Logika, skupovi i diskretna matematika

Daniel Kressner[1]

January 9, 2006

[1]`kressner@math.hr` – corrections and suggestions are always welcome.

Contents

Chapter 1

Sets and Functions

1.1 Basic definitions and properties of sets

A *set* is a well-defined collection of objects (called elements of the set) considered as a whole. A set may be described in words, for example:

set
$\hat{=}$ skup

> A is the set of the four largest Croatian towns.
> B is the set of all primes.

prime
$\hat{=}$ prim broj

A set can also be defined by explicitly listing its elements between braces (also called curly brackets), for example:

$$C = \{4, 2, 1, 3\},$$
$$D = \{\text{Zagreb}, \text{Split}, \text{Rijeka}, \text{Osijek}\}.$$

Two different descriptions may define the same set. For example, for the sets defined above, A and D are identical, since they have precisely the same members. The notation $A = D$ is used to express this equality. Set identity does not depend on the order in which the elements are listed. For example, $\{4, 2, 1, 3\} = \{1, 2, 3, 4\}$. Repeated elements are ignored, e.g., $\{4, 2, 1, 3\} = \{1, 4, 2, 1, 3, 3\}$. A third way to define sets is to include all elements from a *universal set* X that satisfy certain properties, $\{x \in X : x \text{ satisfies properties}\}$. For example,

universal set
$\hat{=}$ univerzalni skup

$$E = \{x \in A : x \text{ is a coastal town}\}.$$

Here, $x \in A$ is used to denote that x is an element of A. Otherwise, if x is not an element of A, $x \notin A$. The set which contains no element is called the *empty set* and is denoted by $\emptyset$.

empty set
$\hat{=}$ prazan skup

The five most important sets in mathematics are:

$\mathbb{N} =$ the set of all natural numbers, $\mathbb{N} = \{1, 2, 3, \dots\}$;

natural number
$\hat{=}$ prirodan broj

$\mathbb{Z} =$ the set of all integers, $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$;

integer
$\hat{=}$ cijeli broj

$\mathbb{Q} =$ the set of all rational numbers, $\mathbb{Q} = \{q : q = a/b, a \in \mathbb{Z}, b \in \mathbb{N}\}$;

$\mathbb{R} =$ the set of all real numbers;

$\mathbb{C} =$ the set of all complex numbers, $\mathbb{C} = \{z : z = x + \mathrm{i}y, x \in \mathbb{R}, y \in \mathbb{R}\}$, where i is the *imaginary unit* defined by the relationship $\mathrm{i}^2 = -1$.
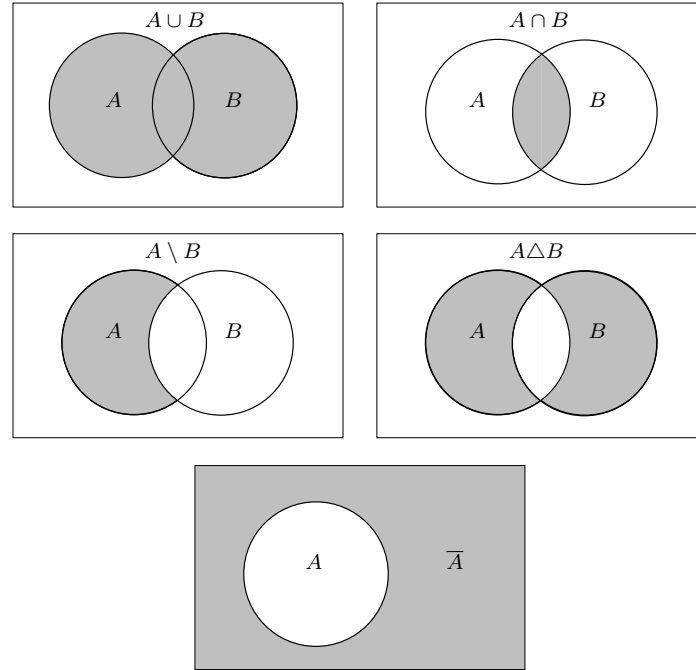
imaginary unit
$\hat{=}$ imaginara jedinica

1

Figure 1.1. *Venn diagrams of set operations.*

If every element of a set B is also an element of A then B is called a *subset* of A. This is denoted by $B \subseteq A$. If additionally $A \neq B$ (i.e., there is at least one element in A which is not an element of B) then B is called a *proper subset* of A and we write $B \subset A$. For the number sets defined above, we have $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{Z}$.

| proper subset |
| $\hat{=}$ pravi podskup |

Given two sets A and B, both subsets of a universal set X, the following operations can be defined:

union: $A \cup B = \{x \in X : \ x \in A \text{ or } x \in B\}$;

| intersection |
| $\hat{=}$ presjek |

intersection: $A \cap B = \{x \in X : \ x \in A \text{ and } x \in B\}$;

| set theoretic difference |
| $\hat{=}$ razlika skupova |

difference: $A \setminus B = \{x \in X : \ x \in A \text{ and } x \notin B\}$;

symmetric difference: $A \triangle B = \{x \in X : \ \text{either } x \in A \text{ or } x \in B\}$;

complement: $\overline{A} = \{x \in X : \ x \notin A\}$.

Venn diagrams as displayed in Figure 1.1 are often used to visualize such operations and to show relationships between sets.

| disjoint |
| $\hat{=}$ disjunktan |

Two sets A and B are called *disjoint* if $A \cap B = \emptyset$. Accordingly, a family $A_1, A_2, \ldots, A_n$ of sets is called disjoint if $A_i \cap A_j = \emptyset$ for all $i \neq j$.

C/C++ excursion 1. *The C++ standard template library provides the container class* `set` *for defining and manipulating sets.*

```
// Initialize A = {2,4,6,8} and B = {3,5,7,8}
int s1[] = {2, 4, 6, 8}; int s2[] = {3, 5, 7, 8};
set A (s1, s1 + sizeof s1 / sizeof *s1);
set B (s2, s2 + sizeof s2 / sizeof *s2);
```

The functions `set_difference`, `set_intersection`, `set_symmetric_difference`, *and* `set_union` *compute* $A \setminus B$, $A \cap B$, $A \triangle B$, *and* $A \cup B$, *respectively.*

```
set<int> c;
insert_iterator<set<int>> ins_c (c, c.begin ());
// Compute union of A and B.
set_union(A.begin(), A.end(), B.begin(), B.end(), ins_c);
```

Theorem 1.1. *Given three sets* $A, B, C \subseteq X$, *the following relations hold:*

1. $A \cup A = A$, $A \cap A = A$;   (idempotency)

2. $(A \cup B) \cup C = A \cup (B \cup C)$, $(A \cap B) \cap C = A \cap (B \cap C)$;   (associativity)

3. $A \cup B = B \cup A$, $A \cap B = B \cap A$;   (commutativity)

4. $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$, $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$; (distributivity)

5. $\overline{A \cup B} = \overline{A} \cap \overline{B}$, $\overline{A \cap B} = \overline{A} \cup \overline{B}$;   (DeMorgan's law)

6. $A \cup \emptyset = A$, $A \cap X = A$;   (identity)

7. $A \cup X = X$, $A \cap \emptyset = \emptyset$;

8. $A \cup \overline{A} = X$, $A \cap \overline{A} = \emptyset$;   (complementarity)

9. $\overline{\overline{A}} = A$.   (involution law)

> idempotency
> ≙ idempotentnost

> involution
> ≙ involucija

Proof. EFY.   □

Theorem 1.1 (in particular 1.–5.) provides examples of an important and powerful property of set algebra, namely, the principal of *duality* for sets, which asserts that for any true statement about sets, the dual statement obtained by interchanging unions and intersections, interchanging X and $\emptyset$ and reversing inclusions is also true.

The associativity property allows to write $A \cup B \cup C$ and $A \cap B \cap C$ without ambiguity. In particular, one can write

> ambiguity
> ≙ nejasnost

$$A_1 \cup A_2 \cup \cdots \cup A_n = \bigcup_{j=1}^{n} A_j, \quad A_1 \cap A_2 \cap \cdots \cap A_n = \bigcap_{j=1}^{n} A_j$$

for some sets $A_1, \ldots, A_n$.

Definition 1.2. *The set of all subsets of a set* X *is called the* power set *and is denoted by* 2^X.

> power set
> ≙ partivni skup

If $X = \{1, 2, 3\}$ then 2^X consists of the elements

$$\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{1, 3\}, \{2, 3\}, \{1, 2, 3\}.$$

The complement of some subset Y of 2^X translates to its subsets, i.e., if $Y \subseteq 2^X$ then $\overline{Y} = \{\overline{A} : A \in X\}$. This is not true for other set operations such as $\cup, \cap$.

Definition 1.3. *Let* $A_1, \ldots, A_n$ *be sets, then the Cartesian product* $\prod_{j=1}^{n} A_j =$

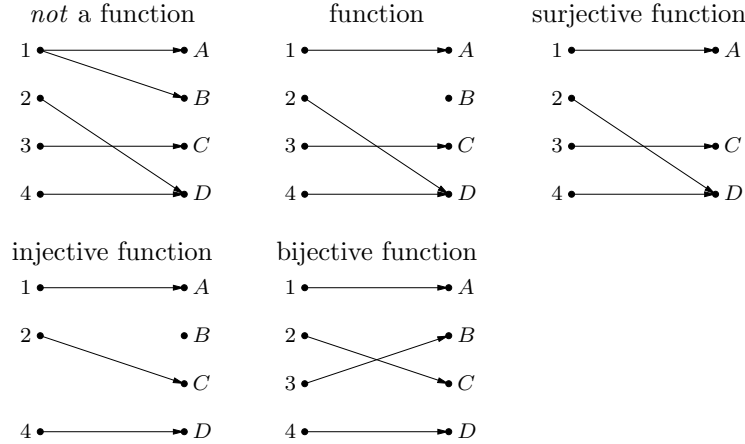> Cartesian product
> ≙ Kartezijev produkt

Figure 1.2. *Several types of functions.*

$A_1 \times A_2 \times \cdots \times A_n$ *is the set which consists of all tuples of the form* $(a_1, \ldots, a_n)$ *with* $a_1 \in A_1, a_2 \in A_2, \ldots, a_n \in A_n$.

tuple
$\hat{=}$ torka

Two-dimensional Cartesian coordinates can be regarded as tuples of $x-$ and $y-$coordinates, i.e., as elements of $\mathbb{R} \times \mathbb{R}$.

1.2 Basic definitions and properties of functions

domain
$\hat{=}$ domena

image
$\hat{=}$ slika

A *function* f assigns to each element of a set A (the *domain*) a unique element of another set B (the *codomain*). This is denoted by $f : A \to B$. Instead of $b = f(a)$ one can also write $f : a \mapsto b$. The *image* of f is the set

$$f(A) = \{b \in B : \text{ there is an } a \in A \text{ such that } b = f(a)\}.$$

The *graph* of f is the subset of $A \times f(A)$ that consists of all tuples of the form $(a, f(a))$ with $a \in A$. Two functions $f_1 : A_1 \to B_1$ and $f_2 : A_2 \to B_2$ are called *identical* (denoted by $f_1 \equiv f_2$) if they have identical domains ($A_1 = A_2$) and codomains ($B_1 = B_2$), and if $f_1(x) = f_2(x)$ for all $x \in A_1$.

C/C++ excursion 2. *In C, a function which returns the maximum of two integers can be defined as follows:*

```
int function1(int number1, int number2)
{
    return max( number1, number2 );
}
```

`function1` *also defines a function in a mathematical sense; it returns the same output value for identical input pairs. The co-domain of this function is given by* $[\texttt{INT\_MIN}, \texttt{INT\_MAX}] \times [\texttt{INT\_MIN}, \texttt{INT\_MAX}]$, *where* `INT_MIN` / `INT_MAX` *is the minimal / maximal value that can be represented by a signed* `int`.

Definition 1.4. *A function* $f : A \to B$ *is called*

surjective *if* $f(A) = B$,

injective *if $a_1 \neq a_2$ implies $f(a_1) \neq f(a_2)$,*

bijective *if f is surjective and injective.*

A bijective function $f : A \to B$ admits the definition of the *inverse function f^{-1} : $B \to A$* as follows:

$$f^{-1} : b \mapsto a \text{ if and only if } f : a \mapsto b.$$

The simplest bijective function is the *identity* function $id : A \to A$, which maps each element to itself, $id : a \mapsto a$.

identity
$\hat{=}$ identiteta

Definition 1.5. *Let $f : A \to B$ and $g : B \to C$ be functions. Then the function $g \circ f$: $A \to C$, defined by $g \circ f : a \to g(f(a))$, is called the* composition *of f and g.*

composition
$\hat{=}$ kompozicija

The inverse of a composition of two bijective functions f and g is the composition of the inverses of these functions in reversed order: $(f \circ g)^{-1} = g^{-1} \circ f^{-1}$.

1.3 Cardinality, equipotent sets

A set A is called *finite* if A contains a finite number of distinct elements. In this case, the elements of A can be numbered, i.e., there is a bijection $f : \{1, \dots, n\} \to A$. The integer n is called the *cardinality* of A, which is denoted by $|A|$. Having defined such a numbering we can write $A = \{a_1, a_2, \dots, a_n\}$ with $a_j = f(j)$ for $j = 1, \dots, n$.

finite set
$\hat{=}$ konačan skup

cardinality
$\hat{=}$ kardinalni broj

Sets that are not finite are called *infinite*. For example, $\mathbb{N}$ and $\mathbb{R}$ are both infinite sets. Intuitively $\mathbb{N}$ and $\mathbb{R}$ seem to have different "levels" of infinity. To catch this difference, the notion of countability is introduced.

infinite set
$\hat{=}$ beskonačan skup

Definition 1.6. *Let A be an infinite set. Then A is called*

countable: *if there is a bijection between $\mathbb{N}$ and A, i.e., we can write $A = \{a_1, a_2, \dots\}$;*

uncountable: *otherwise.*

countable set
$\hat{=}$ prebrojiv skup

From this definition, it is clear that $\mathbb{N}$ is countable. But we will see that also $\mathbb{Z}$ and even $\mathbb{Q}$, the set of rational numbers, are countable sets. In contrast, $\mathbb{R}$ is uncountable.

For finite sets, it is quite simple to define the cardinality for comparing the power of two sets with each other. For infinite sets, particularly for uncountable sets, defining a similarly useful quantity is more difficult.

Definition 1.7. *Two sets A and B are called* equipotent *if there is a bijection $f : A \to B$. In this case, we write $A \sim B$.*

equipotent
$\hat{=}$ ekvipotent

By definition, finite sets are equipotent if and only if they have the same cardinality. Also, a finite set can never be equipotent with an infinite set. The real purpose of equipotency is to compare infinite sets. Equipotency has a number of useful properties, which makes it – as we will later see – an equivalence relation.

equivalence relation
$\hat{=}$ relacija ekvivalencija

Theorem 1.8. *The following holds for any three sets A, B, and C:*

   1. $A \sim A$;   (reflexivity)

2. $A \sim B$ *implies* $B \sim A$;   (symmetry)

3. $A \sim B$ *and* $B \sim C$ *imply* $A \sim C$.   (transitivity)

Proof. All equipotency relations are proven by explicitely constructing a corresponding bijection.

1. The identity $id : A \to A$ is a bijection.

2. Since $A \sim B$, there is a bijection $f : A \to B$. The inverse, $f^{-1} : B \to A$, is also a bijection, which shows $B \sim A$.

3. The assumption implies that there are two bijections $f : A \to B$ and $g : B \to C$. The composition $g \circ f : A \to C$ is another bijection, which shows $A \sim C$.

□

Equipotency also gives rise to a formal extension of the concept of cardinality to infinite sets. Two sets A and B are said to have the same cardinality if they are equipotent, in which case we formally write $|A| = |B|$. If there is an injective function $f : A \to B$ (which is not necessarily surjective), we write $|A| \leq |B|$. If, additionally, A and B are not equipotent then $|A| < |B|$.

Definition 1.9. *Let the set A be countable, then* $|A| = \aleph_0$. *The cardinality of $\mathbb{R}$ is called the* continuum, *denoted by* $|\mathbb{R}| = c$.

Example 1.10.

even number
$\hat{=}$ paran broj

1. *The set $2\mathbb{N}$ of all even natural numbers and the set $\mathbb{N}$ of all natural numbers have the same cardinality, i.e., $|2\mathbb{N}| = |\mathbb{N}| = \aleph_0$. This can be seen by considering the bijection $f : x \mapsto x/2$.*

2. *The set of $\mathbb{Z}$ of all integers and $\mathbb{N}$ are equipotent. A corresponding bijection $f : \mathbb{Z} \to \mathbb{N}$ is given by*

$$f(k) = \begin{cases} 2k & \text{for } k > 0, \\ 2|k| + 1 & \text{for } k \leq 0. \end{cases}$$

3. *The sets $(-1, 1)$ and $\mathbb{R}$ are equipotent, which can be seen by considering the bijection $f : \mathbb{R} \mapsto (-1, 1)$ defined as $f : x \mapsto \tanh x$.*

Theorem 1.11. *Every infinite subset of a countable set is countable.*

Proof. EFY.   □

Surprisingly, also $\mathbb{N} \times \mathbb{N}$ has the same cardinality as $\mathbb{N}$, as shown by the following theorem.

Theorem 1.12. *The Cartesian product of a finite number of countable sets is countable.*

Proof. Let $A_1, A_2, \ldots, A_n$ be countable sets and let $S = A_1 \times A_2 \times \ldots \times A_n$. Since each A_i is countable there exists a bijective function $f_i \colon A_i \to \mathbb{N}$. The function $h \colon S \to \mathbb{N}$ defined by

$$h(a_1, a_2, \ldots a_n) = \prod_{i=1}^{n} p_i^{f_i(a_i)},$$

where p_i is the ith prime is, by the the fundamental theorem of arithmetic, a bijection between S and a subset of $\mathbb{N}$ and therefore S is also countable. $\square$

As $\mathbb{Q}$, the set of all rational numbers, can be identified with an infinite subset of $\mathbb{Z} \times \mathbb{N}$, Theorem 1.12 together with Example 1.10 imply that $\mathbb{Q}$ is countable.

Theorem 1.13 (Cantor's theorem). *The set $\mathbb{R}$ of real numbers is not countable, i.e., $\aleph_0 < c$.*

Proof. Cantor showed that for every given infinite sequence of real numbers $x_1, x_2, x_3, \ldots$ it is possible to construct a real number x that is not on that list. Consequently, it is impossible to enumerate the real numbers; they are uncountable. No generality is lost if we suppose that all the numbers on the list are between 0 and 1. Certainly, if this subset of the real numbers is uncountable, then the full set is uncountable as well.

Let us write our sequence as a table of decimal representations:

decimal representation
$\,\hat=\,$ decimalni prikaz

$$
\begin{array}{llllll}
0. & d_{11} & d_{12} & d_{13} & d_{14} & \ldots \\
0. & d_{21} & d_{22} & d_{23} & d_{24} & \ldots \\
0. & d_{31} & d_{32} & d_{33} & d_{34} & \ldots \\
0. & d_{41} & d_{42} & d_{43} & d_{44} & \ldots \\
 & \vdots & \vdots & \vdots & \vdots & \vdots
\end{array}
$$

where $x_n = 0.\, d_{n1}\, d_{n2}\, d_{n3}\, d_{n4} \ldots$, and the representation avoids an infinite trailing string of the digit 9.

