

Universitext

Guerino Mazzola
G rard Milmeister
Jody Weissmann

Comprehensive Mathematics for Computer Scientists 1

Sets and Numbers, Graphs and Algebra,
Logic and Machines, Linear Geometry

Second Edition

 Springer

Universitext

Springer

Berlin

Heidelberg

New York

Hong Kong

London

Milan

Paris

Tokyo

Guerino Mazzola • Gérard Milmeister •
Jody Weissmann

Comprehensive Mathematics for Computer Scientists 1

Sets and Numbers, Graphs and Algebra,
Logic and Machines, Linear Geometry
(Second Edition)

With 118 Figures

 Springer

Guerino Mazzola
Gérard Milmeister
Jody Weissmann

Department of Informatics
University Zurich
Winterthurerstr. 190
8057 Zurich, Switzerland

The text has been created using $\text{\LaTeX 2\mathcal{E}}$. The graphics were drawn using Dia, an open-source diagramming software. The main text has been set in the Y&Y Lucida Bright type family, the headings in Bitstream Zapf Humanist 601.

Library of Congress Control Number: 2006929906

Mathematics Subject Classification (2000): 00A06

ISBN (First Edition) 3-540-20835-6

ISBN 3-540-36873-6 Springer-Verlag Berlin Heidelberg New York

This work is subject to copyright. All rights are reserved, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilm or in any other way, and storage in data banks. Duplication of this publication or parts thereof is permitted only under the provisions of the German Copyright Law of September 9, 1965, in its current version, and permission for use must always be obtained from Springer-Verlag. Violations are liable for prosecution under the German Copyright Law.

Springer-Verlag is a part of Springer Science+Business Media

springer.com

© Springer-Verlag Berlin Heidelberg 2006

The use of general descriptive names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Cover Design: Erich Kirchner, Heidelberg

Typesetting: Camera ready by the authors

Printed on acid-free paper SPIN: 11803911 40/3142/SPi - 543210

Preface to the Second Edition

A second edition of a book is a success and an obligation at the same time. We are satisfied that a number of university courses have been organized on the basis of the first volume of *Comprehensive Mathematics for Computer Scientists*. The instructors recognized that the self-contained presentation of a broad spectrum of mathematical core topics is a firm point of departure for a sustainable formal education in computer science.

We feel obliged to meet the valuable feedback of the responsible instructors of such courses, in particular of Joel Young (Computer Science Department, Brown University) who has provided us with numerous remarks on misprints, errors, or obscurities. We would like to express our gratitude for these collaborative contributions. We have reread the entire text and not only eliminated identified errors, but also given some additional examples and explications to statements and proofs which were exposed in a too shorthand style.

A second edition of the second volume will be published as soon as the errata, the suggestions for improvements, and the publisher's strategy are in harmony.

Zurich,
June 2006

Guerino Mazzola
G rard Milmeister
Jody Weissmann

Preface

The need for better formal competence as it is generated by a sound mathematical education has been confirmed by recent investigations by professional associations, but also by IT opinion leaders such as Niklaus Wirth or Peter Wegner. It is rightly argued that programming skills are a necessary but by far not sufficient qualification for designing and controlling the conceptual architecture of valid software. Often, the deficiency in formal competence is compensated by trial and error programming. This strategy may lead to uncontrolled code which neither formally nor effectively meets the given objectives. According to the global view such bad engineering practice leads to massive quality breakdowns with corresponding economical consequences.

Improved formal competence is also urged by the object-oriented paradigm which progressively requires a programming style and a design strategy of high abstraction level in conceptual engineering. In this context, the arsenal of formal tools must apply to completely different problem situations. Moreover, the dynamics and life cycle of hard- and software projects enforce high flexibility of theorists and executives on all levels of the computer science academia and IT industry. This flexibility can only be guaranteed by a propaedeutical training in a number of typical styles of mathematical argumentation.

With this in mind, writing an introductory book on mathematics for computer scientists is a somewhat delicate task. On the one hand, computer science delves into the most basic machinery of human thought, such as it is traced in the theory of Turing machines, rewriting systems and grammars, languages, and formal logic. On the other hand, numerous applications of core mathematics, such as the theory of Galois fields (e.g., for coding theory), linear geometry (e.g., for computer graphics), or differential equations (e.g., for simulation of dynamic systems) arise in any

relevant topic of computational science. In view of this wide field of mathematical subjects the common practice is to focus one's attention on a particular bundle of issues and to presuppose acquaintance with the background theory, or else to give a short summary thereof without any further details.

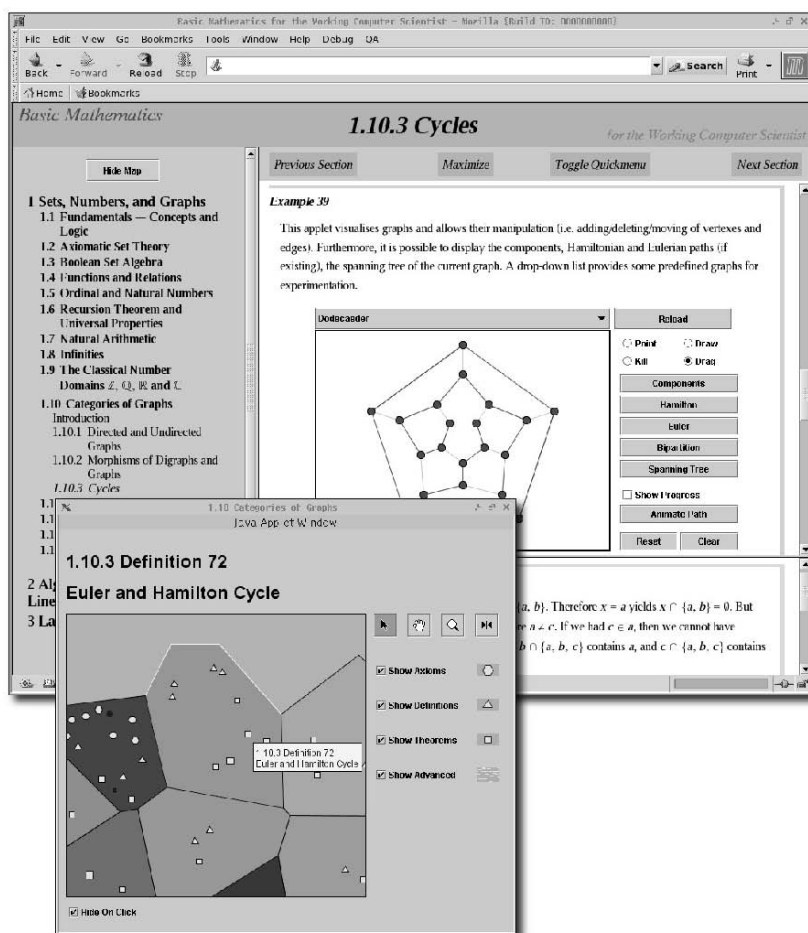
In this book, we have chosen a different presentation. The idea was to set forth and prove the entire core theory, from axiomatic set theory to numbers, graphs, algebraic and logical structures, linear geometry—in the present first volume, and then, in the second volume, topology and calculus, differential equations, and more specialized and current subjects such as neural networks, fractals, numerics, Fourier theory, wavelets, probability and statistics, manifolds, and categories.

There is a price to pay for this comprehensive journey through the overwhelmingly extended landscape of mathematics: We decided to omit any not absolutely necessary ramification in mathematical theorization. Rather it was essential to keep the global development in mind and to avoid an unnecessarily broad approach. We have therefore limited explicit proofs to a length which is reasonable for the non-mathematician. In the case of lengthy and more involved proofs, we refer to further readings. For a more profound reading we included a list of references to original publications. After all, the student should realize as early as possible in his or her career that science is vitally built upon a network of links to further knowledge resources.

We have, however, chosen a modern presentation: We introduce the language of commutative diagrams, universal properties and intuitionistic logic as advanced by contemporary theoretical computer science in its topos-theoretic aspect. This presentation serves the economy and elegance of abstraction so urgently requested by opinion leaders in computer science. It also shows some original restatements of well-known facts, for example in the theory of graphs or automata. In addition, our presentation offers a glimpse of the unity of science: Machines, formal concept architectures, and mathematical structures are intimately related with each other.

Beyond a traditional “standalone” textbook, this text is part of a larger formal training project hosted by the Department of Informatics at the University of Zurich. The online counterpart of the text can be found on <http://math.ifi.unizh.ch>. It offers access to this material and includes interactive tools for examples and exercises implemented by Java

applets and script-based dynamic HTML. Moreover, the online presentation allows switching between textual navigation via classical links and a quasi-geographical navigation on a “landscape of knowledge”. In the latter, parts, chapters, axioms, definitions, and propositions are visualized by continents, nations, cities, and paths. This surface structure describes the top layer of a three-fold stratification (see the following screenshot of some windows of the online version).



On top are the facts, below, in the middle layer, the user will find the proofs, and in the third, deepest stratum, one may access the advanced topics, such as floating point arithmetic, or coding theory. The online counterpart of the book includes two important addenda: First, a list of

errata can be checked out. The reader is invited to submit any error encountered while reading the book or the online presentation. Second, the subject spectrum, be it in theory, examples, or exercises, is constantly updated and completed and, if appropriate, extended. It is therefore recommended and beneficial to work with both, the book and its online counterpart.

This book is a result of an educational project of the E-Learning Center of the University of Zurich. Its production was supported by the Department of Informatics, whose infrastructure we were allowed to use. We would like to express our gratitude to these supporters and hope that the result will yield a mutual profit: for the students in getting a high quality training, and for the authors for being given the chance to study and develop a core topic of formal education in computer science. We also deeply appreciate the cooperation with the Springer Publishers, especially with Clemens Heine, who managed the book's production in a completely efficient and unbureaucratic way.

Zurich,
February 2004

Guerino Mazzola
Gérard Milmeister
Jody Weissmann

Contents

I	Sets, Numbers, and Graphs	1
1	Fundamentals—Concepts and Logic	3
1.1	Propositional Logic	4
1.2	Architecture of Concepts	8
2	Axiomatic Set Theory	15
2.1	The Axioms	17
2.2	Basic Concepts and Results	20
3	Boolean Set Algebra	25
3.1	The Boolean Algebra of Subsets.....	25
4	Functions and Relations	29
4.1	Graphs and Functions	29
4.2	Relations	41
5	Ordinal and Natural Numbers	45
5.1	Ordinal Numbers	45
5.2	Natural Numbers	50
6	Recursion Theorem and Universal Properties	55
6.1	Recursion Theorem.....	56
6.2	Universal Properties	58
6.3	Universal Properties in Relational Database Theory	66
7	Natural Arithmetic	73
7.1	Natural Operations	73
7.2	Euclid and the Normal Forms	76
8	Infinities	79
8.1	The Diagonalization Procedure	79

9	The Classical Number Domains \mathbb{Z}, \mathbb{Q}, \mathbb{R}, and \mathbb{C}	81
9.1	Integers \mathbb{Z}	82
9.2	Rationals \mathbb{Q}	87
9.3	Real Numbers \mathbb{R}	90
9.4	Complex Numbers \mathbb{C}	102
10	Categories of Graphs	107
10.1	Directed and Undirected Graphs	108
10.2	Morphisms of Digraphs and Graphs	114
10.3	Cycles	125
11	Construction of Graphs	129
12	Some Special Graphs	137
12.1	n -ary Trees	137
12.2	Moore Graphs	139
13	Planarity	143
13.1	Euler's Formula for Polyhedra	143
13.2	Kuratowski's Planarity Theorem	147
14	First Advanced Topic	149
14.1	Floating Point Arithmetic	149
14.2	Example for an Addition	154
II	Algebra, Formal Logic, and Linear Geometry	157
15	Monoids, Groups, Rings, and Fields	159
15.1	Monoids	159
15.2	Groups	163
15.3	Rings	171
15.4	Fields	177
16	Primes	181
16.1	Prime Factorization	181
16.2	Roots of Polynomials and Interpolation	186
17	Formal Propositional Logic	191
17.1	Syntactics: The Language of Formal Propositional Logic ..	193
17.2	Semantics: Logical Algebras	196
17.3	Signification: Valuations	200
17.4	Axiomatics	203

Contents	XIII
18 Formal Predicate Logic	209
18.1 Syntactics: First-order Language	211
18.2 Semantics: Σ -Structures	217
18.3 Signification: Models	218
19 Languages, Grammars, and Automata	223
19.1 Languages	224
19.2 Grammars	229
19.3 Automata and Acceptors	243
20 Categories of Matrixes	261
20.1 What Matrixes Are	262
20.2 Standard Operations on Matrixes	265
20.3 Square Matrixes and their Determinant	271
21 Modules and Vector Spaces	279
22 Linear Dependence, Bases, and Dimension	287
22.1 Bases in Vector Spaces	288
22.2 Equations	295
22.3 Affine Homomorphisms	296
23 Algorithms in Linear Algebra	303
23.1 Gauss Elimination	303
23.2 The LUP Decomposition	307
24 Linear Geometry	311
24.1 Euclidean Vector Spaces	311
24.2 Trigonometric Functions from Two-Dimensional Rotations	320
24.3 Gram's Determinant and the Schwarz Inequality	323
25 Eigenvalues, the Vector Product, and Quaternions	327
25.1 Eigenvalues and Rotations	327
25.2 The Vector Product	331
25.3 Quaternions	333
26 Second Advanced Topic	343
26.1 Galois Fields	343
26.2 The Reed-Solomon (RS) Error Correction Code	349
26.3 The Rivest-Shamir-Adleman (RSA) Encryption Algorithm .	353
A Further Reading	357

XIV	Contents
B Bibliography	359
Index	363

Volume II

III Topology and Calculus

Limits and Topology, Differentiability, Inverse and Implicit Functions, Integration, Fubini and Changing Variables, Vector Fields, Fixpoints, Main Theorem of ODEs

IV Selected Higher Subjects

Numerics, Probability and Statistics, Splines, Fourier, Wavelets, Fractals, Neural Nets, Global Coordinates and Manifolds, Categories, Lambda Calculus

PART I

Sets, Numbers, and Graphs

CHAPTER 1

Fundamentals— Concepts and Logic

Die Welt ist alles, was der Fall ist.
Ludwig Wittgenstein

“The world is everything that is the case” — this is the first tractatus in Ludwig Wittgenstein’s *Tractatus Logico-Philosophicus*.

In science, we want to know what is true, i.e., what is the case, and what is not. Propositions are the theorems of our language, they are to describe or denote what is the case. If they do, they are called true, otherwise they are called false. This sounds a bit clumsy, but actually it is pretty much what our common sense tells us about true and false statements. Some examples may help to clarify things:

“This sentence contains five words”

This proposition describes something which is the case, therefore it is a *true* statement.

“Every human being has three heads”

Since I myself have only one head (and I assume this is the case with you as well), this proposition describes a situation which is not the case, therefore it is *false*.

In order to handle propositions precisely, science makes use of two fundamental tools of thought:

- Propositional Logic
- Architecture of Concepts

These tools aid a scientist to construct accurate concepts and to formulate new true propositions from old ones.

The following sections may appear quite diffuse to the reader; some things will seem to be obviously true, other things will perhaps not make much sense to start with. The problem is that we have to use our natural language for the task of defining things in a precise way. It is only by using these tools that we can define in a clear way what a set is, what numbers are, etc.

1.1 Propositional Logic

Propositional logic helps us to navigate in a world painted in black and white, a world in which there is only truth or falsehood, but nothing in between. It is a boiled down version of common sense reasoning. It is the essence of Sherlock Holmes' way of deducing that Professor Moriarty was the mastermind behind a criminal organization ("Elementary, my dear Watson"). Propositional logic builds on the following two propositions, which are declared to be true as basic principles (and they seem to make sense...):

Principle of contradiction (principium contradictionis)

A proposition is never true and false at the same time.

Principle of the excluded third (tertium non datur)

A proposition is either true or false—there is no third possibility.

In other words, in propositional logic we work with statements that are either true (T) or false (F), no more and no less. Such a logic is also known as *absolute* logic.

In propositional logic there are also some operations which are used to create new propositions from old ones:

Logical Negation

The negation of a true proposition is a false proposition, the negation of a false proposition is a true proposition. This operation is also called 'NOT'.

Logical Conjunction

The conjunction of two propositions is true if and only if both propositions are true. In all other cases it is false. This operation is also called 'AND'.

Logical Disjunction

The disjunction of two propositions is true if at least one of the propositions is true. If both propositions are false, the disjunction is false, too. This operation is also known as 'OR'.

Logical Implication

A proposition \mathcal{P}_1 implies another proposition \mathcal{P}_2 if \mathcal{P}_2 is true whenever \mathcal{P}_1 is true. This operation is also known as 'IMPLIES'.

Often one uses so-called truth tables to show the workings of these operations. In these tables, A stands for the possible truth values of a proposition \mathcal{A} , and B stands for the possible truth values of a proposition \mathcal{B} . The rows labeled " A AND B " and " A OR B " contain the truth value of the conjunction and disjunction of the propositions.

A	NOT A	A	B	A AND B	A	B	A OR B	A	B	A IMPLIES B
F	T	F	F	F	F	F	F	F	F	T
T	F	F	T	F	F	T	T	F	T	T
		T	F	F	T	F	T	T	F	F
		T	T	T	T	T	T	T	T	T

Let us look at a few examples.

1. Let proposition \mathcal{A} be "The ball is red". The negation of \mathcal{A} , (i.e., NOT \mathcal{A}) is "It is not the case that the ball is red". So, if the ball is actually green, that means that \mathcal{A} is false and that NOT \mathcal{A} is true.
2. Let proposition \mathcal{A} be "All balls are round" and proposition \mathcal{B} "All balls are green". Then the conjunction \mathcal{A} AND \mathcal{B} of \mathcal{A} and \mathcal{B} is false, because there are balls that are not green.
3. Using the same propositions, the disjunction of \mathcal{A} and \mathcal{B} , \mathcal{A} OR \mathcal{B} is true.
4. For any proposition \mathcal{A} , \mathcal{A} AND (NOT \mathcal{A}) is always false (principle of contradiction).
5. For any proposition \mathcal{A} , \mathcal{A} OR (NOT \mathcal{A}) is always true (principle of excluded third).

In practice it is cumbersome to say: “The proposition *It rains* is true”. Instead, one just says: “It rains.” Also, since the formal combination of propositions by the above operators is often an overhead, we mostly use the common language denotation, such as: “ $2 = 3$ is false” instead of “NOT ($2 = 3$)” or: “It’s raining or/and I am tired.” instead of “It’s raining OR/AND I am tired”, or: “If it’s raining, then I am tired” instead of “It’s raining IMPLIES I am tired.” Moreover, we use the mathematical abbreviation “ A iff B ” for “(A IMPLIES B) AND (B IMPLIES A)”. Observe that brackets (...) are used in order to make the grouping of symbols clear if necessary.

The operations NOT, AND, OR, and IMPLIES have a number of properties which are very useful for simplifying complex combinations of these operations. Let P , Q , and R be truth values. Then the following properties hold:

Commutativity of AND

P AND Q is the same as Q AND P .

Commutativity of OR

P OR Q is the same as Q OR P .

Associativity of AND

$(P$ AND $Q)$ AND R is the same as P AND $(Q$ AND $R)$.

One usually omits the parentheses and writes P AND Q AND R .

Associativity of OR

$(P$ OR $Q)$ OR R is the same as P OR $(Q$ OR $R)$.

One usually omits the parentheses and writes P OR Q OR R .

De Morgan’s Law for AND

NOT $(P$ AND $Q)$ is the same as (NOT P) OR (NOT Q).

De Morgan’s Law for OR

NOT $(P$ OR $Q)$ is the same as (NOT P) AND (NOT Q).

Distributivity of AND over OR

P AND $(Q$ OR $R)$ is the same as $(P$ AND $Q)$ OR $(P$ AND $R)$.

Distributivity of OR over AND

P OR $(Q$ AND $R)$ is the same as $(P$ OR $Q)$ AND $(P$ OR $R)$.

Contraposition

P IMPLIES Q is the same as (NOT Q) IMPLIES (NOT P).

Idempotency of AND

P is the same as $P \text{ AND } P$.

Idempotency of OR

P is the same as $P \text{ OR } P$.

The validity of these properties can be verified by using the truth tables. We will show how this is done for the example of “Distributivity of OR over AND”.

We want to show that $P \text{ OR } (Q \text{ AND } R)$ is the same as $(P \text{ OR } Q) \text{ AND } (P \text{ OR } R)$, for every choice of P , Q , and R . To do so we first write a big truth table which shows the values for P , Q , and R as well as $Q \text{ AND } R$ and $P \text{ OR } (Q \text{ AND } R)$:

P	Q	R	$Q \text{ AND } R$	$P \text{ OR } (Q \text{ AND } R)$
F	F	F	F	F
F	F	T	F	F
F	T	F	F	F
F	T	T	T	T
T	F	F	F	T
T	F	T	F	T
T	T	F	F	T
T	T	T	T	T

Then we write a truth table which shows the values for P , Q , and R as well as $P \text{ OR } Q$, $P \text{ OR } R$, and $(P \text{ OR } Q) \text{ AND } (P \text{ OR } R)$:

P	Q	R	$P \text{ OR } Q$	$P \text{ OR } R$	$(P \text{ OR } Q) \text{ AND } (P \text{ OR } R)$
F	F	F	F	F	F
F	F	T	F	T	F
F	T	F	T	F	F
F	T	T	T	T	T
T	F	F	T	T	T
T	F	T	T	T	T
T	T	F	T	T	T
T	T	T	T	T	T

The truth values of the two expressions we are interested in (shown in bold face) are indeed equal for every possible combination of P , Q , and R .

