0 \end{array}$$

to the situation:

$$\begin{array}{c} \text{halting state} \\ \downarrow \\ 0\underbrace{11...1}_{n+1}0 \end{array}$$

Let us consider the following action table:

| Table | 0 | 1 |
|---|---|---|
| 0 | 1L1 | 1R0 |
| 1 | 0R2 | 1L1 |

This Turing machine moves to the right and it doesn't change 1; it stops at the symbol 0 and changes it to the symbol 1;then it moves to the left till the first 1; finally it moves to the halting state.

**34. Example**

Let us design a Turing machine to add two natural numbers, i.e. to compute the function

$$f : \mathbf{N} \times \mathbf{N} \mapsto \mathbf{N},\ f(n, m) = n + m.$$

The corresponding Turing machine must turn the situation:

current state

$$\downarrow$$

$$0\underbrace{1\ldots1}_{n}0\underbrace{1\ldots1}_{m}0$$

to the situation

halting state

$$\downarrow$$

$$0\underbrace{1\ldots1}_{n}\underbrace{1\ldots1}_{m}0$$

One possible action table is:

| Table | 0 | 1 |
|---|---|---|
| 0 | 1L1 | 1R0 |
| 1 | 0R2 | 1L1 |
| 2 | | 0R3 |

This action table acts as follows: it move to the right, doesn't change 1, it stops when it is reading 0, replaces it by 1, then it moves to the left and replace the left-most 1 by 0, moves one position to the right and finally moves to the halting state. One can design another action table with the same result (by changing the order of some actions):

| Table | 0 | 1 |
|---|---|---|
| 0 | | 0R1 |
| 1 | 1L2 | 1R1 |
| 2 | 0R3 | 1L2 |

This time the Turing machine changes the first (from the left) 1 to 0, then it moves to the right and doesn't change the symbol 1, it changes the first encountered 0 to 1, it moves to the left at the left-most 1 and the moves to the halting state.

**35. The composition of Turing Machines**

Let $M_1$ and $M_2$ be two Turing machines over the same alphabet with $k_1$ and $k_2$ states, respectively. The states of $M_1$ are $\{0, 1, 2, ..., k_1 - 1\}$, with 0 starting state and $k_1$ the halting state.
The states of $M_2$ are $\{k_1, k_1 + 1, ..., k_1 + k_2 - 1\}$, with $k_1$ the starting state and $k_1 + k_2$ the halting state. Then the **composition** of $M_1$ and $M_2$ (in this order) is defined by the action table obtained by putting the action table of $M_2$ below the action table of $M_1$. Sometimes it is possible that the two Turing machines have the same starting state, namely, 0.

**36. Example**

Let us design a Turing machine to compute the function

$$f : \mathbf{N} \mapsto \mathbf{N} \times \mathbf{N},\ f(n) = (n, n)$$

This means to change the situation

$$\begin{array}{c} \text{current state} \\ \downarrow \\ 0\underbrace{1\ ...\ 1}_{n}0 \end{array}$$

to the situation

$$\begin{array}{c} \text{halting state} \\ \downarrow \\ 0\underbrace{1\ ...\ 1}_{n}0\underbrace{1\ ...\ 1}_{n}0 \end{array}$$

We shall use the idea of composing two Turing machines. We shall split the process in two steps: the first one is to turn the situation

$$\begin{array}{c} \text{current state} \\ \downarrow \\ 0\underbrace{1\ ...\ 1}_{n}0 \end{array}$$

to the situation

$$\begin{array}{c}\text{current state}\\ \downarrow\\ 0\underbrace{1\ \ldots\ 1}_{n}0\underbrace{1\ \ldots\ 1}_{n}0\end{array}$$

and the second step is to turn the above situation to

$$\begin{array}{c}\text{halting state}\\ \downarrow\\ 0\underbrace{1\ \ldots\ 1}_{n}0\underbrace{1\ \ldots\ 1}_{n}0\end{array}$$

For the first step, we proceed by induction, by turning the situation:

$$\begin{array}{c}\text{current state}\\ \downarrow\\ 0\underbrace{1\ \ldots\ 1}_{r}\ \underbrace{1\ \ldots\ 1}_{n-r}0\underbrace{1\ \ldots\ 1}_{r}0\end{array}$$

to the situation:

$$\begin{array}{c}\text{current state}\\ \downarrow\\ 0\underbrace{1\ \ldots\ 1}_{r+1}\ \underbrace{1\ \ldots\ 1}_{n-r-1}0\underbrace{1\ \ldots\ 1}_{r+1}0\end{array}$$

The action table for the first Turing machine is:

| Table | 0 | 1 |
|---|---|---|
| 0 | | 0R1 |
| 1 | 0R2 | 1R1 |
| 2 | 1L3 | 1R2 |
| 3 | 0L4 | 1L3 |
| 4 | 1R0 | 1L4 |

The above action table adds a 1 on the right-hand block and moves the head one position to the right. For the second step, one can use the following

action table (the second Turing machine has the same starting state as the first one):

| Table | 0 | 1 |
|---|---|---|
| 0 | 0L5 | |
| 5 | 0R6 | 1L5 |

Consequently, to solve the problem, the action table is:

| Table | 0 | 1 |
|---|---|---|
| 0 | 0L5 | 0R1 |
| 1 | 0R2 | 1R1 |
| 2 | 1L3 | 1R2 |
| 3 | 0L4 | 1L3 |
| 4 | 1R0 | 1L4 |
| 5 | 0R6 | 1L5 |

**37. Example**
Let us design a Turing machine to compute the function

$$f : \mathbf{N} \mapsto \mathbf{N},\ f(n) = 2n.$$

Of course, we shall compose the Turing machines from examples 36 and 34 (in this order). Consequently, we obtain the following two solutions (according to the two action tables found in example 34):

| Table | 0 | 1 |
|---|---|---|
| 0 | 0L5 | 0R1 |
| 1 | 0R2 | 1R1 |
| 2 | 1L3 | 1R2 |
| 3 | 0L4 | 1L3 |
| 4 | 1R0 | 1L4 |
| 5 | 0R6 | 1L5 |
| 6 | 1L7 | 1R6 |
| 7 | 0R8 | 1L7 |
| 8 | | 0R9 |

or

| Table | 0 | 1 |
|---|---|---|
| 0 | 0L5 | 0R1 |
| 1 | 0R2 | 1R1 |
| 2 | 1L3 | 1R2 |
| 3 | 0L4 | 1L3 |
| 4 | 1R0 | 1L4 |
| 5 | 0R6 | 1L5 |
| 6 | | 0R7 |
| 7 | 1L8 | 1R7 |
| 8 | 0R9 | 1L8 |

**The busy beaver problem**

Let $n \in \mathbf{N}$ and let $T$ be a binary Turing machine (i.e. over the binary alphabet) with $n$ states. The action table contains at most $2n$ rules (a table with 2n rows and 2 columns). An instruction of this Turing machine is $a\,S\,k$, with $a \in \{0,1\},\ S \in \{L,R\}$ and $k \in \{0,1,2,...,n-1,n\}$, $n$ being the halting state. Obviously, there are $4(n+1)$ instructions of this type. Consequently, there are at most $(4(n+1))^{2n}$ binary Turing machines with $n$ states. So we proved the following:

**38. Observation**

The set of all binary Turing machines with $n$ states is a finite set.

We denote by $\mathcal{T}_n$ this set. Among these Turing machines there are machines which halt if they start from a blank tape and there are machines that do not halt if they start from the blank tape. We leave to the reader to argue the previous assertion. We denote by $\mathcal{H}_n \subset \mathcal{T}_n$ the set of those Turing machines which halt if they start from a blank tape.

**39. Definition**

For a Turing machine $M \in \mathcal{H}_n$ we denote by $B(M)$ the number of steps necessary to halt if it starts from the blank tape and we define the function:

$$\beta : \mathbf{N} \mapsto \mathbf{N},\ \beta(n) = \max_{M \in \mathcal{H}_n} B(M)$$

The map $\beta$ is called the **busy beaver** function. It is well defined because $\mathcal{H}_n$ is a finite set.

A Turing machine $M \in \mathcal{H}_n$ such that $B(M) = \beta(M)$ is called a "busy beaver" (it is not necessarily unique).

The **busy beaver problem** is to find out whether there is an algorithm (finite procedure , computing programme) to compute the value $\beta(n)$ for every $n \in \mathbf{N}$. In more sophisticated terms the problem is to determine if $\beta$ is a **computable function** (we do not enter into details on computable functions).

**40. Theorem**

There is no algorithm (finite procedure, computing programme) to compute $\beta(n)$ for every $n \in \mathbf{N}$.

Before sketching the proof we want to mention that the above assertion doesn't mean that one cannot compute $\beta(n)$ (in a finite procedure) for a *particular* $n \in \mathbf{N}$; it means that there is no algorithm to compute $\beta(n)$ *for every* $n \in \mathbf{N}$.

**Proof**

We just give the basic ideas of the proof; there are three steps.