For each $n = 1, 2, \ldots$ we choose a digit c_n that is different from d_{nn} and not equal to 9, and consider the real number x with the decimal representation $0.\, c_1\, c_2\, c_3 \ldots$. By construction, this number x is different from every member of the given sequence. For every n, the number x differs from the number x_n in the nth decimal digit. This concludes the proof. $\square$

Definition 1.14. *A real number x with $x \notin \mathbb{Q}$ is called* irrational. *A real number x which is a root of a polynomial $p(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_n x^n$ with integer coefficients $a_0, \ldots, a_n$ is called* algebraic number. *Real numbers that are not algebraic are called* transcendental.

irrational number
$\,\hat=\,$ iracionalan broj

algebraic number
$\,\hat=\,$ algebarski broj

transcendental number
$\,\hat=\,$ transcendentan broj

Being the root of $x^2 - 2$, $\sqrt{2}$ is an algebraic number. Examples of transcendental numbers include e, π, e^π, $2^{\sqrt{2}}$, $\sin 1$, $\ln 2$.

Theorem 1.15. *The set of algebraic numbers is countable.*

Proof. EFY. $\square$

Chapter 2

Mathematical Logic

Mathematical logic provides a general framework for formalizing and verifying statements. In contrast to natural languages, the language of logic leaves no space for ambiguity or doubts. This is of great value in many branches of science, particularly when the verification of a statement is to difficult to be performed by human mind and must be delegated to a computer. An example we will explore in more detail is the formalization and verification of electric circuits. Be careful not to rely on logic reasoning in everyday life. It will make look you deliberately and annoyingly pedantic. In political discussions it is common to conclude some kind of equality between two statements A and B if both, A and B, imply the same statement C. The corresponding logic statement $[(A \Rightarrow C) \wedge (B \Rightarrow C)] \Rightarrow (A \Leftrightarrow B)$, however, is not always true.

> formalize
> $\hat{=}$ formalizirati

We begin with an example how to formalize statements:

(A) If it does not rain and the temperature is reasonable then I go swimming or drink coffee outside.

Let us decompose this compound statement into four simpler statements:

$$A_1 = \text{It rains.}$$
$$A_2 = \text{The temperature is reasonable.}$$
$$A_3 = \text{I go swimming.}$$
$$A_4 = \text{I drink coffee outside.}$$

By substituting these statements into (A), the logical structure becomes more apparent:

(B) If $((not\ A_1)\ and\ A_2)$ then $(A_3\ or\ A_4)$.

Both statements, (A) and (B), are equivalent. Let $A_1, \ldots, A_4$ denote different statements. For example:

$$A_1 = \text{Somebody is at home.}$$
$$A_2 = \text{The door is left open.}$$
$$A_3 = \text{The dog chases the cat.}$$
$$A_4 = \text{The cat chases the dog.}$$

Then B reads as follows:

> *(C) If nobody is at home and the door is left open then the dog chases the cat or*
> *vice versa.*

Although this statement is quite different from (A), it has the same logical structure as (A). Mathematical logic is not concerned with the content but only with the logical structure of statements. From this point of view, (A) and (C) are the same.

statement
$\hat{=}$ sud

truth value
$\hat{=}$ istinitostna vrijednost

Definition 2.1. *A* statement[1] *is defined as a declarative sentence, or part of a sentence, that is capable of having a truth value, such as being true or false. The* truth value *of a statement A is denoted by* $\tau(A)$, *which can take the values* $\tau(A) = \top$ *(if A is true) or* $\tau(A) = \bot$ *(if A is false).*

To simplify notation, we also write $A \equiv \top$ and $A \equiv \bot$.

2.1   Propositional calculus

propositional calculus
$\hat{=}$ algebra sudova

Propositional calculus, also known as sentential calculus, studies ways of combining or altering statements to form more complicated statements. The basic operators are "not" ($\neg$), "and" ($\wedge$), "or" ($\vee$), implication ($\Rightarrow$), equivalence ($\Leftrightarrow$).

negation
$\hat{=}$ negacija

Definition 2.2. *The* negation $\neg A$ *is true if A is false. We have the following truth table[2]:*

A	$\neg A$
$\bot$	$\top$
$\top$	$\bot$

conjunction
$\hat{=}$ konjunkcija

Definition 2.3. *The* conjunction $A \wedge B$ *of two statements A and B is true if both statements are true.*

A	B	$A \wedge B$
$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\bot$
$\top$	$\bot$	$\bot$
$\top$	$\top$	$\top$

disjunction
$\hat{=}$ disjunkcija

Definition 2.4. *The* disjunction $A \vee B$ *of two statements A and B is true if one of the statements is true.*

A	B	$A \wedge B$
$\bot$	$\bot$	$\bot$
$\bot$	$\top$	$\top$
$\top$	$\bot$	$\top$
$\top$	$\top$	$\top$

---

[1]The term proposition is sometimes used synonymously with statement. However, it is also sometimes used to name something abstract that two different statements with the same meaning are both said to "express". In this usage, the two sentences, "Callisto orbits Jupiter" and "Jupiter is orbited by Callisto" would be considered to express the same proposition. However, the nature or existence of propositions as abstract meanings is still a matter of philosophical controversy; in this chapter "statement" and "proposition" are used interchangeably.

[2]Truth tables are a convenient way to determine the truth value of a formula on a given truth assignment: list all the subformulas of the given formula across the top in order of length and then fill in their truth values on the bottom from left to right.

Definition 2.5. *The* implication $A \Rightarrow B$ *is only false if A is true but B is false.*

<div style="float:right">implication
$\hat{=}$ implikacija</div>

$$
\begin{array}{cc|c}
A & B & A \Rightarrow B \\
\hline
\bot & \bot & \top \\
\bot & \top & \top \\
\top & \bot & \bot \\
\top & \top & \top \\
\end{array}
$$

In mathematics, the following statements have the same meaning, and each is translated as $A \Rightarrow B$:

> If A, then B.
> A only if B.
> A implies Q.
> B if A.
> B provided A.
> B when(ever) A.
> A is a sufficient condition for B.
> B is a necessary condition for A.

Definition 2.6. *The* equivalence $A \Leftrightarrow B$ *is true if A and B have the same truth values.*

<div style="float:right">equivalence
$\hat{=}$ ekvivalencija</div>

$$
\begin{array}{cc|c}
A & B & A \Leftrightarrow B \\
\hline
\bot & \bot & \top \\
\bot & \top & \bot \\
\top & \bot & \bot \\
\top & \top & \top \\
\end{array}
$$

In mathematics, the following statements have the same meaning, and each is translated as $A \Rightarrow B$:

> A if and only if B.
> A is equivalent to B.
> A is a necessary and sufficient condition for B.

Using the defined operators, we can rewrite the sentence from the beginning as

$$A = ((\neg A_1) \wedge A_2) \Rightarrow (A_3 \vee A_4).$$

Three other frequently used binary operators are "xor" ($\veebar$), "nand" ($\uparrow$), "nor" ($\downarrow$).

$$
\begin{array}{cc|ccc}
A & B & A \veebar B & A \uparrow B & A \downarrow B \\
\hline
\bot & \bot & \bot & \top & \top \\
\bot & \top & \top & \top & \bot \\
\top & \bot & \top & \top & \bot \\
\top & \top & \bot & \bot & \bot \\
\end{array}
$$

C/C++ excursion 3. *Boolean variables are declared with the keyword* `bool` *and can take one of two values,* `true` *or* `false`.

```
bool A = true;
```

There are four logic operators, "not" (!), "and" (&&), "or" (||), equivalence (==). Other operators must be expressed by combinations of these operators. The truth of a statement is checked by the `if` *clause.*

```
if ( !( A && B ) || ( C == D ) ) {
    cout << "Statement␣is␣true";
} else {
    cout << "Statement␣is␣false";
}
```

If two formulas P and Q have the same truth variables for all possible truth values of their variables then we call P and Q *logically equivalent* and write $P \equiv Q$. An example for a logical equivalence is

$$(A \Leftrightarrow B) \equiv (A \Rightarrow B) \wedge (B \Rightarrow A).$$

To simplify the appearance of formulas we adopt some conventions for omitting parentheses. For arithmetic operations, multiplication is ranked higher than addition, which means that we can write $a \cdot b + c = (a \cdot b) + c$ (if there are no parentheses multiplication is performed before addition) but $(a + b) \cdot c \neq a + b \cdot c$. The symbols

$$\neg (\text{highest}), \ \vee, \ \wedge, \ \Rightarrow, \ \Leftrightarrow (\text{lowest}),$$

are ranked in the given order with $\vee$ and $\wedge$ having equal rank[3]. The examples below illustrate the convention:

$$
\begin{aligned}
A \vee B \Rightarrow C & \quad \text{is short for} \quad (A \vee B) \Rightarrow C; \\
A \wedge B \Rightarrow C & \quad \text{is short for} \quad (A \wedge B) \Rightarrow C; \\
\neg A \Rightarrow \neg B & \quad \text{is short for} \quad (\neg A) \Rightarrow (\neg B); \\
\neg A \Rightarrow B \Leftrightarrow C & \quad \text{is short for} \quad ((\neg A) \Rightarrow B) \Leftrightarrow C.
\end{aligned}
$$

Any logic operator can be expressed by a suitable combination of other operators, for example

$$A \Leftrightarrow B \equiv (\neg A \vee B) \wedge (\neg B \vee A) \equiv \neg(\neg(\neg A \vee B) \vee \neg(\neg B \vee A)),$$

i.e., the logic operator $\Leftrightarrow$ can be expressed by means of the operators $\neg$ and $\vee$. A set of logic operations that admits the expression of all possible logic formulas is called a set of *generators*. Examples for such sets of generators are $\{\neg, \wedge\}$, $\{\neg, \vee\}$, and $\{\neg, \Rightarrow\}$. A set of generators is called *minimal* if none of the generators in the set can be expressed by combinations of the other generators. All the sets defined above are minimal. Interestingly, there are sets of generators consisting of only one elements: $\{\uparrow\}$, $\{\downarrow\}$.

Proving logical equivalences by truth tables is always possible but can be tedious, especially for long formulas with many variables. The following rules considerably simplify propositional calculus.

Theorem 2.7. *Given three statements A, B, C, the following relations hold:*

1. $A \vee A \equiv A, \ A \wedge A \equiv A;$ $\hfill$ (idempotency)

2. $(A \vee B) \vee C \equiv A \vee (B \vee C), \ (A \wedge B) \wedge C \equiv A \wedge (B \wedge C);$ $\hfill$ (associativity)

---

[3]Some books adopt the less common convention that $\vee$ is ranked higher than $\wedge$.

3. $A \vee B \equiv B \vee A$, $A \wedge B \equiv B \wedge A$;   (commutativity)

4. $A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$, $A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$; (distributivity)

5. $\neg(A \vee B) \equiv \neg A \wedge \neg B$, $\neg(A \wedge B) \equiv \neg A \vee \neg B$;   (DeMorgan's law)

6. $A \vee \bot \equiv A$, $A \wedge \top \equiv A$;   (identity)

7. $A \vee \top \equiv \top$, $A \wedge \bot \equiv \bot$;

8. $A \vee \neg A \equiv \top$, $A \wedge \neg A \equiv \bot$;   (complementarity)

9. $\neg\neg A \equiv A$.   (double negation)

Proof. EFY. □

It is informative to compare this theorem with Theorem 1.1; the striking similarity between sets and statements will be investigated more systematically in Section 2.3. As in set calculus, we have to principal of *duality* for statements, which asserts that for any logical equality about statements, one can obtain the dual equality by interchanging $\vee$ and $\wedge$, interchanging $\top$ and $\bot$ and reversing implications.

The law of associativity (Theorem 2.7.2) admits writing $A \vee B \vee C$ and $A \wedge B \wedge C$ without ambiguity, which makes it possible to define the following two operators:

$$\vee_{k=1}^{n} A_k = A_1 \vee A_2 \vee \cdots \vee A_n, \quad \wedge_{k=1}^{n} A_k = A_1 \wedge A_2 \wedge \cdots \wedge A_n.$$

2.2 Tautologies and deduction rules

Definition 2.8. *A formula P is called* tautology *if it is identical to $\top$, i.e., $P \equiv \top$. In this case, we write* $\models P$. *A formula F satisfying $F \equiv \bot$ is called a* contradiction.

tautology
$\hat{=}$ tautologija

contradiction
$\hat{=}$ kontradikcija

For example, $A \Rightarrow A$ is a tautology while $\neg(A \Rightarrow A)$ is a contradiction. The following lemma contains the most prominent examples of tautologies.

Lemma 2.9. *Let A, B, C be statements. Then we have the following tautologies.*

1. $\models A \vee \neg A$

2. $\models (A \Rightarrow B) \wedge (B \Rightarrow C) \Rightarrow (A \Rightarrow C)$

3. $\models \neg(A \wedge \neg A)$

4. $\models \neg\neg A \Leftrightarrow A$

5. $\models (A \Rightarrow B) \Leftrightarrow (\neg B \Rightarrow \neg A)$

6. $\models A \vee (A \wedge B)$ *and its dual* $\models A \wedge (A \vee B)$

Proof. All rules can be deduced from the rules of propositional calculus in Theorem 2.7. For example, if we let $P = (A \Rightarrow B) \wedge (B \Rightarrow C) \Rightarrow (A \Rightarrow C)$ then

$$
\begin{array}{rll}
P & \equiv & (\neg A \vee B) \wedge (\neg B \vee C) \Rightarrow (\neg A \vee C) \qquad\qquad (P \Rightarrow Q \equiv \neg P \vee Q) \\
& \equiv & \neg[(\neg A \vee B) \wedge (\neg B \vee C)] \vee (\neg A \vee C)] \qquad (P \Rightarrow Q \equiv \neg P \vee Q) \\
& \equiv & [\neg(\neg A \vee B) \vee \neg(\neg B \vee C)] \vee (\neg A \vee C)] \qquad\qquad (\text{Theorem 2.7.5}) \\
& \equiv & (A \wedge \neg B) \vee (B \wedge \neg C) \vee (\neg A \vee C) \qquad (\text{Theorem 2.7.2, 5 and 9}) \\
& \equiv & [\neg A \vee (A \wedge \neg B)] \vee [C \vee (B \wedge \neg C)] \qquad\qquad (\text{Theorem 2.7.2}) \\
& \equiv & [\top \wedge (\neg A \vee \neg B)] \vee [(C \vee B) \wedge \top] \qquad (\text{Theorem 2.7.4 and 8}) \\
& \equiv & \top \wedge [(\neg A \vee \neg B) \vee (C \vee B)] \qquad\qquad (\text{Theorem 2.7.4}) \\
& \equiv & \top \wedge \top \equiv \top. \qquad\qquad (\text{Theorem 2.7.2, 6, 7 and 8})
\end{array}
$$

$\square$

The rules of propositional calculus in Theorem 2.7 are powerful enough that there is normally no need to prove logical equivalences by truth tables. This observation motivates the following definition.

| algebra of propositions |
| $\hat{=}$ algebra sudova |

Definition 2.10. *The* algebra of propositions *is a set of statements S, together with two binary operations $\vee$, $\wedge$ and one unary operation $\neg$ such that the rules of Theorem 2.7 are satisfied.*

This definition conceptualizes propositional calculus but it also allows to cover other frameworks such as binary calculus, see Section 2.4.

In the following, we will introduce and formalize proof techniques (also called *deduction rules*). The probably most important technique is conclusion.

| logical conclusion |
| $\hat{=}$ logički posljedak |

Definition 2.11. *We call a statement A a* (logical) conclusion *from some statements $P_1, \ldots, P_n$ if the truth of $P_1, \ldots, P_n$ implies that A is true, and write*

$$\boxed{P_1, \ldots, P_n \models A}$$

| premise |
| $\hat{=}$ pretpostavka |

The statements $P_1, \ldots, P_n$ are called premises *and A is called* consequence *or simply* conclusion.

Theorem 2.12. *If $P_1, \ldots, P_n \models A$ then $\models \wedge_{k=1}^{n} P_k \Rightarrow A$, and vice versa.*

Proof. Let $P_1, \ldots, P_n \models A$. Assume that $\wedge_{k=1}^{n} P_k \Rightarrow A$ is *not* a tautology. Then there are truth values for $P_1, \ldots, P_n$ such that $\wedge_{k=1}^{n} P_k \equiv \top$ but $A \equiv \bot$. However, this is not possible as this implies $P_1 \equiv \cdots \equiv P_n \equiv \top$, which, together with $A \equiv \bot$, contradicts the definition of $P_1, \ldots, P_n \models A$

To show that the converse holds, let $\wedge_{k=1}^{n} P_k \Rightarrow A$ be a tautology. Then $P_1 \equiv \cdots \equiv P_n \equiv \top$ implies $A \equiv \top$. This proves $P_1, \ldots, P_n \models A$. $\square$

The first part of this proof illustrates another technique: proof by contradiction, to which we will come back below. With the help of Theorem 2.12, we can rewrite Lemma 2.9.2 as

$$A \Rightarrow B, B \Rightarrow C \models A \Rightarrow C.$$

| syllogism |
| $\hat{=}$ silogizm |

An alternative, more schematic way of writing this so called *syllogism* is as follows.

$$A \Rightarrow B$$
$$B \Rightarrow C$$
$$\overline{A \Rightarrow C}$$

Let us consider a stupid example:

Premise:	If the weather is nice, I go swimming.
Premise:	If I go swimming, I get wet.
Conclusion:	If the weather is nice, I get wet.

(Remember not to use mathematical logic in everyday life. EFY: What is the reason for this apparent contradiction?)

Lemma 2.13. *For two statements A and B we have*

$$A, A \Rightarrow B \models B,$$

also called modus ponens *(MP)*.

Proof. EFY. □

Let us modify the stupid example accordingly:

Premise:	If the weather is nice, I go swimming.
Premise:	The weather is nice.
Conclusion:	I go swimming.

Two other deduction rules are *modus tollens*

$$A \Rightarrow B, \neg B \models \neg A$$

and the *law of excluded middle* (tertium non datur in Latin)

law of excluded middle
$\hat{=}$ pravilo isključenja trećeg

$$A \vee B, \neg B \models A.$$

These correspond to

Premise:	If the weather is nice, I go swimming.
Premise:	I don't go swimming.
Conclusion:	The weather is not nice.

and

Premise:	It rains or I go swimming.
Premise:	I don't go swimming.
Conclusion:	It rains.

A very common proof technique is *proof by contradiction* (also called *indirect proof*). If negating the statement A leads to a contradiction then A must be true:

$$\neg A \Rightarrow \bot \models A.$$

The following lemma illustrates this technique.

Lemma 2.14. *Let n be an integer. If n^2 is even then also n is even.*

Proof. The assumption is that n^2 is even. Let A be the statement: n is even. If we show that negating A leads to a false statement then, by contradiction, A must be true.

The statement $\neg A$ means n is odd, i.e., there is an integer k such that $n = 2k + 1$. Squaring this equation leads to $n^2 = 2(2k^2 + 2k) + 1$, i.e., n^2 is odd. This contradicts the assumption and therefore n is even. $\square$

2.3   Relationship between statements and sets

Each row of the following table lists two notions for sets and statements which are closely related to each other.

disjunction	$A \vee B$	$A \cup B$ union
conjunction	$A \wedge B$	$A \cap B$ intersection
negation	$\neg A$	$\overline{A}$ complement
exclusive disjunction	$A \veebar B$	$A \triangle B$ symmetric difference
true	$\top$	X universal set
false	$\bot$	$\emptyset$ empty set

The relationships become apparent when reconsidering the definitions for the set operations:

$$A \cap B = \{x \in X : \ x \in A \wedge x \in B\},$$
$$A \cup B = \{x \in X : \ x \in A \vee x \in B\},$$
$$\overline{A} = \{x \in X : \ \neg(x \in A)\},$$
$$A \triangle B = \{x \in X : \ x \in A \veebar x \in B\},$$
$$X = \{x \in X : \ \top\},$$
$$\emptyset = \{x \in X : \ \bot\}.$$

If two statements A_P and A_Q satisfy $A_P \equiv A_Q$, then the two sets $P = \{x \in X : \ \text{statement } A_P \text{ is true}\}$ and $Q = \{x \in X : \ \text{statement } A_Q \text{ is true}\}$ are equal. In particular, the statements in Theorem 1.1 follow directly from the corresponding statements in Theorem 2.7.