The verification of the remaining properties is left as an exercise for the reader.

1.2 Architecture of Concepts

In order to formulate unambiguous propositions, we need a way to describe the concepts we want to make statements about. An architecture of concepts deals with the question: “How does one build a concept?” Such an architecture defines ways to build new concepts from already existing concepts. Of course one has to deal with the question where to anchor the architecture, in other words, what are the basic concepts and how are they introduced. This can be achieved in two different ways. The first uses the classical approach of undefined primary concepts, the second avoids primary concepts by circular construction. This second approach is the one that is used for building the architecture of set theory in this book.

1. A concept has a *name*, for example, “Number” or “Set” are names of certain concepts.
2. Concepts have *components*, which are concepts, too. These components are used to construct a concept.
3. There are three fundamental principles of how to combine such components:
 - *Conceptual Selection*: requires one component
 - *Conceptual Conjunction*: requires one or two components
 - *Conceptual Disjunction*: requires two components
4. Concepts have *instances* (examples), which have the following properties:
 - Instances have a name
 - Instances have a value

The construction principles mentioned above are best described using instances:

The value of an instance of a concept constructed as a selection is the collection of the references to selected instances of the component.

The value of an instance of a concept constructed as a conjunction is the sequence of the references to the instances of each component.

The value of an instance of a concept constructed as a disjunction is a reference to an instance of one of the components.

Perhaps some examples will clarify those three construction principles.

A selection is really a selection in its common sense meaning: You point at a thing and say, "I select this", you point at another thing and say "I select this, too" and so on.

One example for a conjunction are persons' names which (at least in the western world) always consists of a first name and a family name. Another example is given by the points in the plane: every point is defined by an x - and a y -coordinate.

A disjunction is a simple kind of "addition": An instance of the disjunction of all fruits and all animals is either a fruit or an animal.

Notation

If we want to write about concepts and instances, we need an expressive and precise notation.

Concept: `ConceptName.ConstructionPrinciple(Component(s))`

This means that we first write the concept's name followed by a dot. After the dot we write the construction principle (Selection, Conjunction, or Disjunction) used to construct the concept. Finally we add the component or components which were used for the construction enclosed in brackets.

Instance: `InstanceName@ConceptName(Value)`

In order to write down an instance, we write the instance's name followed by an '@'. After this, the name of the concept is added, followed by a value enclosed in brackets. In the case of a disjunction, a semicolon directly following the value denotes the first component, and a semicolon preceding the value denotes the second component.

Very often it is not possible to write down the entire information needed to define a concept. In most cases one cannot write down components and values explicitly. Therefore, instead of writing the concept or instance, one only writes its name. Of course, this presupposes that these

objects can be identified by a name, i.e., there are enough names to distinguish these objects from one another. Thus if two concepts have identical names, then they have identical construction principles and identical components. The same holds for instances: identical names mean identical concepts and identical values.

By identifying names with objects one can say “let X be a concept” or “let z be an instance”, meaning that X and z are the names of such objects that refer to those objects in a unique way.

Here are some simple examples for concepts and instances:

CitrusFruits.Disjunction(Lemons, Oranges)

The concept **CitrusFruit** consists of the concepts **Lemons** and **Oranges**.

MyLemon@Citrusfruits(Lemon2;)

MyLemon is an instance of the concept **CitrusFruit**, and has the value **Lemon2** (which is itself an instance of the concept **Lemons**).

YourOrange@Citrusfruits(; Orange7)

YourOrange is an instance of the concept **CitrusFruits**, and has the value **Orange7** (which is itself an instance of the concept **Oranges**).

CompleteNames.Conjunction(FirstNames, FamilyNames)

The concept **CompleteNames** is a conjunction of the concept **FirstNames** and **FamilyNames**.

MyName@CompleteNames(John; Doe)

MyName is an instance of the concept **CompleteNames**, and has the value **John; Doe**.

SmallAnimals.Selection(Animals)

The concept **SmallAnimals** is a selection of the concept **Animals**.

SomeInsects@SmallAnimals(Ant, Ladybug, Grasshopper)

SomeInsects is an instance of the concept **SmallAnimals** and has the value **Ant, Ladybug, Grasshopper**.

Mathematics

The environment in which this large variety of concepts and propositions is handled is Mathematics.

With the aid of set theory Mathematics is made conceptually precise and becomes the foundation for all formal tools. Especially formal logic is only possible on this foundation.

In Mathematics the existence of a concept means that it is conceivable without any contradiction. For instance, a set exists if it is conceivable without contradiction. Most of the useful sets exist (i.e., are conceivable without contradiction), but one may conceive sets which don't exist. An example of such a set is the subject of the famous paradox advanced by Bertrand Russell: the set containing all sets that do not contain themselves—does this set contain itself, or not?

Set theory must be constructed successively to form an edifice of concepts which is conceivable without any contradictions.

In this section we will first show how one defines natural numbers using concepts and instances. After that, we go on to create set theory from “nothing”.

Naive Natural Numbers

The natural numbers can be conceptualized as follows:

Number.Disjunction(Number, Terminator)
Terminator.Conjunction(Terminator)

The concept **Number** is defined as a disjunction of itself with a concept **Terminator**, the concept **Terminator** is defined as a conjunction of itself (and nothing else). The basic idea is to define a specific natural number as the successor of another natural number. This works out for 34, which is the successor of 33, and also for 786657, which is the successor of 786656. But what about 0? The number zero is not the successor of any other natural number. So in a way we use the **Terminator** concept as a starting point, and successively define each number (apart from 0) as the successor of the preceding number. The fact that the concept **Terminator** is defined as a conjunction of itself simply means: “**Terminator** is a thing”. This is a first example of a circular construction used as an artifice to ground the definition of natural numbers.

Now let us look at some instances of these concepts:

t@Terminator(t)

In natural language: the value of the instance **t** of **Terminator** is itself.
This is a second application of circularity.

0@Number(; t)

The instance of **Number** which we call **0** has the value **t**;

1@Number(0;)

The instance of **Number** which we call **1** has the value **0**;

2@Number(1;)

The instance of **Number** which we call **2** has the value **1**;

If we expand the values of the numbers which are neither **0** nor **t**, we get

- the value of **1** is **0**;
- the value of **2** is **1**; which is **0**;
- the value of **3** is **2**; which is **0**;;
- etc.

This could be interpreted by letting the semicolon stand for the operation “successor of”, thus **3** is the successor of the successor of the successor of **0**.

Pure Sets

The pure sets are defined in the following circular way:

Set.Selection(Set)

Here, we say that a set is a selection of sets. Since one is allowed to select nothing in a conceptual selection, there is a starting point for this circularity. Let us look at some instances again:

∅@Set()

Here we select nothing from the concept **Set**. We therefore end up with the empty set.

1@Set(∅)

Since \emptyset is a set we can select it from the concept **Set**. The value of **1** is a set consisting of one set.

2@Set(\emptyset , 1)

Here we select the two sets we have previously defined. The value of 2 is a set consisting of two sets.

Elements of the Mathematical Prose

In Mathematics, there is a “catechism” of true statements, which are named after their relevance in the development of the theory.

An *axiom* is a statement which is not proved to be true, but supposed to be so. In a second understanding, a theory is called *axiomatic* if its concepts are abstractions from examples which are put into generic definitions in order to develop a theory from a given type of concepts.

A *definition* is used for building—and mostly also for introducing a symbolic notation for—a concept which is described using already defined concepts and building rules.

A *lemma* is an auxiliary statement which is proved as a truth preliminary to some more important subsequent true statement. A *corollary* is a true statement which follows without significant effort from an already proved statement. Ideally, a corollary should be a straightforward consequence of a more difficult statement. A *sorite* is a true statement, which follows without significant effort from a given definition. A *proposition* is an important true statement, but less important than a *theorem*, which is the top spot in this nomenclature.

A mathematical *proof* is the logical deduction of a true statement B from another true statement C . Logical deduction means that the theorems of absolute logic are applied to establish the truth of B , knowing the truth of C . The most frequent procedure is to use as the true statement C the truth of A and the truth of A IMPLIES B , in short, the truth of A AND (A IMPLIES B). Then B is true since the truth of the implication with the true antecedent A can only hold with B also being true. This is the so-called *modus ponens*. This scheme is also applied for *indirect proofs*, i.e., we use the true fact (NOT B) IMPLIES (NOT A), which is equivalent to A IMPLIES B (contraposition, see also properties on page 6). Now, by the principle of the excluded third and the principle of contradiction, either B or NOT B will be true, but not both at the same time. Then the truth of NOT B enforces the truth of NOT A . But by the principle of contradiction, A and NOT A cannot be both true, and since A is true, NOT B cannot hold, and therefore, by the principles of the excluded

third and of contradiction, B is true. There are also more technical proof techniques, such as the proof by induction, but logically speaking, they are all special cases of the general scheme just described.

In this book, the end of a proof is marked by the symbol \square .

Axiomatic Set Theory

Axiomatic set theory is the theory of pure sets, i.e., it is built on a set concept which refers to nothing else but itself. One then states a number of axioms, i.e., propositions which are supposed to be true for sets (i.e., instances of the set concept). On this axiomatic basis, the whole mathematical concept framework is built, leading from elementary numbers to the most complex structures, such as differential equations or manifolds.

The concept of “pure sets” was already given in our introduction 1.2:

Set.Selection(Set)

and the instance scheme

SetName@Set(Value)

where the value is described such that each reference is uniquely identified.

Notation 1 *If a set X has a set x amongst its values, one writes “ $x \in X$ ” and one says “ x is an element of X ”. If it is not the case that “ $x \in X$ ”, one writes “ $x \notin X$ ”.*

If it is possible to write down the elements of a set explicitly, one uses curly brackets: Instead of “ $A@Set(a, b, c, \dots z)$ ” one writes¹ “ $A = \{a, b, c, \dots z\}$ ”. For example, the empty set is $\emptyset = \{\}$.

If there is an attribute \mathcal{F} which characterizes the elements of a set A one writes “ $A = \{x \mid \mathcal{F}(x)\}$ ”, where “ $\mathcal{F}(x)$ ” stands for “ x has attribute \mathcal{F} ”.

¹ In the context of set theory, this traditional notation is preferred, but can always be reduced to the more generic @-notation.

Definition 1 (Subsets and equality of sets) Let A and B be sets. We say that A is a subset of B , and we write " $A \subset B$ " if for every set x the proposition ($x \in A$ IMPLIES $x \in B$) is true.

One says that A equals B and writes " $A = B$ " if the proposition ($A \subset B$ AND $B \subset A$) is true. If " $A \subset B$ " is false, one writes " $A \not\subset B$ ". If " $A = B$ " is false, one writes " $A \neq B$ ". If $A \subset B$, but $A \neq B$, one also writes " $A \subsetneq B$ ".

A subset $A \subset B$ is said to be the smallest subset of B with respect to a property P , if it has this property and is a subset of every subset $X \subset B$ having this property P .

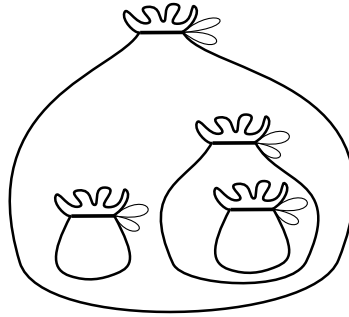


Fig. 2.1. In order to give the reader a better intuition about sets, we visualize them as bags, while their elements are shown as smaller bags—or as symbols for such elements—contained in larger ones. For example, \emptyset is drawn as the empty bag. The set in this figure is $\{\emptyset, \{\emptyset\}\}$.

Example 1 Let $A = \{a, b, \{c, d\}\}$. Then $a \in A$ and $\{c, d\} \in A$, but $c \notin A$; $\{a, b\} \subset A$ and $\{\{c, d\}\} \subset A$, but $\{c, d\} \not\subset A$.

In these examples, sets are specified by enumerating their elements. For an example of the use of an attribute for characterizing the members of a set consider the propositional attribute $\mathcal{A}(x)$ defined by " x is a member of the Apollo 11 crew". Then $\{x \mid \mathcal{A}(x)\} = \{\text{Armstrong, Aldrin, Collins}\}$. The number of objects specified by an attribute need not be limited. Thus the set $\{x \mid \mathcal{N}(x)\}$, where $\mathcal{N}(x)$ is defined by " x is a number", cannot be written down by means of enumeration alone.

For any set X we have $\{\} \subset X$. To prove this, one has to show that the proposition ($x \in \{\}$ IMPLIES $x \in X$) is true. Since the empty set does not have any elements, the left hand side of the implication is false. A

quick glance at the truth tables on page 5 shows that in this case the implication as a whole is true, irrespective of the truth value of the right hand side. This reasoning is to be kept in mind when dealing with the empty set.

Example 2 Two empty sets $A = \{\}$ and $B = \{\}$ are equal. This is the case because the previous example tells us that $A \subset B$ and $B \subset A$, and this is simply the definition for $A = B$.

It is impossible to decide whether two circular sets $I = \{I\}$ and $J = \{J\}$ are equal.

2.1 The Axioms

Axiomatic set theory is defined by two components: Its objects are pure sets, and the instances of such sets are required to satisfy a number of properties, the axioms, which cannot be deduced by logical reasoning, but must be claimed. It is hard to show that such axioms lead to mathematically existing sets. We circumvent this problem by stating a list of common axioms.

The following collection of axioms is a variant of the collection proposed by Ernst Zermelo and Abraham Fraenkel (ZFC, for short). However, we do not include the axiom of foundation since modern theoretical computer science has a need for circular sets which this axiom excludes. Finally, we replace the axiom of extentionality by the axiom of equality, which respects more properly the difference between equality and identity. For a discussion of the classical ZFC axioms, see [20].

Axiom 1 (Axiom of Empty Set) *There is a set, denoted by \emptyset , which contains no element, i.e., for every set x , we have $x \notin \emptyset$, or, differently formulated, $\emptyset = \{\}$.*

Axiom 2 (Axiom of Equality) *If a, x, y are sets such that $x \in a$ and $x = y$, then $y \in a$.*

Axiom 3 (Axiom of Union) *If a is a set, then there is a set*

$$\{x \mid \text{there exists an element } b \in a \text{ such that } x \in b\}.$$

This set is denoted by $\bigcup a$ and is called the union of a .

Notation 2 If $a = \{b, c\}$, or $a = \{b, c, d\}$, respectively, one also writes $b \cup c$, or $b \cup c \cup d$, respectively, instead of $\cup a$.

Axiom 4 (Axiom of Pairs) If a and b are two sets, then there is the pair set $c = \{a, b\}$.

Notation 3 If ϕ is a propositional attribute for sets, then, if $\phi(x)$ is true, we simply write " $\phi(x)$ " to ease notation within formulas.

Axiom 5 (Axiom of Subsets for Propositional Attributes) If a is a set, and if ϕ is a propositional attribute for all elements of a , then there is the set $\{x \mid x \in a \text{ and } \phi(x)\}$; it is called the subset of a for ϕ , and is denoted by $a|\phi$.

Axiom 6 (Axiom of Powersets) If a is a set, then there is the powerset 2^a , which is defined by $2^a = \{x \mid x \subset a\}$, i.e., the propositional attribute $\phi(x) = "x \subset a"$. The powerset of a is also written $\mathcal{P}(a)$.

Example 3 The powerset of $c = \{a, b\}$ is $2^c = \{\emptyset, \{a\}, \{b\}, \{a, b\}\}$. If the inclusion relation is drawn as an arrow from x to y if $x \subset y$ then the powerset of c can be illustrated as in figure 2.2.

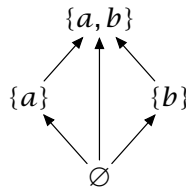


Fig. 2.2. The powerset of $\{a, b\}$.

For the next axiom, one needs the following proposition:

Lemma 1 For any set a , there is the set $a^+ = a \cup \{a\}$. It is called the successor of a .

Proof Axiom 6 states that for a given set a , the powerset of a exists. Since $a \subset a$, $\{a\} \in 2^a$, therefore $\{a\}$ exists. Axiom 3 then implies that $a \cup \{a\}$ exists. \square

Axiom 7 (Axiom of Infinity) There is a set w with $\emptyset \in w$ and such that $x \in w$ implies $x^+ \in w$.

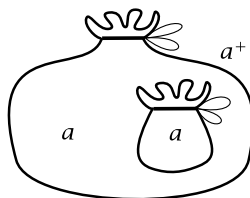


Fig. 2.3. The successor a^+ of a set a .

Remark 1 Axiom 1 is a consequence of axioms 5 and 7, but we include it, since axiom 7 is very strong (the existence of an infinite set is quite hypothetical for a computer scientist).

Definition 2 For two sets a and b , the set $\{x \mid x \in a \text{ and } x \in b\}$ is called the intersection of a and b and is denoted by $a \cap b$. If $a \cap b = \emptyset$, then a and b are called disjoint.

Axiom 8 (Axiom of Choice) Let a be a set whose elements are all non-empty, and such that any two different elements $x, y \in a$ are disjoint. Then there is a subset $c \subset \bigcup a$ (called choice set) such that for every non-empty $x \in a$, $x \cap c$ has exactly one element (see figure 2.4).

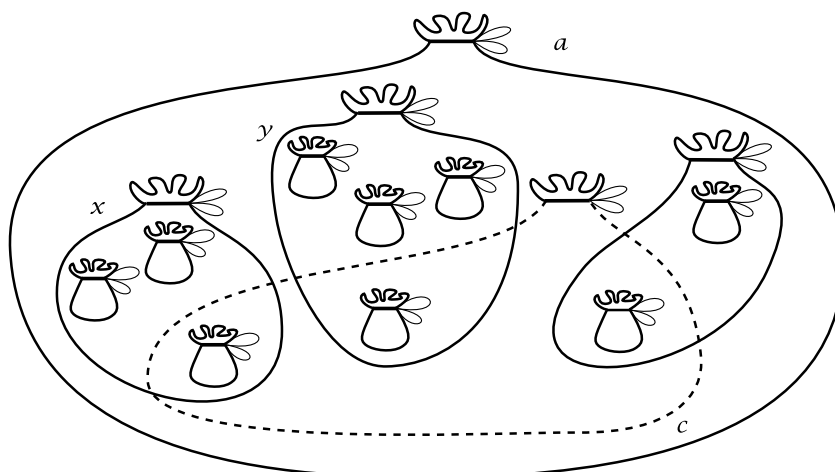


Fig. 2.4. Axiom of Choice: c is a choice set of the sets $x, y, \dots \in a$.

2.2 Basic Concepts and Results

We shall now develop the entire set theory upon these axioms. The beginning is quite hard, but we shall be rewarded with a beautiful result: all numbers from classical mathematics, all functions and all complex structures will emerge from this construction.

Sorte 2 For any three sets a, b, c , we have

- (i) $a \subset a$
- (ii) If $a \subset b$ and $b \subset a$, then $a = b$.
- (iii) If $a \subset b$ and $b \subset c$, then $a \subset c$.

Proof (i) If $x \in a$ then, trivially, $x \in a$. (ii) This is true by definition 1. (iii) Let $x \in a$. This implies $x \in b$, because $a \subset b$. Moreover, $b \subset c$ implies $x \in c$. So, $x \in a$ implies $x \in c$ for any x , and this means $a \subset c$ by definition. \square

Proposition 3 For any sets a, b, c, d :

- (i) (Commutativity of unions) the set $a \cup b$ exists and equals $b \cup a$,
- (ii) (Associativity of unions) the sets $(a \cup b) \cup c$ and $a \cup (b \cup c)$ exist and are equal, we may therefore write $a \cup b \cup c$ instead,
- (iii) $(a \cup b \cup c) \cup d$ and $a \cup (b \cup c \cup d)$ exist and are equal, we may therefore write $a \cup b \cup c \cup d$ instead.

Proof (i) By axiom 4 the set $x = \{a, b\}$ exists. By axiom 3 both unions exist and we have $a \cup b = \bigcup x = \{c \mid \text{there is a } m \in x \text{ such that } c \in m\} = \{c \mid c \in a \text{ or } c \in b\}$. On the other hand, $b \cup a = \bigcup y$, where $y = \{b, a\} = x$, so the two unions are equal.

(ii) $(a \cup b) \cup c = \{x \mid x \in a \cup b\} \cup c = \{x \mid x \in a \cup b \text{ or } x \in c\} = \{x \mid x \in a \text{ or } x \in b \text{ or } x \in c\}$. On the other hand, $a \cup (b \cup c) = \{x \mid x \in a \text{ or } x \in b \cup c\} = \{x \mid x \in a \text{ or } x \in b \text{ or } x \in c\}$, so the two are equal.

(iii) follows from (ii) by replacing d with c and b with $b \cup c$ in the proof. \square

Remark 2 The set whose elements are all sets x with $x \notin x$ does not exist, in fact both, the property $x \in x$, as well as $x \notin x$ lead to contradictions. Therefore, by axiom 5, the set of all sets does not exist.

Proposition 4 If $a \neq \emptyset$, then the set $\{x \mid x \in z \text{ for all } z \in a\}$ exists, it is called the intersection of a and is denoted by $\bigcap a$. However, for $a = \emptyset$, the attribute $\Phi(x) = "x \in z \text{ for all } z \in a"$ is fulfilled by every set x , and therefore $\bigcap \emptyset$ is inexistent, since it would be the non-existent set of all sets.