The first step: the function $\beta$ is strictly increasing, i.e.

$$\beta(n+1) > \beta(n),\ \forall n \in \mathbf{N}$$

Let $M \in \mathcal{H}_n$ such that $\beta(n) = B(M)$. We define a new Turing machine $M' \in \mathcal{H}_{n+1}$ by adding to the action table of $M$ new instructions, for example:

| | | |
|---|---|---|
| n | 1 L (n+1) | 1 L (n+1) |

Here, as usual, $n$ is the halting state of $M$ (and it is a state of $M'$) and $n+1$ is the halting state of $M'$. It can be checked that $B(M') = \beta(n) + 1$, so

$$\beta(n) + 1 = B(M') \leq \beta(n+1),$$

and the first step is proved.

The second step: it can be proved that any algorithm (computer programme) can be described in terms of a Turing machine, so the busy beaver

problem reduces to prove that there is no Turing machine to compute $\beta(n)$ for every $n \in \mathbf{N}$.

The third step is to prove that the assumption that there exists a Turing machine to compute $\beta(n)$, for every $n \in \mathbf{N}$ leads to a contradiction. For this we define a set of Turing machines as follows:

1) There exists a Turing machine with 2 states, $M_1$ to compute the function $f : \mathbf{N} \mapsto \mathbf{N},\ f(n) = n + 1$ (as in example 33).

2) There exists a Turing machine with 9 states, $M_2$ to compute the function $f : \mathbf{N} \mapsto \mathbf{N},\ f(n) = 2n$ (as in example 37).

3) Suppose on the contrary that a Turing machine $M$ exists to compute the function $\beta(n)$, $\forall n \in \mathbf{N}$; let us suppose that $M$ has $k$ states.

4) We now define the Turing machine $T_m = M_1 M_2^m M$; the notation $M_2^m$ means "a sequence of $m$ copies of $M_2$". The Turing machine $T_m$ starts with $M_1$, followed by $m$ copies of $M_2$ and it ends with $M$. It is easy to check that $T(m)$, when started with a blank tape will halt with $\beta(2^m)$ of successive 1's on the tape, so it will take at least $\beta(2^m)$ steps to halt. Moreover, the number of states of $T(m)$ is $2 + 9m + k$. It results:

$$\beta(2^m) \leq B(T(m)) \leq \beta(2 + 9m + k)$$

Obviously, for all sufficiently large $m \in \mathbf{N}$, we have the inequality

$$2^m > 1 + 9m + k,$$

so, by applying the first step we get

$$\beta(2^m) > \beta(2 + 9m + k),$$

which is in contradiction with the above inequality. It results that there exists no Turing machine to compute the busy beaver function, so, according to the second step, the proof is completed.

**41. The halting problem**

The result obtained for the busy beaver function can be used to solve the following **halting problem**:

Is there an algorithm (finite procedure, computing programme) which can determine for any Turing machine and for any input tape, if the Turing machine will halt?

The answer is negative; as in the previous proof (second step) it is enough to prove that there exists no Turing machine to complete this task. Let us suppose, on the contrary, that such a Turing machine exists and let it be denoted by $M$. For every $n \in \mathbf{N}$, by using $M$, one can find out those Turing machines in $\mathcal{T}_n$ which halt when started with the blank tape; this is the subset $\mathcal{H}_n$. This is a finite set, so we can now compute $B(H)$, $\forall H \in \mathcal{H}_n$; consequently, we compute $\beta(n)$, $\forall n \in \mathbf{N}$, which is a contradiction with the conclusion of the busy beaver problem.

# Chapter 5

# Boolean Algebras

## 5.1 Boolean Calculus

**1. Definition**

A **Boolean algebra** is a system $\aleph = \{\mathcal{B}, \vee, \cdot, ', 0, 1\}$, where:

$\mathcal{B}$ is a non-empty set;

$\vee : \mathcal{B} \times \mathcal{B} \mapsto \mathcal{B}$
and

$\cdot : \mathcal{B} \times \mathcal{B} \mapsto \mathcal{B}$
are operations on $\mathcal{B}$ called **disjunction** and,
respectively, **conjunction**;

$' : \mathcal{B} \mapsto \mathcal{B}$
is a map called **negation**;

0 and 1 are two elements in $\mathcal{B}$ with $0 \neq 1$
such that for every $x, y, z \in \mathcal{B}$, we have:

$$\begin{array}{rcll} x \vee y & = & y \vee x, & (1.1) \\ xy & = & yx, & (1.1') \\ (x \vee y) \vee z & = & x \vee (y \vee z), & (1.2) \\ (xy)z & = & x(yz), & (1.2') \\ x \vee xy & = & x, & (1.3) \\ x(x \vee y) & = & x, & (1.3') \\ x \vee yz & = & (x \vee y)(x \vee z), & (1.4) \\ x(y \vee z) & = & xy \vee xz, & (1.4') \\ x \vee 1 & = & 1, & (1.5) \\ x0 & = & 0, & (1.5') \\ x \vee x' & = & 1, & (1.6) \\ xx' & = & 0, & (1.6'). \end{array}$$

Hence, the disjunction and the conjunction are **commutative**, (1.1 and 1.1′), **associative**, (1.2 and 1.2′) and verify the **absorption laws**, (1.3 and 1.3′); a system $\{\mathcal{B}, \vee, \cdot, \}$ with properties 1.1 to 1.3′ is called a **lattice**.

To be a Boolean algebra, this lattice must be **distributive**, (1.4 and 1.4′), to have a **unit element**, (1.5) and a **zero element**, (1.5′), and to be **complemented**, (1.6 and 1.6′). The element $x'$ is called the **complement** of $x$.

**2. Examples**

**(i)** Let $\mathcal{B}_2 = \{0, 1\}$ and let $\vee$, $\cdot$ and $'$ be defined as follows:

$$0 \vee 0 = 0,\ 0 \vee 1 = 1,\ 1 \vee 1 = 1$$

$$0 \cdot 0 = 0,\ 0 \cdot 1 = 0,\ 1 \cdot 1 = 1$$

$$0' = 1,\ 1' = 0$$

The properties 1.1 to 1.6′ can be easily verified. $\mathcal{B}_2$ is called **the two-element algebra**.

**(ii)** The set of all propositions is a Boolean algebra with the usual logical connectors.

**(iii)** Let $X$ be a non-empty set and let $\mathcal{P}(X) = \{A\,;\, A \subseteq X\}$ be the set of all subsets of $X$. We consider the usual operations of sets:

$$A \cup B = \{x \in X\,;\, x \in A \text{ or } x \in B\}.$$

$$A \cap B = \{x \in X\,;\, x \in A \text{ and } x \in B\}.$$

$$A' = \{x \in X\,;\, x \notin A\}.$$

Then $\{\mathcal{P}(X),\, \cup,\, \cap,\, ',\, \emptyset,\, X\}$ is a Boolean algebra.
**(iv)** More generally, if $\mathcal{H} \subseteq \mathcal{P}(X)$ such that:

$$\emptyset \in \mathcal{H},$$

$$A, B \in \mathcal{H} \Rightarrow A \cup B \in \mathcal{H},$$

$$A, B \in \mathcal{H} \Rightarrow A \cap B \in \mathcal{H},$$

$$A \in \mathcal{H} \Rightarrow A' \in \mathcal{H},$$

then $\{\mathcal{H}, \cup,\, \cap,\, ',\, \emptyset,\, X\}$ is a Boolean algebra.
**(v)** Let $\aleph = \{\mathcal{B},\, \vee,\, \cdot,\, ',\, 0,\, 1\}$ be a Boolean algebra; then the system $\{\mathcal{B},\, \cdot,\, \vee,\, ',\, 1,\, 0\}$ is also a Boolean algebra, called the **dual** of $\aleph$ and denoted by $\aleph^d$. The assertion results because in the definition of a Boolean algebra the axioms are grouped into pairs, by interchanging $\vee$ with $\cdot$ and 0 with 1.
**(vi)** We now give an example of a lattice which is not a Boolean algebra.
Let $M$ be a non-empty set and let

$$\mathcal{F}(M) = \{f\,;\, f : M \mapsto [0,1]\}.$$

An element $f \in \mathcal{F}(M)$ is called a **fuzzy set**. For every $f, g \in \mathcal{F}(M)$,we define:

$$f \vee g : M \mapsto [0,1],\ (f \vee g)(t) = \text{maximum}\{f(t),\, g(t)\},$$

$$f \cdot g : M \mapsto [0,1],\ (f \cdot g)(t) = \text{minimum}\{f(t),\, g(t)\}.$$

It results that $\{\mathcal{F}(M),\, \vee,\, \cdot\}$ is a distributive lattice.
In order to define a structure of a Boolean algebra on $\mathcal{F}(M)$, a natural choice is to consider:

$$0 : M \mapsto [0,1],\ 0(t) = 0,$$

$$1 : M \mapsto [0,1],\ 1(t) = 1,$$

$$f' = 1 - f.$$

However, the system $\{\mathcal{F}(M),\, \vee,\, \cdot,\, ',\, 0,\, 1\}$ is not a Boolean algebra because if $f$ is a constant function (fuzzy set), $f \neq 0$ and $f \neq 1$, then there is no $g \in \mathcal{F}(M)$ such that $f \vee g = 1$ and $f \cdot g = 0$.