2.4   Boolean algebras

The similarities between Theorem 1.1 and Theorem 2.7 motivate the following abstract definition of an algebraic structure.

Boolean algebra
$\hat{=}$ Booleova algebra

Definition 2.15. *A* Boolean algebra *is a set B, supplied with two binary operations $+$ and $\cdot$, an unary operation $\overline{}$ and two elements 0 and 1, such that, for all elements a, b and c of B, the following axioms hold:*

1. $a + a = a$, $a \cdot a = a$;   (idempotency)

2. $(a + b) + c = a + (b + c)$, $(a \cdot b) \cdot c = a \cdot (b \cdot c)$;   (associativity)

3. $a + b = b + a$, $a \cdot b = b \cdot a$;   (commutativity)

4. $a \cdot (b + c) = (a \cdot b) + (a \cdot c)$, $a + (b \cdot c) = (a + b) \cdot (a + c)$;   (distributivity)

5. $\overline{a + b} = \bar{a} \cdot \bar{b}$, $\overline{a \cdot b} = \bar{a} + \bar{b}$;   (DeMorgan's law)

6. $a + 0 = a$, $a \cdot 1 = a$;   (identity)

7. $a + 1 = 1$, $a \cdot 0 = 0$;

8. $a + \bar{a} = 1$, $a \cdot \bar{a} = 0$;   (complementarity)

9. $\bar{\bar{a}} = a$.   (involution law)

Including the two Boolean algebras we already know, the following list contains the most popular examples for Boolean algebras.

1. The simplest Boolean algebra, the so called *two-element Boolean algebra* has only two elements, 0 and 1, and is defined by the rules:

$$
\begin{array}{c|cc}
+ & 0 & 1 \\
\hline
0 & 0 & 1 \\
1 & 1 & 1
\end{array}, \qquad
\begin{array}{c|cc}
\cdot & 0 & 1 \\
\hline
0 & 0 & 0 \\
1 & 0 & 1
\end{array}
$$

   We will see that this Boolean algebra is isomorph to the algebra of propositions, interpreting 0 as false, 1 as true, $+$ as $\vee$, $\cdot$ as $\wedge$, and $^-$ as $\neg$.

2. By Theorem 1.1, the power set of any given set A forms a Boolean algebra with the two operations $+ := \cup$ and $\cdot = \cap$. The smallest element 0 is the empty set and the largest element 1 is the set S itself.

3. Other examples of Boolean algebras arise, e.g., in quantum logic.

Lemma 2.16. *The elements* 0 *and* 1 *of a Boolean algebra* B *are uniquely defined. Moreover, the operations* $+$ *and* $\cdot$ *satisfy the absorption laws:*

absorption laws
$\hat{=}$ pravila apsorpcije

$$a + a \cdot b = a, \quad a \cdot (a + b) = a.$$

Proof. Let 0_1 and 0_2 be two zeros in B. From Definition 2.15.6, we have $0_1 + 0_2 = 0_1$ and $0_2 + 0_1 = 0_2$. By Definition 2.15.3, this implies $0_1 = 0_2$. The proof of the uniqueness of 1 is left as exercise.

The first absorption laws follows by writing $a + a \cdot b = a \cdot 1 + a \cdot b = a \cdot (1 + b) = a \cdot 1 = a$. Similarly, $a \cdot (a + b) = (a + 0) \cdot (a + b) = a + (0 \cdot b) = a + 0 = a$.   □

The two-element Boolean algebra, the algebra of subsets of $\{x\}$ as well as the algebra of statements are all Boolean algebras with two elements, which can be identified with each other by means of an isomorphism.

Definition 2.17. *Let* B_1 *and* B_2 *be two Boolean algebras. Then the function* $f : B_1 \to B_2$ *is called an* isomorphism *between* B_1 *and* B_2 *if* f *is bijective and*

$$f(a \cdot b) = f(a) \cdot f(b), \quad f(\bar{a}) = \overline{f(a)} \tag{2.1}$$

holds for all $a, b \in B_1$. *If there exists such a function* f *then* B_1 *and* B_2 *are called* isomorphic.

Using DeMorgan's law, the equalities (2.1) also imply

$$f(a + b) = f\left(\overline{\overline{a} \cdot \overline{b}}\right) = \overline{f(\overline{a} \cdot \overline{b})} = \overline{f(\overline{a}) \cdot f(\overline{b})} = \overline{\overline{f(a)} \cdot \overline{f(b)}} = f(a) + f(b).$$

As an example for such an isomorphism, let us consider the two-element Boolean algebra $B_1 = \{0, 1\}$, the algebra of propositions $B_2 = \{\bot, \top\}$, and the function $f : B_1 \to B_2$ defined as $f(0) = \bot$ and $f(1) = \top$. Trivially, f is a bijection and

$$f(a \cdot b) = f(a) \wedge f(b), \quad , f(\overline{a}) = \neg f(a),$$

which shows that f is an isomorphism between B_1 and B_2.

<div style="float:left; border:1px solid;">subalgebra
$\hat{=}$ podalgebra</div>

Definition 2.18. *A subalgebra of a Boolean algebra B is a Boolean algebra that consists of a subset of B supplied with the same operations.*

If f is an isomorphism and B_1 is a subalgebra then the image of f is again a subalgebra.

2.5   Boolean functions

In the following, B denotes the two-element Boolean algebra $B = \{0, 1\}$.

Definition 2.19. *A Boolean function is a function $F : B \times B \times \cdots \times B \to B$.*

Any Boolean function can be interpreted as an n-ary logical operation. For example, the operations $+$ and $\cdot$ in the Boolean algebra are Boolean functions with $n = 2$. Another interpretation is to consider a Boolean function as an input-output-system, having n inputs and 1 outputs.

Lemma 2.20. *There are 2^{2^n} different Boolean functions $F : B^n \to B$.*

Proof. The set B^n contains 2^n different members (as there are 2^n binary numbers of length n). Each member can be assigned a 0 or a 1, which gives 2^{2^n} different possibilities taking all members together.   □

Here is a table of all Boolean functions for $n = 2$:

x_1	x_2	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1
0	1	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

This table illustrates that every Boolean function can be encoded by a single binary number of length 2^n. Furthermore, by the isomorphism between the two-element algebra and the algebra of propositions, we can identify $F_2, F_7, F_8, F_{10}, F_{14}$ with the logical operations $\wedge, \underline{\vee}, \vee, \Leftrightarrow, \Rightarrow$, respectively.

Lemma 2.21. *The set of all Boolean functions $f : B^n \to B$ (n is fixed) supplied*

with the operations

$$(F + G)(x_1, \ldots, x_n) = F(x_1, \ldots, x_n) + G(x_1, \ldots, x_n),$$
$$(F \cdot G)(x_1, \ldots, x_n) = F(x_1, \ldots, x_n) \cdot G(x_1, \ldots, x_n),$$
$$\overline{F}(x_1, \ldots, x_n) = \overline{F(x_1, \ldots, x_n)},$$

is a Boolean algebra.

2.6   Normal forms

To check whether a Boolean function $F(x_1, x_2, \ldots, x_n)$ satisfies $F \equiv 1$ (which means that the isomorphic logical statement is a tautology) requires to test all 2^n assignments of the variables $x_1, \ldots, x_n$. This task can be simplified if F is a product of Boolean functions

$$F(x_1, \ldots, x_n) = F_1(x_1, \ldots, x_n) \cdot F_2(x_1, \ldots, x_n) \cdots F_k(x_1, x_2, \ldots, x_n).$$

Then $F \equiv 1$ if and only if each of the k factors satisfies $F_j \equiv 1$. Moreover, if the factors take the form

$$x_{i_1} + x_{i_2} \cdots + x_{i_{n_j}} + \overline{x_{l_1}} + \overline{x_{l_2}} + \cdots + \overline{x_{l_{m_j}}}, \tag{2.2}$$

then $F_j \equiv 1$ if and only if $i_p = l_p$ for some p, i.e., if a variable appears twice in F_j, in its original form and in the negated form. This is much simpler than checking the value of F for all possible assignments of the variables. In the following, we will show that any function F can be rewritten in the form described above.

The following definition provides an important intermediate form.

Definition 2.22. *An expression for a Boolean function F is said to be in* negation normal form *(NNF) if the unary operation $^{-}$ is only taken w.r.t. variables.*

For example, the expression

$$F_1(x_1, x_2, x_3, x_4) = \overline{x_1} + \overline{x_2} \cdot \overline{x_3} + x_4$$

is in negation normal form while

$$F_2(x_1, x_2, x_3, x_4) = \overline{x_1 \cdot x_2 + x_3} + x_4 \tag{2.3}$$

is not.

Any expression can be transformed into negation normal form by repeated application of DeMorgan's law, see Definition 2.15.5. For example, after one application of DeMorgan's law we obtain from (2.3) that

$$F_2(x_1, x_2, x_3) = \overline{x_1 \cdot x_2} \cdot \overline{x_3} + x_4,$$

and after another application,

$$F_2(x_1, x_2, x_3) = \overline{x_1} + \overline{x_2} \cdot \overline{x_3} + x_4,$$

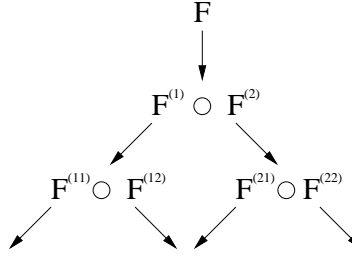which shows that F_1 and F_2 are actually equal.

Definition 2.23. *An expression for a Boolean function F is said to be in* conjunctive normal form *(CNF) if*

$$F(x_1, \ldots, x_n) = F_1(x_1, \ldots, x_n) \cdot F_2(x_1, \ldots, x_n) \cdots F_k(x_1, x_2, \ldots, x_n),$$

where each factor F_j is an elementary disjunction *of the form*

$$x_{i_1} + x_{i_2} + \cdots + x_{i_{n_j}} + \overline{x_{l_1}} + \overline{x_{l_2}} + \cdots + \overline{x_{l_{m_j}}}.$$

Any Boolean function F can be expressed in CNF. To obtain this form, we first assume (without loss of generality) that the expression for F is already in NNF. The CNF is obtained in a recursive fashion. Since F is in NNF, it either is already an elementary disjunction or its expression can be decomposed into $F = F^{(1)} \circ F^{(2)}$, where where $\circ \in \{\cdot, +\}$. If not consisting of elementary disjunctions, $F^{(1)}$ and $F^{(2)}$ are further decomposed as $F^{(1)} = F^{(11)} \circ F^{(12)}$ and $F^{(2)} = F^{(21)} \circ F^{(22)}$. This process is continued until all F is completely decomposed into. Finally, one obtains a decomposition tree of the following form with all leafs being (negated) variables[4]:

$$
\begin{array}{c}
\mathbf{F} \\
\downarrow \\
\mathbf{F}^{(1)} \circ \mathbf{F}^{(2)} \\
\swarrow \qquad \searrow \\
\mathbf{F}^{(11)} \circ \mathbf{F}^{(12)} \qquad \mathbf{F}^{(21)} \circ \mathbf{F}^{(22)} \\
\swarrow \quad \searrow \qquad \swarrow \quad \searrow
\end{array}
$$

Now, we work from bottom to top as follows. Assuming that $F^{(j1)}$ and $F^{(j2)}$ are in CNF, we already have a CNF for $F^{(j)} = F^{(j1)} \circ F^{(j1)}$ if $\circ = \cdot$. If $\circ = +$, we obtain a CNF from the law of distributivity (see Theorem 2.15.4):

$$
\begin{aligned}
F^{(j)} &= F^{(j1)} + F^{(j2)} \\
&= \prod_{k=1}^{n_1} V^{(k)} + \prod_{l=1}^{n_2} W^{(l)} \\
&= \prod_{k=1}^{n_1} \prod_{l=1}^{n_2} \left(V^{(k)} + W^{(l)} \right),
\end{aligned}
$$

where $V^{(k)}$ and $W^{(l)}$ denote elementary disjunctions. Let us illustrate this technique by an example. If $F = (p \cdot \overline{q}) + (q + \overline{p})$ then $F = F^{(1)} + F^{(2)}$ with $F^{(1)} = p \cdot \overline{q}$ and $F^{(2)} = q + \overline{p}$. Similarly, $F^{(1)} = F^{(11)} \cdot F^{(12)}$ with $F^{(11)} = p$ and $F^{(12)} = \overline{q}$. Now, F is completely decomposed into the elementary disjunctions $F^{(11)}$, $F^{(12)}$, and $F^{(2)}$. Also, there is nothing to do for $F^{(1)}$, as $F^{(1)} = F^{(11)} \cdot F^{(12)}$ is already in CNF. To obtain a CNF for F, we have to apply the distributivity law:

$$F = (p \cdot \overline{q}) + (q + \overline{p}) = (p + q + \overline{p}) \cdot (\overline{q} + q + \overline{p}).$$

From the discussion in the introduction, we can now easily see that F is a tautology.

There is a dual form to the CNF, which is obtained by interchanging the roles of $+$ and $\cdot$ in the algorithm described above.

---

[4]It suffices to decompose the expression until all leafs are in CNF

Definition 2.24. *An expression for a Boolean function F is said to be in* disjunctive normal form *(DNF) if*

$$F(x_1, \ldots, x_n) = F_1(x_1, \ldots, x_n) + F_2(x_1, \ldots, x_n) + F_k(x_1, x_2, \ldots, x_n),$$

where each summand F_j is an elementary conjunction *of the form*

$$x_{i_1} \cdot x_{i_2} \cdots \cdots \cdot x_{i_{n_j}} \cdot \overline{x_{l_1}} \cdot \overline{x_{l_2}} \cdot \cdots \cdot \overline{x_{l_{m_j}}}.$$

disjunctive
$\hat{=}$ disjuktivna

2.7 Logic circuits

A *logic circuit* is an electric circuit whose output depends upon the input in a way that can be expressed as a Boolean function $F : B^n \to B$ in symbolic logic. Logic circuits that perform particular functions are called *logic gates.* Basic logic circuits include the AND[5] gate, the OR gate, and the NOT gate, which perform the operations $\cdot$, $+$, and $^{-}$. Nowadays, circuits are often combined from integrated logic circuits that perform more complex Boolean functions.

logic circuit
$\hat{=}$ logički sklop

name	symbol	Boolean	statement
AND gate		$x \cdot y$	$X \wedge Y$
OR gate		$x + y$	$X \vee Y$
inverter		$\overline{x}$	$\neg X$
NAND gate		$\overline{x \cdot y}$	$X \uparrow Y$
NOR gate		$\overline{x + y}$	$X \downarrow Y$

These symbols can be extended in a direct manner to accept three or more inputs and to denote that some of the inputs are negated. For example, the expressions $x + y + z$ and $\overline{x} + \overline{y} + \overline{z}$ are represented by:

To give an example, the logic circuit corresponding to $((y + z) \cdot \overline{x}) + \overline{x + y}$ is
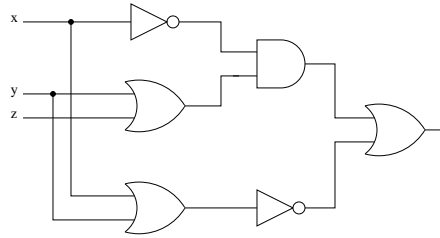
2.8 Predicate calculus

First-order *predicate calculus* or first-order logic (FOL) permits the formulation of

predicate calculus
$\hat{=}$ predikatni račun

---

[5]Nikola Tesla received the first patents for AND logic gates in July 1900.

quantified statements such as "there exists an x such that..." ($\exists x$) or "for any x, it is the case that..." ($\forall x$), where x is a member of the domain under consideration. First-order logic is mathematical logic that is distinguished from higher-order logic in that it does not allow quantification over properties; i.e. it cannot express statements such as "for every property P, it is the case that..." ($\forall P$) or "there exists a property P such that..." ($\exists P$). Nevertheless, first-order logic is strong enough to formalize all of set theory and thereby virtually all of mathematics.

Definition 2.25.   A predicate *is a function $P(\cdot)$ which maps every element of a given set D (the domain of P) to a statement $P(x)$.*

two-adic
$\hat{=}$ dvo-mjesni

A *two-adic predicate* has a domain of the form $D = D_1 \times D_2$ and we can write $P(x_1, x_2)$ in place of $P(x)$ with $x = (x_1, x_2) \in D$. Similarly, the domain of an *n-adic predicate* takes the form $D = D_1 \times \cdots \times D_n$ and we write $P(x_1, \ldots, x_n)$ with the variables $x_1 \in D_1, \ldots, x_n \in D_n$. Apart from the already introduced logic operators $\neg$, $\wedge$, and $\vee$, predicate calculus admits the use of two other operators.

universal quantifier
$\hat{=}$ univerzalni kvantifikator

The phrase "for every x" is called the *universal quantifier* and we shall write it as ==$\forall x$==. For example if $P(x)$ is the statement "x is greedy" on the domain of human beings then $\forall x\, P(x)$ means "All human beings are greedy." Note that although $\forall x$ means for *every* x, the restriction to the given domain (of human beings) is understood.

existential quantifier
$\hat{=}$ egzistencijalni kvant.

The phrase "there exists an x such that" is called the *existential quantifier* and we shall write it as ==$\exists x$==. For example, with $P(x)$ as above, $\exists x\, P(x)$ means "There is a greedy human being." Again the restriction to some given domain is understood.

Two or more quantifiers may be used in tandem. In the following examples we write a statement in symbolic form and give possible translations.

$$\forall x \forall y (x + y = y + x)$$
$=$ For every x, for every y, $x + y = y + x$.
$=$ Addition of numbers is commutative.

$$\forall x \forall y \exists z (x + y = z)$$
$=$ For every x, for every y, there is a z such that $x + y = z$.
$=$ The sum of any two numbers is a number.

In the statement

$$\sum_{k=1}^{5} k^2 = 55 \tag{2.4}$$

k is a dummy variable. When (2.4) is written out as

$$1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 55$$

bound variable
$\hat{=}$ vezana varijabla

k does not appear at all. In (2.4), k can be changed to some other variable without altering the meaning. In logic, such a dummy variable is called a *bound variable*. In the formula

$$\exists x (x < y) \tag{2.5}$$

x is a bound variable while y is not a bound variable. When (2.5) is written as

There is a number less than y

x disappears, but y is still present. If we think of y as being some fixed number, (2.5) says something about y. A variable that is not bound is *free*. y is a free variable in (2.5).

A variable may be bound in different ways, but in the following we shall consider only variables bound by quantifiers. An *occurrence* of a variable v in a formula Q is bound if and only if it is the explicit occurrence in a quantifier $\forall v$ or $\exists x$, or if it is in the scope of a quantifier $\forall v$ or $\exists v$. The same variable may be both free and bound in the same formula. In the formula

$$(\exists x(x < 7)) \wedge (x + z = 8)$$

the first two occurrences of x are bound, but the third occurrence of x is free because it is not in the scope of the quantifier $\exists x$. As far as meaning is concerned, the free and bound occurrences of the same variable in a formula have nothing to do with each other. Such situations are confusing and should be avoided.