Proof If $a \neq \emptyset$, and if $b \in a$ is one element satisfying the attribute Φ , the required set is also defined by $\{x \mid x \in b \text{ and } x \in z \text{ for all } z \in a\}$. So this attribute selects a subset of b defined by Φ , which is a legitimate set according to axiom 5. If $a = \emptyset$, then the attribute $\Phi(x)$ alone is true for every set x , which leads to the inexistent set of all sets. \square

Definition 3 For two sets a and b , the complement of a in b or the difference of b and a is the set $\{x \mid x \notin a \text{ and } x \in b\}$. It is denoted by $b - a$.

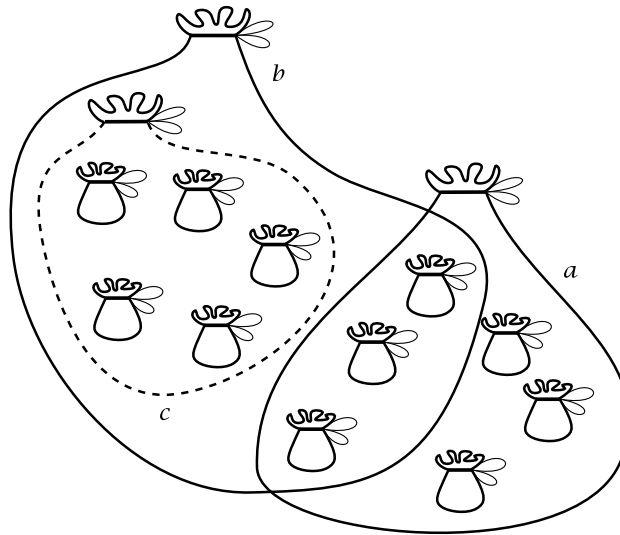


Fig. 2.5. The complement c of a in b , or $c = b - a$.

Sorite 5 For any three sets a, b, c we have

- (i) $(c - a) \subset c$,
- (ii) If $a \subset c$, then $c - (c - a) = a$,
- (iii) $c - \emptyset = c$,
- (iv) $c - c = \emptyset$,
- (v) $a \cap (c - a) = \emptyset$,
- (vi) If $a \subset c$, then $a \cup (c - a) = c$,
- (vii) $c - (a \cup b) = (c - a) \cap (c - b)$,
- (viii) $c - (a \cap b) = (c - a) \cup (c - b)$,
- (ix) $c \cap (a - b) = (c \cap a) - (c \cap b)$.

Proof We shall only prove part of the statements; the proof of the remaining statements is left as an exercise for the reader.

(ii)

$$\begin{aligned}
c - (c - a) &= \{x \mid x \in c \text{ and } x \notin (c - a)\} \\
&= \{x \mid x \in c \text{ and not } (x \in c \text{ and } x \notin a)\} \\
&= \{x \mid x \in c \text{ and } (x \notin c \text{ or not } x \notin a)\} & (1) \\
&= \{x \mid x \in c \text{ and } (x \notin c \text{ or } x \in a)\} \\
&= \{x \mid (x \in c \text{ and } x \notin c) \text{ or } (x \in c \text{ and } x \in a)\} & (2) \\
&= \{x \mid x \in c \text{ and } x \in a\} & (3) \\
&= \{x \mid x \in a\} & (4) \\
&= a
\end{aligned}$$

Equality (1) follows from de Morgan's law, equality (2) from distributivity of AND over OR. Equality (3) holds because the condition $(x \in c \text{ and } x \notin c)$ is always false. Equality (4) holds because $a \subset c$ means that $x \in a$ already implies $x \in c$, which therefore can be omitted. For the rules of transformation and simplification used here, see also the discussion of truth tables on page 5.

(iv) By definition, $c - c = \{x \mid x \in c \text{ and } x \notin c\}$. Obviously, there is no x which can fulfill both of these contradictory conditions, so $c - c = \emptyset$.

(v)

$$\begin{aligned}
a \cap (c - a) &= \{x \mid x \in a \text{ and } x \in c - a\} \\
&= \{x \mid x \in a \text{ and } (x \in c \text{ and } x \notin a)\} \\
&= \{x \mid x \in a \text{ and } x \notin a \text{ and } x \in c\} & (*) \\
&= \emptyset
\end{aligned}$$

Here we use the commutativity and associativity of AND to regroup and reorder the terms of the propositional attribute. In line (*) the attribute contains the conjunction of a statement and its negation, which is always false, therefore the result is the empty set.

(vii)

$$\begin{aligned}
c - (a \cup b) &= \{x \mid x \in c \text{ and } x \notin (a \cup b)\} \\
&= \{x \mid x \in c \text{ and not } (x \in a \text{ or } x \in b)\} \\
&= \{x \mid x \in c \text{ and } (x \notin a \text{ and } x \notin b)\} \\
&= \{x \mid x \in c \text{ and } x \notin a \text{ and } x \notin b\} & (1) \\
&= \{x \mid x \in c \text{ and } x \notin a \text{ and } x \in c \text{ and } x \notin b\} & (2) \\
&= \{x \mid (x \in c \text{ and } x \notin a) \text{ and } (x \in c \text{ and } x \notin b)\} & (3) \\
&= \{x \mid x \in c - a \text{ and } x \in c - b\} & (4)
\end{aligned}$$

$$\begin{aligned} &= \{x \mid x \in c - a\} \cap \{x \mid x \in c - b\} \\ &= (c - a) \cap (c - b) \end{aligned} \tag{5}$$

Equalities (1) and (3) hold because AND is associative. Equality (2) holds because $P \text{ AND } Q = P \text{ AND } P \text{ AND } Q$ for any truth values P and Q . Equality (4) is the definition of the set difference. Equality (5) is the definition of the intersection of two sets. \square

Boolean Set Algebra

In this chapter, we shall give a more systematic account of the construction of sets by use of union, intersection and complement. The structures which emerge in this chapter are prototypes of algebraic structures which will appear throughout the entire course.

3.1 The Boolean Algebra of Subsets

Lemma 6 *For two sets a and b , the union $a \cup b$ is a least upper bound, i.e., $a, b \subset a \cup b$, and for every set c with $a, b \subset c$, we have $a \cup b \subset c$. This property uniquely determines the union.*

Dually, the intersection $a \cap b$ is a greatest lower bound, i.e., $a \cap b \subset a, b$, and for every set c with $c \subset a, b$, we have $c \subset a \cap b$. This property uniquely determines the intersection.

Proof Clearly, $a \cup b$ is a least upper bound. If x and y are any two least upper bounds of a and b , then by definition, we must have $x \subset y$ and $y \subset x$, therefore $x = y$. The dual statement is demonstrated by analogous reasoning. \square

Summarizing the previous properties of sets, we have the following important theorem, stating that the powerset 2^a of a set a is a *Boolean algebra*. We shall discuss this structure in a more systematic way in chapter 17.

Proposition 7 (Boolean Algebra of Subsets) *For a given set a , the powerset 2^a has the following properties. Let x, y, z be any elements of 2^a , i.e., subsets of a ; also, denote $x' = a - x$. Then:*

- (i) (Reflexivity) $x \subset x$,
- (ii) (Antisymmetry) if $x \subset y$ and $y \subset x$, then $x = y$,
- (iii) (Transitivity) if $x \subset y$ and $y \subset z$, then $x \subset z$,
- (iv) we have a “minimal” set $\emptyset \in 2^a$ and a “maximal” set $a \in 2^a$, and $\emptyset \subset x \subset a$,
- (v) (Least upper bound) the union $x \cup y$ verifies $x, y \subset x \cup y$, and for every z , if $x, y \subset z$, then $x \cup y \subset z$,
- (vi) (dually: Greatest lower bound) the intersection $x \cap y$ verifies $x \cap y \subset x, y$, and for every z , if $z \subset x, y$, then $z \subset x \cap y$,
- (vii) (Distributivity) $(x \cup y) \cap z = (x \cap z) \cup (y \cap z)$ and (dually) $(x \cap y) \cup z = (x \cup z) \cap (y \cup z)$,
- (viii) we have $x \cup x' = a, x \cap x' = \emptyset$

Proof (i) is true for any set, see sorite 2.

(ii) is the very definition of equality of sets.

(iii) is immediate from the definition of subsets.

(iv) is clear.

(v) and (vi) were proved in lemma 6.

(vii) is clear.

(viii) follows immediately from the definition of the complement x' . □

Example 4 Figure 3.1 uses Venn diagrams to illustrate distributivity of \cap over \cup . In this intuitive representation, sets are drawn circular areas or parts thereof.

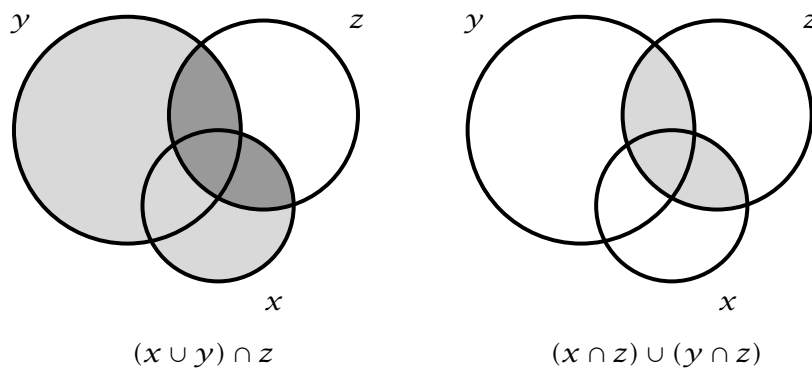


Fig. 3.1. Distributivity of \cap over \cup .

Exercise 1 Given a set $a = \{r, s, t\}$ consisting of pairwise different sets, give a complete description of 2^a and the intersections or unions, respectively, of elements of 2^a .

Here is a second, also important structure on the powerset of a given set a :

Definition 4 For $x, y \in 2^a$, we define $x + y = (x \cup y) - (x \cap y)$ (symmetric set difference). We further define $x \cdot y = x \cap y$. Both operations are illustrated using Venn diagrams in figure 3.2.

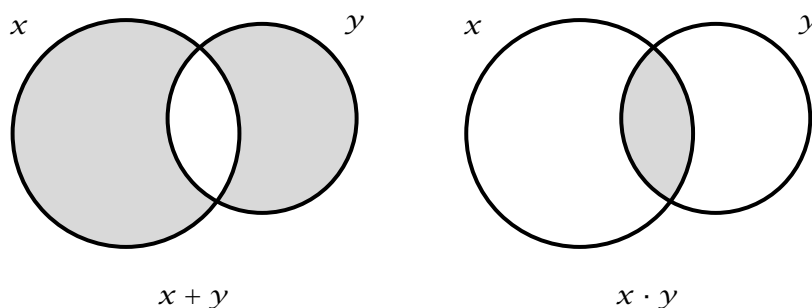


Fig. 3.2. Symmetric difference and intersection of sets.

Proposition 8 For a set a , and for any three elements $x, y, z \in 2^a$, we have:

- (i) (Commutativity) $x + y = y + x$ and $x \cdot y = y \cdot x$,
- (ii) (Associativity) $x + (y + z) = (x + y) + z$, $x \cdot (y \cdot z) = (x \cdot y) \cdot z$, we therefore also write $x + y + z$ and $x \cdot y \cdot z$, respectively,
- (iii) (Neutral elements) we have $x + \emptyset = x$ and $x \cdot a = x$,
- (iv) (Distributivity) $x \cdot (y + z) = x \cdot y + x \cdot z$,
- (v) (Idempotency) $x \cdot x = x$,
- (vi) (Involution) $x + x = \emptyset$,
- (vii) the equation $x + y = z$ has exactly one solution w for the unknown x , i.e., there is exactly one set $w \subset a$ such that $w + y = z$.

Proof (i) follows from the commutativity of the union $a \cup b$ and the intersection $a \cap b$, see also proposition 3.

(ii) associativity also follows from associativity of union and intersection, see again proposition 3.

(iii) we have $x + \emptyset = (x \cup \emptyset) - (x \cap \emptyset) = x - \emptyset = x$, $x \cdot a = x \cap a = x$.

(iv)

$$\begin{aligned} x \cdot (y + z) &= x \cap ((y \cup z) - (y \cap z)) \\ &= x \cap (y \cup z) - x \cap y \cap z \\ &= ((x \cap y) \cup (x \cap z)) - x \cap y \cap z \end{aligned}$$

whereas

$$\begin{aligned} x \cdot y + x \cdot z &= ((x \cap y) \cup (x \cap z)) - ((x \cap y) \cap (x \cap z)) \\ &= ((x \cap y) \cup (x \cap z)) - x \cap y \cap z \end{aligned}$$

and we are done.

(v) and (vi) are immediate from the definitions.

(vii) in view of (vi), $w = y + z$ is a solution. For any two solutions $w + y = w' + y$, one has $w = w + y + y = w' + y + y = w'$. \square

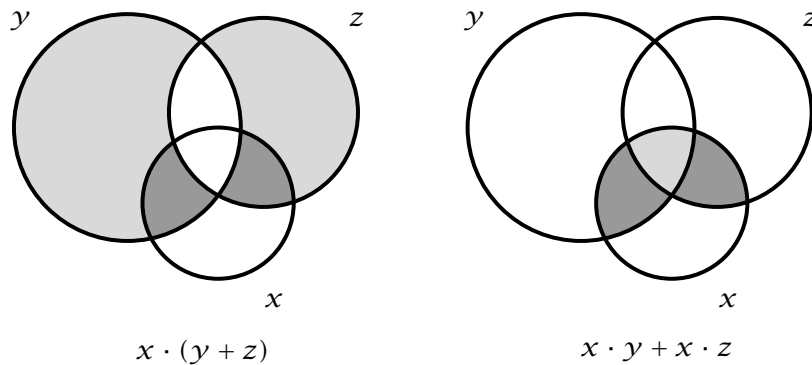


Fig. 3.3. Distributivity of \cdot over $+$.

Remark 3 This structure will later be discussed as the crucial algebraic structure of a commutative ring, see chapter 15.

Exercise 2 Let $a = \{r, s, t\}$ as in exercise 1. Calculate the solution of $w + y = z$ within 2^a for $y = \{r, s\}$ and $z = \{s, t\}$.

Exercise 3 Let $a = \{\emptyset\}$. Calculate the complete tables of sums $x + y$ and products $x \cdot y$, respectively, for $x, y \in 2^a$, use the symbols $0 = \emptyset$ and $1 = a$. What do they remind you of?

Functions and Relations

We have seen in chapter 1 that the conceptual architecture may be a selection, conjunction, or disjunction. Sets are built on the selection type. They are however suited for simulating the other types as well. More precisely, for the conjunction type, one needs to know the position of each of two conceptual coordinates: Which is the first, which is the second. In the selective type, however, no order between elements of a set is given, i.e., $\{x, y\} = \{y, x\}$. So far, we have no means for creating order among such elements. This chapter solves this problem in the framework of set theory.

4.1 Graphs and Functions

Definition 5 *If x and y are two sets, the ordered pair (x, y) is defined to be the following set:*

$$(x, y) = \{\{x\}, \{x, y\}\}$$

Observe that the set (x, y) always exists, it is a subset of the powerset of the pair set $\{x, y\}$, which exists according to axiom 4.

Here is the essence of this definition:

Lemma 9 *For any four sets a, b, c, d , we have $(a, b) = (c, d)$ iff $a = c$ and $b = d$. Therefore one may speak of the first and second coordinate a and b , respectively, of the ordered pair (a, b) .*

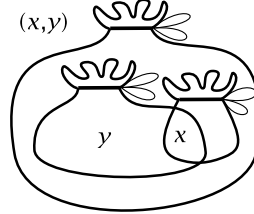


Fig. 4.1. The bag representation of the ordered pair (x, y) .

Proof The ordered pair (x, y) has one single element $\{x\}$ iff $x = y$, and it has different elements $\{x\} \neq \{x, y\}$ iff $x \neq y$. So, if $(a, b) = (c, d)$, then either $a = b$ and $c = d$, and then $\{\{a\}\} = (a, b) = (c, d) = \{\{c\}\}$, whence $a = c$. Or else, $a \neq b$ and $c \neq d$. But then the only element with one element in (a, b) is $\{a\}$. Similarly the only element with one element in (c, d) is $\{c\}$. So $(a, b) = (c, d)$ implies $a = c$. Similarly, the other element $\{a, b\}$ of (a, b) must be equal to $\{c, d\}$. But since $a = c$ and $a \neq b$, we have $b = d$, and we are done. The converse implication is evident. \square

Exercise 4 Defining $(x, y, z) = ((x, y), z)$, show that $(x, y, z) = (u, v, w)$ iff $x = u$, $y = v$, and $z = w$.

Lemma 10 Given two sets a and b , there is a set

$$a \times b = \{(x, y) \mid x \in a \text{ and } y \in b\},$$

it is called the Cartesian product of a and b .

Proof We have the set $v = a \cup b$. Let $P = 2^{(2^v)}$ be the powerset of the powerset of v , which also exists. Then an ordered pair $(x, y) = \{\{x\}, \{x, y\}\}$, with $x \in a$ and $y \in b$ is evidently an element of P . Therefore $a \times b$ is the subset of those $p \in P$ defined by the propositional attribute $\Phi(p) = \text{“there are } x \in a \text{ and } y \in b \text{ such that } p = (x, y)\text{.”}$ \square

Sorite 11 Let a, b, c, d be sets. Then:

- (i) $a \times b = \emptyset$ iff $a = \emptyset$ or $b = \emptyset$,
- (ii) if $a \times b \neq \emptyset$, then $a \times b = c \times d$ iff $a = c$ and $b = d$.

Proof The first claim is evident. As to the second, if $a \times b \neq \emptyset$, then we have $a \cup b = \bigcup(\bigcup(a \times b))$, as is clear from the definition of ordered pairs. Therefore we have the subset $a = \{x \mid x \in a \cup b, \text{ there is } z \in a \times b \text{ with } z = (x, y)\}$. Similarly for b , and therefore also $a = c$ and $b = d$. \square

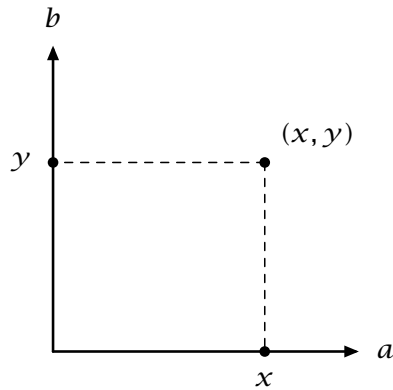


Fig. 4.2. It has become common practice to represent Cartesian products $a \times b$ intuitively by two axes, the horizontal axis representing a , i.e., $x \in a$ is drawn as a point on this axis, and the vertical axis representing the set b , i.e., $y \in b$ is drawn as a point on this axis. In traditional language, the horizontal axis is called the *abscissa*, while the vertical axis is called the *ordinate*. The element (x, y) is drawn as a point on the plane, whose coordinates x and y are obtained by projections perpendicular to the respective axes.

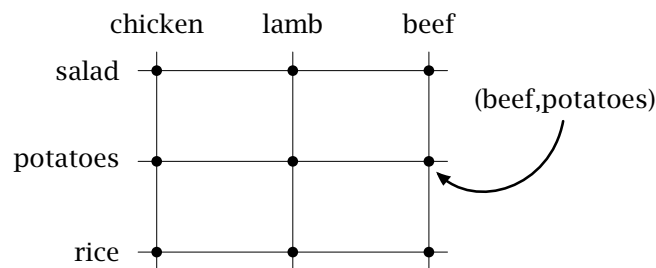


Fig. 4.3. The idea behind Cartesian products is to define sets which are composed from two given sets by simultaneous specification of two elements in order to define one of the Cartesian product. Here, to define a meal, we are given two intuitive sets: *meat dish* and accompanying *side dish*. In OOP, the meal class would have two instance variables, `meat_dish` and `side_dish`, both allowing appropriate character strings as values, “chicken”, “beef”, “lamb”, and “salad”, “rice”, “potatoes”, respectively. Logically, the information encoded in a Cartesian product is that of a conjunction: To know an object of a Cartesian product is to know its first coordinate AND its second coordinate.

Definition 6 If a Cartesian product $a \times b$ is non-empty, we call the uniquely determined sets a and b its first and second projection, respectively, and we write $a = pr_1(a \times b)$ and $b = pr_2(a \times b)$.

The following concept of a graph is a formalization of the act of associating two objects out of two domains, such as the pairing of a man and a woman, or associating a human and its bank accounts.

Lemma 12 The following statements about a set g are equivalent:

- (i) The set g is a subset of a Cartesian product set $a \times b$.
- (ii) Every element $x \in g$ is an ordered pair $x = (u, v)$.

Proof Clearly, (i) implies (ii). Conversely, if g consists of ordered pairs, we may take $P = \bigcup(\bigcup g)$, and then immediately see that $g \subset P \times P$. \square

Definition 7 A set which satisfies one of the two equivalent properties of lemma 12 is called a graph.

Example 5 For any set a , the *diagonal graph* is the graph

$$\Delta_a = \{(x, x) \mid x \in a\}.$$

Lemma 13 For a graph g , there are two sets

$$pr_1(g) = \{u \mid (u, v) \in g\} \text{ and } pr_2(g) = \{v \mid (u, v) \in g\},$$

and we have $g \subset pr_1(g) \times pr_2(g)$.