**3. Theorem (the principle of duality)**

Let $\aleph = \{B, \vee, \cdot, ', 0, 1\}$ be a Boolean algebra and let $P$ be a property expressed using $\vee$, $\cdot$, $'$, 0 and 1. Let $P^d$ be the property (usually called the dual property of $P$) obtained by interchanging (in $P$) $\vee$ with $\cdot$ and 0 with 1. Then we have:

$$P \text{ is true} \Leftrightarrow P^d \text{ is true.}$$

**Proof**

If $P$ is a true property in an arbitrary Boolean algebra $\aleph$, then $P$ must be true in the dual algebra, $\aleph^d$, (see example 2(v)). But the property $P$ in $\aleph^d$ is exactly the property $P^d$ in $\aleph$.

**4. Theorem**

Let $\{\mathcal{B}, \vee, \cdot, ', 0, 1\}$ be a Boolean algebra; then, for every $x, y \in \mathcal{B}$, we have:

$$\begin{array}{lcll} x \vee x & = & x & (1.7) \\ x \cdot x & = & x & (1.7') \\ x \vee 0 & = & x & (1.8) \\ x \cdot 1 & = & x & (1.8') \\ x \vee y = 0 & \Leftrightarrow & x = y = 0 & (1.9) \\ x \cdot y = 1 & \Leftrightarrow & x = y = 1 & (1.9') \end{array}$$

We call (1.7) and (1.7′) the laws of **idempotency**.

**Proof**

We shall prove only (1.7), (1.8) and (1.9); the rest will follow by using the principle of duality.

From (1.3) and (1.3′), we get:

$$x = x \vee x(x \vee y) = x \vee x.$$

From (1.5′) and (1.3), we get:

$$x \vee 0 = x \vee x0 = x.$$

We now prove (1.8); from the idempotency, we get $0 \vee 0 = 0$. Conversely, if $x \vee y = 0$, by using (1.3′) and (1.5′), we get:

$$x = x(x \vee y) = x0 = 0.$$

Of course, a similar argument gives $y = 0$.

**5. Lemma**
Let $x, y \in \mathcal{B}$.

$$\text{If } x \vee y = 1 \text{ and } xy = 0, \text{ then } y = x'.$$

We say that the complementation in a Boolean algebra is unique.
**Proof**
We have:

$$y = y1 = y(x \vee x') = yx \vee yx' = x'y \vee 0 = x'y \vee x'x =$$

$$= x'(y \vee x) = x'1 = x'.$$

**6. Theorem**
For every $x, y \in \mathcal{B}$, we have:

$$\begin{array}{rcll} (x \vee y)' & = & x'y', & (1.10) \\ (xy)' & = & x' \vee y', & (1.10') \\ (x')' & = & x, & (1.11) \\ x \vee x'y & = & x \vee y, & (1.12) \\ x\,(x' \vee y) & = & xy, & (1.12'). \end{array}$$

We call (1.10) and (1.10′) **De Morgan laws**, (1.11) the law of **double negation** and (1.12) and (1.12′) the laws of **Boolean absorption**.
**Proof**
We shall prove only the properties (1.10), (1.11) and (1.12); the others follow by using the principle of duality.
From (1.6) and (1.6′), we have:

$$x' \vee (x')' = 1 \text{ and } x'\,(x')' = 0,$$

hence, $(x')' = x$ by the above lemma. By using the same lemma, to prove De Morgan laws, we need to prove

$$(x \vee y) \vee x'y' = 1 \text{ and } (x \vee y)x'y' = 0.$$

We have:

$$(x \vee y) \vee x'y' = ((x \vee y) \vee x'))\,((x \vee y) \vee y') =$$

$$= (y \vee (x \vee x'))\,(x \vee (y \vee y')) = (y \vee 1)(x \vee 1) = 1 \vee 1 = 1,$$

and:

$$(x \vee y)x'y' = xx'y' \vee yx'y' = xx'y' \vee x'yy' =$$

$$= 0y' \vee x'0 = 0 \vee 0 = 0.$$

For the Boolean absorption, we have:

$$x \vee y = 1(x \vee y) = (x \vee x')(x \vee y) = (x \vee xy) \vee x'y = x \vee x'y.$$

**7. Definition**
Let $x, y \in \mathcal{B}$. Then, by definition:

$$x \leq y \Leftrightarrow x \vee y = y.$$

We read $\leq$ **"less or equal"**. The converse relation is **"greater or equal"**:

$$x \geq y \Leftrightarrow y \leq x.$$

**8. Observation**
For every $x, y \in \mathcal{B}$, we have:

$$x \leq y \Leftrightarrow xy = x.$$

**Proof** If $x \leq y$, then $x \vee y = y$, hence:

$$xy = x(x \vee y) = x.$$

Conversely, if $xy = y$, then:

$$x \vee y = xy \vee y = y,$$

hence $x \leq y$.

We can extend the principle of duality to relations $\leq$ and $\geq$:

**9. Theorem (the extended principle of duality)**
Let $\aleph = \{\mathcal{B}, \vee, \cdot, ', 0, 1\}$ be a Boolean algebra and let $P$ be a property expressed using $\vee$, $\cdot$, $'$, 0, 1, $\leq$ and $\geq$. Let $P^d$ be the dual property of $P$, obtained by interchanging $\vee$ with $\cdot$, 0 with 1 and $\leq$ with $\geq$. We have:

$$P \text{ is true } \Leftrightarrow P^d \text{ is true.}$$

**Proof**

From theorem 3 and from the above observation, we get:

$$x \leq y \text{ in the algebra } \aleph \Leftrightarrow y \leq x \text{ in the algebra } \aleph^d.$$

**10. Theorem**

Let $\{\mathcal{B}, \vee, \cdot, ', 0, 1\}$ be a Boolean algebra; then for every $x, y, z \in \mathcal{B}$, we have:

$$\begin{array}{lllllll} & x \leq x, & & & & & (1.13) \\ \text{if} & x \leq y & \text{and} & y \leq x, & \text{then} & x = y, & (1.14) \\ \text{if} & x \leq y & \text{and} & y \leq z, & \text{then} & x \leq z, & (1.15). \end{array}$$

We say that the relation $\leq$ is **reflexive**, **antisymmetric** and **transitive**, respectively, hence $\leq$ is a relation of **order** (generally, $\leq$ is not a relation of **total order**).

**Proof**

From $x \vee x = x$, we get $x \leq x$.

If $x \leq y$ and $y \leq x$, then:

$$x = x \vee y = y.$$

From $x \leq y$ and $y \leq z$, we get:

$$x \vee y = y \text{ and } y \vee z = z,$$

hence:

$$x \vee z = x \vee (y \vee z) = (x \vee y) \vee z = y \vee z = z,$$

which proves that $x \leq z$.

Other properties of $\leq$ are given in the following theorem:

**11. Theorem**

For every $x, y, z \in \mathcal{B}$, we have:

$$\begin{array}{llllllll} x & \leq & x \vee y, & & & & & (1.16) \\ xy & \leq & x, & & & & & (1.16') \\ x & \leq & z & \text{and} & y \leq z & \Leftrightarrow & x \vee y \leq z, & (1.17) \\ t & \leq & x & \text{and} & t \leq y & \Leftrightarrow & t \leq xy; & (1.17') \end{array}$$

further:

$$\begin{array}{llllll} \text{if} & x \le y & \text{then} & x \vee z & \le & y \vee z, \qquad (1.18) \\ \text{if} & x \le y & \text{then} & xz & \le & yz, \qquad (1.18') \\ & 0 \le x, & & & & \qquad (1.19) \\ & x \le 1, & & & & \qquad (1.19') \\ & x \le y & \Leftrightarrow & x' \vee y & = & 1, \qquad (1.20) \\ & x \le y & \Leftrightarrow & xy' & = & 0; \qquad (1.20') \end{array}$$

moreover,

$$\begin{array}{lllll} x = y & \Leftrightarrow & (x' \vee y)(x \vee y') & = & 1, \qquad (1.21) \\ x = y & \Leftrightarrow & xy' \vee x'y & = & 0, \qquad (1.21') \\ x \le y & \Leftrightarrow & y' \le x', & & \qquad (1.22) \end{array}$$

It results that $x \vee y$ is the **least upper bound** of $x$ and $y$ and that $xy$ is the **greatest lower bound** of $x$ and $y$. The elements 0 and 1 are the **least** and the **greatest**, respectively, elements of $\mathcal{B}$.

**Proof**

Properties (1.16), (1.17), (1.8) and (1.19) are obvious. We prove (1.20); if $x \le y$, then $x \vee y = y$, hence:

$$x' \vee y = x' \vee (x \vee y) = (x' \vee x) \vee y = 1 \vee y = y.$$

Conversely, if $x' \vee y = 1$, then:

$$x \vee y = (x \vee y) \cdot 1 = (x \vee y)(x' \vee y) = xx' \vee y = 0 \vee y = y,$$

hence $x \le y$. To prove (1.21), we use (1.20) and the antisymmetry:

$$x = y \Leftrightarrow x \le y \text{ and } y \le x \Leftrightarrow$$

$$\Leftrightarrow x' \vee y = 1 \text{ and } x \vee y' = 1 \Leftrightarrow (x' \vee y)(x \vee y') = 1.$$

Finally, (1.22) results directly from (1.20). The rest of the properties follow using the extended principle of duality.