A formula with no free variables can be identified with a statement. A formula with n free variables can be identified with an n-adic predicate P. If v is a free variable in an n-adic predicate P then $\forall v \, P$ is an $(n-1)$-adic predicate.

Definition 2.25 admits the assignment of truth values for quantifiers. If $P(x)$ is a predicate then $\forall x \, P(x)$ is true if and only if $P(x) \equiv \top$ for every x. $\exists x \, P(x)$ is true if there is an x such that $P(x) \equiv \top$.

Lemma 2.26 (DeMorgan's law for quantifiers). *For any predicate $P(x)$ the following equivalences hold:*

$$\neg \forall x \, P(x) \equiv \exists x \, \neg P(x),$$
$$\neg \exists x \, P(x) \equiv \forall x \, \neg P(x).$$

Proof. We only prove the first equivalence, leaving the second one as an EFY. Let D be the domain of P. We must show that $\neg \forall x \, P(x)$ and $\exists x \, \neg P(x)$ have the same truth values. Suppose that $\exists x \, \neg P(x)$ is true. Then $\neg P(x_0)$ is true for some x_0. Consequently, $P(x_0)$ is false and thus $\forall x \, P(x)$ is false, which in turn implies that $\neg \forall x \, P(x)$ is true. Now suppose that $\exists x \, \neg P(x)$ is false. Then $\neg P(x)$ must be false for all x, which implies that $P(x)$ is true for all x. Hence, $\neg \forall x \, P(x)$ is false. $\quad \square$

A tandem of universal quantifiers commutes. So does a tandem of existential quantifiers. However, a mixed tandem of existential and universal quantifiers does in general *not* commute. In summary, we have the following result.

Theorem 2.27. *Let $R(x, y)$ be a predicate. Then the following holds:*

1. $\forall x \forall y \, R(x, y) \equiv \forall y \forall x \, R(x, y)$,

2. $\exists x \exists y \, R(x, y) \equiv \exists y \exists x \, R(x, y)$,

3. $\exists x \forall y \, R(x, y) \models \forall y \exists x \, R(x, y)$,

4. $\forall x \exists y \, R(x, y) \models \exists y \exists x \, R(x, y)$.

Unlike propositional calculus, first-order logic is undecidable. There is provably no decision procedure for determining for an arbitrary formula F, whether or not F is valid (an example for such a problem is the so called *Halting problem*).

Chapter 3

Integers

In this chapter, we will give a brief introduction to the world of integers. The study of integers is at the heart of the mathematical field *number theory*. This field is concerned with wider classes of problems that have arisen naturally from the study of integers. Many famous mathematicians have worked in number theory; to quote Gauss:

> *Mathematics is the queen of the sciences and number theory is the queen of mathematics.*

The typical public picture of a mathematician is that of a number theorist. In fact, many movies featuring mathematicians are about number theorists, see for example **Π** (1998), *Cube* (1997), *Good Will Hunting* (1997), *The Mirror has Two Faces* (1996).

3.1   Divisors, remainders and modular arithmetic

We say $m \mid n$ (read: m divides n, or m is a divisor of n) for any integers m and n iff there exists an integer k such that $n = km$. Thus, divisors can be negative as well as positive. 1 and -1 are divisors of every integer, every integer is a divisor of itself, and every integer is a divisor of 0, while 0 is a divisor only of 0. There are some rules which allow to recognize small divisors of a number from the number's decimal representation; these will be treated in the exercises.

A divisor of n that is not 1, -1, n or $-n$ is known as a non-trivial divisor; numbers with non-trivial divisors are known as *composite numbers*, while prime numbers have no non-trivial divisors.

Definition 3.1. *A prime number is a positive integer $p > 1$ that has no positive integer divisors other than 1 and p itself.*

We call an expression $a = bd$ (a, b, d integers) a factorization of a. A problem on cryptography we will encounter later in this chapter involves factoring large numbers. In particular, consider the following question: What is the fastest general procedure to find the smallest non-trivial positive factor of an integer $n > 0$? This question may seem simple but if n is very large the answer is far from obvious. For example, consider the following 309-digit number

1350664108659952233496032162788059699388814756056670275244851438515226510604859533833940287150571909441798207282164471551373680419703964191743046496589274256239341020864383202110372958725762358509643110564073501508187510676594629205563685529475213500852879416377328533906109750544334999811150056977236890927563

This number is called RSA-1024, since it has 1024 binary digits. The sum of its digits is 1369. If you can factor this number then RSA Security Inc will pay you one hundred thousand dollars! See `http://www.rsasecurity.com/rsalabs/node.asp?id=2093` for more details.

However, if n is relatively small then there are some easy strategies to follow to factor it.

Lemma 3.2 (Basic factoring strategy). *If a positive integer n has a factorization $n = ab$ (a, b integers) then either $a \leq \sqrt{n}$ or $b \leq \sqrt{n}$.*

Proof. Suppose that the statement of this lemma is false, so $a > \sqrt{n}$ and $b > \sqrt{n}$. Then $n = ab > \sqrt{n}\sqrt{n} = n$. This is a contradiction.   □

The result above implies that, if n is composite then the smallest non-trivial factor d of n must satisfy $1 < d \leq \sqrt{n}$, if n has no factor d where $1 < d < \sqrt{n}$ then n must be a prime.

Example 3.3. Let $n = 1233$. By the above fact, to find a positive factor of 1233, we need only check if the integers $1, 2, 3, ..., 35$ divide n since $\sqrt{1233} = 35.11....$ Moreover, any such divisor must be odd is 1233 is odd. Our first try, $1233/3$ turns out to be an integer and we get $1233 = 3 \cdot 411$.

What about the factors of 411? By the above fact, to find a positive factor of 411, we need only check if the integers $1, 2, 3, ..., 20$ divide 441 since $\sqrt{411} = 20.273....$ Again, it must be odd and our first try, $411/3$ turns out to be an integer and we get $411 = 3 \cdot 137$.

To factor 137 we check if $1, 2, 3, ..., 11$ divide 137 since $\sqrt{137} < 12$. Trying all the odd numbers $3, 5, 7, 9, 11$, we see that none of them are divisors of 137. The complete factorization is $1233 = 3 \cdot 3 \cdot 137$.

The next, fundamental result leads to the notion of modular arithmetic and Euclid's algorithm.

Theorem 3.4 (Division algorithm). *Let a and $b > 0$ be integers. There are integers q (the* quotient*) and r (the* remainder*) such that*

$$a = bq + r, \qquad 0 \leq r < b.$$

Furthermore, q and r are unique.

Proof. We begin the proof of the division algorithm by showing that q and r are unique. Suppose $a = bq + r = bq_1 + r_1$, where $0 \leq r < b$, $0 \leq r_1 < b$. Then $0 = b(q - q_1) + (r - r_1)$. Since b divides the left side and the first term of the right side of this equation, b must divide $r - r_1$. But $-b < r - r_1 < b$, so $r - r_1 = 0$. Therefore r is unique. Since $bq + r = bq_1 + r_1$, this in turn implies that q is also unique.

For the proof of the existence of q and r in the division algorithm, consider the set

$$S = a + b\mathbb{Z} = \{a + nb \mid n \in \mathbb{Z}\} = \{..., a - 2b, a - b, a, a + b, a + 2b, ...\}$$

This contains at least one non-negative element, for example, $a + |a|b \geq a + |a| \geq 0$. By the well-ordering principle, S contains a least non-negative element, say r. By definition of S, there is an integer q such that $r = a - qb$. It remains to show $r < b$. Suppose not (to get a contradiction), i.e., suppose $r \geq b$. Then $0 \leq r - b = a + (-q - 1)b \in S$, and $r - b < r$, so r was not the smallest non-negative element of S. This contradiction shows that the hypothesis $r \geq b$ is false. Therefore $r < b$ and the proof is complete. $\square$

About 200 years ago in Germany, at the age of 23 C. F. Gauss published "Disquisitiones Arithmeticae", essentially an expanded version of his PhD thesis, that revolutionized the study of number theory. One piece of new notation which Gauss introduced is the congruence or modulus notation. Let a, b, m be integers. We say that a is congruent to b modulo m, written

$$a \equiv b \mod m$$

if $m \mid (a - b)$. In this case, m is called the *modulus* and b is a *residue* of a modulo m.

The division algorithm may be restated in this new notation as follows.

Theorem 3.5. *If a and $m > 0$ are integers then there is an integer $r \in \{0, 1, ..., m - 1\}$ satisfying*

$$a \equiv r \mod m.$$

In other words, each integer has a residue mod m (called the least residue) which is in the range $0, 1, ..., m - 1$. Furthermore, this residue can be computed using the division algorithm. What is a practical way to find r for large numbers such as $a = 331$ and $m = 81$? First, we compute $331/81 \approx 4.0864...$ The obtained result is truncated (*not* rounded) to yield the largest integer $q = 4$ such that $331 \geq 81 \cdot q$. Then m is given by $331 - q \cdot 81 = 7$.

Example 3.6. *We have*

1. *$71 \equiv 1 \mod 7$,*

2. *$147 \equiv 3 \mod 12$,*

3. *$km \equiv 0 \mod m$ for any integers k and $m > 0$, and*

4. *$-1 \equiv 10 \mod 11$.*

3.2 GCD and LCM

In this section, we are concerned with two already introduced notations in number theory, the greatest common divisor and least common multiple (we already know

that this operations form a Boolean algebra). For the sake of a thorough exposition, let us start with the basic definitions.

Definition 3.7. *Let $a > 0$ and $b > 0$ be integers. The* greatest common divisor *of a, b is the largest integer $d > 0$ satisfying both $d \mid a$ and $d \mid b$. The* greatest common divisor *is denoted by* $\boxed{\gcd(a, b)}$.

In other words, the greatest common divisor of a and b is simply the largest positive integer which divides both a and b. When $gcd(a, b) = 1$ then we say that a, b are *relatively prime*.

Example 3.8. We have $\gcd(12, 15) = 3$, $\gcd(3, 5) = 1$, $\gcd(100, 46) = 2$.

Somewhat analogous to the gcd is the least integer which both a and b divide.

Definition 3.9. *Let $a > 0$ and $b > 0$ be integers. The* least common multiple *of a, b is the smallest integer $m > 0$ satisfying both $a \mid m$ and $b \mid m$. The least common multiple is denoted by $lcm(a, b)$.*

Example 3.10. We have $\mathrm{lcm}(12, 15) = 60$, $\mathrm{lcm}(3, 5) = 15$, $\mathrm{lcm}(100, 46) = 2300$.

The lcm can be computed from the gcd (which can be computed using the Euclidean algorithm) using the following fact.

Lemma 3.11. *Let a, b, c be integers.*

1. *$\gcd(a, b) \cdot \mathrm{lcm}(a, b) = ab$.*

2. *If $a \mid bc$ then $a \mid \gcd(a, b)c$.*

3. *If $a \mid bc$ and $\gcd(a, b) = 1$ then $a \mid c$.*

4. *If $a \mid c$ and $b \mid c$ and $\gcd(a, b) = 1$ then $ab \mid c$.*

5. *$\gcd(ab, c) = 1$ if and only if $\gcd(a, c) = 1$ and $\gcd(b, c) = 1$.*

6. *If $c \mid a$ and $c \mid b$ then $c \mid \gcd(a, b)$.*

7. *If $a \mid c$ and $b \mid c$ then $\mathrm{lcm}(a, b) \mid c$.*

8. *If $d = \gcd(a, b)$ then $\gcd(a/d, b/d) = 1$.*

9. *For any integers m, n we have $\gcd(a, b) \mid (ma + nb)$.*

Proof. Part (1): We will show that $\frac{ab}{\gcd(a,b)} = \mathrm{lcm}(a, b)$. Note that $\frac{b}{\gcd(a,b)} \in \mathbb{Z}$, since $\gcd(a, b)$ divides b. Therefore, $a \cdot \frac{b}{\gcd(a,b)} \in \mathbb{Z}$. Since $\frac{\frac{ab}{\gcd(a,b)}}{a} = \frac{b}{\gcd(a,b)} \in \mathbb{Z}$, we know that $\frac{ab}{\gcd(a,b)}$ is a multiple of a. Similarly, $\frac{\frac{ab}{\gcd(a,b)}}{b} = \frac{a}{\gcd(a,b)} \in \mathbb{Z}$ implies that $\frac{ab}{\gcd(a,b)} \geq \mathrm{lcm}(a, b)$. Now $\frac{ab}{\mathrm{lcm}(a,b)} \in \mathbb{Z}$ and $\frac{a}{\frac{ab}{\mathrm{lcm}(a,b)}} = \frac{\mathrm{lcm}(a,b)}{b} \in \mathbb{Z}$ since b divides $\mathrm{lcm}(a, b)$. Therefore, $\frac{ab}{\mathrm{lcm}(a,b)}$ divides a. Similarly, $\frac{ab}{\mathrm{lcm}(a,b)}$ divides b. Thus:

$$\frac{ab}{\mathrm{lcm}(a, b)} \leq \gcd(a, b) \quad \text{i.e.,} \quad \frac{ab}{\gcd(a, b)} \leq \mathrm{lcm}(a, b).$$

Since we have shown the inequality in both directions, we must have equality.

Part (2): Assume $a \mid bc$. Let $d = \gcd(a, b)$. There are integers x, y such that $ax + by = d$, so $acx + bcy = dc$. Since a divides acx and bcy (by assumption), it divides $acx + bcy$, hence $a \mid \gcd(a, b)c$.

Part (7): We will prove this by contradiction. Suppose that $\mathrm{lcm}(a, b) \nmid c$. Then $\frac{c}{\mathrm{lcm}(a,b)}$ is not an integer, and we may write it as:

$$\frac{c}{\mathrm{lcm}(a, b)} = q + \alpha, \quad \text{where} \quad 0 < \alpha < 1.$$

Multiplying both sides of the above inequality by the positive integer $\mathrm{lcm}(a, b)$ gives: $c = q \cdot \mathrm{lcm}(a, b) + \alpha \cdot \mathrm{lcm}(a, b)$. Let $r = c - q \cdot \mathrm{lcm}(a, b)$. Then:

$$c = q \cdot \mathrm{lcm}(a, b) + r, \quad \text{with} \quad 0 < r < \mathrm{lcm}(a, b).$$

But $a \mid r$ and $b \mid r$, so r is a common multiple of a and b which is less than the least common multiple $\mathrm{lcm}(a, b)$, by the above inequality. Thus we have reached a contradiction, and our original assumption was false. We conclude that $\mathrm{lcm}(a, b) \mid c$.

Part (4) shall be proven later as a consequence of the Fundamental Theorem of Arithmetic. Parts (3), (6), (8), (9) are EFY. □

3.3 The Euclidean algorithm

In this section, we will develop a method for computing $\gcd(a, b)$. This of practical importance. The principle which drives the method is the division algorithm, see Theorem 3.5. Starting with $r_{-1} = a$ and $r_0 = b$, the *Euclidean algorithm* repeatedly applies the division algorithm,

$$r_{i-1} = r_i q_{i+1} + r_{i+1}$$

until $r_k = 0$ for some k. Then $\gcd(a, b) = r_{k-1}$. Before we prove this statement, let us illustrate the progress of this algorithm by an example:

$$
\begin{array}{llllll}
r_{-1} & = & 331 & r_0 & = & 81 \\
r_0 & = & 81 & r_1 & = & 7 \\
r_1 & = & 7 & r_2 & = & 4 \\
r_2 & = & 4 & r_3 & = & 3 \\
r_3 & = & 3 & r_4 & = & 1 \\
\mathbf{r_4} & = & 1 & r_5 & = & 0
\end{array}
$$

Proof. [**Euclidean algorithm**] We can (and do) assume without loss of generality that $0 < b < a$. Since $0 \leq \ldots < r_1 < r_0 = b < r_{-1} = a$, at some point the above algorithm must terminate. Suppose that $r_k = 0$ and k is the smallest such integer. The claim is $\gcd(a, b) = r_{k-1}$. To see that, we first show that $r_{k-1} = ax + by$ for some integers x, y. Indeed, we claim that every r_i ($-1 \leq i < k$) is a integral linear combination of a, b. This may be proven by mathematical induction. (The details are left to the reader as an exercise. The cases $i = -1, 0, 1, 2$ follow from $a = bq_1 + r_1$, $b = r_1 q_2 + r_2$.) Thus $r_{k-1} = ax + by$. The fact $r_{k-1} = ax + by$ implies $\gcd(a, b) \mid r_{k-1}$.

Next, to finish the proof of the above claim, we show that $r_{k-1} \mid a$ and $r_{k-1} \mid b$. Indeed, we claim that r_{k-1} divides every r_i ($-1 \leq i < k$) (Again, the

details are left to the reader as an exercise. The cases $i = k-1, k-2, k-3$ follow from $r_{k-3} = r_{k-2}q_{k-1} + r_{k-1}$, $r_{k-2} = r_{k-1}q_k$.) The fact $r_{k-1} \mid a, r_{k-1} \mid b$ implies $r_{k-1} \mid \gcd(a,b)$.

The claim follows.   □

C/C++ excursion 4. *The remainder of two integers* `a` *and* `b` *in C is computed by* `a % b`. *A straight implementation of the Euclidean algorithm could look as follows:*

```
int gcd(int a, int b) {
  while (b ~= 0) {
    c = b;
        b = a % b;
        a = b;
  }
  return a;
}
```

A more elegant (but less efficient due to stack operations) variant employs recursion:

```
int gcd(int a, int b) {
  if (b == 0)
    return a;
  else
    return gcd(b, a % b);
}
```

As well as being extremely useful in practice, the Euclidean algorithm has important theoretical consequences.

Theorem 3.12. *Let a and b be positive integers, and let $d = \gcd(a,b)$. Then there are integers m and n such that*

$$d = ma + nb.$$

Proof. According to the Euclidean algorithm $d = r_{k-1}$, and using the penultimate equation we have

$$r_{k-1} = r_{k-3} - r_{k-2}q_{k-1}.$$

Thus d can be written in the form $m'r_{k-2} + n'r_{k-3}$, where $m' = -q_{k-1}$ and $n' = 1$. Substituting for r_{k-2} in terms of r_{k-3} and r_{k-4}, we obtain

$$d = m'(r_{k-4} - r_{k-3}q_{k-3}) + n'r_{k-3}$$

which can be written in the form $m''r_{k-3} + n''r_{k-4}$, with $m'' = n' - m'q_{k-2}$ and $n'' = m'$. Continuing this way we eventually obtain an expression for d in the required form.   □

For example, from the calculation used to find the gcd of 331 and 81 we obtain

$$
\begin{array}{rrrrr}
1 = & & 4 - 3 \times 1 = & 1 \times 4 + & (-1) \times 3 \\
= & 4 + & (-1) \times (7 - 4 \times 1) = & (-1) \times 7 + & 2 \times 4 \\
= & -7 + & 2 \times (81 - 7 \times 11) = & 2 \times 81 + & (-23) \times 7 \\
= & 2 \times 81 + & (-23) \times (331 - 81 \times 4) = & (-23) \times 331 + & 94 \times 81.
\end{array}
$$

Thus the required expression $\gcd(a, b) = ma + nb$ is

$$1 = (-23) \times 331 + 94 \times 81.$$

As an important application of Theorem 3.12, the following problem can be addressed:

Suppose that we have two integers $n > 0$ and $\phi > 1$ such that $\gcd(n, \phi) = 1$. Compute an integer s, $0 < s < \phi$, such that $ns \equiv 1 \mod \phi$.