Proof As in the previous proofs, we take the double union $P = \bigcup(\bigcup g)$ and from P extract the subsets $pr_1(g)$ and $pr_2(g)$ as defined in this proposition. The statement $g \subset pr_1(g) \times pr_2(g)$ is then straightforward. \square

Proposition 14 If g is a graph, there is another graph, denoted by g^{-1} and called the inverse graph of g , which is defined by

$$g^{-1} = \{(v, u) \mid (u, v) \in g\}.$$

We have $(g^{-1})^{-1} = g$.

Proof According to lemma 12, there are sets a and b such that $g \subset a \times b$. Then $g^{-1} \subset b \times a$ is the inverse graph. The statement about double inversion is immediate. \square

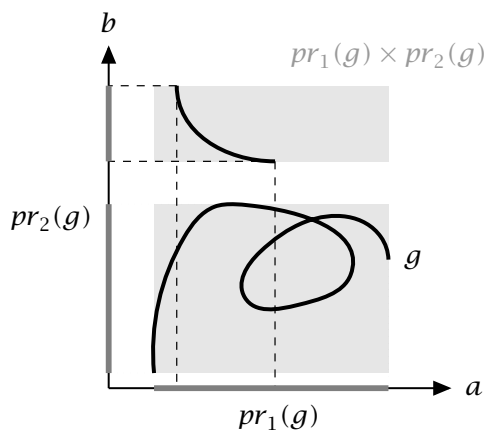


Fig. 4.4. The projections $pr_1(g)$ and $pr_2(g)$ (dark gray segments on the axes) of a graph g (black curves). Note that $g \subset pr_1(g) \times pr_2(g)$ (light gray rectangles).

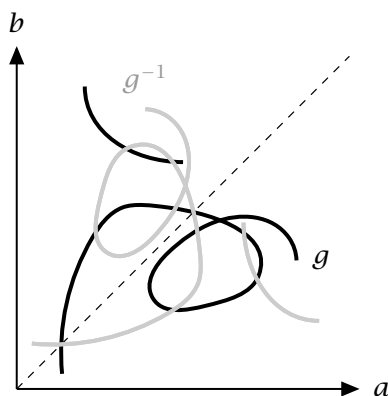


Fig. 4.5. The inverse g^{-1} of a graph g .

Exercise 5 Show that $g = \Delta_{pr_1(g)}$ implies $g = g^{-1}$; give counterexamples for the converse implication.

Definition 8 If g and h are two graphs, there is a set $g \circ h$, the composition of g with h (attention: the order of the graphs is important here), which is defined by

$$g \circ h = \{(v, w) \mid \text{there is a set } u \text{ such that } (v, u) \in h \text{ and } (u, w) \in g\}.$$

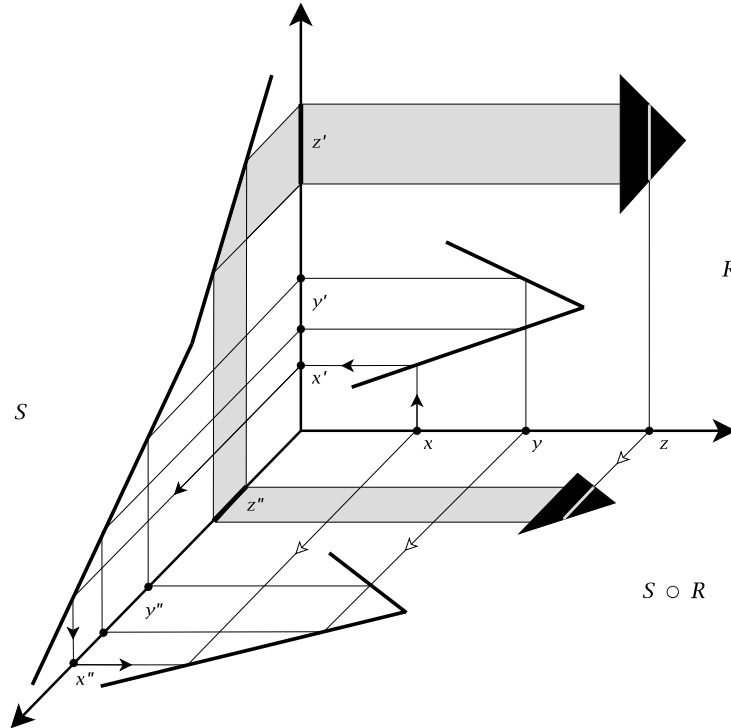


Fig. 4.6. The composition $S \circ R$ of two graphs R and S .

Figure 4.6 illustrates the composition of two graphs R and S . The point x is mapped by R to the point x' , which in turn is mapped by S to the point x'' . R maps the point y to two points both denoted by y' , and these two points are mapped by S to two points y'' . Last, z is mapped by R to a segment z' . This segment is mapped by S to a segment z'' .

Sorite 15 Let f, g, h be three graphs.

- (i) (Associativity) We have $(f \circ g) \circ h = f \circ (g \circ h)$ and we denote this graph by $f \circ g \circ h$.
- (ii) We have $\Delta_{pr_1(g)} \subset g^{-1} \circ g$.

Proof These statements follow from a straightforward application of the definition. \square

Definition 9 A graph g is called functional if $(u, v) \in g$ and $(u, w) \in g$ imply $v = w$.

Exercise 6 Show that the composition $g \circ h$ of two functional graphs g and h is functional.

Example 6 For any sets a and b , the diagonal graph Δ_a is functional, whereas the Cartesian product $a \times b$ is not functional if $a \neq \emptyset$ and if there are sets $x, y \in b$ with $x \neq y$.

Definition 10 A function is a triple (a, f, b) such that f is a functional graph, where $a = pr_1(f)$ and $pr_2(f) \subset b$. The set a is called the domain of the function, the set b is called its codomain, and the set $pr_2(f)$ is called the function's image and denoted by $Im(f)$. One usually denotes a function by a more graphical sign $f : a \rightarrow b$. For $x \in a$, the unique $y \in b$ such that $(x, y) \in f$ is denoted by $f(x)$ and is called the value of the function at the argument x . Often, if the domain and codomain are clear, one identifies the function with the graph sign f , but this is not the valid definition. One then also notates $a = dom(f)$ and $b = codom(f)$.

Example 7 For any set a , the identity function (on a) Id_a is defined by $Id_a = (a, \Delta_a, a)$.

Exercise 7 For the set $1 = \{\emptyset\}$ and for any set a , show that there is exactly one function $(a, f, 1)$. We denote this function by $! : a \rightarrow 1$. (The notation "1" is not quite arbitrary, we shall see the systematic background in chapter 5.) If $a = \emptyset$, and if b is any set, show that there is a unique function (\emptyset, g, b) , also denoted by $! : \emptyset \rightarrow b$.

Definition 11 A function $f : a \rightarrow b$ is called epimorphism, or epi, or surjective or onto if $Im(f) = codom(f)$.

It is called monomorphism, or mono, or injective or one-to-one if $f(x) = f(y)$ implies $x = y$ for all sets $x, y \in dom(f)$.

The function is called isomorphism, or iso, or bijective if it is epi and mono. Isomorphisms are also denoted by special arrows, i.e., $f : a \xrightarrow{\sim} b$.

Example 8 Figure 4.7 illustrates three functions, $f : A \rightarrow B$, $g : B \rightarrow A$ and $h : B \rightarrow C$. An arrow from an element (point) $x \in X$ to an element $y \in Y$ indicates that (x, y) is in the graph $\kappa \subset X \times Y$ of a function $k = (X, \kappa, Y)$. The function f is epi, but not mono, g is mono, but not epi, and h is mono and epi, and thus, iso. The star-shaped points are the "culprits", i.e., the reasons that f is not mono and g is not epi.

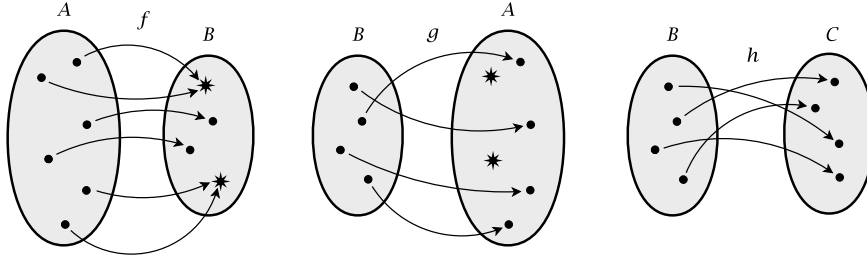


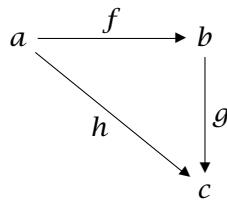
Fig. 4.7. Epimorphism f , monomorphism g and isomorphism h .

Exercise 8 The function $! : a \rightarrow 1$ is epi for $a \neq \emptyset$, the function $! : \emptyset \rightarrow b$ is always mono, and the identity function Id_a is always iso.

Exercise 9 Show that the inverse graph of monomorphism is a functional graph, but not necessarily the graph of a function.

Definition 12 Let $f : a \rightarrow b$ and $g : b \rightarrow c$ be functions, then their composition is the function $g \circ f : a \rightarrow c$.

When dealing with functions, one often uses a more graphical representation of the involved arrows by so-called arrow diagrams. The domains and codomains are shown as symbols on the plane, which are connected by arrows representing the given functions. For example, the functions $f : a \rightarrow b$ and $g : b \rightarrow c$ and their composition $h = g \circ f$ are shown as a triangular diagram. This diagram is *commutative* in the sense that both “paths” $a \xrightarrow{f} b \xrightarrow{g} c$ and $a \xrightarrow{h} c$ define the same function $h = g \circ f$.



Sorite 16 Let $f : a \rightarrow b$, $g : b \rightarrow c$ and $h : c \rightarrow d$ be functions.

- (i) The compositions $(h \circ g) \circ f : a \rightarrow d$ and $h \circ (g \circ f) : a \rightarrow d$ are equal, we therefore denote them by $h \circ g \circ f : a \rightarrow d$.
- (ii) The function $g : b \rightarrow c$ is mono iff the following condition holds: For any two functions $f, f' : a \rightarrow b$, $g \circ f = g \circ f'$ implies $f = f'$.

- (iii) The function $f : a \rightarrow b$ is epi iff the following condition holds: For any two functions $g, g' : b \rightarrow c$, $g \circ f = g' \circ f$ implies $g = g'$.
- (iv) If f and g are epi, mono, iso, respectively, then so is $g \circ f$.
- (v) If f is mono and $a \neq \emptyset$, then there is a—necessarily epi—function $r : b \rightarrow a$ such that $r \circ f = Id_a$, such a function is called a left inverse or retraction of f .
- (vi) If f is epi, then there is a—necessarily mono—function $s : b \rightarrow a$ such that $f \circ s = Id_b$, such a function is called a right inverse or section of f .
- (vii) The function f is iso iff there is a (necessarily unique) inverse, denoted by $f^{-1} : b \rightarrow a$, such that $f^{-1} \circ f = Id_a$ and $f \circ f^{-1} = Id_b$.

Proof (i) follows from the associativity of graph composition, see sorite 15.

(ii) For $x \in a$, $(g \circ f)(x) = (g \circ f')(x)$ means $g(f(x)) = g(f'(x))$, but since g is mono, $f(x) = f'(x)$, for all $x \in a$, whence $f = f'$. Conversely, take $u, v \in b$ such that $g(u) = g(v)$. Define two maps $f, f' : 1 \rightarrow b$ by $f(0) = u$ and $f'(0) = v$. Then $g \circ f = g \circ f'$. So $f = f'$, but this means $u = f(0) = f'(0) = v$, therefore g is injective.

(iii) If f is epi, then for every $y \in b$, there is $x \in a$ with $y = f(x)$. If $g \circ f = g' \circ f$, then $g(y) = g(f(x)) = g'(f(x)) = g'(y)$, whence $g = g'$. Conversely, if f is not epi, then let $z \notin Im(f)$. Define a function $g : b \rightarrow \{0, 1\}$ by $g(y) = 0$ for all $y \in b$. Define $g' : b \rightarrow \{0, 1\}$ by $g'(y) = 0$ for all $y \neq z$, and $g'(z) = 1$. We then have two different functions g and g' such that $g \circ f = g' \circ f$.

(iv) This property follows elegantly from the previous characterization: let f and g be epi, then for $h, h' : c \rightarrow d$, if $h \circ g \circ f = h' \circ g \circ f$, then, since f is epi, we conclude $h \circ g = h' \circ g$, and since g is epi, we have $h = h'$. The same formal argumentation works for mono. Since iso means mono and epi, we are done.

(v) If $f : a \rightarrow b$ is mono, then the inverse graph f^{-1} is also functional and $pr_2(f^{-1}) = a$. Take any element $y \in a$ and take the graph $r = f^{-1} \cup (b - Im(f)) \times \{y\}$. This defines a retraction of f .

(vi) Let f be epi. For every $x \in b$, let $F(x) = \{y \mid y \in a, f(y) = x\}$. Since f is epi, no $F(x)$ is empty, and $F(x) \cap F(x') = \emptyset$ if $x \neq x'$. By the axiom of choice 8, there is a set $q \subset a$ such that $q \cap F(x) = \{q_x\}$ is a set with exactly one element q_x for every $x \in b$. Define $s(x) = q_x$. This defines the section $s : b \rightarrow a$ of f .

(vii) The case $a = \emptyset$ is trivial, so let us suppose $a \neq \emptyset$. Then the characterizations (v) and (vi), together with the fact that “mono + epi = iso”, answer our problem. \square

Remark 4 The proof of statement (vi) in sorite 16 is a very strong one since it rests on the axiom of choice 8.

Definition 13 Let $f : a \rightarrow b$ be a function, and let a' be a set. Then the restriction of f to a' is the function $f|_{a'} : a \cap a' \rightarrow b$, where the graph is $f|_{a'} = f \cap ((a \cap a') \times b)$.

Definition 14 Let $f : a \rightarrow b$ and $g : c \rightarrow d$ be two functions. Then the Cartesian product of f and g is the function $f \times g : a \times c \rightarrow b \times d$ with $(f \times g)(x, y) = (f(x), g(y))$.

Sorite 17 Let $f : a \rightarrow b$ and $g : c \rightarrow d$ be two functions. Then the Cartesian product $f \times g$ is injective (surjective, bijective) if f and g are so.

Proof If one of the domains a or c is empty the claims are obvious. So suppose $a, c \neq \emptyset$. Let f and g be injective and take two elements $(x, y) \neq (x', y')$ in $a \times b$. Then either $x \neq x'$ or $y \neq y'$. In the first case, $(f \times g)(x, y) = (f(x), g(y)) \neq (f(x'), g(y)) = (f \times g)(x', y)$, the second case is analogous. A similar argument settles the cases of epi maps, and the case of iso maps is just the conjunction of the mono and epi cases. \square

The next subject is the basis of the classification of sets. The question is: When are two sets “essentially different”? This is the crucial definition:

Definition 15 A set a is said to be equipollent to b or to have the same cardinality as b iff there is a bijection $f : a \xrightarrow{\sim} b$. We often just write $a \xrightarrow{\sim} b$ to indicate the fact that a and b are equipollent.

Example 9 In figure 4.8, the set A of stars, the set B of crosses and the set C of plusses are equipollent. The functions $f : A \rightarrow B$ and $g : B \rightarrow C$ are both bijections. The composition of g and f is a bijection $h = g \circ f : A \rightarrow C$. The purpose of this example is to show that equipollence is a feature independent of the shape, or “structure” of the set. It only tells us that each element from the first set can be matched with an element from the second set, and vice-versa.

Proposition 18 For all sets a, b and c , we have:

- (i) (Reflexivity) a is equipollent to a .
- (ii) (Symmetry) If a is equipollent to b , then b is equipollent to a .
- (iii) (Transitivity) If a is equipollent to b , and if b is equipollent to c , then a is equipollent to c .

Proof Reflexivity follows from the fact that the identity Id_a is a bijection. Symmetry follows from statement (vii) in sorite 16. Transitivity follows from statement (iv) of sorite 16. \square

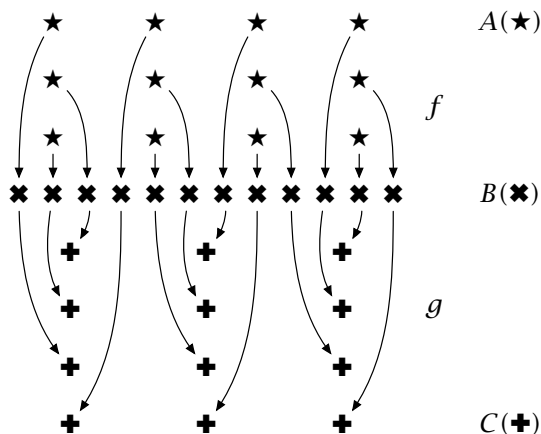


Fig. 4.8. The equipollence of sets A, B and C .

Are there arbitrary large sets? Here are first answers:

Exercise 10 Show that there is an injection $sing : a \rightarrow 2^a$ for each set a , defined by $sing(x) = \{x\}$.

An injection in the reverse direction never exists. We have in fact this remarkable theorem.

Proposition 19 For any set a , a and 2^a are never equipollent.

Proof Suppose that we have a function $f : a \rightarrow 2^a$. We show that f cannot be surjective. Take the subset $g \subset a$ defined by $g = \{x \mid x \in a, x \notin f(x)\}$. We claim that $g \notin Im(f)$. If $z \in a$ with $f(z) = g$, then if $z \notin g$, then $z \in g$, a contradiction. But if $z \in g$, then $z \notin f(z)$ by definition of g , again a contradiction. So $f(z) = g$ is impossible. \square

To see why this contradicts the existence of an injection $2^a \rightarrow a$, we need some further results:

Sorite 20 Let a, b, c, d be sets.

- (i) If $a \sim b$ and $c \sim d$, then $a \times c \sim b \times d$.
- (ii) If $a \sim b$ and $c \sim d$, and if $a \cap c = b \cap d = \emptyset$, then $a \cup c \sim b \cup d$.

Proof (i) If $f : a \rightarrow b$ and $g : c \rightarrow d$ are bijections, then so is $f \times g$ by sorite 17.
 (ii) If $f : a \rightarrow b$ and $g : c \rightarrow d$ are bijections, then the graph $f \cup g \subset (a \cup c) \times (b \cup d)$ clearly defines a bijection. \square

The following is a technical lemma:

Lemma 21 *If $p \cup q \cup r \overset{\sim}{\sim} p$ and $p \cap q = \emptyset$, then $p \cup q \overset{\sim}{\sim} p$.*

Proof This proof is more involved and should be read only by those who really are eager to learn about the innards of set theory. To begin with, fix a bijection $f : p \cup q \cup r \rightarrow p$. If $x \in p \cup q \cup r$, we also write $f(x)$ for $\{f(z) \mid z \in x\}$.

Consider the following propositional attribute $\Psi(x)$ of elements $x \in 2^{p \cup q \cup r}$: We define $\Psi(x) = "q \cup f(x) \subset x"$. For example, we have $\Psi(p \cup q)$. We now show that if $\Psi(x)$, then also $\Psi(q \cup f(x))$. In fact, if $q \cup f(x) \subset x$, then $f(q \cup f(x)) \subset f(x)$, and a fortiori $f(q \cup f(x)) \subset q \cup f(x)$, therefore $q \cup f(q \cup f(x)) \subset q \cup f(x)$, thus $\Psi(q \cup f(x))$, therefore $\Psi(k)$.

Next, let $b \subset 2^{p \cup q \cup r}$ with $\Psi(x)$ for all $x \in b$. Now we show that $\Psi(\bigcap b)$. Denote $k = \bigcap b$. Since $k \subset x$ for all $x \in b$, we also have $f(k) \subset f(x)$ for all $x \in b$, and therefore $q \cup f(k) \subset q \cup f(x) \subset x$ for all $x \in b$. This implies $q \cup f(k) \subset k$.

Now let $b = e = \{x \mid x \in 2^{p \cup q \cup r} \text{ and } \Psi(x)\}$; we know that e is non-empty. Then $k = d = \bigcap e$. From the discussion above, it follows that $\Psi(d)$, i.e., $q \cup f(d) \subset d$. But by the first consideration, we also know that $\Psi(q \cup f(d))$, and, since d is the intersection of all x such that $\Psi(x)$, $d \subset q \cup f(d)$. This means that $q \cup f(d) \subset d \subset q \cup f(d)$, i.e., $q \cup f(d) = d$.