**12. Exercise**

Prove that for every $a, b, c \in \mathcal{B}$, we have:

**(i)** $ab \le c \Leftrightarrow a \le b' \vee c$.

**(ii)** $ab \le c \vee d \Leftrightarrow ac' \le b' \vee d$.

**Solution**
**(i)** If $ab \leq c$, then $(\vee ab')$:

$$ab \vee ab' \leq c \vee ab' \leq c \vee b'.$$

The left member is $a$, hence we get $a \leq b' \vee c$.
Conversely, if $a \leq b' \vee c$, then $(\ \cdot b)$:

$$ab \leq b(b' \vee c) = bc \leq c.$$

**(ii)** If $ab \leq c \vee d$, then, $(\vee ab')$:

$$ab \vee ab' \leq ab' \vee (c \vee d),$$

hence:

$$a \leq ab' \vee c \vee d.$$

It results $(\cdot c')$:

$$ac' \leq ab'c' \vee c'(c \vee d) = ab'c' \vee cd \leq b' \vee cd \leq b'.$$

Conversely, if $ac' \leq b' \vee d$, then we get $(\vee ac)$:

$$a \leq b' \vee d \vee ac,$$

hence $(\cdot b)$:

$$ab \leq bd \vee abc = b(d \vee ac) \leq b(d \vee c) \leq c \vee d.$$

**13. Exercise**
Let $a, b \in \mathcal{B}$; prove that the following assertions are equivalent:
**(i)** $a \leq b$.
**(ii)** $\forall c \in \mathcal{B}$, $ac \leq bc$ and $a \vee c \leq b \vee c$.
**(iii)** $\exists c \in \mathcal{B}$, $ac \leq bc$ and $a \vee c \leq b \vee c$.
**(iv)** $\exists c \in \mathcal{B}$, $ac \leq b$ and $a \leq b \vee c$.
**Solution**
The implications (i)⇒(ii) ⇒(iii) ⇒ (iv) are obvious. We prove (iv)⇒ (i).
Let us suppose that there is $c \in \mathcal{B}$ such that:

$$ac \leq b \text{ and } a \leq b \vee c.$$

Using the exercise 12(i), we get:

$$ac \leq b \Leftrightarrow a \leq c' \vee b.$$

But $a \leq c' \vee b$ and $a \leq b \vee c$ implies :

$$a \leq (c' \vee b)(b \vee c) = c'b \vee b \vee bc = b.$$

**14. Exercise**

Let $a, b \in \mathcal{B}$; the following assertions are equivalent:
**(i)** $a = b$.
**(ii)** $a \vee b \leq ab$.
**(iii)** $a \vee b = ab$.
**(iv)** $\forall c \in \mathcal{B}, ac = bc$ and $a \vee c = b \vee c$.
**(v)** $\exists c \in \mathcal{B}, ac = bc$ and $a \vee c = b \vee c$.

**15. Definition**

Let $\{\mathcal{B}, \vee, \cdot, ', 0, 1\}$ be a Boolean algebra and let $\mathcal{B}_o \subseteq \mathcal{B}$ such that:
**(i)** $0, 1 \in \mathcal{B}_o$.
**(ii)** If $x, y \in \mathcal{B}_o$, then $x \vee y$, $xy$ and $x'$ are in $\mathcal{B}_o$.
We say that $\mathcal{B}_o$ is a **subalgebra** in $\mathcal{B}$.
It results that $\{\mathcal{B}_o, \vee, \cdot, ', 0, 1\}$ is itself a Boolean algebra.

**16. Observation**

It can be proved (exercise) that $\mathcal{B}_o$ is a subalgebra in $\mathcal{B}$ if and only if the following conditions are satisfied:
**(i)** $\mathcal{B}_o \neq \emptyset$.
**(ii)** If $x, y \in \mathcal{B}_o$, then $x \vee y$ and $x'$ are in $\mathcal{B}_o$.

**17. Example**

**(i)** The two-element Boolean algebra, $\mathcal{B}_2$ (see example 2(i)), is a subalgebra in every Boolean algebra.
**(ii)** Let $\mathcal{B}$ be a Boolean algebra different from the two element algebra and let $a \in \mathcal{B}$ such that $a \neq 0$ and $a \neq 1$. Let $\mathcal{B}_a = \{0, a, a', 1\}$. Then $\mathcal{B}_a$ is a subalgebra; it is called the **subalgebra generated by** $a$.
More generally, let $\mathcal{F} \subset \mathcal{B}$; we can define **the subalgebra generated by**

$\mathcal{F}$, denoted $\mathcal{B}_{\mathcal{F}}$, as follows:

$$\mathcal{B}_{\mathcal{F}} = \bigcap \{\mathcal{H}\,;\, \mathcal{H} \text{ subalgebra }\,,\ \mathcal{H} \supseteq \mathcal{F}\}.$$

It results that $\mathcal{B}_{\mathcal{F}}$ is the least subalgebra containing $\mathcal{F}$; of course, it must be shown first that an intersection of subalgebras is a subalgebra.

**18. Definition**

Let $\{\mathcal{B},\, \vee,\, \cdot,\, ',\, 0,\, 1\}$ and $\{\mathcal{G},\, \cup,\, \cap,\, ^\star,\, \odot,\, \dagger\}$ be two Boolean algebras. A map

$$h : \mathcal{B} \mapsto \mathcal{G},$$

is called a **homomorphism** (of Boolean algebras) if for every $x, y \in \mathcal{B}$, the following conditions are satisfied:

$$\begin{array}{rcccc} h(0) & = & \odot & & \\ h(1) & = & \dagger & & \\ h(x \vee y) & = & h(x) & \cup & h(y) \\ h(xy) & = & h(x) & \cap & h(y) \\ h\,(x') & = & (h(x))^\star & & \end{array}$$

If $h$ is a bijective homomorphism, then it it called an **isomorphism** and the Boolean algebras $\mathcal{B}$ and $\mathcal{G}$ are called, in this case, **isomorphic**.

**19. Observation**

It can be proved (exercise) the following properties:
**(i)** The conditions:

$$h(x \vee y) = h(x) \cup h(y) \text{ and } h\,(x') = (h(x))^\star$$

are sufficient for $h$ to be a homomorphism.
**(ii)** If $h$ is an isomorphism, then its inverse, $h^{-1}$, is an isomorphism, too.

**20. Exercise**

Prove that the range of a homomorphism

$$h : \mathcal{B} \mapsto \mathcal{G}$$

is a subalgebra in $\mathcal{G}$.

**21. Definition**

Let $\{\mathcal{B}, \vee, \cdot, ', 0, 1\}$ be a Boolean algebra and let $a, b \in \mathcal{B}$ be two elements such that $a \leq b$. The set:

$$[a, b] = \{x \in \mathcal{B}\,;\, a \leq x \leq b\}$$

is called **the interval** (or the segment) $[a, b]$.
Obviously, $\mathcal{B} = [0, 1]$.

**22. Exercise**

With the above notations, let $a, b \in \mathcal{B}$ such that $a \leq b$. For every $x \in [a, b]$, let

$$x^{\sharp} = a \vee bx'.$$

Prove that $\{[a, b], \vee, \cdot, ^{\sharp}, a, b\}$ is Boolean algebra.
Is it a subalgebra in $\mathcal{B}$?

We conclude this section with a representation theorem for Boolean algebras.

**23. Theorem (M.Stone)**

**(i)** If $\mathcal{B}$ is an arbitrarily **finite** Boolean algebra, then there is a **finite** set $X$ such that $\mathcal{B}$ is isomorphic with the usual Boolean algebra
$\{\mathcal{P}(X), \cup, \cap, ', \emptyset, X\}$; (see example 2(iii)).
**(ii)** If $\mathcal{B}$ is an arbitary Boolean algebra, then there is a set $Y$ and an injective homomorphism

$$h : \mathcal{B} \mapsto \mathcal{P}(Y).$$

This means that the Boolean algebra $\mathcal{B}$ is isomorphic with a subalgebra of $\mathcal{P}(Y)$; of course, this subalgebra is the range of $h$.

**24. Corollary**

**(i)** If $\mathcal{B}$ is a finite Boolean algebra, then there is a natural number $n$ such that the number of elements of $\mathcal{B}$ is $2^n$.
**(ii)** Moreover, all the Boolean algebras with the same number (finite) of elements are isomorphic.

## 5.2 Boolean functions

In the following, $\{\mathcal{B}, \vee, \cdot, ', 0, 1\}$ is, as usual, a Boolean algebra.