We call s the *inverse of n modulo ϕ*. Since $\gcd(n, \phi)$, we can use the method explained in the proof of Theorem 3.12 to find numbers s' and t' such that $s'n + t'\phi = 1$. Then $ns' = -t'\phi + 1$, and, since $\phi > 1$, 1 is the remainder. Thus

$$ns' \equiv 1 \mod \phi.$$

Note that s' is almost the desired value; the problem is that s' may not satisfy $0 < s' < \phi$. However, we can convert s' to the proper value by setting s to the remainder of s under the integer division by ϕ.

3.4 Primes

Recall an integer p is prime if $p > 1$ and the only positive integers dividing p are 1 and p itself. The first few primes are

$$2, 3, 5, 7, 11, 13, 17, 19, \dots \, .$$

The primes form "building blocks" for the integers in some sense (made more precise by the Fundamental Theorem of Arithmetic below). We will later see how primes occur in the encryption of information passed over the Internet.

It has been know since the times of the Greeks that there are infinitely many primes. The following result is one of the oldest and best known results in mathematics!

Theorem 3.13 (Euclid's Second Theorem). *There are infinitely many primes.*

Proof. Suppose that the number of primes is finite. Then the sequence of primes $2, 3, 5, \dots$ must end. Let the last prime in this sequence be p. Let $n = (2 \cdot 3 \cdot 5 \cdots p) + 1$. Since n is larger than p, n is divisible by some prime q. If q is among $2, 3, 5, \dots, p$, then q would be a divisor of n and of the product $2 \cdot 3 \cdot 5 \cdots p$ and thus also of the difference $n - 2 \cdot 3 \cdot 5 \cdots p = 1$, which is impossible. So p is not the largest prime. That is, the sequence of primes does not end. □

In spite of their basic nature and importance, many questions about primes remain unknown. Question: Given a "random" integer n is there a "fast" method of determining if n is a prime or not? It has been a long-standing open problem to find a method which has an execution time that grows at most like a polynomial in the number of digits of n. This problem has only recently been solved by Agrawal, Kayal, and Saxena in 2002. It is still not known how to develop a fast method (i.e., with polynomial execution time growth) for factorizing a given integer. We will later see that the existence of such a method would render many of the encryption schemes used in passing sensible data in the internet insecure.

The rest of this section will be concerned with the factorization problem. For example,

$$825 = 3 \times 5 \times 5 \times 11.$$

Finding a factorization is not always as easy as this example suggests. Even a relatively small number such as 1807 presents some difficulty and a number such as $2^{67} - 1$ is really quite hard.

Given any positive integer $n > 1$, there is a prime factorization of n. This is a consequence of the following argument. Suppose there is a "bad" positive integer (a number greater than 1 that cannot be expressed as a product of primes). Then consider the smallest "bad" number, m. Now m cannot be a prime p, because if so we have the trivial factorization $m = p$. So $m = rs$ where $1 < r < m$ and $1 < s < m$. Since m is the smallest "bad" number, both r and s do have factorizations. But in that case the equation $m = rs$ yields an expression for m as a product of primes, contradicting the fact that m is "bad". Hence there are no "bad" numbers.

We now turn to the question of uniqueness. If we are asked to factorize 990, we might proceed as follows:

$$990 = 2 \times 495 = 2 \times 5 \times 99 = \dots.$$

On the other hand, we might start in a different way:

$$990 = 11 \times 90 = 11 \times 2 \times 45 = \dots.$$

You are probably confident that the answers will be the same, although possibly the order of the factors will differ. However, this fact must be proved. The key step for this proof is the following result.

Lemma 3.14. *If p is a prime and $x_1, x_2, \ldots, x_n$ are integers such that*

$$p \mid x_1 x_2 \cdots x_n$$

then $p \mid x_i$ for some x_i $(1 \le i \le n)$.

Proof. We use the principle of induction. The result is plainly true when $n = 1$ but for reasons that will appear, it is convenient to start by proving the case $n = 2$.

Suppose then that $p \mid x_1 x_2$. We shall prove that if p does not divide x_1 then p must divide x_2. Now, if p does not divide x_1 then (since 1 and p are the only positive divisors of p) we must have $\gcd(p, x_1) = 1$. From Theorem 3.12 there are integers r and s such that $rp + sx_1 = 1$. Hence

$$x_2 = (rp + sx_1)x_2 = (rx_2)p + s(x_1 x_2).$$

Since p divides both terms it follows that $p \mid x_2$, as required.

Suppose the result holds when $n = k$, and consider the case $n = k + 1$, that is when p is a divisor of the product $x_1 x_2 \cdots x_k x_{k+1}$. Define $X = x_1 x_2 \cdots x_k$ so that $p \mid X x_{k+1}$. If $p \mid X$ then, by the induction hypothesis, $p \mid x_i$ for some x_i in the range $1 \le i \le k$. On the other hand, if p does not divide X, then by the result for the case $n = 2$, we must have $p \mid x_{k+1}$. Thus the induction step is done and the results holds for all n.   □

Note that the above result holds only for *primes p* (and not for other positive integers). This is a crucial property of primes, which makes prime factorizations

unique. It is not possible to obtain unique factorizations by other sets of positive
integers.

Theorem 3.15 (The Fundamental Theorem of Arithmetics). *A positive
integer $n \geq 2$ has a unique prime factorization, apart from the order of the factors.*

Proof. We already know the existence of prime factorizations, it remains to prove
their uniqueness. If there is a number for which we have non-uniqueness, then there
is a smallest one N. That is,

$$N = p_1 p_2 \cdots p_k, \quad \text{and} \quad N = q_1 q_2 \cdots q_l,$$

where the p_i and the q_j are primes, not necessarily distinct. Write

$$N = p_1 N', \quad \text{where} \quad N' = p_2 \cdots p_k.$$

Since $p_1 \mid N$ and $N = q_1 q_2 \cdots q_l$, it follows from Lemma 3.14 that p_1 divides one of
the factors, say q_j. In fact, since q_j is a prime $p_1 = q_1$ (see exercises). Thus we can
cancel the terms p_1 and q_j from the two expressions for N, obtaining

$$N' = p_2 \cdots p_k = q_1 q_2 \cdots q_{j-1} q_{j+1} \cdots q_l,$$

So N' has two prime factorizations. But $N' < N$, so the factorizations must be the
same, apart from the order of the factors. If we now re-introduce the (equal) factors
p_1 and q_j, we conclude that the original factorizations of N must be the same, apart
from the order of the factors. This contradicts the definition of N. Hence, there
can be no such N, and the theorem is true for $n \geq 2$. $\square$

We usually collect equal primes in the factorization of n and write

$$n = p_1^{e_1} p_2^{e_2} \cdots p_r^{e_r},$$

where $p_1, p_2, \ldots, p_r$ are distinct primes and $e_1, e_2, \ldots, e_r$ are positive integers. For
example, $7000 = 2^3 \times 5^3 \times 7$.

Many important results in number theory are proved using prime factoriza-
tions. The following lemma is an immediate consequence of the Fundamental The-
orem of Arithemetic.

Lemma 3.16. *Let $a = p_1^{e_1} p_2^{e_2} \cdots p_k^{e_k}$ and $b = p_1^{f_1} p_2^{f_2} \cdots p_k^{f_k}$ be any positive
integers with the given prime factorizations ($e_i \geq 0$, $f_i \geq 0$, for all i, $1 \leq i \leq k$).
Then the following is true:*

1. $\gcd(a,b) = p_1^{m_1} p_2^{m_2} \cdots p_k^{m_k}$, *where* $m_i = \min(e_i, f_i)$,

2. $\mathrm{lcm}(a,b) = p_1^{M_1} p_2^{M_2} \cdots p_k^{M_k}$, *where* $M_i = \max(e_i, f_i)$.

3.5   The Sieve of Eratosthenes

In this section, we briefly present a simple general method for determining if a
number is prime. There are much more efficient (and complicated) primality tests
than the one we discuss here.

The key fact that is used for the method discussed in this section is the fact that if n is composite then it must have a prime factor which is less than or equal to $\sqrt{n}$, see Lemma 3.2.

The method to produce all primes up to $N > 2$:

1. List all integers $2, ..., n$.

2. Let $a = 2$.

3. Cross out all multiples of a except for a itself.

4. If all integers between a and N are crossed out then stop. Otherwise, replace a by then next largest integer which has not been crossed out. If this new a is greater than $\sqrt{N}$ then stop.

5. Go to step 3.

This process must terminate after at most $\sqrt{N}$ steps. Here is an example for $N = 20$ ($\sqrt{N} \approx 4.5$).

Step 1:

$2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20$

Step 2: Cross out multiples of 2:

$2, 3, \cancel{4}, 5, \cancel{6}, 7, \cancel{8}, 9, \cancel{10}, 11, \cancel{12}, 13, \cancel{14}, 15, \cancel{16}, 17, \cancel{18}, 19, \cancel{20}$

Step 3: Cross out multiples of 3:

$2, 3, \cancel{4}, 5, \cancel{6}, 7, \cancel{8}, \cancel{9}, \cancel{10}, 11, \cancel{12}, 13, \cancel{14}, \cancel{15}, \cancel{16}, 17, \cancel{18}, 19, \cancel{20}$

All the remaining numbers are prime.

3.6   Encrypting messages

Suppose that Alice wants to send Bob an e-mail that noone else should be able to read. Sending a plain text message is not an option; always remember that sending an e-mail is like sending a picture card, easily readable for everyone who has access to one of the many servers the message must pass before it is transmitted from Alice to Bob. To keep the message secret, Alice (the sender) *encrypts* the message into a seemingly meaningless sequence of letters, and Bob (the receiver) *decrypts* this sequence back to the original message. If the cryptosystem is secure, unauthorized persons will be unable to discover the decryption technique, so even if they read the encrypted message, they will be unable to decrypt it. For example, if a credit card number is sent over the Internet, it is important for the number to be read only by the intended recipient. In this section, we look at some algorithms that support secure communication.

In one of the oldest and simplest systems, the sender and receiver each have a key that defines a substitute character for each potential character to be sent. Moreover, the sender and receiver do not disclose the key. Such keys are said to be private.

Example 3.17. If a key is defined as (note that the first character in the following sequence is the space)

```
characters:          ABCDEFGHIJKLMNOPQRSTUVWXYZ
replaced by:         EIJFUAXVHWP GSRKOBTQYDMLZNC
```

the message `SEND MONEY` would be encrypted as `QARUESKRAN`. The encrypted message `SKRANEKRELIN` would be decrypted as `MONEY ON WAY`.

This system has two fundamental flaws:

1. *Long messages can be easily deciphered by frequency analysis.* On average, every letter appears with a certain frequency in an English text. For example, "e" and "z" have the frequencies 12.7% and 0.1%, respectively. This means that in a text of 1000 letters it can be expected that e appears approximately 127 times while z appears only approximately 10 times. The longer the text the more precise these estimates become. Now, if in an encrypted text of 1000 letters the letter "a" appears 120 times, the chances are quite high that "a" stands for "e" in the original text ("t", the second most frequent letter, has a frequency of 9.1%). This technique can be refined by exploiting various other properties of long texts, e.g., "the" is a frequently encountered word.

2. *The need to exchange private keys.* The sender needs to send the key to the receiver in a secure fashion. This might not be a problem for Alice and Bob, who can meet each other regularly and can exchange keys from person to person. But key distribution is a difficult problem for companies. Imagine a bank wants to send some confidential data to a client. The bank picks some key (because of 1, it cannot use the same key frequently) and encrypts the message. How does the bank inform the client of the key? The only secure way to send a set of keys is to hand it over in person, an impracticable and time-consuming task.

Of the two flaws, the second is the one that really hurts. The first disadvantage can be addressed by using fancy, complicated encryption schemes such as DES, but until the mid-1970's it was believed that the problem of key distribution was unsolvable. But then Ronald A. Rivest, Adi Shimar and Leonard M. Adleman came up with the so called RSA public-key cryptosystem which avoids the problem of key distribution by using different keys for encrypting and decrypting a message. The idea is that Bob generates a public and a private key, the first is for encrypting messages and the second for decrypting messages. He keeps the private key secret (on his computer) and sends the public key to Alice, who uses this key to encrypt her message. Only Bob knows the private key, so noone besides him (not even Alice) is able to decrypt the message. While the described concept might sound pretty simple, its realization requires many of the concepts from number theory we have had in this chapter.

In the RSA system, messages are represented as numbers. For example, each character might be represented as a number. If a blank space is represented as 1, "A" as 2, "B" as 3, and so on, the message `SEND MONEY` would be represented as 20, 6, 15, 5, 1, 14, 16, 15, 6, 26. The integers can be combined into the single integer

$$20061505011416150626$$

by adding leading zeros to all single-digit numbers.

We next describe how RSA works, present a concrete example, and then discuss why it works. Each prospective recipient (Bob) chooses two primes p and q and computes $z = pq$. Since the security of the RSA system rests primarily on the inability of anyone knowing the value of z to discover the numbers p and q, the numbers p and q are typically chosen so that each has 100 or more digits. Next,

the prospective recipient computes $\phi = (p-1)(q-1)$ and chooses an integer n such that $\gcd(n, \phi) = 1$. In practice, n is often chosen to be a prime. The pair z, n is then made public. Finally, the prospective recipient computes the unique number s, $0 < s < \phi$, satisfying $ns \equiv 1 \mod \phi$ (an efficient way for computing s is given in Section 3.3). The number s is kept secret and used to decrypt messages.

To send the integer a, $0 \le a \le z - 1$ (in practice, messages are broken into smaller parts corresponding to integers smaller than z), to the holder of the public key z, n (Bob), the sender (Alice) computes the remainder c of a^n under division by z and sends c (how to compute this efficiently will be discussed in the exercises). To decrypt the message, the recipient computes the remainder of c^s under division by z, which can be shown to be equal to a.

Example 3.18. Suppose that we choose $p = 23, q = 31$, and $n = 29$. Then $z = pq = 713$ and $\phi = (p - 1)(q - 1) = 660$. Now $s = 569$ since

$$29 \times 569 = 16501 \equiv 1 \mod 660.$$

Bob makes the pair $z = 713, n = 29$ publicly available. For Alice to transmit the message $a = 572$ to Bob, she computes

$$a^n = 572^{29} \equiv \underbrace{113}_{=:c} \mod 713$$

and sends $c = 113$. Bob computes

$$c^s = 113^{569} \equiv 572 \mod 713$$

in order to decrypt the message.

The main result that makes encryption and decryption work is that

$$a^u \equiv a \mod z, \quad \text{for all} \quad 0 \le a < z \text{ and } u \equiv 1 \mod \phi. \tag{3.1}$$

Using this result and the following theorem, we will show that decryption produces the correct result.

Theorem 3.19. *If a, b, and z are positive integers and $a \equiv r \mod z$, $b \equiv q \mod z$, then*

$$ab \equiv rq \mod z.$$

Since $ns \equiv 1 \mod \phi$, using (3.1) and Theorem 3.19 gives

$$c^s \mod z = (a^n \mod z)^s \mod z = a^{ns} \mod z = a,$$

wich proves that Bob indeed decrypts a. The security of RSA relies heavily on the fact that currently there is no efficient algorithm for factoring integers.

Cryptology is a fascinating subject still under development. One of the challenging problems is to hide an encrypted message in an innocently looking piece of information with noone even able to decide whether the innocent information contains an not-so-innocent encrypted message, leave alone extracting the encrypted message. An example in this direction are water mark technologies in digital images.

Chapter 4

Relations

Relations generalize the notion of functions. The presence of an ordered pair (a, b) is interpreted as indicating a relationship from a to b. But in contrast to functions there might be some $c \neq b$ for which there is also a relationship from a to c.

Definition 4.1. *A* (binary) relation R *from a set* X *to a set* Y *is a subset of the Cartesian product* $X \times Y$. *If* $(x, y) \in R$, *we write* $x\,R\,y$ *and say that* x *is related to* y. *If* $X = Y$, *we call* R *a relation on* X.

Similarly as for functions, the set

$$\{x \in X : (x, y) \in R \text{ for some } y \in Y\}$$

is called the *domain* of R. The set

$$\{y \in Y : (x, y) \in R \text{ for some } x \in X\}$$

is called the *range* of R. A function f is a special type of relation, additionally satisfying the following two properties

  1. The domain of f is equal to X.

  2. For each $x \in X$, there is exactly one $y \in Y$ such that $(x, y) \in f$.

Example 4.2. Let R be the relation on $X = \{1, 2, 3, 4\}$ defined by $(x, y) \in R$ if $x \leq y, y \in X$. Then

$$R = \{(1, 1), (1, 2), (1, 3), (1, 4), (2, 2), (2, 3), (2, 4), (3, 3), (3, 4), (4, 4)\}.$$

The domain and range of R are both equal to X.

An informative way to picture a relation on a set is to draw its *digraph*. To draw the digraph of a relation on a set X, we first draw dots (called *vertices*) to represent the elements of X. Next, if the element (x, y) is in the relation, we draw an arrow (called a *directed edge*) from x to y. In Figure 4.1, we have drawn directed edges to represent the members of the relation of Example 4.2. Note that an element of the form (x, x) in a relation corresponds to a directed edge from x to x. Such an edge is called a *loop*.
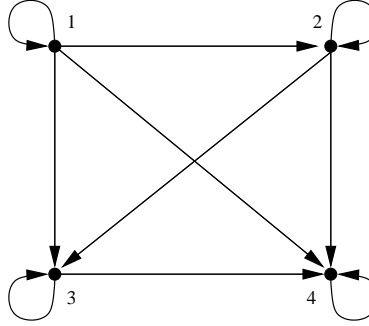
Figure 4.1. *The digraph of the relation of Example 4.2.*

We next define several properties a relation may have.

Definition 4.3. *A relation R on a set X is called* reflexive *if $(x, x) \in R$ for all $x \in X$.*

The relation of Example 4.2 is reflexive. The digraph of a reflexive relation has a loop at every vertex.

Definition 4.4. *A relation R on a set X is called* symmetric *if for all $x, y \in X$ with $(x, y) \in R$ we also have $(y, x) \in R$.*

The relation of Example 4.2 is not symmetric. For example $(1, 3) \in R$ but $(3, 1) \notin R$. Edges in the digraph of a symmetric relation always go in both directions (usually denoted by $\longleftrightarrow$).

Definition 4.5. *A relation R on a set X is called* antisymmetric *if for all $x, y \in X$ with $(x, y) \in R$ and $x \neq y$, we have $(y, x) \notin R$.*

The relation of Example 4.2 is antisymmetric. Edges in the digraph of a antisymmetric relation never go in both directions.

Remark 4.6. *A relation which has* no *members of the form (x, y) with $x \neq y$ is, by the definition, antisymmetric.*

Definition 4.7. *A relation R on a set X is called* transitive *if for all $x, y, z \in X$ with $(x, y) \in R$ and $(y, z) \in R$, we also have $(x, z) \in R$.*

The relation of Example 4.2 is transitive. To formally verify that this relation satisfies Definition 4.7, we can list all pairs (x, y) and (y, z) with $x \neq y$ and $y \neq z$, and then verify that $(x, z) \in R$:

(x, y)	(y, z)	(x, z)
$(1, 2)$	$(2, 3)$	$(1, 3)$
$(1, 2)$	$(2, 4)$	$(1, 4)$
$(1, 3)$	$(3, 4)$	$(1, 4)$
$(2, 3)$	$(3, 4)$	$(2, 4)$

This method of listing will be tedious for large X and impossible for infinite X. A more elegant of proof that the relation of Example 4.2 is transitive could be as follows:

$$(x, y), (y, z) \in R \;\Rightarrow\; x \leq y, y \leq z \;\Rightarrow\; x \leq z \;\Rightarrow\; (x, z) \in R.$$

4.1   Partial order

Relations can be used to order elements of a set. For example, the relation R defined on the set of integers by

$$(x, y) \in R \qquad \text{if } x \leq y$$

orders the integers. Note that the relation R is reflexive, antisymmetric, and transitive.