Moreover, $d \cap (p - f(d)) = \emptyset$. In fact, $d \cap (p - f(d)) = (q \cup f(d)) \cap (p - f(d)) = (q \cap (p - f(d))) \cup (f(d) \cap (p - f(d))) = \emptyset \cup \emptyset = \emptyset$ because we suppose $p \cap q = \emptyset$. So we have a disjoint union $q \cup p = d \cup (p - f(d))$. Now, we have a bijection $f : d \overset{\sim}{\sim} f(d)$ and a bijection (the identity) $p - f(d) \overset{\sim}{\sim} p - f(d)$. Therefore by sorite 20 (ii), we obtain the required bijection. \square

This implies a famous theorem:

Proposition 22 (Bernstein-Schröder) *Let a, b, c be three sets such that there exist two injections $f : a \rightarrow b$ and $g : b \rightarrow c$. If a and c are equipollent, then all three sets are equipollent.*

Proof We apply lemma 21 as follows. Let $f : a \rightarrow b$ and $g : b \rightarrow c$ be injections and $h : a \rightarrow c$ a bijection. Then we may take the image sets $a' = g(f(a))$ and $b' = g(b)$ instead of the equipollent sets a and b , respectively. Therefore without loss of generality we may show the theorem for the special situation of subsets $a \subset b \subset c$ such that a is equipollent to c . To apply our technical lemma we set $p = a$, $q = b - a$ and $r = c - b$. Therefore $c = p \cup q \cup r$ and $b = p \cup q$. In these terms, we are supposing that p is equipollent to $p \cup q \cup r$. Therefore the lemma yields that p is equipollent to $p \cup q$, i.e., a is equipollent to b . By transitivity of equipollence, b and c are also equipollent. \square

In particular:

Corollary 23 *If $a \subset b \subset c$, and if a is equipollent to c , then all three sets are equipollent.*

Corollary 24 *For any set a , there is no injection $2^a \rightarrow a$.*

Proof If we had an injection $2^a \rightarrow a$, the existing reverse injection $a \rightarrow 2^a$ from exercise 10 and proposition 22 would yield a bijection $a \xrightarrow{\sim} 2^a$ which is impossible according to proposition 19. \square

4.2 Relations

Until now, we have not been able to deal with “relations” in a formal sense. For example, the properties of reflexivity, symmetry, and transitivity, as encountered in proposition 18, are only properties of single pairs of sets, but the whole set of all such pairs is not given. The theory of relations will deal with such problems.

Definition 16 *A binary relation on a set a is a subset $R \subset a \times a$; this is a special graph, where the domain and codomain coincide and are specified by the choice of a . Often, instead of “ $(x, y) \in R$ ”, one writes “ xRy ”.*

Example 10 The empty relation $\emptyset \subset a \times a$, the complete relation $R = a \times a$, and the diagonal graph Δ_a are relations on a . For each relation R on a , the inverse relation $R^{-1} = \{(y, x) \mid (x, y) \in R\}$ (the inverse graph) is a relation on a . If R and S are two relations on a , then the composed graph $R \circ S$ defines the composed relation on a . In particular, we have the second power $R^2 = R \circ R$ of a relation R .

Notation 4 *Often, relation symbols are not letters, but special symbols such as $<, \leq, \prec, \dots$. Their usage is completely dependent on context and has no universal meaning. Given relations $<$ and \leq , the corresponding inverse relations are denoted by $>$ and \geq , respectively.*

Definition 17 *Let \leq be a relation on a . The relation is called*

- (i) reflexive iff $x \leq x$ for all $x \in a$;
- (ii) transitive iff $x \leq y$ and $y \leq z$ implies $x \leq z$ for all $x, y, z \in a$;
- (iii) symmetric iff $x \leq y$ implies $y \leq x$ for all $x, y \in a$;
- (iv) antisymmetric iff $x \leq y$ and $y \leq x$ implies $x = y$ for all $x, y \in a$;
- (v) total iff for any two $x, y \in a$, either $x \leq y$ or $y \leq x$.

(vi) equivalence relation, iff it is reflexive, symmetric, and transitive. In this case, the relation is usually denoted by \sim instead of \leq .

Example 11 We shall illustrate these properties with examples from the real world. The relation “ x is an ancestor of y ” is transitive, but neither reflexive nor symmetric. The “subclass” relation of object-oriented programming is reflexive, transitive, antisymmetric, but not symmetric.

The relation “ x lives within 10 kilometers from y ” is reflexive, symmetric, but not transitive.

The relation “ x is a sibling of y ” is symmetric and transitive. It is not reflexive. None of these relations is total.

A total, transitive relation, is, for instance, “ x is not taller than y ”.

Definition 18 Given a binary relation $R \subset X \times X$, we call the smallest set R_r , such that $R \subset R_r$ and R_r is reflexive, the reflexive closure of R . The smallest set R_s , such that $R \subset R_s$ and R_s is symmetric, is called the symmetric closure of R . The smallest set R_t , such that $R \subset R_t$ and R_t is transitive, is called the transitive closure of R . Finally, the smallest equivalence relation R_e containing R is called the equivalence relation generated by R .

Proposition 25 If \sim is an equivalence relation on a , and if $s \in a$, then a subset

$$[s] = \{r \mid r \in a \text{ and } s \sim r\}$$

is called an equivalence class with respect to \sim . The set of equivalence classes—a subset of 2^a —is denoted by a/\sim . It defines a partition of a , i.e., for any two elements $s, t \in a$, either $[s] = [t]$ or $[s] \cap [t] = \emptyset$, and $a = \bigcup(a/\sim)$.

Proof Since for every $s \in a$, $s \sim s$, we have $s \in [s]$, whence $a = \bigcup(a/\sim)$. If $u \in [s] \cap [t]$, then if $r \sim s$ we have $r \sim s \sim u \sim t$, whence $[s] \subset [t]$, the converse inclusion holds with the roles of s and t interchanged, so $[s] = [t]$. \square

Example 12 For an equivalence relation, consider “ x has the same gender as y ”. The set of equivalence classes of this relation partitions mankind into two sets, the set of males and the set of females.

Exercise 11 Show that the reflexive, symmetric, transitive closure $((R_r)_s)_t$ of a relation R is the smallest equivalence relation R_e containing R . Hint:

First, show that $R_e \supset ((R_r)_s)_t$. Then, show that $((R_r)_s)_t$ is an equivalence relation. The only critical property is symmetry. Show that the transitive closure of a symmetric relation is symmetric.

Example 13 A relation $R \subset A \times A$, where $A = \{a, b, c, d, e\}$ is shown in figure 4.9 in a graphical way. An element $(x, y) \in R$ is represented by a point at the intersection of the vertical line through x and the horizontal line through y . For the reflexive, symmetric and transitive closure, added elements are shown in gray.

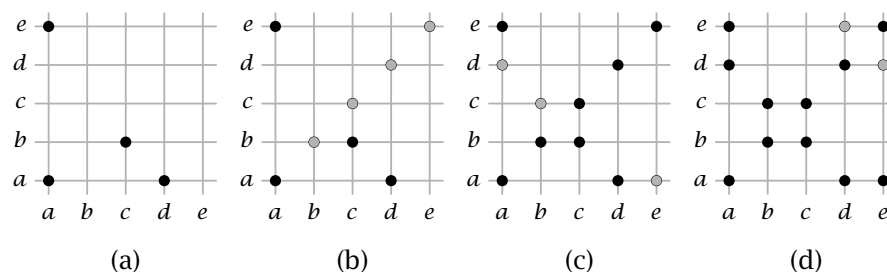


Fig. 4.9. A relation R (a), its reflexive closure R_r (b), the reflexive, symmetric closure $(R_r)_s$ (c), and the reflexive, symmetric, transitive closure $R_e = ((R_r)_s)_t$ (d).

Definition 19 A binary relation \leq on a set a is called a partial ordering iff it is reflexive, transitive and antisymmetric. A partial ordering is called linear iff it is total. A linear ordering is called a well-ordering iff every non-empty set $b \subset a$ has a minimal element m , i.e., $m \leq x$ for all $x \in b$.

Example 14 We will later see that the set of natural numbers $(0, 1, 2 \dots)$ and the set of integers $(\dots -2, -1, 0, 1, 2 \dots)$ are both linearly ordered by “ x is less than or equal to y ”. However, whereas the natural numbers are well-ordered with respect to this relation, the integers are not.

The aforementioned “subclass” relation is a partial ordering. It is not linear, because two classes can be completely unrelated, or derive from the same class in the hierarchy without one being a subclass of the other. Another example of a partial, but not linear, ordering is the inclusion relation on sets.

Lemma 26 *Let \leq be a binary relation on a set a . Denoting $x < y$ iff $x \leq y$ and $x \neq y$, the following two statements are equivalent:*

- (i) *The relation \leq is a partial ordering.*
- (ii) *The relation \leq is reflexive, the relation $<$ is transitive, and for all $x, y \in a$, $x < y$ excludes $y < x$.*

If these equivalent properties hold, we have $x \leq y$ iff $x = y$ or else $x < y$. In particular, if we are given $<$ with the properties (ii), and if we define $x \leq y$ by the preceding condition, then the latter relation is a partial ordering.

Proof (i) implies (ii): If \leq is a partial ordering, then it is reflexive by definition. The relations $a < b < c$ imply $a \leq c$, but $a = c$ is excluded since \leq is antisymmetric. The last statement is a consequence of the transitivity of $<$.

(ii) implies (i): \leq is reflexive by hypothesis. It is transitive, since $<$ is so, and the cases of equality are obvious. Finally, if $x < y$, then $y \leq x$ is impossible since equality is excluded by the exclusion of the simultaneous validity of $x < y$ and $y < x$, and inequality is by definition excluded by the same condition. \square

Definition 20 *If R is a relation on a , and if a' is any set, the induced relation $R|_{a'}$ is defined to be the relation $R \cap (a \cap a') \times (a \cap a')$ on $a \cap a'$.*

Exercise 12 Show that the induced relation $R|_{a'}$ is a partial ordering, a linear ordering, a well-ordering, if R is so.

Exercise 13 Given a relation R on a and a bijection $f : a \rightarrow b$, then we consider the image R_f of the induced bijection $(f \times f)|_R$ in $b \times b$. This new relation is called “structural transport” of R . Show that R_f inherits all properties of R , i.e., it is a partial ordering, a linear ordering, a well-ordering, iff R is so.

The strongest statement about relations on sets is this theorem (due to Ernst Zermelo):

Proposition 27 (Zermelo) *There is a well-ordering on every set.*

Proof We shall not prove this quite involved theorem, but see [46]. \square

Remark 5 If every set admits a well-ordering, the axiom of choice is a consequence hereof. Conversely, the proposition 27 of Zermelo is proved by use of the axiom 8 of choice. In other words: Zermelo’s theorem and the axiom of choice are equivalent.

Ordinal and Natural Numbers

Until now, our capabilities to produce concrete sets were quite limited. In particular, we were only capable of counting from zero to one: from the empty set $0 = \emptyset$ to the set $1 = \{0\}$. We are not even able to say something like: “For $n = 0, 1, \dots$ ”, since the dots have no sense up to now! This serious lack will be abolished in this chapter: We introduce the basic construction of natural numbers—together with one of the most powerful proof techniques in mathematics: proof by infinite induction.

5.1 Ordinal Numbers

We shall now construct the basic sets needed for every counting and number-theoretic task in mathematics (and all the sciences, which count on counting, be aware of that!)

Definition 21 A set a is called transitive if $x \in a$ implies $x \subset a$.

Example 15 The sets 0 and 1 are trivially transitive, and so is any set $J = \{J\}$ (if it exists).

Exercise 14 Show that if a and b are transitive, then so is $a \cap b$.

Definition 22 A set a is called alternative if for any two elements $x, y \in a$, either $x = y$, or $x \in y$, or $y \in x$.

Exercise 15 Show that, if a is alternative and $b \subset a$, then b is alternative.

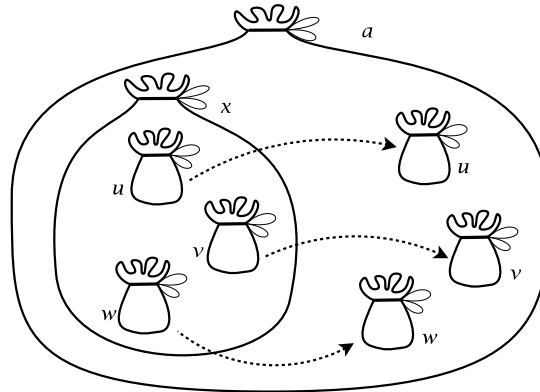


Fig. 5.1. Transitivity: Elements of an element $x \in a$ of a transitive set a are themselves elements of a .

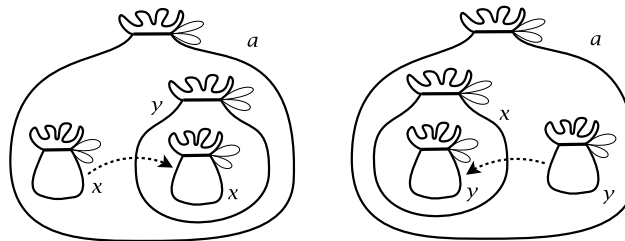


Fig. 5.2. Alternativity: In an alternative set a , if $x \neq y$ are two elements of a , then either $x \in y$ or $y \in x$.

Definition 23 A set a is called founded if for each non-empty $b \subset a$, there is $x \in b$ with $x \cap b = \emptyset$.

Example 16 The sets 0 and 1 are founded. What does it mean for a set not to be founded. Consider the negation of foundedness: If a set a is not founded, then there is “bad” non-empty subset $b \subset a$, such that for all $x \in b$, we have $x \cap b \neq \emptyset$, in other words: every element of b has an element, which already is in b . This means that there is an endless chain of elements of b , each being an element of the previous one. The simplest example is the “circular” set defined by the equation $J = \{J\}$ (if it exists). Here the chain is $J \ni J \ni J \dots$ The property of foundedness ensures that a set does not contain a bottomless pit.

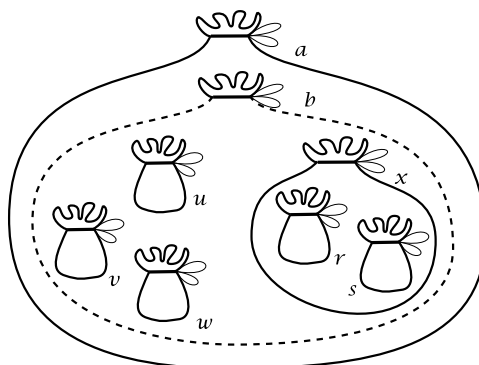


Fig. 5.3. Foundedness: If $b \subset a$ is non-empty, and a is founded, then there is $x \in b$ such that $x \cap b = \emptyset$; here we have $x = \{r, s\}$, $b = \{x, u, v, w\}$ with $r \neq u, v, w$ and $s \neq u, v, w$.

In modern computer science, however, such circular sets play an increasingly important role, a prominent example being the modeling of circular data structures, see [3].

The circular non-founded set is a special case of the following result:

Proposition 28 *Suppose that a is founded, then $x \in a$ implies $x \notin x$.*

Proof For $x \in a$, consider the subset $\{x\} \subset a$. Then $x \in x$ contradicts the fact that $\{x\} \cap x$ must be empty because a is founded. \square

Lemma 29 *If d is founded, and if $a, b, c \in d$ such that $a \in b$ and $b \in c$, then $a \neq c$ and $c \notin a$.*

Proof Consider the subset $\{a, b\} \in d$. If $x = b$, then $a \in x \cap \{a, b\}$. Therefore $x = a$ yields $x \cap \{a, b\} = \emptyset$. But $a = c$ enforces that b is in this intersection. Therefore $a \neq c$. If we had $c \in a$, then we cannot have $a \cap \{a, b, c\} = \emptyset$. By the hypothesis of the lemma, $b \cap \{a, b, c\}$ contains a , and $c \cap \{a, b, c\}$ contains b . Therefore the set $\{a, b, c\}$ contradicts the foundedness of d and so $c \notin a$. \square

Here is the fundamental concept for creating numbers:

Definition 24 *A set is called ordinal if it is transitive, alternative, and founded.*

The following series of results is destined to control the basic properties of ordinals.

Lemma 30 *Let d be an ordinal. If $a \subset d$ is non-empty, then there is $x_0 \in a$ such that whenever $x \in a$, then either $x_0 = x$ or $x_0 \in x$.*

Proof Since d is an ordinal, it is founded, and there is an element $x_0 \in a$ such that $x_0 \cap a = \emptyset$. Let $x \in a$ be any element different from x_0 . Since d is alternative, either $x \in x_0$ or $x_0 \in x$. But $x \in x_0$ contradicts $x_0 \cap a = \emptyset$, and we are done. \square

Proposition 31 *If d is ordinal, then $x \in d$ implies that x is ordinal.*

Proof Let $x \in d$. Then by transitivity of d , $x \subset d$. Hence x is alternative and founded. Let $b \in x$. Then $a \in b$ implies $a \in x$. In fact, $x \subset d$, therefore $b \in d$ and $a \in b \in d$. Since d is transitive, $a \in d$. So $a, x \in d$. Hence either $a = x$, or $a \in x$, or $x \in a$. But by lemma 29, applied to the element chain $a \in b \in x$, we must have $a \in x$. \square

Proposition 32 *A transitive proper subset c of an ordinal d is an element of that ordinal.*

Proof Since d is founded, there is $y \in d - c$ with $y \cap (d - c) = \emptyset$. We claim that $c = y$. Since d is transitive, $y \subset d$, and by construction of y , $y \subset c$. Conversely, let $b \in c$, then $b \in d$. Since d is alternative, either $b \in y$ or $b = y$ or $y \in b$. But $b = y$ implies $b \in d - c$, a contradiction. Further, $y \in b$ and $b \in c$ imply $y \in c$ by transitivity of c . Again, a contradiction. Therefore we have $b \in y$. \square

Corollary 33 *If d is an ordinal, then a set a is an element of d iff it is an ordinal and $a \subsetneq d$.*

Proof Clearly, an element of an ordinal is a proper subset. The converse follows from proposition 32. \square

Exercise 16 Show that if d is a non-empty ordinal, then $\emptyset \in d$.

Exercise 17 Show that if c is ordinal, then $a \in b$ and $b \in c$ imply $a \in c$.

Exercise 18 Show that if a and b are ordinals, then $a \in b$ implies $b \notin a$.

Proposition 34 *If a and b are ordinals, then either $a \subset b$ or $b \subset a$.*

Proof Suppose both conclusions are wrong. Then the intersection $a \cap b$ is a proper subset of both a and b . But it is evidently transitive, hence an ordinal element of both, a and b , hence an element of itself, a contradiction. \square

Corollary 35 *If a and b are ordinals, then exclusively either $a \in b$, or $a = b$, or $b \in a$.*

Proof Follows from propositions 32 and 34. \square

Corollary 36 *If all elements $a \in b$ are ordinals, then b is alternative.*

Proof Follows from corollary 35. \square

Proposition 37 *If every element $x \in u$ of a set u is ordinal, then there is exactly one element $x_0 \in u$ such that for every $x \in u$, we have either $x_0 \in x$ or $x_0 = x$.*

Proof Uniqueness is clear since all elements of u are ordinal.

Existence: If $u = \{a\}$, then take $x_0 = a$. Else, there are at least two different elements $c, c' \in u$, and either $c \in c'$ or $c' \in c$. So either $c \cap u$ or $c' \cap u$ is not empty. Suppose $c \cap u \neq \emptyset$. By lemma 30, there is an $x_0 \in c \cap u$ such that either $x_0 = x$ or $x_0 \in x$ for all $x \in c \cap u$. Take any $y \in u$. Then either: $c \in y$, but $x_0 \in c$, and therefore $x_0 \in y$. Or: $c = y$, whence $x_0 \in y$, or else $y \in c$, whence $y \in c \cap u$, and $x_0 = y$ or $x_0 \in y$ according to the construction of x_0 . \square

Corollary 38 *If all elements of a set u are ordinals, then u is founded.*

Proof Let $c \subset u$ be a non-empty subset. Take the element $x_0 \in c$ as guaranteed by proposition 37. Then clearly $x_0 \cap c = \emptyset$. \square

Corollary 39 *A transitive set a is ordinal iff all its elements are so.*

Proof Follows immediately from the corollaries 36 and 38. \square

Remark 6 There is no set *Allord* containing all ordinal sets. In fact, it would be transitive, and therefore ordinal, i.e., we would have the absurd situation $Allord \in Allord$.

Proposition 40 *For any set a , the successor set a^+ is non-empty, we have $a \in a^+$, and a is ordinal iff a^+ is so.*

Proof By definition, $a \in a^+$, thus a successor is never empty. If a^+ is ordinal, so is its element a . Conversely, all the elements of a^+ are ordinal. Moreover, a^+ is transitive, hence ordinal by corollary 39. \square

Lemma 41 *If a and b are ordinals with $a \in b$, then*

- (i) *either $a^+ \in b$ or $a^+ = b$;*
- (ii) *$a^+ \in b^+$;*
- (iii) *there is no x such that $a \in x$ and $x \in a^+$.*

Proof Since a^+ and b are ordinal, we have $a^+ \in b$ or $a^+ = b$ or $b \in a^+$. But the latter yields a contradiction to $a \in b$. The second statement follows from the first. The third follows from the two impossible alternatives $x = a$ or $x \in a$. \square

Proposition 42 *If a and b are ordinals, then $a = b$ iff $a^+ = b^+$.*

Proof Clearly, $a = b$ implies $a^+ = b^+$. Conversely, $a^+ = b^+$ implies $b = a$ or $b \in a$, but the latter implies $b^+ \in a^+$, a contradiction. \square

Corollary 43 *If two ordinals a and b are equipollent, then so are their successors.*

Proof We have a bijection $f : a \xrightarrow{\sim} b$. We define the following bijection $g : a^+ \xrightarrow{\sim} b^+$: On elements x of a , $g(x) = f(x)$. For $a \in a^+$ and $b \in b^+$, we know that $a \notin a$ and $b \notin b$, therefore we set $g(a) = b$. \square

Proposition 44 *Let Φ be an attribute of sets such that whenever it holds for all elements $x \in a$ of an ordinal a , then it also holds for a . Then Φ holds for all ordinals.*

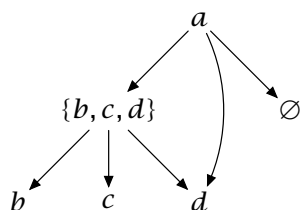
Proof Observe that in particular, Φ holds for \emptyset . Suppose that there is an ordinal b such that $\Phi(b)$ does not hold. Then the subset $b' = \{x \mid x \in b, \text{NOT } \Phi(x)\}$ of b is not empty (since $\text{NOT } \Phi(b)$) and a proper subset of b (since \emptyset is not in b'). According to proposition 37, there is a minimal element $x_0 \in b'$, i.e., $\text{NOT } \Phi(x_0)$, but every element of x_0 is element of $b - b'$. This is a contradiction to the hypothesis about Φ . \square

5.2 Natural Numbers

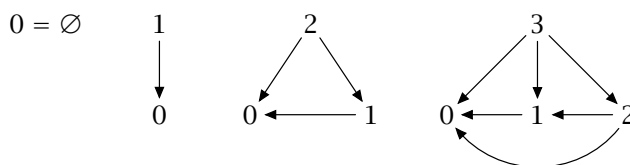
The natural numbers, known from school in the common writing $(0,) 1, 2, 3, 4, 5, \dots$ are constructed from the ordinal sets as follows. This construction traditionally stems from an axiomatic setup, as proposed by Giuseppe Peano. In the set-theoretical framework, the Peano axioms appear as propositional statements. This is why proposition 45 and proposition 46 are named "Peano Axioms".