**25. Definition**
By a **Boolean function** of $n$ variables we mean any function

$$f : \mathcal{B}^n \mapsto \mathcal{B},$$

which can be obtained by the following rules:
**(i)** For every $a \in \mathcal{B}$, the **constant function**, $f_a$, defined by:

$$f_a : \mathcal{B}^n \mapsto \mathcal{B},\ f_a(x_1, .., x_n) = a,$$

is a Boolean function.
**(ii)** For every $i \in \{1, 2, .., n\}$, the **projection function**, $p_i$, defined by:

$$p_i : \mathcal{B}^n \mapsto \mathcal{B},\ p_i(x_1, .., x_n) = x_i,$$

is a Boolean function.
**(iii)** For every Boolean functions of $n$ variables, $f$ and $g$, the following functions are Boolean functions, too:

$$f \vee g : \mathcal{B}^n \mapsto \mathcal{B},\ (f \vee g)(x_1, ..x_n) = f(x_1, .., x_n) \vee g(x_1, .., x_n)$$

$$f \cdot g : \mathcal{B}^n \mapsto \mathcal{B},\ (fg)(x_1, .., x_n) = f(x_1, .., x_n)g(x_1, .., x_n)$$

$$f' : \mathcal{B}^n \mapsto \mathcal{B},\ f'(x_1, .., x_n) = (f(x_1, .., x_n))'.$$

Of course, the functions $f \vee g$, $f \cdot g$ and $f'$ are called the **disjunction of** $f$ **and** $g$, the **conjunction of** $f$ **and** $g$ and the **negation of** $f$, respectively.
**(iv)** Any Boolean function of $n$ variables is obtained by applying the rules **(i)**,**(ii)** and **(iii)** a finite number of times.

A Boolean function is called a **simple Boolean function** if it is obtained by the rules **(ii)**, **(iii)** and:
**(v)** Any simple Boolean function of $n$ variables is obtained by applying the rules **(ii)** and **(iii)** a finite number of times.
This means that a constant function is simple Boolean function if and only if it is 0 or 1.

We mention that there are a functions which are not Boolean functions.

**26. Notations**

In the following, we shall denote by $X, Y, ..$ arbitrary vectors in $\mathcal{B}^n$:

$$X = (x_1, .., x_n) \text{ and } Y = (y_1, .., y_n).$$

If the components of a vector belong to $\{0, 1\}$, then it is called an **elementary vector**. We denote elementary vectors by the letters $A, B, ..$ and their components by $\alpha_1, \alpha_2, ..$:

$$A = (\alpha_1, .., \alpha_n), \text{ where } \alpha_i \in \{0, 1\},$$

or, equivalently,

$$A \in \{0, 1\}^n.$$

Of course, there are $2^n$ elementary vectors. For an elementary vector $A = (\alpha_1, .., \alpha_n)$, we denote by $A'$ the vector:

$$A' = (\alpha_1', .., \alpha_n').$$

If $\alpha \in \{0, 1\}$ and $x \in \mathcal{B}$, the notation $x^\alpha$ means:

$$x^\alpha = \begin{cases} x' & \text{if} \quad \alpha = 0 \\ x & \text{if} \quad \alpha = 1 \end{cases}$$

If $X = (x_1, .., x_n)$ and $A = (\alpha_1, .., \alpha_n)$, the notation $X^A$ means the conjunction of all the elements $x_1^{\alpha_1}, .., x_n^{\alpha_n}$:

$$X^A = x_1^{\alpha_1} \cdots x_n^{\alpha_n}.$$

The notation $\left(\bigvee X^A\right)$ means the disjunction of all the elements $x_1^{\alpha_1}, .., x_n^{\alpha_n}$:

$$\left(\bigvee X^A\right) = x_1^{\alpha_1} \vee ... \vee x_n^{\alpha_n}.$$

If $\{a_A\}_{A \in \{0,1\}^n}$ is a set of $2^n$ elements in $\mathcal{B}$, then, the expressions

$$\bigvee_A a_A X^A \text{ and } \bigwedge_A \left(a_A \vee \left(\bigvee X^A\right)\right),$$

mean that the disjunction and the conjunction, (denoted here by $\bigwedge$) respectively, is over all the $2^n$ elementary vectors $A \in \{0, 1\}^n$.

**27. Examples**

**(i)** If $n = 1$, then $X = x \in \mathcal{B}$ and the elementary vectors are $A = 1$ and $A = 0$. We have:

$$\bigvee_A a_A X^A = ax \vee bx',$$

$$\bigvee_A a_A X^A = (c \vee x)(b \vee x'),$$

where, $a, b, c, d \in \mathcal{B}$.

**(ii)** If $n = 2$, we have $X = (x, y)$ and the elementary vectors are:

$$A = (0,0),\ B = (0,1),\ C = (1,0),\ D = (1,1).$$

We have:

$$X^A = x'y',\ X^B = x'y,\ X^C = xy',\ X^D = xy.$$

and:

$$\bigvee_A a_A X^A = ax'y' \vee bx'y \vee cx'y \vee dxy,$$

$$\bigwedge_A \left(a_A \vee \left(\bigvee X^A\right)\right) = (s \vee x' \vee y')(t \vee x' \vee y \vee u \vee x \vee y')(v \vee x \vee y),$$

where, $a, b, c, d, s, t, u, v \in \mathcal{B}$.

The following theorem is a fundamental result in the study of Boolean functions (without proof).

**28. Theorem (canonical forms)**

**(i)** A function $f : \mathcal{B}^n \mapsto \mathcal{B}$ is a Boolean function if and only if it can be written in the **canonical disjunctive form**:

$$f(X) = \bigvee_A a_A X^A,$$

the coefficients $a_A$ being

$$a_A = f(A).$$

**(ii)** A function $f : \mathcal{B}^n \mapsto \mathcal{B}$ is a Boolean function if and only if it can be written in the **canonical conjunctive form**:

$$f(X) = \bigwedge_A \left(b_A \vee \left(\bigvee X^A\right)\right),$$

the coefficients $b_A$ being

$$b_A = f(A').$$

Of course, assertion **(ii)** is the dual of **(i)**.

**29. Corollary**

**(i)** A Boolean function is determined by its values on the elementary vectors.

**(ii)** The values of a Boolean function $f$ satisfy the inequalities:

$$\bigwedge_A f(A) \leq f(X) \leq \bigvee_A f(A), \forall X \in \mathcal{B}^n.$$

**(iii)** A Boolean function $f$ is a simple Boolean function if and only if for every elementary vector $A$, we have:

$$f(A) \in \{0, 1\}.$$

**30. Examples**

**(i)** Let $f(x) = ax \vee bx' \vee c$. We compute:

$$f(0) = b \vee c,\ f(1) = a \vee c,$$

hence the canonical disjunctive form is:

$$f(x) = (b \vee c)x' \vee (a \vee c)x,$$

and the canonical conjunctive form is:

$$f(x) = (b \vee c \vee x)(a \vee c \vee x').$$

**(ii)** Let $f(x) = (a \vee x)(b'x \vee x')$. We compute:

$$f(0) = a, \text{ and } f(1) = b',$$

hence the canonical disjunctive form is:

$$f(x) = ax' \vee b'x,$$

and the canonical conjunctive form is:

$$f(x) = (a \vee x)(b' \vee x').$$

**(iii)** Let $f : \mathcal{B}^2 \mapsto \mathcal{B}$,

$$f(x,y) = ab'(x \vee y')(a'xy \vee b(x' \vee y)') \vee a \vee (x \vee b).$$

We compute:

$$f(0,0) = a \vee b,\ f(0,1) = a \vee b,\ f(1,0) = 1,\ f(1,1) = a,$$

hence the canonical forms are:

$$f(x,y) = (a \vee b)x'y' \vee (a \vee b)x'y \vee xy' \vee axy,$$

and:

$$f(x,y) = ((a \vee b) \vee x \vee y)((a \vee b) \vee x \vee y')(1 \vee x \vee y')(a \vee x' \vee y') =$$

$$= ((a \vee b) \vee x \vee y)((a \vee b) \vee x \vee y')(a \vee x' \vee y').$$

**(iv)** Let $f : \mathcal{B}^3 \mapsto \mathcal{B}$,

$$f(x,y,z) = (x \vee y)(a \vee x' \vee (xz)').$$

We compute:

$$f(0,0,0) = 0,\ f(0,0,1) = 0,\ f(0,1,0) = 1,\ f(1,0,0) = 1,$$

$$f(0,1,1) = 1,\ f(1,0,1) = a,\ f(1,1,0) = 1,\ f(1,1,1) = a.$$

The canonical forms are:

$$f(x,y,z) = x'yz' \vee xy'z' \vee x'yz \vee axy'z \vee xyz' \vee axyz,$$

and

$$f(x,y,z) = (x \vee y \vee z)(x \vee y \vee z')(a \vee x' \vee y \vee z')(a \vee x' \vee y' \vee z').$$

**31. Exercise**
Let $f : \mathcal{B}^4 \mapsto \mathcal{B}$, defined by:

$$f(a,b,x,y,z) = axy \vee bxy' \vee b'x'y \vee a'x'y'.$$

Prove that:

$$f(a,b,x,y) = f(x,y,a,b) = f(a,b',y,x) =$$

$$= f(a',b',x',y') = f(x,y',b,a) = f(x',y',a',b').$$

## 5.3 Boolean equations

As usual, $\{\mathcal{B}, \vee, \cdot, ', 0, 1\}$ is a Boolean algebra.