Definition 4.8. *A relation R on a set X is called a* partial order *if R is reflexive, antisymmetric, and transitive.*

Example 4.9. The relation defined on the positive integers by

$$(x, y) \in R \qquad \text{if } x \text{ divides } y$$

is reflexive $(x \mid x)$, antisymmetric $(x \mid y, y \mid x \;\Rightarrow\; x = y)$, and transitive $(x \mid y, y \mid z \;\Rightarrow\; x \mid z)$. Thus, R is a partial order.

If R is a partial order on a set X, the notation $x \prec y$ is sometimes used to indicate that $(x, y) \in R$. This notation suggests that we are interpreting the relation as an ordering of the elements in X.

Suppose that R is a partial order on a set X. If $x, y \in X$ and $x \prec y$ or $y \prec x$, we say that x and y are *comparable*. If $x, y \in X$ and $x \nprec y$ and $y \nprec x$, we say that x and y are *incomparable*. If every pair of elements in X is comparable, we call R a *total order*. The less than or equal relation of Example 4.2 is a total order since, if x and y are integers, we have $x \leq y$ or $y \leq x$. The "divides" relation of Example 4.9 has both comparable and incomparable elements and is therefore not a total order. For example, 2 and 3 are incomparable (since 2 does not divide 3 and 3 does not divide 2).

4.2   Equivalence relations

Suppose we have a set X of 10 balls, each of which is either red, blue, or green. If we divide the balls into sets R, B, and G according to their color, then we have a partition of X. More formally, a partition of a set is defined as follows.

Definition 4.10. *A collection $\mathcal{S}$ of nonempty subsets of a set X is said to be a* partition *of the set X if every element in X belongs to* exactly one *member of $\mathcal{S}$*

Example 4.11. If $X = \{1, 2, 3, 4, 5, 6, 7, 8\}$ then

$$\mathcal{S}_1 = \{\{1, 4, 5\}, \{2, 6\}, \{3\}, \{7, 8\}\}$$

is a partition of X but

$$\mathcal{S}_2 = \{\{1, 3, 5, 7\}, \{2, 4, 6\}\}, \quad \mathcal{S}_3 = \{\{1, 3, 5, 7\}, \{2, 4, 6, 8\}, \{3, 4\}\}$$

are not.

Theorem 4.12. *Let $\mathcal{S}$ be a partition of a set X. Define the relation $x\,R\,y$ to mean that for some set S in $\mathcal{S}$, both x and y belong to S. Then R is reflexive, symmetric, and transitive.*

Proof. Let $x \in X$. By the definition of a partition, x belongs to some member S of $\mathcal{S}$. Thus, $x\,R\,x$ and R is reflexive.

Suppose that $x\,R\,y$. Then both x and y belong to some set $S \in \mathcal{S}$. Since both y and x belong to S, we also have $y\,R\,x$ and R is symmetric.

Finally, suppose $x\,R\,y$ and $y\,R\,z$. Then both x and y belong to some set $S \in \mathcal{S}$ and both y and z belong to some set $T \in \mathcal{S}$. Since y belongs to exactly one set of the partition, we must have $S = T$. Therefore, both y and z belong to S, which implies $x\,R\,z$ and consequently R is transitive.   □

Example 4.13. Consider the partition $\mathcal{S}_1$ from Example 4.11. Then the relation R on X given by Theorem 4.12 contains the ordered pairs $(1,1), (1,4), (1,5)$ because $\{1,4,5\}$ is in $\mathcal{S}_1$. The complete relation is given by

$$R = \{(1,1), (1,4), (1,5), (4,1), (4,4), (4,5), (5,1), (5,4), (5,5),$$
$$(2,2), (2,6), (6,2), (6,6), (3,3), (7,7), (7,8), (8,7), (8,8)\}.$$

Let $\mathcal{S}$ and R be as in Theorem 4.12. If $S \in \mathcal{S}$, we can regard the members of S as equivalent in the sense of relation R, which motivates calling relations that are reflexive, symmetric, and transitive *equivalence relations*. In the starting example, the relation is "is the same color as"; hence equivalent means that two balls have the same color.

Given an equivalence relation on a set X, we can partition X by grouping related members of X. The following theorem, which can be thought as a converse of Theorem 4.12, gives the details of this idea.

Theorem 4.14. *Let R be an equivalence relation on a set X. For each $a \in X$, let*

$$[a] = \{x \in X : x\,R\,a\}.$$

Then $\mathcal{S} = \{[a] : a \in X\}$ is a partition of X.

Proof. We must show that every element in X belongs to exactly one member of $\mathcal{S}$. Let $a \in X$. Since $a\,R\,a$, we have $a \in [a]$. Thus every element of X belongs to *at least one* member of $\mathcal{S}$. It remains to show that every element in X belongs to *exactly one* member of $\mathcal{S}$, that is

$$\text{if } x \in X \text{ and } x \in [a] \cap [b], \text{ then } [a] = [b]. \tag{4.1}$$

We first show that for all $c, d \in X$, if $c\,R\,d$, then $[c] = [d]$. Suppose that $c\,R\,d$. Let $x \in [c]$. Then $x\,R\,c$. Since $c\,R\,d$ and R is transitive, $x\,R\,d$. Therefore, $x \in [d]$ and $[c] \subseteq [d]$. The argument that $[d] \subseteq [c]$ is the same as that just given, but with the roles of c and d interchanged. Thus $[c] = [d]$. We now prove (4.1). Assume that $x \in X$ and $x \in [a] \cap [b]$. Then $x\,R\,a$ and $x\,R\,b$. Our preceeding result shows that $[x] = [a]$ and $[x] = [b]$. Thus $[a] = [b]$.   □

Definition 4.15. *Let R be an equivalence relation on a set X. The sets $[a]$ defined in Theorem 4.14 are called the* equivalence classes *of X given by the relation R.*
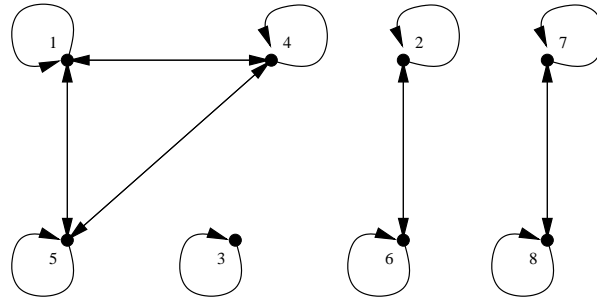
Example 4.16. Let us consider the equivalence relation R from Example 4.13:

$$R = \{(1,1), (1,4), (1,5), (4,1), (4,4), (4,5), (5,1), (5,4), (5,5),$$
$$(2,2), (2,6), (6,2), (6,6), (3,3), (7,7), (7,8), (8,7), (8,8)\}.$$

The equivalence class $[1]$ containing 1 consists of all x such that $(x,1) \in R$. Therefore, $[1] = \{1,4,5\} = [4] = [5]$. The remaining equivalence classes are found similarly:

$$[2] = [6] = \{2,6\}, \quad [3] = \{3\}, \quad [7] = [8] = \{7,8\}.$$

The equivalence classes appear quite clearly in the digraph of the relation:

The three equivalence classes appear as subgraphs, each of which is completely connected (in each subgraph there is an edge from every vertex to every other vertex). The subgraphs themselves, however, are completely disconnected from each other, i.e., there is no edge from one subgraph to the other.

4.3   Matrices of relations

A matrix is a convenient way to represent a relation R from X to Y if both sets are finite. Such a representation can be used by the computer to analyse a relation. We label the rows with the elements of X (in some arbitrary order), and we label the columns with the elements of Y (in some arbitrary order). We then set the entry in row x and column y to 1 if $x\,R\,y$ and to 0 otherwise. This matrix is called the *matrix of the relation R*.

Example 4.17. The matrix of the relation

$$R = \{(1,b), (1,d), (2,c), (3,c), (3,b), (4,a)\}$$

from $X = \{1,2,3,4\}$ to $Y = \{a,b,c,d\}$ relative to the orderings $1,2,3,4$ and a,b,c,d is

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{cccc} a & b & c & d \\ \left[\begin{array}{cccc} 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 \end{array}\right]. \end{array}$$

A different ordering of the elements in X and Y yields a different matrix. For example if we use the orderings $2, 3, 4, 1$ and d, b, a, c then

$$
\begin{array}{c}
 \\
2 \\
3 \\
4 \\
1
\end{array}
\begin{array}{cccc}
d & b & a & c \\
\left[\begin{array}{cccc}
0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 \\
1 & 1 & 0 & 0
\end{array}\right].
\end{array}
$$

When we write the matrix of a relation R on a set X (i.e., from X to X), we use the same ordering for the rows as we do for the columns.

Example 4.18. The matrix of the relation

$$R = \{(a, a), (b, b), (c, c), (d, d), (b, c), (c, b)\}$$

on $\{a, b, c, d\}$ relative to the ordering a, b, c, d is given by

$$
\begin{array}{c}
 \\
a \\
b \\
c \\
d
\end{array}
\begin{array}{cccc}
a & b & c & d \\
\left[\begin{array}{cccc}
1 & 0 & 0 & 0 \\
0 & 1 & 1 & 0 \\
0 & 1 & 1 & 0 \\
0 & 0 & 0 & 1
\end{array}\right].
\end{array}
$$

Note that the matrix of a relation on a set X is always a square matrix.

We can quickly determine whether a relation R on a set X is reflexive by examining the matrix A of R. The relation R is reflexive if and only if A has 1's on the diagonal. We can also quickly determine whether a relation R is symmetric. The relation R is symmetric if and only if for all i and j, the $(i, j)th$ entry of A equals the (j, i)th entry of A. It is more difficult to check whether a given relation is transitive. For this purpose, we introduce the composition of two relations.

Definition 4.19. *If R_1 is a relation from X to Y and R_2 is a relation from Y to Z, the composition of R_1 and R_2, denoted by $R_2 \circ R_1$, is the relation from X to Z defined by*

$$R_2 \circ R_1 = \{(x, z) : \ (x, y) \in R_1 \ and \ (y, z) \in R_2 \ for \ some \ y \in Y\}.$$

The matrix representing the relation $R_2 \circ R_1$ can be obtained by multiplying the matrices representing R_2 and R_1. Before we state this result formally, let us consider an example.

Example 4.20. Let R_1 be the relation from $X = \{1, 2, 3\}$ to $Y = \{a, b\}$ defined by

$$R_1 = \{(1, a), (2, b), (3, a), (3, b)\},$$

and let R_2 be the relation from Y to $Z = \{x, y, z\}$ defined by

$$R_2 = \{(a, x), (a, y), (b, y), (b, z)\}.$$

Matrices A_1 and A_2 representing R_1 and R_2 are given by

$$A_1 = \begin{array}{c} 1 \\ 2 \\ 3 \end{array} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 1 \end{bmatrix}, \quad A_2 = \begin{array}{c} a \\ b \end{array} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}.$$

where columns of A_1 are labelled $a\ b$ and columns of A_2 labelled $x\ y\ z$.

The product of these matrices is given by

$$A_1 A_2 = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 2 & 1 \end{bmatrix}.$$

The exact values of this matrix are not so much of interest; it is only important whether an entry is zero or nonzero. We therefore set all nonzeros entries of this matrix to one and obtain the following matrix representing the relation $R_2 \circ R_1$,

$$\begin{array}{c} 1 \\ 2 \\ 3 \end{array} \begin{bmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}.$$

with columns labelled $x\ y\ z$.

Theorem 4.21. *Let R_1 be a relation from X to Y and let R_2 be a relation from Y to Z. Choose orderings of X, Y, and Z. Let A_1 be the matrix of R_1 and let A_2 be the matrix of R_2 with respect to the orderings selected . The matrix of the relation $R_2 \circ R_1$ with respect to these orderings is obtained by replacing each nonzero term in the matrix product $A_1 A_2$ by 1.*

Theorem 4.21 gives a quick test for determining whether a given relation is transitive. If A is the matrix of R (relative to some ordering), we compute A^2. We then compare A and A^2. The relation R is transitive if and only if whenever an entry (i, j) in A^2 is nonzero, the entry (i, j) in A is also nonzero. The reason is that an entry (i, j) in A^2 is nonzero if and only if there are elements (i, k) and (k, j) in R (this follows from the rules of matrix-matrix multiplication). Now R is transitive if and only if whenever (i, k) and (k, j) are in R, then (i, j) is in R. But (i, j) is in R if and only if the entry (i, j) in A is nonzero.

Example 4.22. The matrix of the relation in Example 4.18 is given by

$$\begin{array}{c} a \\ b \\ c \\ d \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

with columns labelled $a\ b\ c\ d$.

Its square is

$$A^2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 2 & 2 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

We see that whenever an entry (i, j) in A^2 is nonzero, the entry (i, j) in A is also nonzero. Thus, the underlying relation is transitive.

Example 4.23. The matrix of the relation

$$R = \{(a, a), (b, b), (c, c), (d, d), (a, c), (c, b)\}$$

on $\{a, b, c, d\}$ relative to the ordering a, b, c, d is given by

$$A = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

Its square is

$$A^2 = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

The entry $(1, 2)$ of A^2 is nonzero but the corresponding entry in A is zero. Thus, R is not transitive.

Chapter 5

Principles of Counting

A major theme of this course is the development of effective techniques for counting a finite set X. When X arises in a complex problem we may require sophisticated counting methods far removed from the say-and-point technique of constructing a bijective correspondence between $\mathbb{N}$ and X. But before, we recall one of the most important techniques for proving counting principles. It is also important in virtually any branch of mathematics and computing science.

5.1 Proof by induction

The principle of *mathematical induction* can be formally stated in terms of a predicate $P(\cdot)$ on the set of natural numbers.

Axiom 5.1. *Suppose that $P(n)$ is a statement with the following properties:*

  (i) *$P(1)$ is true.*

  (ii) *if $P(k)$ is true then $P(k+1)$ is true for every $k \in \mathbb{N}$.*

Then $P(n)$ is true for all $n \in \mathbb{N}$.

Example 5.2. Suppose we want to prove that the following statement $P(n)$ is true for all $n \in \mathbb{N}$:
$$P(n) = \{n^3 + 5n \text{ is a multiple of } 6\}.$$

We have to check that the two properties of Axiom 5.1 are satisfied:

  (i) $P(1)$ is true.

  (ii) if $P(k)$ is true then $P(k+1)$ is true for every $k \in \mathbb{N}$.

Property (i) is satisfied since $1^3 + 5 \cdot 1 = 6$ is trivially a multiple of 6. Showing property (ii) requires more work (this is a typical "feature" of proofs by induction). Suppose that $P(k)$ is true; that is, $k^3 + 5k = 6m$, where m is a natural number. We have to deduce that $P(k+1)$ is true. Inserting $n = k+1$ in the expression $n^3 + 5n$, we obtain
$$(k+1)^3 + 5(k+1) = (k^3 + 3k^2 + 3k + 1) + (5k + 5).$$

In order to use the assumption that $k^3 + 5k = 6m$, we rewrite this as follows:

$$(k^3 + 5k) + 3(k^2 + k + 2) = 6m + 3k(k+1) + 6.$$

It remains to show that this is a multiple of 6. Now, we already know that $k(k+1)$ is always an even number, say $2r$. This shows that the last expression is equal to $6(m + r + 1)$. To summarize, we have shown that $P(1)$ is true and that if $P(k)$ is true, then $P(k + 1)$ is true. Applying the principle of induction (Axiom 5.1), it follows that $P(n)$ is true for all $n \in \mathbb{N}$.

Often it is convenient to use the following terminology. The statement $P(1)$ is called the *induction basis*, the assumption that $P(k)$ is called the *induction hypothesis*, and the proof that $P(k)$ implies $P(k + 1)$ is called the *induction step*.

5.2   The addition principle

Our first rule of counting has been used in practice since the dawn of civilization (not in such a formalized form, though).

Theorem 5.3. *If A and B are non-empty finite sets, and A and B are disjoint, then*
$$|A \cup B| = |A| + |B|.$$

Proof. Since A and B are non-empty and finite, we can list A and B in the standard way as
$$A = \{a_1, a_2, \ldots, a_r\}, \quad B = \{b_1, b_2, \ldots, b_s\}.$$
Since A and B are disjoint, $A \cup B$ can be listed in the standard way as
$$A \cup B = \{c_1, c_2, \ldots, c_r, c_{r+1}, \ldots, c_{r+s}\},$$
where
$$c_i = \begin{cases} a_i & \text{if } i \leq r, \\ b_{i-r} & \text{if } i > r. \end{cases}$$
Hence $|A \cup B| = r + s = |A| + |B|$, as claimed.   □ It is clear that the rule is still valid if A, or B, or both A and B, are empty. Furthermore, the rule can be extended to the union of any number of mutually disjoint sets $A_1, A_2, \ldots, A_n$ in the obvious way,
$$|A_1 \cup A_2 \cup \cdots \cup A_n| = |A_1| + |A_2| + \cdots + |A_n|.$$
This result can be proven by mathematical induction using Theorem 5.3 for the induction step.

A simple application of this rule leads to the so called *pigeonhole principle* Suppose that a number of objects are distributed into n boxes, and A_i denotes the set of objects which are in box i, $1 \leq i \leq n$. Since the sets A_i are disjoint (no object can be in two boxes at the same time), the total number of objects is $|A_1| + |A_2| + \cdots + |A_n|$, and if no box contains more than r objects this number is at most
$$r + r + \cdots + r = nr.$$
Putting this argument in reverse, we have the pigeonhole principle:

> *If m objects are distributed into n boxes and m > nr, then at least one box*
> *contains at least r + 1 objects.*

Example 5.4. Suppose we want to show that in any set of six people there are
either three mutual friends or three mutual strangers.

For this purpose, let a be any one of the people, and distribute the other five
people into two "boxes", box 1 containing the people who are friends with a, and
box 2 containing those who are strangers to a. Since $5 > 2 + 2$, one of these boxes
contains at least three people. Suppose box 1 contains β, γ, δ (and possibly other
people). If any two of $\{\beta, \gamma, \delta\}$ are friends, say β and γ, then $\{\alpha, \beta, \gamma\}$ is a set of
three mutual friends. On the other hand, if no pair of $\{\beta, \gamma, \delta\}$ are friends, then
$\{\beta, \gamma, \delta\}$ is a set of three mutual strangers.

If it happens that box 2 contains three or more people, a parallel argument
with friends and strangers interchanged leads to the same conclusions.

5.3 Counting pairs and the multiplication principle

Quite often we have to count things which can be described more naturally as pairs,
rather than single objects. To illustrate such situations, consider the following menu
of a junk food restaurant:

Appetizers	*Main Courses*	*Beverages*
Nachos	Hamburger	Tea
Salad	Cheeseburger	Milk
	Fish Filet	Cola
		Root Beer

If we list all possible dinners consisting of one main course and one beverage,

$$\text{HT, HM, HC, HR, CT, CM, CC, CR, FT, FM, FC, FR,}$$

we see that there are 12 different dinners. Notice that there are three main courses
and four beverages, and $12 = 3 \cdot 4$. Similarly, there are 24 possible dinners consisting
of one appetizer, one main course, and one beverage. Notice that there are two
appetizers, three main courses and four beverages, and $12 = 2 \cdot 3 \cdot 4$.