Definition 25 *A natural number is an ordinal set n which is either \emptyset or a successor m^+ of an ordinal number m and such that every element x of n is either $x = \emptyset$ or a successor $x = y^+$ of an ordinal number y .*

The membership relation of sets can be illustrated by a diagram where an arrow from a set x to a set y means that x contains y as an element, or $y \in x$. The following picture, for example, represents the set $a = \{\{b, c, d\}, d, \emptyset\}$:



Using this method, the natural numbers $0 = \emptyset$, $1 = \{\emptyset\}$, $2 = \{\emptyset, \{\emptyset\}\}$, $3 = \{\emptyset, \{\emptyset\}, \{\emptyset, \{\emptyset\}\}\}$, ... can be drawn:



Proposition 45 (Peano Axioms 1 through 4) We denote the empty set by the symbol 0 if we consider it as an ordinal.

- (i) The empty set 0 is a natural number.
- (ii) If n is natural, then so is n^+ .
- (iii) The empty set 0 is not a successor.
- (iv) If n and m are natural, then $n^+ = m^+$ implies $n = m$.

Proof This is straightforward from the propositions following the definition, lemma 1 in chapter 2, of a successor. \square

Proposition 46 (Peano Axiom 5) Let Ψ be an attribute of sets with these properties:

- (i) We have $\Psi(0)$.
- (ii) For every natural number n , $\Psi(n)$ implies $\Psi(n^+)$.

Then Ψ holds for every natural number.

Proof Let m be a natural number with NOT $\Psi(m)$. Since $m \neq 0$, i.e., $m = n^+$, by hypothesis, the subset $\{x \mid x \in m, \text{NOT } \Psi(x)\}$ contains n . Take the minimum x_0 within this subset. This is a successor $x_0 = y^+$ such that $\Psi(y)$, a contradiction. \square

Remark 7 Proposition 46 yields a proof scheme called “proof by induction”. One is given an attribute Ψ of sets, usually only specified for natural numbers. For other sets, we tacitly set Ψ to be false, but that value is irrelevant for the given objective, which is to prove that Ψ holds for all natural numbers. To this end, one proves (i) and (ii) in proposition 46 and concludes as required.

The following proposition is the first result which follows from proposition 46, Peano's fifth axiom:

Proposition 47 *Let n be natural, and let $a \subsetneq n$. Then there is $m \in n$ such that $a \sim m$.*

Proof We prove the proposition by induction: For any natural number n , let $\Psi(n)$ be the proposition that for every proper subset $y \subset n$, there is a natural m , which is equipollent to y . Clearly, $\Psi(0)$. Suppose that $n = m^+$ and $\Psi(m)$ holds. Let y be a proper subset of n . If $y \subset m$, either $y = m$ and we are done, or y is a proper subset of m and the induction hypothesis gives us the required bijection. Else, if $m \in y$, there is an element $u \in m - y$. But then $y' = (y - \{m\}) \cup \{u\}$ is equipollent to y , and we are redirected to the above case. \square

Proposition 48 *Let n be a natural number, and suppose that an ordinal a is equipollent to n . Then $a = n$. In particular, two natural numbers m and n are equipollent iff they are equal.*

Proof We prove the claim by induction and consider the property $\Psi(n)$ for natural numbers defined by $\Psi(n)$ iff for any ordinal a , the fact that a is equipollent to n implies $a = n$. Suppose that there is a counterexample m ; evidently $m \neq 0$. Let $a \neq m$ be ordinal, but equipollent to m . Let $\bar{m} = \{x \mid x \in m, \Psi(x)\}$, which is not empty, since it contains 0.

1. $\bar{m} = m$. Since a and m are ordinal, either $a \in m$ or $m \in a$. In the first case, $\Psi(a)$, which contradicts the choice of m . So $m \in a$ and $m \subset a$ is a proper subset. Let $f : a \rightarrow m$ be a bijection. Then $f(m)$ is a proper subset of m . By proposition 47, there is an element $n \in m$ which is equipollent to $f(m)$ and therefore also to m . But we know that $\Psi(n)$, whence $m = n$, which contradicts $n \in m$.

2. $m - \bar{m} \neq \emptyset$. Take the smallest x_0 in this difference set. There is an ordinal $b \neq x_0$, but equipollent to x_0 . Then either $b \in x_0$ or $x_0 \in b$. In the first case, b is a natural number in m since $x_0 \subset m$. So $b = x_0$ by construction of x_0 , a contradiction. If $x_0 \in b$, then since b is ordinal, $x_0 \subset b$ is a proper subset, and we may proceed as in the first case above. This concludes the proof. \square

Remark 8 This means that natural numbers describe in a unique way certain cardinalities of ordinals. Each natural number represents exactly one cardinality, and two different natural numbers are never equipollent.

Definition 26 A set a is called finite if it is equipollent to a natural number. This number is then called cardinality of a and denoted by $\text{card}(a)$, by $\#(a)$, or by $|a|$, depending on the usage.

This definition is justified by the fact that the cardinality of a finite set is a unique number which is equipollent to that set, i.e., which in our previous terminology has the same cardinality as the given set.

Corollary 49 A subset of a finite set is finite.

Proof We may suppose that we are dealing with a subset of a natural number, where the claim is clear. \square

Corollary 50 An ordinal set is finite iff it is a natural number.

Proof This follows right from proposition 48. \square

Corollary 51 If an ordinal a is not finite, it contains all natural numbers.

Proof For any natural number n , we have either $a \in n$ or $a = n$ or $n \in a$. The middle case is excluded by definition of a . The left alternative is excluded since every subset of a finite set is finite. So every n is an element of a . \square

Corollary 52 A finite set is not equipollent to a proper subset.

Proof In fact, a proper subset of a natural number n is equipollent to an element of n , hence equal to this element, a contradiction. \square

From the axiom 7 of infinity, we derive this result:

Proposition 53 There is an ordinal \mathbb{N} whose elements are precisely the natural numbers.

Proof This follows immediately from the axiom 7. \square

Notation 5 The relation $n \in m$ for elements of \mathbb{N} defines a well-ordering among natural numbers. We denote it by $n < m$ and say “ n is smaller than m ” or else “ m is larger than n ”.

Exercise 19 With the well-ordering \leq among natural numbers defined by $<$, show that we have the following facts: Every non-empty set of natural numbers has a (uniquely determined) minimal element. Let b be a limited non-empty set of natural numbers, i.e., there is $x \in \mathbb{N}$ such that $y < x$ for all $y \in b$. Then there is a (uniquely determined) maximal element $z \in b$, i.e., $y \leq z$ for all $y \in b$. Hint: Proceed by induction on b .

Recursion Theorem and Universal Properties

Before developing the arithmetic of natural numbers more specifically, we describe some crucial methods for the construction of sets and functions on sets. These properties are the basis of a fundamental branch in mathematics, called topos theory, a branch which is of great importance to computer science too, since it provides a marriage of formal logic and geometry (see [36]).

Proposition 54 *If a and b are two sets, then there is a set, denoted by $Set(a, b)$, whose elements are exactly the functions $f : a \rightarrow b$.*

Proof The elements of the required set $Set(a, b)$ are triples (a, f, b) , where $f \subset a \times b$ is a functional graph. So $(a, f, b) = ((a, f), b) \in (2^a \times 2^{a \times b}) \times 2^b$, whence $Set(a, b) \subset (2^a \times 2^{a \times b}) \times 2^b$ is a subset selected by a straightforward propositional attribute. \square

Notation 6 *The set $Set(a, b)$ of functions is also denoted by b^a if we want to stress that it is a set, without emphasis on the specific nature, i.e., that its elements are the functions $f : a \rightarrow b$. This distinction may seem superfluous now, but we will understand this point when we will deal with more general systems of objects and “functions” between such objects.*

Example 17 Setting $a = 0 (= \emptyset)$, we have $Set(0, b) = \{! : \emptyset \rightarrow b\}$. Setting $b = 1 (= \{0\})$, we have $Set(a, 1) = \{! : a \rightarrow 1\}$. If $a \neq 0$, then $Set(a, 0) = \emptyset$.

6.1 Recursion Theorem

The set of functions allows a very important application of the fifth Peano axiom (which is a theorem following from our set of axioms), namely construction by induction. To this end, we need to look at functions $f : \mathbb{N} \rightarrow a$ for any non-empty set a . If $n \in \mathbb{N}$, we have the restriction $f|_n : n \rightarrow a$ as declared in definition 13. Observe that $f|_n$ is defined for all natural numbers strictly smaller than n , but not for n . If $g : n^+ \rightarrow a$ is a function, then we denote by g^* the function $g^* : \mathbb{N} \rightarrow a$ with $g^*(m) = g(n)$ for $m > n$ and $g^*(m) = g(m)$ for $m \leq n$. If g is $! : 0 \rightarrow a$, we pick an element $g_0 \in a$ and set $g^*(m) = g_0$ for all natural numbers m . Here is the general recursion theorem:

Proposition 55 (Recursion Theorem) *Let a be a set, and let $\Phi : a^{\mathbb{N}} \rightarrow a^{\mathbb{N}}$ be a function such that for every natural number n , if $f, g \in a^{\mathbb{N}}$ are such that $f|_n = g|_n$, then $\Phi(f)(n) = \Phi(g)(n)$. Then Φ has a unique fixpoint $L_\Phi \in a^{\mathbb{N}}$, which means that $\Phi(L_\Phi) = L_\Phi$. Consider the function $\Phi_n : a^n \rightarrow a^n$ which evaluates to $\Phi_n(g) = \Phi(g^*)(n)$. Then we have*

$$\begin{aligned} L_\Phi(0) &= \Phi_0(! : 0 \rightarrow a) \\ L_\Phi(n^+) &= \Phi_{n^+}(L_\Phi|_{n^+}). \end{aligned}$$

Proof There is at most one such fixpoint. In fact, let L and M be two such fixpoints, $\Phi(L) = L$ and $\Phi(M) = M$, and suppose that they are different. Then there is a smallest value n_0 such that $L(n_0) \neq M(n_0)$. This means that $L|_{n_0} = M|_{n_0}$. But then $\Phi(L)(n_0) = \Phi(M)(n_0)$, a contradiction. So there is at most one such fixpoint. For every $n \in \mathbb{N}$, let $S(n) \subset a^n$ be the set of those functions $f : n \rightarrow a$ such that for all $m \in n$, $f(m) = \Phi_m(f|_m)$. Clearly, either $S(n)$ is empty or $S(n)$ contains precisely one function g_n . The set $S(0)$ is not empty. Let N^+ be the smallest natural number such that $S(N^+)$ is empty. We may define a function $h : N^+ \rightarrow a$ by $h|_N = g_N$ and $h(N) = \Phi_N(h|_N)$. But this is a function in $S(N^+)$, so every $S(n)$ is non-empty. Now define $L(n) = g_{n^+}(n)$. Clearly, this function is our candidate for a fixpoint: To begin with, if $n < m$, then evidently, by the uniqueness of the elements of $S(n)$, $g(m)|_n = g(n)$. Therefore, $L|_n = g_n$ for all n . And L is a fixpoint, in fact: $L(n) = g_{n^+}(n) = \Phi_n(g_{n^+}|_n) = \Phi_n(g_n) = \Phi(g_n^*)(n) = \Phi(L)(n)$ since $L|_n = g_n = g_n^*|_n$. The claimed formula then follows by construction. \square

Remark 9 Very often, the above formal context will be treated in a rather sloppy way. The common wording is this: One wants to define objects (functions, sets of whatever nature) by the following data: one knows

which object O_0 you have to start with, i.e., the natural number 0. Then you suppose that you have already “constructed” the objects O_m , $m \leq n$ for a natural number n , and your construction of O_{n^+} for the successor n^+ is given from O_0, O_1, \dots, O_n and some “formula” Φ . Then you have defined the objects O_n for every $n \in \mathbb{N}$.

Example 18 In order to clarify the rather abstract recursion theorem, we shall apply it to the definition of the function $c+?: \mathbb{N} \rightarrow \mathbb{N}$ which will be discussed in depth in the next chapter. Intuitively, we want this function to behave as follows:

$$\begin{aligned} c + 0 &= c \\ c + b^+ &= (c + b)^+ \end{aligned}$$

This corresponds to a function $\Phi : \mathbb{N}^{\mathbb{N}} \rightarrow \mathbb{N}^{\mathbb{N}}$ given by $\Phi(f)(0) = c$, and $\Phi(f)(n^+) = (f(n))^+$. This Φ satisfies the condition for the recursion theorem: Let $f, g \in \mathbb{N}^{\mathbb{N}}$ be two functions with $f|_{n^+} = g|_{n^+}$. In particular, this implies that $f(n) = g(n)$. Then, $\Phi(f)(0) = c = \Phi(g)(0)$, and $\Phi(f)(n^+) = (f(n))^+ = (g(n))^+ = \Phi(g)(n^+)$. The recursion theorem now states that Φ has a fixpoint $L_\Phi \in \mathbb{N}^{\mathbb{N}}$ with the property that

$$L_\Phi(0) = \Phi_0(! : 0 \rightarrow \mathbb{N}) = \Phi(! : 0 \rightarrow \mathbb{N})^*(0) = c,$$

and

$$\begin{aligned} L_\Phi(n^+) &= \Phi_{n^+}(L_\Phi|_{n^+}) \\ &= \Phi((L_\Phi|_{n^+})^*)(n^+) \\ &= ((L_\Phi|_{n^+})^*(n))^+ \\ &= (L_\Phi|_{n^+}(n))^+ \\ &= L_\Phi(n)^+. \end{aligned}$$

This is exactly the behavior we requested for our function $c+?$.

Exercise 20 Assuming the availability of the multiplication of natural numbers (which will be introduced in the next chapter), let the factorial function $fact : \mathbb{N} \rightarrow \mathbb{N}$ be given by

$$fact(0) = 1 \quad \text{and} \quad fact(n^+) = n^+ \cdot fact(n).$$

Explicitly write down the function Φ corresponding to this recursive definition of $fact$, show that Φ satisfies the condition for the application of the recursion theorem, and show that the function L_Φ is equal to the function $fact$.

6.2 Universal Properties

While the recursion theorem describes a constructive method for the definition of new sets, universal properties comprise a more declarative methodology for the construction of sets, which are uniquely characterized by certain functions on such sets. Such sets are universal in the sense that their concrete construction is secondary compared with their characteristic behavior.

Definition 27 A set b is called *final* iff for every set a , $|\text{Set}(a, b)| = 1$. A set a is called *initial* iff for every set b , $|\text{Set}(a, b)| = 1$.

Proposition 56 (Existence of Final and Initial Sets) A set b is final iff $|b| = 1$. The only initial set is \emptyset .

Proof This is immediate. □

We shall usually pick the set $1 = \{\emptyset\}$ to represent the final sets.

Notation 7 Given two sets a and b , denote by $pr_a : a \times b \rightarrow a$ and $pr_b : a \times b \rightarrow b$ the functions $pr_a(x, y) = x$ and $pr_b(x, y) = y$ for elements $(x, y) \in a \times b$. The functions pr_a, pr_b are called *projections onto a, b* , respectively.

Proposition 57 (Universal Property of Cartesian Product) Given two sets a and b and any set c , the function

$$\beta : \text{Set}(c, a \times b) \xrightarrow{\sim} \text{Set}(c, a) \times \text{Set}(c, b)$$

defined by $\beta(u) = (pr_a \circ u, pr_b \circ u)$ is a bijection.

The following commutative diagram shows the situation:

$$\begin{array}{ccccc}
 & & c & & \\
 & \swarrow & \downarrow & \searrow & \\
 & pr_a \circ u & u & pr_b \circ u & \\
 & \swarrow & \downarrow & \searrow & \\
 a & \longleftarrow & a \times b & \longrightarrow & b \\
 & pr_a & & pr_b &
 \end{array}$$

Proof If $u : c \rightarrow a \times b$, then for every $x \in c$, $u(x) = (pr_a(u(x)), pr_b(u(x)))$, so u is determined by its projections, and thus β is injective. If we are given $v : c \rightarrow a$ and $w : c \rightarrow b$, then define $u(x) = (v(x), w(x))$. Then evidently, $\beta(u) = (v, w)$, so β is surjective. □

Exercise 21 Suppose that a set q , together with two functions $p_a : q \rightarrow a$ and $p_b : q \rightarrow b$ has the property that

$$\beta : \text{Set}(c, q) \xrightarrow{\sim} \text{Set}(c, a) \times \text{Set}(c, b)$$

defined by $\beta(u) = (p_a \circ u, p_b \circ u)$ is a bijection. Show that there is a unique bijection $i : q \xrightarrow{\sim} a \sqcup b$ such that $pr_a \circ i = p_a$ and $pr_b \circ i = p_b$.

Definition 28 Given two sets a and b , the disjoint sum or coproduct $a \sqcup b$ of a and b is the set $a \sqcup b = (\{0\} \times a) \cup (\{1\} \times b)$, together with the injections $in_a : a \rightarrow a \sqcup b$, and $in_b : b \rightarrow a \sqcup b$, where $in_a(x) = (0, x)$ and $in_b(y) = (1, y)$ for all $x \in a$ and $y \in b$.

Evidently, the coproduct $a \sqcup b$ is the disjoint union of the two subsets $\{0\} \times a$ and $\{1\} \times b$. Here is the universal property for the coproduct corresponding to the universal property of the Cartesian product proved in proposition 57:

Proposition 58 (Universal Property of Coproduct) Given two sets a and b and any set c , the function

$$\gamma : \text{Set}(a \sqcup b, c) \xrightarrow{\sim} \text{Set}(a, c) \times \text{Set}(b, c)$$

defined by $\gamma(u) = (u \circ in_a, u \circ in_b)$ is a bijection.

The following commutative diagram shows the situation:

$$\begin{array}{ccccc}
 & & c & & \\
 & \nearrow & \uparrow & \nwarrow & \\
 a & & a \sqcup b & & b \\
 \xrightarrow{in_a} & & & & \xleftarrow{in_b}
 \end{array}$$

Proof Clearly, a map $u : a \sqcup b \rightarrow c$ is determined by its restrictions to its partitioning subsets $\{0\} \times a$ and $\{1\} \times b$, which in turn is equivalent to the pair $u \circ in_a$ and $u \circ in_b$ of maps. So γ is injective. Conversely, if $v : a \rightarrow c$ and $w : b \rightarrow c$ are any two functions, then we define $u((0, x)) = v(x)$ and $u((1, y)) = w(y)$, which shows the surjectivity of γ . \square

Exercise 22 Suppose that a set q , together with two functions $i_a : a \rightarrow q$ and $i_b : b \rightarrow q$ has the property that

$$\gamma : \text{Set}(q, c) \xrightarrow{\sim} \text{Set}(a, c) \times \text{Set}(b, c)$$

defined by $\gamma(u) = (u \circ i_a, u \circ i_b)$ is a bijection. Show that there is a unique bijection $i : a \sqcup b \xrightarrow{\sim} q$ such that $i \circ in_a = i_a$, $i \circ in_b = i_b$.

Exercise 23 Use the universal property of coproducts to show that, for three sets a , b and c , the coproducts $(a \sqcup b) \sqcup c$ and $a \sqcup (b \sqcup c)$ are equipollent. Therefore we can write $a \sqcup b \sqcup c$.¹

The following proposition characterizes the set of functions c^b as the solution to a property of functions defined on a Cartesian product of sets.

Proposition 59 (Universal Property of Exponentials) *If a , b and c are sets, there is a bijection*

$$\delta : \text{Set}(a \times b, c) \xrightarrow{\sim} \text{Set}(a, c^b)$$

defined by

$$\delta(f)(\alpha)(\beta) = f((\alpha, \beta))$$

for all $\alpha \in a$ and $\beta \in b$, and $f \in \text{Set}(a \times b, c)$. This bijection is called the natural adjunction.