**32. Definition**

By a **Boolean equation in $n$ unknowns** we mean an equation of the form

$$f(X) = g(X),$$

where $f$ and $g$ are Boolean functions of $n$ variables.

The vector $X = (x_1, .., x_n)$ is the **unknown**. A **solution** of the equation is any vector $X_o$ such that $f(X_o) = g(X_o)$.

By a **Boolean inequality in $n$ unknowns** we mean an inequality of the form

$$f(X) \leq g(X),$$

with $f$ and $g$ as above; of course, a solution of the inequality is any vector $X_o$ such that $f(X_o) \leq g(X_o)$.

By a **system of Boolean equations in $n$ unknowns** we mean a system of the form:

$$f_i(X) = g_i(X), \ \forall i \in \{1, 2, .., m\},$$

where, for every $i \in \{1, 2, .., m\}$, $f_i$ and $g_i$ are Boolean functions of $n$ variables.

A solution of the system is any vector $X_o$ which satisfies all the equations of the system.

Analogously is defined the system of Boolean inequalities.

**To solve** a Boolean equation (or system, etc,) it means to determine all the solutions.

An equation (or system, etc) is called **consistent** if it has at least a solution (or, equivalently, the set of all its solutions is non-empty). Otherwise, the equation (or system, etc) is called **inconsistent**.

Two equations (or systems, etc) are said to be **equivalent** if they have the same sets of solutions.

**32. Theorem**

Every Boolean equation of $n$ unknowns (or inequality, or system of equations, or system of inequalities) is equivalent to a **single Boolean equation** of the form:

$$f(X) = 0,$$

where $f$ is a Boolean function of $n$ variables.

**Proof**

Let us consider the Boolean equation

$$f(X) = g(X).$$

By using the equivalence: $x = y \Leftrightarrow xy' \vee x'y = 0$, (section 1, theorem 11, $(1.21')$) it results that the equation is equivalent with:

$$f(X)g'(X) \vee f'(X)g(X) = 0.$$

Let us consider now the inequality:

$$f(X) \leq g(X).$$

By using the equivalence: $x \leq y \Leftrightarrow xy' = 0$, (section 1, theorem 11 $(1.20')$), it results that the inequality is equivalent with:

$$f(X)g'(X) = 0.$$

It results that every system of equations (or inequalities) is equivalent with a system of equations of the form:

$$f_i(X) = 0, \ \forall i \in \{1, 2, .., m\}.$$

By using the equivalence $x = y = 0 \Leftrightarrow x \vee y = 0$, (section 1,theorem 4 $(1.9)$), it results that the system is equivalent with the single equation:

$$f_1(X) \vee .... \vee f_m(X) = 0.$$

As a consequence of the above theorem, we shall restrict our study (without loss of generality) to equations of the form $f(X) = 0$.

**33. Examples**

**(i)** The equation

$$x \vee yz' = x \vee z'$$

is equivalent with:

$$(x \vee yz')(x \vee z')' \vee (x \vee yz')'(x \vee z') = 0,$$

$$(x \vee yz')x'z \vee x(y' \vee z)(x \vee z') = 0,$$

$$xy' \vee xy'z' \vee xz = 0,$$
$$xy' \vee xz = 0,$$

**(ii)** The system:

$$x' \vee yz \leq xz',\ xy \vee z' = 1,$$

is equivalent with:

$$(x' \vee yz)(xz')' = 0,\ (xy \vee z')' = 0,$$
$$x' \vee yz = 0,\ x'z \vee y'z = 0.$$

Of course, in this moment we can solve the system:

$$x' = 0,\ yz = 0,\ y'z = 0,$$

and the solution is:

$$x = 1,\ z = 0,\ y \in \mathcal{B}.$$

**34. Exercise**
Let $a, b, c, p, r, s \in \mathcal{B}$; then the equation:

$$ax \vee bx' \vee c = px \vee rx' \vee s,$$

is equivalent with:

$$[(a \vee c)p's' \vee a'c'(p \vee s)]x \vee [(b \vee c)r's' \vee b'c'(r \vee s)] = 0.$$

**Solution**

We put first both members of the equation in the canonical disjunctive form, hence the equation is equivalent with:

$$(a \vee c)x \vee (b \vee c)x' = (p \vee s)x \vee (r \vee s)x'.$$

Further:

$$[(a \vee c)x \vee (b \vee c)x']\,[(p \vee s)x \vee (r \vee s)x']' \cdot$$
$$\cdot [(a \vee c)x \vee (b \vee c)x']'\,[(p \vee s)x \vee (r \vee s)x'] = 0.$$

We compute:

$$[(a \vee c)x \vee (b \vee c)x']' = (a'c' \vee x')(b'c' \vee x) =$$

$$= a'c'x \vee b'c'x' \vee a'b'c'.$$

Analogously:

$$[(p \vee s)x \vee (r \vee s)x']' = p's'x \vee r's'x \vee p'r's'.$$

Introducing the above expressions into the equation, we get the desired form.

**35. Exercise**
Let $a, b \in \mathcal{B}$; then the system:

$$xy' = ab',\ x'y = a'b$$

is equivalent with the equation:

$$(a'b \vee ab')xy \vee (a' \vee b)xy' \vee (a \vee b')x'y \vee (ab' \vee a'b)x'y' = 0.$$

**Solution**
The system is equivalent with:

$$xy'(ab')' \vee (xy')'ab' = 0,\ x'y(a'b)' \vee (x'y)'a'b = 0;$$

by using De Morgan laws, we get:

$$xy'(a' \vee b) \vee (x' \vee y)ab' = 0,\ x'y(a \vee b') \vee (x \vee y')a'b = 0.$$

We reduce to a single equation:

$$a'bx \vee ab'x' \vee ab'y \vee a'by' \vee (a' \vee b)xy' \vee (a \vee b'x'y) = 0.$$

If we bring the left member of this equation to the canonical disjunctive form, then we get the required conclusion.

**36. Exercise**
Let $a, b, c \in \mathcal{B}$; then the system:

$$a \leq x,\ b \leq y,\ xy = c,$$

is equivalent with:

$$c'xy \vee (b \vee c)xy' \vee (a \vee c)x'y \vee (a \vee b \vee c)x'y' = 0.$$

**Solution**

The system is equivalent with the system:

$$ax' = 0,\ by' = 0,\ c'xy \vee cx' \vee cy' = 0,$$

hence, by reducing to a single equation, we get:

$$(a \vee c)x' \vee (b \vee c)y' \vee c'xy = 0.$$

The canonical disjunctive form of the left member of the above equation is the required form of the equation.

**37. Exercises**

Reduce the following systems to a single equation of the form

$$f(X) = 0.$$

**(i)** $yz = a \vee bc,\ zx = b \vee ca,\ xy = c \vee ab.$
**(ii)** $xy = a,\ x \vee y = b.$
**(iii)** $x(y \vee z) = b \vee c,\ y(z \vee x) = c \vee a,\ z(x \vee y) = a \vee b.$

We shall study now the **Boolean equation in one unknown**, written in the canonical disjunctive form:

$$ax \vee bx' = 0.$$

If we denote $f(x) = ax \vee bx'$, then we have $a = f(1)$ and $b = f(0)$.

**38. Lemma**

The following assertions are equivalent:
**(i)** $ax \vee bx' = 0.$
**(ii)** $b \leq x \leq a'.$
**(iii)** $x = a'x \vee bx'.$

**Proof**

**(i)**⇒ **(ii)**. The equality:

$$ax \vee bx' = 0$$

is equivalent (see theorem 4(1.9)) with the equalities:

$$ax = 0 \text{ and } bx' = 0,$$

which are equivalent (see section1, theorem 11(1.20′)) with:

$$x \leq a' \text{ and } b \leq x,$$

hence we proved **(ii)**.

**(ii)** ⇒**(iii)**. From the inequalities:

$$b \leq x \text{ and } x \leq a',$$

we get:

$$bx' = 0 \text{ and } a'x = x,$$

hence:

$$a'x \vee bx' = x \vee 0 = x,$$

which proves **(iii)**.