If we let X denote the set of all appetizers, Y the set of all main courses, and
Z the set of all beverages, then each element in the set $X \times Y \times Z$ is a possible
dinner consisting of one appetizer, one main course, and one beverage. What we
have observed above can be stated formally as

$$|X \times Y \times Z| = |X| \cdot |Y| \cdot |Z|.$$

Some dinner combinations could be a matter of taste. Let us say a guest excludes
certain combinations and comes up with the following table of admissible combina-
tions

	Hamburger	Cheeseburger	Fish Filet	
Tea	$-$	$-$	$+$	1
Milk	$-$	$+$	$+$	2
Cola	$+$	$+$	$+$	3
Root Beer	$+$	$+$	$-$	2
	2	3	4	

Here, $+$ denotes an admissible combination. The numbers on the right and on the bottom denote the number of admissible combinations in each row and column, respectively. Note that the sums of these numbers yield the same result $1+2+3+2 = 2+3+3 = 8$, which happens to be the number of all admissible combinations of one main course with one beverage.

All observations made above are formalized and summarized in the following theorem.

Theorem 5.5. *Let X and Y be finite non-empty sets, and let S be a subset of $X \times Y$. Then the following results hold*

1. *The size of S is given by*

$$|S| = \sum_{x \in X} r_x(S) = \sum_{y \in Y} c_y(S),$$

   where $r_x(S)$ and $c_y(S)$ are the row and columns totals defined as

$$r_x(S) = |\{y : (x,y) \in S\}|, \quad c_y(S) = |\{x : (x,y) \in S\}|.$$

2. *The size of $X \times Y$ is given by $|X \times Y| = |X| \cdot |Y|$.*

5.4   Euler's function

In this section we shall prove an important and useful theorem using only the most basic counting principles.

The theorem is concerned with the divisibility properties of integers. Recall that two integers x and y are relatively prime if $\gcd(x,y) = 1$; and for each $n \geq 1$ let $\phi(n)$ denote the number of integers x in the range $1 \leq x \leq n$ such that x and n are relatively prime. We can calculate the first few values of $\phi(n)$ by making a table:

n	relatively prime to n	$\phi(n)$
1	1	1
2	1	1
3	$1, 2$	2
4	$1, 3$	2
5	$1, 2, 3, 4$	4
6	$1, 5$	2
7	$1, 2, 3, 4, 5, 6$	6
8	$1, 3, 5, 7$	4

The function ϕ is called *Euler's function* after Leonhard Euler (1707–1783). When n is prime, say $n = p$, each one of the integers $1, 2, \ldots, p-1$ is relatively prime to p, so we have

$$\phi(p) = p - 1, \quad (p \text{ prime}).$$

In the following, we will show that the sum of the values $\phi(d)$ taken over all divisors d of a given positive integer n is again n. For example when $n = 12$, the divisors d are $1, 2, 3, 4, 6$, and 12, and we find that

$$\phi(1) + \phi(2) + \phi(3) + \phi(4) + \phi(6) + \phi(12) = 1 + 1 + 2 + 2 + 2 + 4 = 12.$$

Theorem 5.6. *For any positive integer n,*

$$\sum_{d|n} \phi(d) = n.$$

Proof. Let S denote the set of pairs of integers (d, f) satisfying

$$d \mid n, \quad 1 \le f \le d, \quad \gcd(f, d) = 1.$$

Using the terminology of Theorem 5.5, we find that $r_d(S) = \phi(d)$ and we thus have

$$|S| = \sum_{d|n} \phi(d).$$

In order to show $|S| = n$ we shall construct a bijection β from S to $[1, n]$. Given a pair $(d, f) \in S$, we define

$$\beta(d, f) = fn/d.$$

Since $d|n$, the value of β is an integer, and since $1 \le f \le d$, it lies in $\mathbb{N}$. To show that β is an injection we remark that

$$\beta(d, f) = \beta(d', f') \quad \Rightarrow \quad fn/d = f'n/d' \quad \Rightarrow \quad fd' = f'd.$$

But f and d are relatively prime, as are f' and d', so we can conclude $d = d'$ and $f = f'$. To show that β is a surjection, suppose we are given $x \in \mathbb{N}$. Let g_x denote the gcd of x and n, and let

$$d_x = n/g_x, \qquad f_x = x/g_x.$$

Since g_x is a divisor of x and n, both d_x and f_x are integers, and since it is the gcd, d_x and f_x are relatively prime. Now

$$\beta(d_x, f_x) = f_x n/d_x = x,$$

and so β is a surjection. Thus β is a bijection and $|S| = n$, as required. $\square$

5.5   Functions, words, and selections

We shall consider functions (not necessarily bijections) defined on a set $[1, m]$, and with values in a given set Y. The values of such a function determine an m-tuple

$$(f(1), f(2), \ldots, f(m))$$

of elements of Y. According to the general definition of a product set, this m-tuple belongs to the set $Y \times Y \times \cdots \times Y$ (m factors), which is also denoted by Y^m. Each element of Y^m is an m-tuple $(y_1, y_2, \ldots, y_m)$ and corresponds to a function f from $[1, m]$ to Y defined by the equation

$$f(1) = y_1, \quad f(2) = y_2, \quad \ldots \quad, f(m) = y_m.$$

These remarks lead us to the conclusion that a function from $[1, m]$ to Y is logically the same thing as an element of the product set Y^m.

There is another way of looking at this relationship, which is very useful in practice. If we think of the members of Y as the letters of an alphabet, then the sequence $f(1), f(2), \ldots, f(m)$ can be regarded as the m letters of a *word*. For example, if Y is the simple alphabet $\{a, b, c, d\}$ the words *cab* and *dad* correspond to the functions f and g defined by

$$f(1) = c, \quad f(2) = a, \quad f(3) = b, \quad g(1) = d, \quad g(2) = a, \quad g(3) = d.$$

The function f, the 3-tuple (c, a, b), and the word *cab* are formally identical, and so we shall define a *word* of *length m* in the *alphabet* to be a function from $[1, m]$ to Y.

Theorem 5.7. *Let X and Y be non-empty finite sets, and let F denote the set of functions from X to Y. If $|X| = m$ and $|Y| = n$, then*

$$|F| = n^m.$$

Proof. Let $X = \{x_1, x_2, \ldots, x_m\}$. Each member f of the set F is a function from X to Y, and is uniquely determined by the m-tuple of its values $(f(1), f(2), \ldots, f(m))$. This tuple belongs to Y^m, and so $|F| = |Y^m| = |Y|^m = n^m$.   □

Equivalently, we may say that the number of words of length m in an alphabet Y of n symbols is n^m. For example, there are $26^3 = 17576$ three-word letters in the usual Roman alphabet. Note that each word represents an ordered selection of letters from the alphabet, with repetitions as many times as required. In general we can say that a function from $[1, m]$ to Y is a mathematical model of an *ordered selection with repetition of m objects from the set Y*. (In the subsequent sections, we will be concerned with selections which may be ordered or unordered, and with or without repetition.)

Example 5.8. The developed framework provides a clean proof that the power set 2^X (the set of all subsets of X) has cardinality 2^n, where n is the cardinality of X. For this purpose, suppose $X = \{x_1, x_2, \ldots, x_n\}$, and let Y be the alphabet $\{0, 1\}$. Any subset S of X corresponds to a word of length n in Y, defined by the function

$$S(i) = \begin{cases} 0 & \text{if } x_i \notin S, \\ 1 & \text{if } x_i \in S. \end{cases}$$

For example, if $n = 7$ and $S = \{x_1, x_2, x_3\}$, the word is 0101100. Consequently, the number of distinct subsets of X is the same as the number of words of length n in the alphabet $\{0, 1\}$, that is 2^n.

5.6   Injections as ordered selections without repetition

In many situations we have to make a selection *without* repetition. For example, if we are selecting a team for soccer, then no available player can be selected more than once. The language of functions provides a ready model for this situation.

We have seen that an ordered selection of m objects from a set Y corresponds to a function f from $[1, m]$ to Y, where $f(1)$ is the first member of Y selected, and so on. When repetition is allowed, it is possible that the same object is selected twice, so that $f(r) = f(s)$ for distinct r and s in $[1, m]$. If this is prohibited then f is an injection, yielding a mathematical model of ordered selection without repetition.

Theorem 5.9. *The number of ordered selections, without repetition, of m objects from a set Y of size n is the same as the number of injections from $[1, m]$ to Y, and it is given by*

$$n(n-1)(n-2)\cdots(n-m+1).$$

Proof. Each injection from $[1, m]$ to Y is uniquely determined by the ordered selection of distinct values $i(1), i(2), \ldots, i(m)$. The first selection can be any one of the n objects in Y. Since repetition is not allowed, the second selection $i(2)$ must be one of the remaining $n - 1$ objects. Similarly, there are $n - 2$ possibilities for $i(3)$, and so on. When we come to select $i(m)$, $m - 1$ objects have already been selected, and so $i(m)$ must be one of the remaining $n - (m - 1)$ objects. Hence, the total number is as stated.   □ For example if we have a pool of 20 soccer players

the number of ways of selecting the 11 players to their individual positions in the game is

$$20 \times 19 \times \cdots 10 = 6704425728000.$$

5.7   Permutations

A permutation of a non-empty finite set X is a bijection from X to X. Frequently we take X from $[1, m] = \{1, 2, \ldots, m\}$. For example, a typical permutation of $[1, 5]$ is the function α defined by

$$\alpha(1) = 2, \quad \alpha(2) = 4, \quad \alpha(3) = 5, \quad \alpha(4) = 1, \quad \alpha(5) = 3. \tag{5.1}$$

A bijection from a finite set to itself is necessarily an injection, and conversely we already know that any such injection is also a bijection. Thus the number of permutations of a set of cardinality n is the same as the number of injections from $[1, n]$ and to itself, and by Theorem 5.9 this number is

$$n(n-1)\cdots 1 = n!.$$

We denote the set of all permutations of $[1, n]$ by S_n. For example, S_3 contains the following $3! = 6$ permutations

$$
\begin{array}{ccc}
1 \ 2 \ 3 & \quad 1 \ 2 \ 3 & \quad 1 \ 2 \ 3 \quad 1 \ 2 \ 3 \quad 1 \ 2 \ 3 \quad 1 \ 2 \ 3 \\
\downarrow \downarrow \downarrow & \downarrow \downarrow \downarrow & \downarrow \downarrow \downarrow \quad \downarrow \downarrow \downarrow \quad \downarrow \downarrow \downarrow \quad \downarrow \downarrow \downarrow \\
1 \ 2 \ 3 & 1 \ 3 \ 2 & 2 \ 1 \ 3 \quad 2 \ 3 \ 1 \quad 3 \ 1 \ 2 \quad 3 \ 2 \ 1.
\end{array}
$$

Permutations are functions; they are combined as in the usual composition of functions. To illustrate this, let α as in (5.1) and consider $\beta \in S_5$ defined by

$$\beta(1) = 3, \quad \beta(2) = 5, \quad \beta(3) = 1, \quad \beta(4) = 4, \quad \beta(5) = 2.$$

The composite function $\beta\alpha$ is the permutation defined by $\beta\alpha(i) = \beta(\alpha(i))$, $1 \leq i \leq 5$, that is

$$\beta\alpha(1) = 5, \quad \beta\alpha(2) = 4, \quad \beta\alpha(3) = 2, \quad \beta\alpha(4) = 3, \quad \beta\alpha(5) = 1.$$

There are four important properties of the composition of permutations, which are summarized in the following theorem.

Theorem 5.10. *The following properties hold in the set S_n of all permutations of $\{1, 2, \ldots, n\}$.*

1. *If π and σ are in S_n, so is $\pi\sigma$.*

2. *For any permutations π, σ, τ in S_n, $(\pi\sigma)\tau = \pi(\sigma\tau)$.*

3. *The identity function denoted by* id *and defined by* $\mathrm{id}(r) = r$ *for all $r \in [1n]$, is a permutation and for any $\sigma \in S_n$ we have* $\mathrm{id}\,\sigma = \sigma\,\mathrm{id} = \sigma$.

4. *For every permutation $\pi \in S_n$ there is an inverse permutation $\pi^{-1} \in S_n$ such that $\pi\pi^{-1} = \pi\pi^{-1} = \mathrm{id}$.*

5.8   Binomial numbers

The mathematical model of an *unordered selection without repetition* is very simple. When we are given a set X with n members and we select r of them, the result is a subset Y of X with $|Y| = r$. It must be stressed that in this model it is the result of the selection (the subset Y) which is important, rather than the process of selection. Also, there is no possibility of repetition, since each member of X is either in Y or not, and no member can be selected twice. Thus the number of unordered selections, without repetition, of r objects from a set X of size n is just the number of subsets having cardinality r. For example, there are six unordered selections, without repetition, of two objects from the set $\{a, b, c, d\}$; they correspond to the subsets

$$\{a, b\}, \quad \{a, c\}, \quad \{a, d\}, \quad \{b, c\}, \quad \{b, d\}, \quad \{c, d\}.$$

In general, the number of subsets Y with $|Y| = r$ from a set X with $|X| = r$ is denoted by the symbol

$$\binom{n}{r}.$$

This is often spoken as n *choose* r, and will be referred to as a *binomial number*. For example, we have just checked that there are six subsets of cardinality 2 from a set of cardinality 4, and so

$$\binom{4}{2} = 6.$$

The calculation of binomial numbers in general depends on the following result.

Lemma 5.11. *If n and r are positive integers satisfying $1 \leq r \leq n$ then*

$$\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}.$$

Proof. Let Z be a set with $n-1$ elements. Choose $a \notin Z$. Then $\binom{n}{r}$ is the number of subsets of cardinality r from $X = Z \cup \{a\}$. Now these subsets can be divided into two disjoint classes:

1. Subsets of X not containing a.

2. Subsets of X containing a.

The subsets of class 1 are just subsets of Z and there are $\binom{n-1}{r}$ of these. Each subset of class 2 consists of a subset of Z having cardinality $r-1$ together with a; there are $\binom{n-1}{r-1}$ of these. Since both classes are disjoint the addition principle implies $\binom{n}{r} = \binom{n-1}{r-1} + \binom{n-1}{r}$. □

Lemma 5.11 provides a recursive method for calculating binomial numbers. If the numbers $\binom{n-1}{k}$ are known for $0 \le k \le n-1$, then the numbers $\binom{n}{k}$ can be computed. This calculation is often displayed in the form of a triangle as follows:

$$
\begin{array}{ccccccccccccc}
 & & & & & & 1 & & & & & & \\
 & & & & & 1 & & 1 & & & & & \\
 & & & & 1 & & 2 & & 1 & & & & \\
 & & & 1 & & 3 & & 3 & & 1 & & & \\
 & & 1 & & 4 & & 6 & & 4 & & 1 & & \\
 & 1 & & 5 & & 10 & & 10 & & 5 & & 1 & \\
1 & & 6 & & 15 & & 20 & & 15 & & 6 & & 1 \\
1 & 7 & & 21 & & 35 & & 35 & & 21 & & 7 & 1.
\end{array}
$$

This is sometimes called Pascal's triangle, after Blaise Pascal (1623–1662). If only one individual coefficient is of interest, it can be more convenient to use the following explicit formula.

Theorem 5.12. *If n and r are positive integers satisfying $1 \le r \le n$, then*

$$
\binom{n}{r} = \frac{n(n-1)\cdots(n-r+1)}{r!} = \frac{n!}{r!(n-r)!}.
$$

Proof. We use the principle of induction. For the induction basis, we remark that the result is true when $n = 1$, since $\binom{1}{1} = 1$ and the formula reduces to $1/1! = 1$.

For the induction hypothesis suppose the result is true when $n = k$. Then, by Lemma 5.11 and the induction hypothesis,

$$
\begin{aligned}
\binom{k+1}{r} &= \binom{k}{r-1} + \binom{k}{r} \\
&= \frac{k(k-1)\cdots(k-r+2)}{(r-1)!} + \frac{k(k-1)\cdots(k-r+1)}{r!} \\
&= \frac{k(k-1)\cdots(k-r+2)}{(r-1)!}\left(1 + \frac{k-r+1}{r}\right) \\
&= \frac{(k+1)k(k-1)\cdots(n-r+2)}{r!}.
\end{aligned}
$$

It follows that the result is true when $n = k+1$, and so, by the principle of induction, it is true for all positive integers n. □

It is simple to show the formulas

$$(a + b)^2 = a^2 + 2ab + b^2, \qquad (a + b)^3 = a^3 + 3a^2b + 3ab^2 + b^3.$$

The general result giving a formula for $(a + b)^n$ is known as the *binomial theorem.*

Theorem 5.13. *Let n be a positive integer. The coefficient of the term $a^{n-r}b^r$ in the expansion of $(a + b)^n$ is the binomial number $\binom{n}{r}$. Explicitly, we have*

$$(a + b)^n = \binom{n}{0}a^n + \binom{n}{1}a^{n-1}b + \binom{n}{2}a^{n-2}b^2 + \cdots + \binom{n}{n}b^n.$$

Proof. Consider what happens when we multiply n factors

$$(a + b)(a + b) \cdots (a + b).$$

A term in the product is obtained by selecting either a or b. The number of terms $a^{n-r}b^r$ is just the number of ways of selecting r b's (and consequently $n - r$ a's), and by definition this is the binomial number $\binom{n}{r}$. □

The coefficients in the expansion may therefore be calculated by using the recursion for the binomial numbers (Pascal's triangle) or by using the formula. For example,

$$(a + b)^6 = \binom{6}{0}a^6 + \binom{6}{1}a^5b + \binom{6}{2}a^4b^2 + \binom{6}{3}a^3b^3 + \binom{6}{4}a^2b^4 + \binom{6}{1}ab^5 + \binom{6}{0}b^6$$

$$= a^6 + 6a^5b + 15a^4b^2 + 20a^3b^3 + 15a^2b^4 + 6ab^5 + b^6.$$

The binomial theorem can also be used to derive identities involving binomial numbers.

Lemma 5.14. *For any positive integer n, we have*

$$\binom{n}{0}^2 + \binom{n}{1}^2 + \binom{n}{2}^2 + \cdots + \binom{n}{n}^2 = \binom{2n}{n}.$$

Proof. We use the identiy

$$(1 + x)^n (1 + x)^n = (1 + x)^{2n}.$$

According to the binomial theorem the left-hand side is the product of two factors both equal to

$$1 + \binom{n}{1}x + \cdots \binom{n}{r}x^r + \cdots + x^n.$$

When then two factors are multiplied, a term in x^n is obtained by taking a term $\binom{n}{r}x^r$ from the first factor and a term $\binom{n}{n-r}x^{n-r}$ from the second factor. Hence the coefficient of x^n in the product is

$$\binom{n}{0}\binom{n}{n} + \binom{n}{1}\binom{n}{n-1} + \binom{n}{2}\binom{n}{n-2} + \cdots + \binom{n}{n}\binom{n}{0}.$$

Since $\binom{n}{n-r} = \binom{n}{r}$, we see that this is the left-hand side of the required identity. But the right-hand side is $\binom{2n}{n}$ which is also the coefficient of x^n in the expansion of $(1 + x)^{2n}$, and so we have the equality stated. □

5.9 Unordered selection with repetition

The binomial number $\binom{n}{r}$ is defined to be the number of unordered selections *without* repetition of r objects from a set of n objects. We now turn to unordered selections *with* repetition. When the numbers involved are small, it is easy to list all the possibilities. For example, there are 15 unordered selections of four objects from the set $\{a, b, c\}$, with repetition allowed, and they are

$$
\begin{array}{ccccc}
aaaa & aaab & aaac & aabb & aabc \\
aacc & abbb & abbc & abcc & accc \\
bbbb & bbbc & bbcc & bccc & cccc.
\end{array}
$$

We will show that there is a general formula for the number of such selections, involving binomial numbers. The proof of this fact involves the representation of such selections as words in the alphabet $\{0, 1\}$; for example, the selection *abcc* will be represented by the word 101011. The zeros are markers which seoarate the different types of objects, and the ones tell us how many of each object there are, according to the scheme

$$
\begin{array}{cccccc}
a & & b & & c & c \\
1 & 0 & 1 & 0 & 1 & 1.
\end{array}
$$

Since there are two markers which can be placed in any of the two positions, the total number of selections in this case is $\binom{6}{2} = 15$, as we found by listing them.