Proof The map δ is evidently injective. On the other hand, if $g : a \rightarrow c^b$, then we have the map $f : a \times b \rightarrow c$ defined by $f(\alpha, \beta) = g(\alpha)(\beta)$, and then $\delta(f) = g$, thus δ is surjective. \square

For the next concepts we need to know what the fiber of a function is:

Definition 29 *If $f : a \rightarrow b$ is a function, and if $c \subset b$, then one calls the set $\{x \mid x \in a \text{ and } f(x) \in c\}$ “fiber of f over c ”, or preimage or inverse image of c under f , and denotes it by $f^{-1}(c)$. For a singleton $c = \{\kappa\}$, one writes $f^{-1}(\kappa)$ instead of $f^{-1}(\{\kappa\})$.*

We have this commutative diagram, where the horizontal arrows are the inclusions:

$$\begin{array}{ccc} f^{-1}(c) & \longrightarrow & a \\ \downarrow f|_{f^{-1}(c)} & & \downarrow f \\ c & \longrightarrow & b \end{array}$$

In axiom 6 we had introduced the somewhat strange notation 2^a for the powerset of a . Here is the explanation for this choice.

¹ This corresponds to the procedure known as “currying” in λ -calculus.

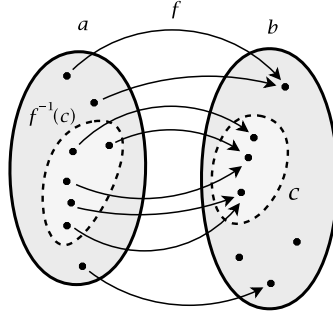


Fig. 6.1. The fiber $f^{-1}(c) \subset a$ of f over $c \subset b$.

Proposition 60 (Subobject Classifier) *The natural number $2 = \{0, 1\}$ is a subobject classifier, i.e., for every set a , there is a bijection*

$$\chi : 2^a \xrightarrow{\sim} \text{Set}(a, 2)$$

defined in the following way:

If $b \subset a$ is an element of 2^a , then $\chi(b)(\alpha) = 0$ if $\alpha \in b$, and $\chi(b)(\alpha) = 1$ else. The function $\chi(b)$ is called the characteristic function of b . The inverse of χ is the zero fiber, i.e., $\chi^{-1}(f) = f^{-1}(0)$.

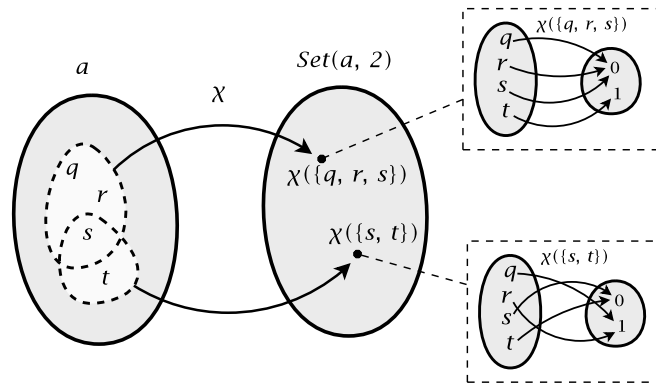


Fig. 6.2. Subobject classifier: The values of χ for just two elements are shown. For example, the subset $\{q, r, s\}$ of $a = \{q, r, s, t\}$ is mapped by χ to its characteristic function $\chi(\{q, r, s\})$ illustrated in the upper right of the figure. Note that $\chi^{-1}(\{q, r, s\}) = \{q, r, s\} = \chi(q, r, s)^{-1}(0)$, i.e., the zero fiber of the characteristic function.

Proof This result is immediate. \square

Observe that now the subsets of a , elements of 2^a , are identified with functions $a \rightarrow 2$, i.e., elements of $\text{Set}(a, 2)$ which we also denote by 2^a , and this is now perfectly legitimate by the above proposition.

A generalization of the Cartesian product is given by so-called families of sets.

Definition 30 A family of sets is a surjective function $f : a \rightarrow b$. The images $f(x)$ are also denoted by f_x , and the function is also notated by $(f_x)_{x \in a}$ or by $(f_x)_a$. This means that the elements of b are “indexed” by elements of a .

If $c \subset a$, then the subfamily $(f_x)_{x \in c}$ is just the restriction $f|_c$, together with the codomain being the image $\text{Im}(f|_c)$.

The Cartesian product $\prod_{x \in a} f_x$ of a family $(f_x)_{x \in a}$ of sets is the subset of $(\bigcup b)^a$ consisting of all functions $t : a \rightarrow \bigcup b$ such that $t(x) \in f_x$ for all $x \in a$. Such a function is also denoted by $(t_x)_{x \in a}$ and is called a family of elements. We shall always assume that when a family of elements is given, that there is an evident family of sets backing this family of elements, often without mentioning these sets explicitly.

For a given index x_0 , we have the x_0 -th projection $pr_{x_0} : \prod_{x \in a} f_x \rightarrow f_{x_0}$ which sends $(t_x)_{x \in a}$ to t_{x_0} .

Example 19 Figure 6.3 shows a family of sets $f : a \rightarrow b$, with $a = \{x, y\}$ and $b = \{\{q, r, s\}, \{s, t\}\}$. It is defined as $f_x = \{q, r, s\}$ and $f_y = \{s, t\}$. The Cartesian product $\prod_{x \in a} f_x$ is given by the functions $t_i : a \rightarrow \{q, r, s, t\}$ with $t_1 = \{(x, q), (y, s)\}$, $t_2 = \{(x, r), (y, s)\}$, $t_3 = \{(x, s), (y, s)\}$, $t_4 = \{(x, q), (y, t)\}$, $t_5 = \{(x, r), (y, t)\}$, $t_6 = \{(x, s), (y, t)\}$, where the graphs of the functions have been used to describe them.

One defines the *union of the family* $(f_x)_a$ by $\bigcup_a f_x = \bigcup \text{Im}(f)$, i.e., by the union of all its member sets. Similarly one defines the *intersection of the family* $(f_x)_a$ by $\bigcap_a f_x = \bigcap \text{Im}(f)$, i.e., by the intersection of all its member sets. The intersection however exists only for a non-empty family.

Sortite 61 Let $(f_x)_a$ be a non-empty family of sets, i.e., $a \neq \emptyset$.

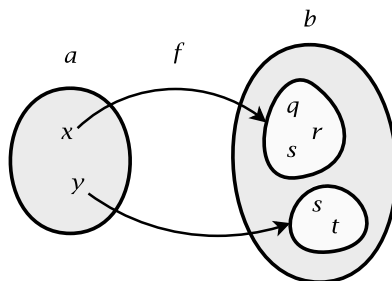


Fig. 6.3. A family of sets $f : a \rightarrow b$.

- (i) The Cartesian product $\prod_{x \in a} f_x$ is non-empty iff each f_x is non-empty.
- (ii) If all sets f_x coincide and are equal to c , then $\prod_{x \in a} f_x = c^a$.
- (iii) If $a = 2$, then $\prod_{x \in 2} f_x \cong f_0 \times f_1$.
- (iv) If $(u_x : d \rightarrow f_x)_{x \in a}$ is a family of functions, then there is a unique function $u : d \rightarrow \prod_{x \in a} f_x$ such that $u_x = pr_x \circ u$ for all $x \in a$.

Proof (i) By definition, the Cartesian product is non-empty for $a \neq \emptyset$ iff each f_x is so.

(ii) This is an immediate consequence of the definition of the Cartesian product and of the fact that here, $c = \bigcup_{x \in a} c$.

(iii) The functions $g : 2 \rightarrow f_0 \cup f_1$, where $g(0) \in f_0$ and $g(1) \in f_1$, are in bijection with the pairs of their evaluations $(g(0), g(1)) \in f_0 \times f_1$.

(iv) The reader should think about the set where the family of functions is deduced. But if this is done, the statement is immediate. \square

Example 20 Families where the index set a is a natural number n or the set \mathbb{N} of all natural numbers are called *sequences*. For $a = n$, we have the *finite* sequences $(t_i)_{i \in n}$, or, in an equivalent notation, $(t_i)_{i < n}$. One then often writes $t_i, i = 0, 1, \dots, n-1$ instead, or also t_0, t_1, \dots, t_{n-1} . For $a = \mathbb{N}$, one writes also $(t_i)_{i=0,1,2,\dots}$ or else t_0, t_1, \dots . The *length of a sequence* $(t_i)_{i \in n}$ is the (uniquely determined) number n . One also calls such a sequence an *n-tuple*.

In computer science, one often calls sequences *lists*, and it is also agreed that list indexes start with 0, as do natural number indexes. The *empty list* is also that sequence with index set $a = 0$, the empty set.

Exercise 24 Prove that there is exactly one empty family, i.e., a family with an empty index set.

Cartesian products $\prod_{x \in a} f_x$ also admit linear orderings if their members do so. Here is the precise definition of the so-called “lexicographic ordering”:

Definition 31 Suppose that we are given a family $(f_x)_a$ of sets such that each f_x bears a linear ordering $<_x$, and such that the index set a is well-ordered by the relation $<$. Then, for two different families $(t_x)_a, (s_x)_a \in \prod_{x \in a} f_x$ the relation

$$(t_x)_a < (s_x)_a \text{ iff the smallest index } y, \text{ where } t_y \neq s_y, \text{ has } t_y <_y s_y$$

is called the lexicographic ordering on $\prod_{x \in a} f_x$.

Lemma 62 The lexicographic ordering is a linear ordering.

Proof According to lemma 26, we show that $<$ is transitive, antisymmetric and total. Let $(t_x)_a < (s_x)_a < (u_x)_a$. If the smallest index y , where these three families differ, is the same, then transitivity follows from transitivity of the total ordering at this index. Else, one of the two smallest indexes is smaller than the other, let $y_1 < y_2$ for the index y_1 of the left pair $(t_x)_a < (s_x)_a$. Then the inequalities at this index are $t_{y_1} <_{y_1} s_{y_1} = u_{y_1}$, whence $t_{y_1} <_{y_1} u_{y_1}$, i.e., $(t_x)_a < (u_x)_a$; similarly for the other situation, namely, $y_2 < y_1$. The same argument works for antisymmetry. As to totality: Let $(t_x)_a$ and $(s_x)_a$ be any two families. If they are different, then the smallest index y where they differ has either $t_y <_y s_y$ or $s_y <_y t_y$ since $<_y$ is total. \square

Exercise 25 Show that the lexicographic ordering on $\prod_{n \in \mathbb{N}} f_n$ is a well-ordering iff each linear ordering $<_n$ on f_n is so. The same is true for a finite sequence of sets, i.e., for $\prod_{n < N} f_n$, where $N \in \mathbb{N}$.

Exercise 26 Suppose that we are given a finite alphabet set \mathcal{A} of “letters”. Suppose that a bijection $u : \mathcal{A} \xrightarrow{\sim} N$ with the natural number $N = \text{card}(\mathcal{A})$ is fixed, and consider the ordering of letters induced by this bijection, i.e., $X < Y$ iff $u(X) < u(Y)$. Suppose that an element $\sqsubset \in \mathcal{A}$ is selected. Consider now the restriction of the lexicographic ordering on $\mathcal{A}^{\mathbb{N}}$ to the subset $\mathcal{A}^{(\mathbb{N})}$ consisting of all sequences $(\tau_n)_{\mathbb{N}}$ such that $\tau_n = \sqsubset$ for all but a finite number of indexes. Show that this set may be identified with the set of all finite words in the given alphabet. Show that the

induced lexicographic ordering on $\mathcal{A}^{(\mathbb{N})}$ coincides with the usual lexicographic ordering in a dictionary of words from the alphabet \mathcal{A} ; here the special sign $_$ plays the role of the empty space unit.

The name “lexicographic” effectively originates in the use of such an ordering in compiling dictionaries. As an example we may consider words of length 4, i.e., the set \mathcal{A}^4 . Let the ordering on \mathcal{A} be $_ < A < B \dots Z$. Then, writing the sequence $(t_i)_{i=0,1,2,3}$, $t_i \in \mathcal{A}$ as $t_0 t_1 t_2 t_3$, we have, for instance:

BALD $<$ BAR $_$ $<$ BASH $<$ I $_$ AM $<$ MAN $_$ $<$ MANE $<$ MAT $_$ $<$ SO $_$ $_$ $<$ SORE

The minimal element of \mathcal{A}^4 is $_ _ _ _$, the maximal element is ZZZZ.

Definition 32 *If $(f_x)_a$ is a family of sets, where each set f_x bears a binary relation R_x , then the Cartesian product $\prod_{x \in a} f_x$ bears the product relation $R = \prod_{x \in a} R_x$ which is defined “coordinatewise”, i.e.,*

$$(t_x)_a R (s_x)_a \text{ iff } t_x R_x s_x \text{ for each } x \in a.$$

Attention: Even if each binary relation on the set f_x is a linear ordering. Therefore *product relation* is not, in general, a linear ordering, so the lexicographic ordering is a remarkable construction since it “preserves” linear orderings.

Until now, we only considered binary relations. By use of Cartesian products of families of sets, one can now introduce the concept of an n -ary relation for $n \in \mathbb{N}$ as follows:

Definition 33 *If n is a natural number and a is a set, an n -ary relation on a is a subset R of the n -fold Cartesian product a^n , the binary relation being the special case of $n = 2$.*

Not every binary relation is an equivalence relation, but very often, one is interested in a kind of minimal equivalence relation which contains a given relation. Here is the precise setup:

Lemma 63 *If $(R_x)_a$ is a non-empty family of equivalence relations on a set b , then the intersection $\bigcap_a R_x$ is an equivalence relation. It is the largest equivalence relation (for the subset inclusion relation), which is contained in all relations R_x , $x \in a$.*

Proof This is straightforward to check. □

Proposition 64 *Given a relation R on a set a , the smallest equivalence relation \sim containing R consists of all pairs (x, y) such that either $x = y$ or there exists a finite sequence $x = x_0, x_1, \dots, x_n = y$ with $x_i R x_{i+1}$ or $x_{i+1} R x_i$ for all $i = 0, 1, \dots, n$.*

Proof Clearly, the smallest equivalence relation must contain these pairs. But these pairs visibly define an equivalence relation, and we are done. \square

Definition 34 *The equivalence relation \sim defined in proposition 64 is called the equivalence relation generated by the relation R .*

6.3 Universal Properties in Relational Database Theory

Relational database theory serves as a very useful example of the universal properties of Cartesian product constructions. More than that: It even requires a construction which is slightly more general than the Cartesian product: the fiber product. Let us first introduce it, before we discuss a concrete database process implemented in the relational database management system language SQL (Structured Query Language). SQL is an ANSI (American National Standards Institute) standard computer language for accessing and manipulating database systems. SQL statements are used to retrieve and update data in a database (see [24] for a reference to SQL and [18] for relational database theory).

Definition 35 (Universal property of fiber products) *Given three sets a , b and c and two maps $f : a \rightarrow c$ and $g : b \rightarrow c$, a couple of maps $s_a : d \rightarrow a$ and $s_b : d \rightarrow b$ is called a fiber product, or pullback, with respect to f and g iff $f \circ s_a = g \circ s_b$, and if for every couple of maps $u : x \rightarrow a$ and $v : x \rightarrow b$ such that $f \circ u = g \circ v$, there is exactly one map $l : x \rightarrow d$ such that $s_a \circ l = u$ and $s_b \circ l = v$. Compare the commutative diagram in figure 6.4 for this situation.*

The existence and uniqueness of fiber products is easily shown, see the following proposition. But one special case is already at our hands: Suppose that $c = \{\emptyset\} = 1$. Then, by its universal property as a final set, there is always exactly one couple of maps $! : a \rightarrow 1$ and $! : b \rightarrow 1$. Further, the commutativity conditions $f \circ s_a = g \circ s_b$, $s_a \circ l = u$ and $s_b \circ l = v$ are also automatically fulfilled. So, if we set $d = a \times b$, this is a fiber product in

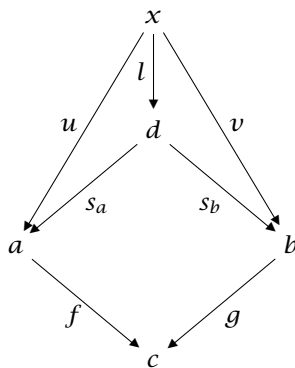


Fig. 6.4. Universal property of fiber products.

this case. In other words, the Cartesian product is the special case of a fiber product with $c = 1$, the final set.

Proposition 65 *Given three sets a, b and c and two maps $f : a \rightarrow c$ and $g : b \rightarrow c$, there exists a fiber product $s_a : d \rightarrow a$ and $s_b : d \rightarrow b$ with respect to f and g , and for any fiber product $s'_a : d' \rightarrow a$ and $s'_b : d' \rightarrow b$ with respect to f and g , there is a unique bijection $t : d \xrightarrow{\sim} d'$ such that $s'_a \circ t = s_a$ and $s'_b \circ t = s_b$. The fiber product is denoted by $d = a \times_c b$, the two maps f and g being implicitly given.*

More explicitly, such a fiber product is constructed as follows: Take the Cartesian product $a \times b$, together with its projections pr_a and pr_b . Then consider the subspace $a \times_c b \subset a \times b$ consisting of all couples (x, y) such that $f(x) = g(y)$. On this set, take the restriction of the projections, i.e., $s_a = pr_a|_{a \times_c b}$ and $s_b = pr_b|_{a \times_c b}$.

Exercise 27 The easy proof is left to the reader.

Exercise 28 Given two subsets $a \subset c$ and $b \subset c$, show that $a \times_c b \xrightarrow{\sim} a \cap b$.

Exercise 29 Given a map $f : a \rightarrow c$ and a subset $g : b \subset c$, prove that the fiber product of these two maps is the fiber $s_a : f^{-1}(c) \subset a$, with the second map $s_b = f|_{f^{-1}(c)}$.

With this small extension to theory, the relational database structure is easily described. We make an illustrative example and interpret its mech-

anisms in terms of the fiber product and other set-theoretical constructions.

To begin with, one is given domains from where values can be taken. However, these domains also have a name, not only values. We therefore consider sets with specific names X , but also isomorphisms $X \xrightarrow{\sim} V_X$ with given sets of values. Then, whenever we need to take elements $x \in X$, their values will be taken to lie in V_X , so that we may distinguish the elements, but nevertheless compare their values. In our example, we consider these sets of values: $\text{INTEGER} = \{1, 2, 3, 4, 5 \dots\}$, this is a finite set of numbers (we assume that these are given, we shall discuss the precise definition of numbers later), whose size is defined by the standard implementation of numbers on a given operating system. Next, we are given a set

$$\text{TEXT} = \{\text{Apples, Cookies, Oranges, Donald Duck, Mickey Mouse, Goofy, Bunny, Shrek, \dots}\}$$

of words, which also depends on the computer memory (again, words in a formal sense will be defined later). We now need the sets:

$$\begin{aligned} \text{ORDER_ID} &\xrightarrow{\sim} \text{INTEGER} \\ \text{CUSTOMER_ID} &\xrightarrow{\sim} \text{INTEGER} \\ \text{PRODUCT} &\xrightarrow{\sim} \text{TEXT} \\ \text{NAME} &\xrightarrow{\sim} \text{TEXT} \\ \text{ADDRESS} &\xrightarrow{\sim} \text{TEXT} \end{aligned}$$

We consider two subsets ORDERS and CUSTOMERS , called *relations* in database theory,

$$\begin{aligned} \text{ORDERS} &\subset \text{ORDER_ID} \times \text{PRODUCT} \times \text{CUSTOMER_ID} \\ \text{CUSTOMERS} &\subset \text{CUSTOMER_ID} \times \text{NAME} \times \text{ADDRESS} \end{aligned}$$

which we specify as follows, to be concrete. The set ORDERS :

ORDER_ID	PRODUCT	CUSTOMER_ID
7	Apples	3
8	Apples	4
11	Oranges	3
13	Cookies	3
77	Oranges	7

and the set CUSTOMERS:

CUSTOMER_ID	NAME	ADDRESS
3	Donald Duck	Pond Ave.
4	Mickey Mouse	Cheeseway
5	Goofy	Dog Street
6	Bunny	Carrot Lane
7	Shrek	Swamp Alley

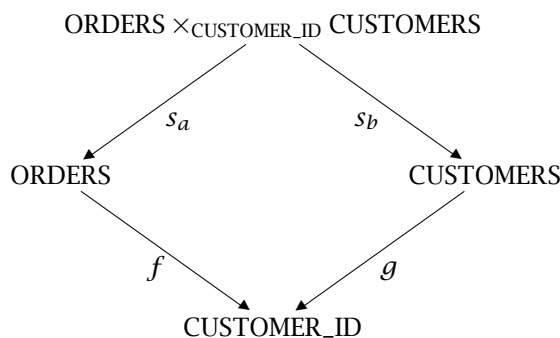
The rows of each table are its single records. A first operation on such tables is their so-called *join*. The *join* operation bears no relation to the *join* operator in mathematical lattice theory, but the terminology is common in database theory. Mathematically speaking, it is a fiber product, which works as follows: Observe that the two relations ORDERS and CUSTOMERS are subsets of Cartesian products where factor spaces with common values appear, for example CUSTOMER_ID of ORDERS and CUSTOMER_ID of CUSTOMERS. In the join, we want to look for records having the same value for their CUSTOMER_ID coordinate. We therefore consider the composed maps

$$F : \text{ORDER_ID} \times \text{PRODUCT} \times \text{CUSTOMER_ID} \rightarrow \text{CUSTOMER_ID}$$

and

$$G : \text{CUSTOMER_ID} \times \text{NAME} \times \text{ADDRESS} \rightarrow \text{CUSTOMER_ID}$$

derived from the first projections and the identification bijections for the values. Their restrictions $f = F|_{\text{ORDERS}}$ and $g = G|_{\text{CUSTOMERS}}$ yield the situation needed for a fiber product, and the join is exactly this construction:



In the SQL syntax, the join is defined by the command

```

ORDERS JOIN CUSTOMERS
ON (ORDERS.CUSTOMER_ID = CUSTOMERS.CUSTOMER_ID)

```

The dot notation `ORDERS.CUSTOMER_ID` means the choice of the coordinate `CUSTOMER_ID`, i.e., this is the definition of the arrow f , whereas `CUSTOMERS.CUSTOMER_ID` defines g in the fiber product, and the equality sign “=” means that these two arrows f and g are taken for the fiber product construction.