**(iii)**⇒**(i)** By conjugation with $x'$ and with $ax$ of the equality:

$$x = a'x \vee bx',$$

we deduce (respectively):

$$0 = bx' \text{ and } ax = 0,$$

hence:

$$ax \vee bx' = 0.$$

**39. Theorem**

The Boolean equation in one unknown:

$$ax \vee bx' = 0$$

is consistent if and only if

$$ab = 0.$$

If this condition is satisfied, then the set of all the solutions of the equation is the interval:

$$x \in [b, a'],$$

or, equivalently, in a parametric form:

$$x = a't \vee b, \text{ where } t \in \mathcal{B}.$$

**Proof**

Let us assume that the equation is consistent, hence there is $x_o \in \mathcal{B}$ such that:

$$ax_o \vee bx_o' = 0.$$

From the above lemma we get $b \leq a'$, hence:

$$ab = 0.$$

Conversely if $ab = 0$, then $x = b$ is a solution of the equation:

$$ab \vee bb' = 0.$$

Let us now suppose that the condition $ab = 0$ is fulfilled. The fact that the set of all the solutions is the interval $[b, a']$ was proved in the above lemma. To prove the parametric formula of the solutions, let us consider an arbitrary $t \in \mathcal{B}$; then $x = a't \vee b$ is a solution:

$$a(a't \vee b) \vee b(a't \vee b)' = ab \vee b(a't)'b' = 0,$$

because $ab = 0$.
Conversely, if $x \in \mathcal{B}$ is an arbitrary solution of the equation, then, by lemma 8 we have:

$$b \leq x \leq a',$$

hence $x$ can be written as:

$$x = x \vee b = a'x \vee b.$$

It results that there is $t \in \mathcal{B}$ such that $x = a't \vee b$; in fact, $t = x$.

We apply now the above method to solve several equations (inequalities) in one unknown.

**40. Examples**

**(i)** Solve the inequality:

$$x \vee c \geq s$$

The inequality is equivalent with the equation:

$$(x \vee c)'s = 0,$$

hence we must solve:

$$sc'x' = 0.$$

We observe that the consistency condition is fulfilled:

$$0 \cdot (sc') = 0.$$

The solution is:

$$x \in [sc', 1],$$

or, in parametric form:

$$x = t \vee sc',\ t \in \mathcal{B}.$$

**(ii)** Let us consider the equation:

$$x \vee c = s,$$

which is equivalent with:

$$s'x \vee s'c \vee sc'x' = 0,$$

or, in the canonical disjunctive form:

$$s'x \vee (s'c \vee sc')x' = 0.$$

The consistency condition is:

$$s'c = 0,$$

which is not always fulfilled; if it holds, the solution is:

$$x \in [sc', s],$$

or in parametric form:

$$x = st \vee sc',\ t \in \mathcal{B}.$$

**(iii)** Let us solve the equation:

$$ax = s.$$

It is equivalent with:

$$(s'a \vee sa')x \vee sx' = 0.$$

The consistency condition is:

$$sa' = 0.$$

If the above condition is satisfied, the solution is:

$$x \in [s, s \vee a'],$$

or, in parametric form:

$$x = (s \vee a')t \vee s,\ t \in \mathcal{B}.$$

**(iv)** More generally, let us solve the equation:

$$ax \vee c = s.$$

The equation is equivalent with:

$$[s'(a \vee c) \vee sa'c']x \vee (s'c \vee sc')x' = 0.$$

The consistency condition is:

$$s'c \vee a'c's = 0,$$

and the solution (if the above condition holds) is:

$$x \in [sc', s \vee a'],$$

or, in parametric form:

$$x = (s \vee a')t \vee sc',\ t \in \mathcal{B}.$$

**(v)** Let us solve now:

$$ax \vee bx' \vee c = 0.$$

The canonical disjunctive form is (see section 2,example 6(i)):

$$(a \vee c)x \vee (b \vee c)x' = 0.$$

The equation is consistent if and only if:

$$(a \vee c)(b \vee c) = 0,$$

or:

$$ab \vee c = 0,$$

hence:

$$ab = 0 \text{ and } c = 0.$$

If these conditions are satisfied, the solution is:

$$x \in [b, a'],$$

or, equivalently:

$$x = a't \vee b,\ b \in \mathcal{B}.$$

**(vi)** Let us solve the equation:

$$ax = bx.$$

The equation is equivalent with:

$$(ab' \vee a'b)x = 0,$$

which is always consistent; the solution is:

$$x \in [0, ab \vee a'b'],$$

or, in parametric form:

$$x = (ab \vee a'b')t,\ t \in \mathcal{B}.$$

**41. Example**

Let us solve now the general equation:

$$ax \vee bx' \vee c = px \vee rx' \vee s.$$

The canonical disjunctive form is (see example 40):

$$[(a \vee c)p's' \vee a'c'(p \vee s)]x \vee [(b \vee c)r's' \vee b'c'(r \vee s)] = 0.$$

The consistency condition is:

$$(ab \vee c)p'r's' \vee ab'c'p'rs' \vee a'bc'pr's' \vee a'b'c'(pr \vee s) = 0.$$

If this condition is fulfilled, the solution is:

$$x \in [(b \vee c)r's' \vee b'c'(r \vee s), a'c'p's' \vee (a \vee c)(p \vee s),$$

or, in parametric form:

$$x = [a'c'p's' \vee (a \vee c)(p \vee s)]t \vee (b \vee c)r's' \vee b'c'(r \vee s).$$

We now return to the case of several variables. The consistency condition is given in the following theorem.

**42. Theorem**

Let $f : \mathcal{B}^n \mapsto \mathcal{B}$, be a Boolean function of $n$ variables. The equation (in $n$ unknowns):

$$f(X) = 0$$

is consistent if and only if:

$$\bigwedge_{A \in \{0,1\}^n} f(A) = 0.$$

The result is a generalization to $n$ variables of the consistency condition of the case of a single variable.

**43. Examples**

**(i)** Let us consider the general equation in two unknowns:

$$axy \vee bxy' \vee cx'y \vee dx'y'.$$

The consistency condition is:

$$abcd = 0$$

**(ii)** Let us consider the equation:

$$cxy \vee ayz \vee bxz = 0.$$

If we denote:

$$f(x, y, z) = cxy \vee ayz \vee bxz,$$

then the consistency condition is:

$$f(0,0,0)f(0,0,1)f(0,1,0)f(1,0,0)f(1,1,0)f(1,0,1)f(0,1,1)f(1,1,1) = 0.$$

We have:

$$f(0,0,0)=0,\ f(0,0,1)=0,\ f(0,1,0)=0,\ f(1,0,0)=0,$$

$$f(1,1,0)=c,\ f(1,0,1)=b,\ f(0,1,1)=a,\ f(1,1,1)=a\vee b\vee c.$$

The canonical disjunctive form for $f$ is:

$$f(x,y,z)=cxyz'\vee bxy'z\vee ax'yz\vee(a\vee b\vee c)xyz,$$

and the consistency condition for the equation $f(x,y,z)=0$ is always true.

**44. Proposition**

The range of a Boolean function of $n$ variables:

$$f:\mathcal{B}^n\mapsto\mathcal{B}$$

is the interval:

$$f\left(\mathcal{B}^n\right)=[\bigwedge_{A\in\{0,1\}^n}f(A)\,,\ \bigvee_{A\in\{0,1\}^n}f(A)].$$

**Proof**

It was shown (see section 2,corollary 5) that

$$f(X)\in[\bigwedge_{A\in\{0,1\}^n}f(A)\,,\ \bigvee_{A\in\{0,1\}^n}f(A)],\ \forall X\in\mathcal{B}^n.$$

To prove the converse, we have to prove that for every

$$c\in[\bigwedge_{A\in\{0,1\}^n}f(A)\,,\ \bigvee_{A\in\{0,1\}^n}f(A)],$$

the equation

$$f(X)=c$$

is consistent. The above equation is equivalent with:

$$c'f(X)\vee cf'(X)=0.$$

By using theorem 12, the consistency condition for this equation is:

$$\bigwedge_{A\in\{0,1\}^n}[c'f(A)\vee cf'(A)]=0,$$

which is equivalent with:

$$c' \bigwedge_{A\in\{0,1\}^n} f(A) \vee c \bigwedge_{A\in\{0,1\}^n} f'(A) = 0.$$

This last equality is fulfilled because:

$$\bigwedge_{A\in\{0,1\}^n} f(A) \leq c \Leftrightarrow c' \bigwedge_{A\in\{0,1\}^n} f(A) = 0,$$

and:

$$c \leq \bigvee_{A\in\{0,1\}^n} f(A) \Leftrightarrow c \bigwedge_{A\in\{0,1\}^n} f'(A) = 0.$$

The main result for solving Boolean equations in several variables is the **method of successive eliminations**, which is presented below.