Theorem 5.15. *The number of unordered selections, with repetition, of r objects from a set of n objects is*

$$
\binom{n + r - 1}{r}.
$$

Proof. Since the selections are unordered, we may arrange matters so that, within each selection, all the objects of one type come first, followed by the objects of another type, and so on. When this can be done, we can assign to each selection a word of length $n + (r - 1)$ in the alphabet $\{0, 1\}$, by the method explained above. That is, if there k_i objects of the ith kind ($1 \leq i \leq n$), then the first k_1 letters of the word are 1's, followed by a single 0, followed by k_2 1's, another 0, and so on. The function defined by this rule is a bijection from the set of selections to the set of words of length $n + r - 1$ which contain exactly $n - 1$ zeros. The zeros can occupy any of the $n + r - 1$ positions, so the number of words is

$$
\binom{n + r - 1}{n - 1} = \binom{n + r - 1}{r},
$$

as required. □

The results of this chapter so far are summarized in Table 5.1.

5.10 The sieve principle

The most basic counting principle asserts that $|A \cup B|$ is the sum of $|A|$ and $|B|$ when A and B are disjoint sets. If A and B are not disjoint, the result of adding

	Ordered	Unordered
Without repetition	$\dfrac{n!}{r!}$	$\dbinom{n}{r}$
With repetition	n^r	$\dbinom{n+r-1}{r}$

Table 5.1. *Number of selections – ordered and unordered, with and without repetition – of r objects from a set of n objects.*

$|A|$ and $|B|$ is that all members in $|A \cap B|$ are counted twice. So in order to obtain the correct answer, we must subtract $|A \cap B|$:

$$|A \cup B| = |A| + |B| - |A \cap B|.$$

A similar method can be applied to three sets. When we add $|A|$, $|B|$, and C the members of $A \cap B$, $B \cap C$, and $C \cap A$ are counted twice (if they are not in all three sets). To correct for this we subtract $|A \cap B|$, $|B \cap C|$, and $|C \cap A|$. But now the members of $|A \cap B \cap C|$, originally counted three times, have been deducted three times. So, in order to obtain the correct answer, we must *add* $|A \cap B \cap C|$. Thus,

$$|A \cup B \cup C| = \alpha_1 - \alpha_2 + \alpha_3,$$

where

$$\alpha_1 = |A| + |B| + |C|, \quad \alpha_2 = |A \cap B| + |B \cap C| + |C \cap A|, \quad \alpha_3 = |A \cap B \cap C|.$$

This result is a simple case of what is called the *sieve principle*.

Theorem 5.16. *If $A_1, A_2, \quad , A_n$ are finite sets then*

$$|A_1 \cup A_2 \cup \cdots \cup A_n| = \alpha_1 - \alpha_2 + \alpha_3 - \cdots + (-1)^{n-1}\alpha_n,$$

where α_i is the sum of the cardinalities of all intersections of i sets.

A simple corollary of Theorem 5.16 is often very useful in practice. Suppose that $A_1, A_2, \ldots, A_n$ are subsets of a given set X with $|X| = N$. Then the number of members of X which are *not* in any of these subsets is

$$|X \setminus (A_1 \cup A_2 \cup \cdots \cup A_n)| = |X| - |A_1 \cup A_2 \cup \cdots \cup A_n|$$
$$= N - \alpha_1 + \alpha_2 - \cdots + (-1)^n \alpha_n.$$

Example 5.17. There are 73 students in a musics class. Among them a total of 52 can play the piano, 25 can play the violin, and 20 can play the flute; 17 can play both piano and violin, 12 can play piano and flute and 7 can play violin and flute; but only one can play all three instruments. How many in this class cannot play any of these instruments?

Solution: Let P, V, and F denote the sets of students who can play the piano, violin, and flute, respectively. Using the information given above we have

$$\alpha_1 = |P| + |V| + |F| = 52 + 25 + 20 = 97,$$
$$\alpha_2 = |P \cap V| + |P \cap F| + |V \cap F| = 17 + 12 + 7 = 36$$
$$\alpha_3 = |P \cap V \cap F| = 1.$$

Hence the number of students who do not belong to any of the sets P, V, F is

$$73 - 97 + 36 - 1 = 11.$$

Chapter 6

Recurrence Relations

A *recurrence relation* for a sequence $f_0, f_1, \ldots$ is an equation that relates f_n to certain of its predecessors $f_0, f_1, \ldots, f_{n-1}$. *Initial conditions* for the sequence $a_0, a_1, \ldots$ are explicitly given values for a finite number of terms of the sequence.

recurrence relation
$\hat{=}$ rekurzivna relacija

In the following, we illustrate the concept of recurrence relations by some examples.

Example 6.1. A person invests €1000 at 12 percent interest compounded annually. If A_n represents the amount at the end of n years then A_n consists of the amount A_{n-1} plus the interest. Thus

$$A_n = A_{n-1} + 0.12 \times A_{n-1} = 1.12 \times A_{n-1}, \quad n \geq 1. \tag{6.1}$$

The initial condition is given by $A_0 = 1000$. This allows us to compute the value of A_n for any n. For example,

$$A_3 = 1.12 \times A_2 = 1.12 \times 1.12 \times A_1 = 1.12 \times 1.12 \times 1.12 \times A_0 = (1.12)^3 \times 1000 = 1404.93.$$

Thus, at the end of the third year, the amount is €1404.93. This computation can be carried out for an arbitrary value of n to obtain

$$A_n = (1.12)^n A_0.$$

Note that it is not always possible to find an *explicit formula* for a recurrence relation in such an easy manner.

Example 6.2. The *Fibonacci sequence* is defined by the recurrence relation

$$f_n = f_{n-1} + f_{n-2}, \qquad n \geq 2,$$

and initial conditions $f_0 = 0, f_1 = 1$. The first elements of this sequence are as follows:

n	0	1	2	3	4	5	6	7	8	9	10	11	12	13	$\cdots$
f_n	0	1	1	2	3	5	8	13	21	34	55	89	144	233	$\cdots$

6.1   Linear homogenous recursions

In general, it is very difficult to provide an explicit formula for the elements of a recurrence relation. However, for some classes there is a general way how such formulas can be obtained. In the following, we are concerned with *linear homogeneous recurrence relations with constant coefficients* of order k:

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} + \cdots + c_k a_{n-k}, \quad c_k \neq 0, \quad n \geq k. \tag{6.2}$$

Examples 6.1 and 6.2 both belong to this class with $k = 1$ and $k = 2$, respectively. For $k = 2$, we have the following result.

Theorem 6.3. *Let*

$$a_n = c_1 a_{n-1} + c_2 a_{n-2} \tag{6.3}$$

be a linear homogeneous recurrence relation of order 2. *If two sequences S and T solve (6.3) then $U = bS + dT$ is also a solution of (6.3).*

If r is a root of the so called characteristic equation

$$\lambda^2 - c_1 \lambda - c_2 = 0 \tag{6.4}$$

then the sequence r^n, $n = 0, 1, \ldots$, is a solution of (6.3).

If we additionally have initial conditions $a_0 = C_0, a_1 = C_1$ and r_1, r_2 are roots of (6.4) with $r_1 \neq r_2$, then there exist constants b and d such that

$$a_n = b r_1^n + d r_2^n, \quad n = 0, 1, \ldots.$$

Proof. Since S and T are solutions of (6.3),

$$S_n = c_1 S_{n-1} + c_2 S_{n-2}, \qquad T_n = c_1 T_{n-1} + c_2 T_{n-2}.$$

If we multiply the first equation by b and the second by d and add, we obtain

$$U_n = bS_n + dT_n = c_1(bS_{n-1} + dT_{n-1}) + c_2(bS_{n-2} + dT_{n-2})$$
$$= c_1 U_{n-1} + c_2 U_{n-2}.$$

Therefore, U is a solution of (6.3), which proves the first part.

Since r is a root of (6.4), we have $r^2 = c_1 r + c_2$. Now

$$c_1 r^{n-1} + c_2 r^{n-2} = r^{n-2}(c_1 r + c_2) = r^{n-2} r^2 = r^n,$$

which proves the second part.

If we set $U_n = b r_1^n + d r_2^n$, then we know from the first two parts that U is a solution of (6.3). To meet the initial conditions, we must have

$$U_0 = b + d = C_0, \qquad U_1 = b r_1 + d r_2 = C_1.$$

If we multiply the first equation by r_1 and subtract, we obtain

$$d(r_1 - r_2) = r_1 C_0 - C_1.$$

Since $r_1 - r_2 \neq d$, we can solve for d. Similarly, we can solve for b. With these choices for b and d, we have $U_0 = C_0$ and $U_1 = C_1$, which completes the proof. $\square$

Example 6.4. Considering the Fibonacci sequence, see Example 6.2, we have the characteristic equation

$$\lambda^2 - \lambda - 1 = 0.$$

The roots of this equation are given by

$$r_1 = \frac{1 - \sqrt{5}}{2}, \qquad r_2 = \frac{1 + \sqrt{5}}{2}$$

To account for the initial conditions $a_0 = 0$ and $a_1 = 1$, we have to solve

$$c + d = 0, \qquad cr_1 + dr_2 = 1,$$

which gives $1 = c(r_1 - r_2) = -\sqrt{5}c$ and therefore $c = -1/\sqrt{5}$ and $d = 1/\sqrt{5}$. Hence, the explicit formula for the Fibonacci sequence is given by

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

It is possible to extend the results of Theorem 6.3 to general k. The quadratic characteristic equation (6.3) changes to

$$\lambda^k - c_1 \lambda^{k-1} - c_2 \lambda^{k-2} - \cdots - c_{k-1} \lambda - c_k = 0.$$

This polynomial has k roots $r_1, r_2, \ldots, r_k$ yielding the general solution

$$U_n = b_1 r_1^n + b_2 r_2^n + \cdots + b_k r_k^n.$$

To determine the coefficients $b_1, b_2, \ldots, b_k$, k instead of 2 initial conditions must be provided.

Chapter 7

Algorithms

An *algorithm* is a step-by-step method of solving some problem. Although algorithms can be found in many other fields, we will be only concerned with those that can executed on a computer. Algorithms typically have the following characteristics:

Input The algorithm receives input.

Output The algorithm produces output.

Precision The steps are precisely stated.

Determinism The intermediate results of each step of execution are unique and are determined only by the inputs and the results of the preceding steps.

Finiteness The algorithm *terminates*; that is, it stops after finitely many instructions have been executed.

Correctness The output produced by the algorithm is correct; that is, the algorithm correctly solves the problem.

Generality The algorithm applies to a nontrivial set of inputs.

Example 7.1. Consider the following algorithm that finds the maximum of three numbers a, b, and c:

1. $large = a$.

2. If $b > large$, then $large = b$.

3. If $c > large$, then $large = c$.

Although ordinary language is sometimes adequate to describe an algorithm, most computer scientists prefer *pseudocode* because of its precision, structure, and universality. Pseudocode is so named because it resembles the actual code of computer language such as C++ and Java but does not include language-specific definitions and overhead which would distract from understanding the algorithm. There are many versions of pseudocode, any of which is acceptable as long as its instructions are unambiguous. As our first example, we rewrite Example 7.1 in pseudocode:

Algorithm 7.2 (Maximum of three numbers).

```
Input:    a,b,c
Output:   large (the largest of a, b, and c)
```

```
1  max3(a,b,c) {
2     large = a
3     if (b > large)
4         large = b
5     if (c > large)
6         large = c
7     return large
8  }
```

Our algorithms consists of a title, a brief description of the algorithms, the input to and output from the algorithm, and the functions containing the instructions of the algorithm. To make it convenient to refer to individual lines, we number the lines consecutively. The method of Algorithm 7.2 can be used to find the largest number in a sequence of n numbers.

Algorithm 7.3 (Maximum of a sequence $s_1, s_2, \ldots, s_n$).

```
Input:    s, n
Output:   large (the largest value in the sequence s)
```

```
1  max3(s, n) {
2     large = s[1]
3     for i = 2 to n do
4         if s[i] > large
5             large = s[i]
6     return large
7  }
```

We verify that Algorithm 7.3 is correct by proving that

$$large \text{ is the largest value in the subsequence } s_1, \ldots, s_i$$

is a *loop invariant* using induction on i.

For the induction basis ($i = 1$), we note that just before the for loop begins executing, *large* is set to s_1, so large is surely the largest value in the subsequence s_1.

Assume that *large* is the largest value in the subsequence $s_1, \ldots, s_i$. If $i < n$ is true (so that the loop body executes again), i becomes $i + 1$. Suppose first that $s_{i+1} > large$. It then follows that s_{i+1} is the largest value in the subsequence $s_1, \ldots, s_i, s_{i+1}$. In this case, the algorithm assigns *large* the value s_{i+1}. In this case, the algorithm assigns *large* the value of s_{i+1}. Now *large* is equal to the largest value of the subsequence $s_1, \ldots, s_i, s_{i+1}$. Suppose next that $s_{i+1} \leq large$. It then follows that *large* is already the largest value of the subsequence $s_1, \ldots, s_i, s_{i+1}$. In this case, the algorithm does not change the value of *large*, thus *large* remains the largest value of the subsequence $s_1, \ldots, s_i, s_{i+1}$. We have proved the inductive step, and thus

$$large \text{ is the largest value in the subsequence } s_1, \ldots, s_i$$

is a loop invariant.

7.1 Text searching

One of the most frequent tasks performed on a computer is searching. This area has revived great interest with the advent of the internet (e.g., search engines) and bioinformatics (e.g., finding gene expressions in a DNA sequence).

Suppose that we are given a text t (e.g., a word processor document) and we want to find the first occurence of a pattern p in t or determine that p does not occur in t. We index the characters in t starting at 1. One of the simplest approaches to searching for p is to check whether p occurs at index 1. If so, we stop, having found the first occurence of p in t. If not, we check whether p occurs at index 2 in t. If not, we next check whether p occurs at index 3 in t, and so on.

Algorithm 7.4 (Text search).

```
Input:    p (indexed from 1 to m), m, t (index from 1 to n), n
Output:   i (the index of the first occurence of p in t, if
             it does not occur, i is set to 0)
```

```
1   text_search(p,m,t,n) {
2      for i = 1 to n-m+1 {
3         j = 1
4         while (t[i+j-1] == p[j]) {
5            j = j + 1
6            if (j > m)
7               return i
8         }
9      }
10     return 0
11  }
```

7.2 Efficiency of algorithms

Roughly speaking, the efficiency of an algorithm is determined by the relationship between the *effort* required to solve any specific case of a problem and the *size* of that case. In this section we shall attempt to clarify this vague idea.

In order to measure the effort required we usually count the number of significant operations which the algorithm performs. For example, in Algorithm 7.3 we need $n-1$ comparisons and at most n assignments, so we might say that the effort of the algorithms grows proportionally with n. In contrast, the effort of Algorithm 7.4 heavily depends on the input itself (and not only on its size). If the algorithm finds the pattern p at the first position of t then only m comparisons are necessary. In the worst case, the pattern p is not contained in t and the number of comparisons to discover this fact may grow proportionally with $(n-m+1)m$. (There are much better algorithms for searching.)

The minimum time needed by an algorithm among all possible inputs is called the *best-case time*. The maximum time is called the *worst-case time*. The average time among all inputs (with the possibility of weighting the influence of inputs differently) gives the *average-case time*. Usually, we are less interested in the exact best-case or worst-case times but only in an approximate estimate how the time increases as the size of the input increases. For example, assuming $n \gg m$, we can say that the worst-case time of Algorithm 7.4 grows like nm (or like n if we are only interested in the influence of n). The following definition provides useful tools for finding such approximate estimates.

Definition 7.5. *Let f and g be functions with domain $\{1, 2, 3, \ldots\}$. We write*

$$f(n) = O(g(n))$$

and say $f(n)$ is of order at most $g(n)$ if there exists a positive constant C_1 such that $|f(n)| \leq C_1|g(n)|$ for all but finitely many positive integers n. We write

$$f(n) = \Omega(g(n))$$

and say $f(n)$ is of order at least $g(n)$ if there exists a positive constant C_2 such that $|f(n)| \geq C_2|g(n)|$ for all but finitely many positive integers n. We write

$$f(n) = \Theta(g(n))$$

and say that $f(n)$ is of order $g(n)$ if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

According to this definition, if $f(n) = O(g(n))$, all we can conclude is that, except for a constant factor and a finite number of exceptions, f is bounded above by g, so g grows at least as fast as f. For example, if $f(n) = n$ and $g(n) = 2^n$, then $f(n) = O(g(n))$, but g grows considerably faster than f. The statement $f(n) = O(g(n))$ says nothing about a lower bound for f.

Example 7.6. Since

$$60n^2 + 5n + 1 \leq 60n^2 + 5n^2 + n^2 = 66n^2,$$

for all $n \geq 1$, we can take $C_1 = 66$ in Definition 7.5 to obtain

$$60n^2 + 5n + 1 = O(n^2).$$

Since $60n^2 + 5n + 1 \geq 60n^2$, we also have $60n^2 + 5n + 1 = \Omega(n^2)$. Hence,

$$60n^2 + 5n + 1 = \Theta(n^2).$$

A similar method as in Example 7.6 can be used to show the following result.

Lemma 7.7. *If f is a polynomial of degree k,*

$$f(n) = a_k n^k + a_{k-1} n^{k-1} + \cdots + a_1 n + a_0$$

with $a_k \neq 0$, then $f(n) = \Theta(n^k)$.

In the analysis of algorithms one is often confronted with of integers. The following result considerably simplifies the task of obtaining simplified expressions.

Theorem 7.8. *Let $f(k) \geq 0$ for all $k \geq 1$. Then*

$$f(1) + f(2) + \cdots + f(n) = \Theta(g(n)),$$

where g is a primitive of f.

primitive
$\hat{=}$ osnovna funkcija

For example, we have

$$1 + 2 + \cdots + n = \Theta\left(\int k \ dk\right) = \Theta(n^2),$$

$$1^2 + 2^2 + \cdots + n^2 = O\left(\int k^2 \ dk\right) = \Theta(n^3),$$

$$\ln n! = \ln 1 + \ln 2 + \cdots + \ln n = O\left(\int \ln k \ dk\right) = \Theta(n \ln n).$$

Many algorithms have logarithmic complexity. Because of $\log_b n = \log_b a \cdot \log_a n$, we have $\log_a n = \Theta(\log_b n)$, i.e., we do not need to worry about the base of logarithms in estimating the complexity.

Example 7.9. As a simple example, let us consider the following algorithm.

```
for i = 1 to n
   for j = 1 to i
      x = x + 1
```

The total number of times the inner loop is executed is

$$1 + 2 + 3 + \cdots + n = \Theta(n^2).$$

Hence, the worst- and best-case times of this algorithm are of order n^2.

Index