**45. Theorem (the method of successive eliminations)**
Let $f : \mathcal{B}^n \mapsto \mathcal{B}$ be a Boolean function and let

$$f(X) = 0$$

be the associated Boolean equation.
For every $p \in \{1, 2, .., n\}$, we define:

$$f_p(x_1, .., x_p) = \bigwedge_{(\alpha_{p+1},..,\alpha_n)\in\{0,1\}^{n-p}} f(x_1, .., x_p, \alpha_{p+1}, .., \alpha_n),$$

thus, in particular, we have:

$$f_n(x_1, .., x_n) = f(x_1, .., x_n),$$

$$f_{n-1}(x_1, .., x_{n-1}) = f(x_1, .., x_{n-1}, 0)f(x_1, .., x_{n-1}, 1).$$

The method (algorithm) starts with the equation $f(X) = 0$, written as:

$$f_n(x_1, .., x_n) = 0.$$

We write this equation in the equivalent form:

$$f_n(x_1, ..x_{n-1}, 1)x_n \vee f(x_1, .., x_{n-1}, 0)x'_n = 0.$$

If we consider the above equation as an equation in the single variable $x_n$, then its solution is:

$$f_n(x_1, .., x_{n-1}, 0) \leq x_n \leq f'_n(x_1, .., x_{n-1}, 1),$$

if and only if the consistency condition holds:

$$f_n(x_1, .., x_{n-1}, 1) \cdot f(x_1, .., x_{n-1}, 0) = 0.$$

But the above equation (in the unknowns $x_1, .., x_{n-1}$) is precisely:

$$f_{n-1}(x_1, .., x_{n-1}) = 0.$$

By repeating the above step $n-1$ times, we finally get the equation:

$$f_1(x_1) = 0,$$

whose solution is:

$$f_1(0) \leq x_1 \leq f'_1(1),$$

if and only if the consistency condition holds:

$$f_1(0) \cdot f_1(1) = 0.$$

It can be proved that the original equation

$$f(X) = 0$$

is consistent if and only if $f_1(0)f_1(1) = 0$. If this condition is fulfilled, the solutions can be written as recurrent inequalities:

$$f_p(x_1, .., x_{p-1}, 0) \leq x_p \leq f'_p(x_1, .., x_{p-1}, 1),\ \forall p = 1, 2, ..n,$$

or, equivalently, in recurrent parametric form:

$$x_p = f'_p(x_1, .., x_{p-1}, 1)t_p \vee f_p(x_1, .., x_{p-1}, 0),\ \forall t_p \in \mathcal{B},\ \forall p = 1, 2, .., n.$$

**46. Example**

Let us apply the method of successive eliminations to solve the general Boolean equation in two unknowns:

$$axy \vee bxy' \vee cx'y \vee dx'y' = 0.$$

We have:

$$f_2(x, y) = f(x, y),$$

which can be written in the equivalent form:

$$(ax \vee cx')y \vee (bx \vee dx')y' = 0$$

The solution is:

$$bx \vee dx' \leq y \leq (ax \vee cx')' = a'x \vee c'x',$$

provided the consistency condition holds:

$$(ax \vee cx')(bx \vee dx') = 0.$$

This last equation in $x$ is equivalent with:

$$abx \vee cdx' = 0.$$

The consistency condition is:

$$abcd = 0$$

and the solution is:

$$cd \leq x \leq a' \vee b',$$
$$bx \vee dx' \leq y \leq a'x \vee c'x'.$$

**47. Example**

**(i)** Let us solve the system:

$$xy' = ab',\ x'y = a'b.$$

It was shown in exercise 5 that the system is equivalent with the equation:

$$(a'b \vee ab')xy \vee (a' \vee b)xy' \vee (a \vee b')x'y \vee (ab' \vee a'b)x'y' = 0.$$

To reduce $y$, we write the equation as:

$$[(ab' \vee a'b)x \vee (a \vee b')x']y \vee [(a' \vee b)x \vee (ab' \vee a'b)x']y' = 0.$$

The consistency condition for the above equation is:

$$[(ab' \vee a'b)x \vee (a \vee b')x'][(a' \vee b)x \vee (ab' \vee a'b)x'] = 0,$$

hence the equation in $x$ is:

$$[(a'b \vee ab')(a' \vee b)]x \vee [(a \vee b')(ab' \vee a'b)]x' = 0,$$

or in equivalent form:

$$a'bx \vee ab'x' = 0.$$

The consistency condition is always fulfilled:

$$ab' \cdot a'b = 0.$$

The solution is:

$$ab' \leq x \leq a \vee b'.$$

We return now to the equation in $y$, which can be written as:

$$[ab'x \vee (a \vee b')x']y \vee [(a' \vee b)x \vee a'bx']y' = 0,$$

because, from the original system we get:

$$ab'x' = 0 \text{ and } a'bx = 0.$$

The solution is:

$$(a' \vee b)x \vee a'bx' \leq y \leq [ab'x \vee (a \vee b')x']',$$

or, in equivalent form,

$$(a' \vee b)x \vee (a'b)x' \leq y \leq (a' \vee b)x \vee a'b.$$

By using the identity:

$$(a' \vee b)x \vee (a'b)x' = (a' \vee b)x \vee a'b,$$

we get:

$$y = (a' \vee b)x \vee a'b.$$

Finally, the system has the solutions:

$$x \in [ab' \, , \, a \vee b'],$$

$$y = (a' \vee b)x \vee a'b,$$

or, in parametric form:

$$x = (a \vee b')t \vee ab',$$
$$y = (a' \vee b)[(a \vee b')t \vee ab'] \vee a'b = (a' \vee b)t \vee a'b,\ t \in \mathcal{B}.$$

**(ii)** Let us solve now the same system

$$xy' = ab',\ x'y = a'b,$$

in the Boolean subalgebra generated by the elements $a$ and $b$, (see section 1, example 17(ii)). The intersection of the Boolean subalgebra generated by $a$ and $b$ and the interval $[ab'\,,\ a \vee b']$ is the set:

$$\{ab'\,,\ a\,,\ b'\,,\ a \vee b'\}.$$

It results that the solutions of the system in this case are:

$$x_1 = ab',\ y_1 = a'b,$$

$$x_2 = a,\ y_2 = b,$$
$$x_3 = b',\ y_3 = a',$$
$$x_4 = a \vee b',\ y_4 = a' \vee b.$$

**48. Exercise**

**(i)** Let us solve the system:

$$a \leq x,\ b \leq y,\ xy = c.$$

It was shown in exercise 6 that the above system is equivalent with the equation:

$$c'xy \vee (b \vee c)xy' \vee (a \vee c)x'y \vee (a \vee b \vee c)x'y' = 0,$$

which can be written as:

$$[c'x \vee (a \vee c)x']y \vee [(b \vee c)x \vee (a \vee b \vee c)x']y' = 0.$$

The consistency condition gives the equation in $x$:

$$bc'x \vee (a \vee c)x' = 0,$$

whose consistency condition is:

$$abc' = 0.$$

If this last condition holds, then solution is:

$$(a \vee c) \leq x \leq b' \vee c.$$

By using the equality:

$$(a \vee c)x' = 0,$$

the equation in the unknown $y$ is equivalent with:

$$c'xy \vee [(b \vee c)x \vee bx']y' = 0,$$

or, equivalently:

$$c'xy \vee (b \vee cx)y' = 0.$$

Finally, the solution is:

$$b \vee cx \leq y \leq c \vee x'.$$

**(ii)** Let us solve now the same system:

$$a \leq x,\ b \leq y,\ xy = c,$$

in the Boolean subalgebra generated by the elements $a$, $b$ and $c$.
If the consistency condition is satisfied:

$$abc' = 0,$$

then we have:

$$a \vee c = abc \vee ab'c \vee ab'c' \vee a'bc \vee a'b'c,$$

$$b' \vee c = a \vee c \vee a'b'c'.$$

It results that the intersection of the Boolean subalgebra generated by $a$,$b$, and $c$ with the interval $[a \vee c\,,\ b' \vee c]$ is the set:

$$\{a \vee c\,,\ b' \vee c\}.$$

For $x = a \vee c$, we get $b \vee c \leq y \leq a' \vee c$. With the same method as above, (or by using the symmetry $a \leftrightarrow b$), we obtain for $y$ the solutions:

$$y = b \vee c \text{ and } y = a' \vee c.$$

For $x = b' \vee c$, we get $b \vee c \leq y \leq b \vee c$, hence $y = b \vee c$.
It results that the solutions of the system in the algebra generated by $a$, $b$ and $c$ are:

$$x_1 = a \vee c,\ y_1 = b \vee c,$$
$$x_2 = a \vee c,\ y_2 = a' \vee c,$$
$$x_3 = b' \vee c,\ y_3 = b \vee c.$$

**49. Exercise**
**(i)** Find the solutions of the system:

$$yz = a \vee bc,\ zx = b \vee ca,\ xy = c \vee ab.$$

**(ii)** Find the solutions of the above system in the algebra generated by $a$, $b$ and $c$.

## BIBLIOGRAPHY

1. W.W. Chen *Discrete Mathematics*, Imperial College University of London, 2003.

2. M.B. Finan *Lecture Notes in Discrete Mathematics*, Arkansas Tech University, 2008.

3. John Hopcroft , Jeffrey Ullman *Introduction to Automata Theory, Languages and Computation* , Addison-Wesley, 1979.

4. S. Lang *Undergraduate Algebra*, Springer Verlag, 1987.

5. Lewis Harry R., C.Papadimitriou *Elements of the theory of computation*, Prentice Hall, Englewood Cliffs,N.J. 1981.

6. I. Lovasz, K. Vesztergombi, *Discrete Mathematics*, Lecture Notes, Yale University, Spring 1999.

7. S. Rudeanu *Boolean Functions and Equations* North-Holland, Amsterdam/London 1974; Kogaku Tosho, Tokyo, 1984.

8. Ju.A.Schreider *Equality, Resemblance and Order*,
Mir Publishers, 1974.

9. O.Stanasila *Notiuni si tehnici de matematica discreta*
Editura Stiintifica si Enciclopedica, Bucuresti, 1985.