We therefore obtain the following fiber product table:

ORDER_ID	PRODUCT	CUSTOMER_ID	NAME	ADDRESS
7	Apples	3	Donald Duck	Pond Ave.
11	Oranges	3	Donald Duck	Pond Ave.
13	Cookies	3	Donald Duck	Pond Ave.
8	Apples	4	Mickey Mouse	Cheeseway
77	Oranges	7	Shrek	Swamp Alley

For the next operation on this join we consider subsets thereof which are defined by the additional code `WHERE . . .` as in the following example:

```

ORDERS JOIN CUSTOMERS
ON (ORDERS.CUSTOMER_ID = CUSTOMERS.CUSTOMER_ID)
WHERE
    ORDERS.PRODUCT = 'Apples'
OR
    ORDERS.PRODUCT = 'Oranges'

```

This means that one chooses all elements in the join such that their projection to the coordinate `ORDERS` is either “Apples” or “Oranges”. Mathematically, this again implies fiber products: We first consider the fiber of the projection to the factor `PRODUCT`:

$$p_{\text{PRODUCT}} : \text{ORDERS} \times_{\text{CUSTOMER_ID}} \text{CUSTOMERS} \rightarrow \text{PRODUCT}.$$

Then we consider the fiber of the singleton {“Apples”}:

$$\begin{array}{ccc}
 p_{\text{PRODUCT}}^{-1}(\text{“Apples”}) & \xrightarrow{\text{inclusion}} & \text{ORDERS} \times_{\text{CUSTOMER_ID}} \text{CUSTOMERS} \\
 \downarrow & & \downarrow p_{\text{PRODUCT}} \\
 \{\text{“Apples”}\} & \xrightarrow{\text{inclusion}} & \text{PRODUCT}
 \end{array}$$

This gives us all elements of the join which have the PRODUCT coordinate “Apples”. The same is done with the “Oranges” coordinate:

$$\begin{array}{ccc}
 p_{\text{PRODUCT}}^{-1}(\text{“Oranges”}) & \xrightarrow{\text{inclusion}} & \text{ORDERS} \times_{\text{CUSTOMER_ID}} \text{CUSTOMERS} \\
 \downarrow & & \downarrow p_{\text{PRODUCT}} \\
 \{\text{“Oranges”}\} & \xrightarrow{\text{inclusion}} & \text{PRODUCT}
 \end{array}$$

So we have obtained two sets in the join set which we now may combine with the Boolean operation “OR”, which amounts to taking the union $p_{\text{PRODUCT}}^{-1}(\text{“Apples”}) \cup p_{\text{PRODUCT}}^{-1}(\text{“Oranges”})$ of these two fibers.

Concluding this example, one then chooses a number of coordinates and omits the others in the union set by the prepended SELECT command

```

SELECT PRODUCT, NAME
FROM
ORDERS JOIN CUSTOMERS
ON (ORDERS.CUSTOMER_ID = CUSTOMERS.CUSTOMER_ID)
WHERE
    ORDERS.PRODUCT = 'Apples'
OR
    ORDERS.PRODUCT = 'Oranges'

```

Mathematically, we take the image

$$p_{\text{NAME,PRODUCT}}(p_{\text{PRODUCT}}^{-1}(\text{“Apples”}) \cup p_{\text{PRODUCT}}^{-1}(\text{“Oranges”}))$$

of the union under the projection

$$p_{\text{NAME,PRODUCT}} : \text{ORDERS} \times_{\text{CUSTOMER_ID}} \text{CUSTOMERS} \rightarrow \text{NAME} \times \text{PRODUCT}$$

which gives us this list:

PRODUCT	NAME
Apples	Donald Duck
Oranges	Donald Duck
Apples	Mickey Mouse
Oranges	Shrek

Natural Arithmetic

This chapter is central insofar as the basic arithmetic operations, i.e., addition, multiplication, and exponentiation of natural numbers are introduced, operations which are the seed of the entire mathematical calculations.

7.1 Natural Operations

All these operations are defined by recursion, i.e., by applying the recursion theorem 55.

Definition 36 *Given a natural number a , addition to a is recursively defined as the function¹ $a + ? : \mathbb{N} \rightarrow \mathbb{N}$ which evaluates to*

$$a + 0 = a \text{ and } a + b^+ = (a + b)^+.$$

Supposing that addition is defined, multiplication with a is defined as the function $a \cdot ? : \mathbb{N} \rightarrow \mathbb{N}$ which evaluates to

$$a \cdot 0 = 0 \text{ and } a \cdot (b^+) = (a \cdot b) + a.$$

Supposing that addition and multiplication are defined, exponentiation of $a \neq 0$ is defined as the function $a^? : \mathbb{N} \rightarrow \mathbb{N}$ which evaluates to

$$a^0 = 1 \text{ and } a^{(b^+)} = (a^b) \cdot a.$$

If $a = 0$, we define $0^0 = 1$ and $0^b = 0$ for $b \neq 0$.

¹ When defining function symbols, the question mark is used to indicate the position of the arguments.

Evidently, $a^+ = a + 1$, and from now on we identify these two expressions. The number $a + b$ is called the sum of a and b . The number $a \cdot b$ is called the product of a and b . These operations share the following properties. All these properties can be demonstrated by induction.

Sorte 66 Let a, b, c be natural numbers. We have these laws:

- (i) (Additive neutral element) $a + 0 = 0 + a = a$,
- (ii) (Additive associativity) $a + (b + c) = (a + b) + c$, which is therefore written as $a + b + c$,
- (iii) (Additive commutativity) $a + b = b + a$,
- (iv) (Multiplicative neutral element) $a \cdot 1 = 1 \cdot a = a$,
- (v) (Multiplicative associativity) $a \cdot (b \cdot c) = (a \cdot b) \cdot c$, which is therefore written as $a \cdot b \cdot c$,
- (vi) (Multiplicative commutativity) $a \cdot b = b \cdot a$,
- (vii) (Multiplication distributivity) $a \cdot (b + c) = a \cdot b + a \cdot c$,
- (viii) (Exponential neutral element) $a^1 = a$,
- (ix) (Exponentiation (+)-distributivity) $a^{b+c} = a^b \cdot a^c$,
- (x) (Exponentiation (\cdot)-distributivity) $(a \cdot b)^c = a^c \cdot b^c$,
- (xi) (Additive monotony) if $a < b$, then $a + c < b + c$,
- (xii) (Multiplicative monotony) if $c \neq 0$ and $a < b$, then $a \cdot c < b \cdot c$,
- (xiii) (Exponential base monotony) if $c \neq 0$ and $a < b$, then $a^c < b^c$,
- (xiv) (Exponential exponent monotony) if $c \neq 0, 1$ and $a < b$, then $c^a < c^b$,
- (xv) (Ordering of operations) if $a, b > 1$, then $a + b \leq a \cdot b \leq a^b$.

Proof (i) We have $a + 0 = a$ by definition, while $0 + (a^+) = (0 + a)^+ = a^+$ by recursion on a .

(ii) This is true for $c = 0$ by (i). By recursion on c , we have $a + (b + c^+) = a + ((b + c)^+) = (a + (b + c))^+ = ((a + b) + c)^+ = (a + b) + c^+$.

(iii) This is true for $b = 0$ by (i). We also have $a + 1 = 1 + a$, in fact, this is true for $a = 0$, and by recursion, $(a^+)^+ = (a + 1)^+ = (a + (1^+)) = a + (1 + 1) = (a + 1) + 1 = (a^+) + 1$. By recursion on b , we have $a + b^+ = (a + b)^+ = (b + a)^+ = b + (a^+) = b + (a + 1) = b + (1 + a) = (b + 1) + a = b^+ + a$.

(iv) We have $a \cdot 1 = (a \cdot 0) + a = 0 + a = a$. Therefore $1 \cdot 0 = 0 = 0 \cdot 1$, while $1 \cdot a^+ = (1 \cdot a) + 1 = a + 1 = a^+$.

(vii) We have $a \cdot (b + c) = a \cdot b + a \cdot c$ for $c = 0$. By recursion on c we have $a \cdot (b + c^+) = a \cdot ((b + c)^+) = a \cdot (b + c) + a = a \cdot b + a \cdot c + a = a \cdot b + a \cdot c^+$.

- (v) For $c = 0, 1$ we have associativity by the previous results. By recursion on c we have $a \cdot (b \cdot c^+) = a \cdot (b \cdot c + b) = a \cdot (b \cdot c) + a \cdot b = (a \cdot b) \cdot c + (a \cdot b) \cdot 1 = (a \cdot b) \cdot c^+$.
- (vi) Commutativity is known for $b = 1$. By recursion on b we have $a \cdot b^+ = a \cdot (b + 1) = a \cdot b + a = b \cdot a + a = 1 \cdot a + b \cdot a = (1 + b) \cdot a = b^+ \cdot a$.
- (viii) We have $a^1 = a^0 \cdot a = 1 \cdot a = a$.
- (ix) We have $a^{(b+0)} = a^b = a^b \cdot a^0$, and $a^{(b+c^+)} = a^{((b+c)^+)} = a^{(b+c)} \cdot a = a^b \cdot a^c \cdot a = a^b \cdot (a^c \cdot a) = a^b \cdot a^{(c^+)}$.
- (x) We have $(a \cdot b)^0 = 1 = 1 \cdot 1 = a^0 \cdot b^0$, and $(a \cdot b)^{(c^+)} = (a \cdot b)^c \cdot (a \cdot b) = a^c \cdot b^c \cdot a \cdot b = (a^c \cdot a) \cdot (b^c \cdot b) = a^{(c^+)} \cdot b^{(c^+)}$.
- (xi) If $a < b$, then $a + 0 < b + 0$, and $a + c^+ = (a + c)^+ < (b + c)^+ = b + c^+$, since $x < y$ implies $x^+ < y^+$ by lemma 41 (ii).
- (xii) If $a < b$, then $a \cdot 1 < b \cdot 1$, and $a \cdot c^+ = a \cdot c + a < b \cdot c + a < b \cdot c + b = b \cdot c^+$.
- (xiii) For $a = 0$ or $c = 1$ it is clear, suppose $a \neq 0$ and do recursion on c . Then $a^{(c^+)} = a^c \cdot a < b^c \cdot a < b^c \cdot b = b^{(c^+)}$.
- (xiv) For $b = a^+$ it is clear, so suppose $b = d^+$, $a < d$. Then by recursion on b , $c^a < c^d$ and therefore $c^d < c^d \cdot c = c^b$.
- (xv) To begin with denote $1^+ = 2$ (attention: we still have not introduced the common notation for natural numbers) and take $b = 2$. Then $a + 2 \leq a \cdot 2 \leq a^2$ is easily proved by induction on a , starting with the famous equality $2 + 2 = 2 \cdot 2 = 2^2$. We then prove the inequalities by induction on b , the details being left to be completed by the reader. \square

Proposition 67 *If a and b are natural numbers such that $a \leq b$, then there is exactly one natural number c such that $a + c = b$.*

Proof We use induction on b . If $b = a$, then $c = 0$ solves the problem. If $b > a$, then $b = d^+$ with $d \geq a$. Then, since $d < b$, we can use the induction hypothesis to find e , such that $a + e = d$. We set $c = e^+$, and we have by definition of addition $a + c = a + e^+ = (a + e)^+ = d^+ = b$. If c and c' are two different solutions, we must have $c < c'$, for example. Then monotony implies $a + c < a + c'$, i.e., $b < b$, a contradiction. \square

If n is a natural number different from 0, we may look for the unique m , such that $m + 1 = n$. Clearly, it is the m such that $n = m^+$. This is the *predecessor of n* , which we denote by $n - 1$, a notation which will become clear later, when subtraction has been defined.

7.2 Euclid and the Normal Forms

The following theorem is Euclid's so-called "division theorem". It is a central tool for the common representation of natural, and also rational and real numbers in the well-known decimal format. Moreover, it is the central step in the calculation of the greatest common divisor of two natural numbers by the Euclidean algorithm², see chapter 16.

Proposition 68 (Division Theorem) *If a and b are natural numbers with $b \neq 0$, then there is a unique pair q and r of natural numbers with $r < b$ such that*

$$a = q \cdot b + r.$$

Proof Existence: Let t be the minimal natural number such that $a < t \cdot b$. For example, according to sorite 66, $a < a \cdot b$, so t exists and evidently is non-zero, $t = q^+$. This means that $q \cdot b \leq a$. So by proposition 67, there is r such that $a = q \cdot b + r$. If $b \leq r$ we have $r = b + p$, and by the choice of t , $a = q \cdot b + b + p = (q + 1)b + p = t \cdot b + p > a$, a contradiction. So the existence is proved.

Uniqueness: If we have two representations $a = q \cdot b + r = q' \cdot b + r'$ with $q' \geq q^+$, then we have $a = q' \cdot b + r' \geq r \cdot b + b + r' > q \cdot b + r = a$, a contradiction. So $q = q'$, and equality of r and r' follows from proposition 67. \square

Proposition 69 *If a and b are natural numbers with $a \neq 0$ and $b \neq 0, 1$, then there is a unique triple c, s, r of natural numbers with $r < b^c$ and $0 < s < b$ such that*

$$a = s \cdot b^c + r.$$

Proof Let t be the minimal natural number such that $a < b^t$, and clearly $t = w + 1$. Such a t exists since $a < b + a \leq b \cdot a \leq b^a$ by sorite 66. Therefore $a \geq b^w$. We now apply proposition 68 and have $a = r \cdot b^w + s, s < b^w$. Now, if $r = b + p$, then we also have $a = (b + p) \cdot b^w + s = b^t + p \cdot b^w + s$, a contradiction to the choice of t . So we have one such desired representation. Uniqueness follows by the usual contradiction from different s coefficients, and then from different r 's for equal s coefficients. \square

In order to define the b -adic representation of a natural number, of which the decimal (10-adic) representation is a special case, we need to define

² An *algorithm* is a detailed sequence of actions (steps), starting from a given input, to perform in order to accomplish some task, the output. It is named after Al-Khwarizmi, a Persian mathematician who wrote a book on arithmetic rules about A.D. 825.

what is the sum of a finite sequence $(a_i)_{i < n}$ of length n of natural numbers a_i .

Definition 37 Given a finite sequence $(a_i)_{i \leq n}$ of natural numbers, its sum is denoted by $\sum_{i \leq n} a_i$, by $a_0 + a_1 + \dots + a_n$, or by $\sum_{i=0,1,\dots,n} a_i$, and is defined by recursion on the sequence length as follows:

$$\begin{aligned} n = 0 & : \sum_{i \leq 0} a_i = a_0 \\ n > 0 & : \sum_{i \leq n} a_i = \left(\sum_{i \leq n-1} a_i \right) + a_n \end{aligned}$$

Because of the associative law of addition, it is in fact not relevant how we group the sum from a_0 to a_n .

Proposition 70 (Adic Normal Form) If a and b are non-zero natural numbers and $b \neq 1$, then there is a uniquely determined finite number n and a sequence $(s_i)_{i=0,\dots,n}$, with $s_n \neq 0$ and $s_i < b$ for all i , such that

$$a = \sum_{i=0,\dots,n} s_i \cdot b^i. \quad (7.1)$$

Proof This immediately results from proposition 69 and by induction on the (unique) remainder r in that proposition. \square

Definition 38 Given non-zero natural numbers a and b , and $b \neq 1$ as in proposition 70, the number b which occurs in the representation (7.1), is called the base of the adic representation, and the representation is called the b -adic representation of a . It is denoted by

$$a =_b s_n s_{n-1} \dots s_1 s_0 \quad (7.2)$$

or, if the base is clear, by

$$a = s_n s_{n-1} \dots s_1 s_0.$$

Remark 10 In computer science, the term *-adic* is usually replaced by the term *-ary*.

Example 21 For the basis $b = 2$, we have the 2-adic representation, which is also known as the *dual* or *binary* representation. Here, the representation $a =_b s_n s_{n-1} \dots s_1 s_0$ from formula (7.2) reduces to a sequence of 1s and 0s.

With the well-known notations $3 = 2 + 1$, $4 = 3 + 1$, $5 = 4 + 1$, $6 = 5 + 1$, $7 = 6 + 1$, $8 = 7 + 1$, $9 = 8 + 1$, $Z = 9 + 1$, we have the *decadic* representation

$$a =_Z s_n s_{n-1} \dots s_1 s_0 \quad 0 \leq s_i \leq 9$$

with special cases $Z = 10$, $Z^2 = 100$, $Z^3 = 1000$, and so on.

For the *hexadecimal* base $H =_Z 16$, one introduces the symbols $1, 2, \dots, 9$, $A =_Z 10$, $B =_Z 11$, $C =_Z 12$, $D =_Z 13$, $E =_Z 14$, $F =_Z 15$.

For example $x =_Z 41663$ becomes, in the hexadecimal base, $x =_H A2BF$, and, in the binary representation, $x =_2 1010001010111111$.

Infinites

We already know that the powerset 2^a of any set a has larger cardinality than a itself, i.e., there is an injection $a \rightarrow 2^a$ but no injection in the other direction. This does not mean, however, that constructions of larger sets from given ones always lead to strictly larger cardinalities.

8.1 The Diagonalization Procedure

We first have to reconsider the universal constructions of the Cartesian product and coproduct. Given a set a and a non-zero natural number n , we have the n -th power a^n , but we could also define recursively $a^{\times 1} = a$ and $a^{\times n+1} = a^{\times n} \times a$, and it is clear that $a^{\times n} \overset{\sim}{\sim} a^n$. Dually, we define $a^{\sqcup 1} = a$ and $a^{\sqcup n+1} = a^{\sqcup n} \sqcup a$.

Proposition 71 *If a is a set that has the cardinality of \mathbb{N} (in which case a is called denumerable), then for any positive natural number n , the sets a , $a^{\times n}$ and $a^{\sqcup n}$ have the same cardinality, i.e., are equipollent.*

Proof The proof of this proposition depends on the Bernstein-Schröder theorem 22, which we apply to the situation of pairs of injections $a \rightarrow a^{\times n} \rightarrow a$ and $a \rightarrow a^{\sqcup n} \rightarrow a$. Now, an injection $a \rightarrow a^{\sqcup n}$ is given by the known injection to the last cofactor defined in definition 28. An injection $a \rightarrow a^{\times n}$ is also given by the identity $a \rightarrow a$ on each factor. So we are left with the injections in the other direction. We may obviously suppose $n = 2$ and deduce from this the general case by induction on n . We also may suppose $a = \mathbb{N}$.

Now, a map $f : \mathbb{N} \sqcup \mathbb{N} \rightarrow \mathbb{N}$ is given as follows: for x in the left cofactor, we define $f(x) = x \cdot 2$, for x in the right cofactor, we set $f(x) = x \cdot 2 + 1$. By

the uniqueness in proposition 68, this is an injection. To obtain an injection $g : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$, we consider any pair $(x, y) \in \mathbb{N}^2$. We may then associate each pair (x, y) with the pair $(x, x + y) = (x, n) \in \mathbb{N}^2$ with $0 \leq x \leq n$, $n \in \mathbb{N}$. Call $\mathbb{N}^{<2}$ the set of these pairs. Then we have a bijection $u : \mathbb{N}^2 \xrightarrow{\sim} \mathbb{N}^{<2}$. Consider the function $f : \mathbb{N}^{<2} \rightarrow \mathbb{N}$ defined by $f(x, n) = 2^n + x$. We have $f(0, 0) = 1$, and for $0 < n$, $x \leq n < 2 \cdot n \leq 2^n$, so the uniqueness part of proposition 69 applies, and we have an injection (see figure 8.1). \square

(x, n)	0	1	2	3	\dots
0	1	2	4	8	\dots
1		3	5	9	
2			6	10	
3				11	
\vdots					

Fig. 8.1. The entries in the table are the values of the injection f from $\mathbb{N}^{<2}$ to \mathbb{N} . The arrows indicate the order on the natural numbers.

Remark 11 The proof of proposition 71 uses the so-called “diagonal procedure”, which is a central tool in aligning n -tuples in a linear ordering. This procedure also works for non-denumerable infinite sets, i.e., the proposition is also true for any infinite sets, but this is not relevant for general computer science.

Remark 12 The equivalence of the axiom of choice and the proposition that every set can be well-ordered has a special meaning for finite or denumerable sets: If a set is denumerable, then it can be well-ordered by the ordering among natural numbers, and therefore, the axiom of choice easily follows from this well-ordering for denumerable sets. In other words, for denumerable sets, the axiom of choice is a theorem by the very definition of denumerability.

The Classical Number Domains \mathbb{Z} , \mathbb{Q} , \mathbb{R} , and \mathbb{C}

This chapter is the recompensation for the abstract initialization of mathematics: it will give us all the central number domains as they are constructed from the natural numbers. In general, there are different methods for the construction of the same number domains. We have decided to present the most direct methods which also lead to effective representations of numbers in computer environments.

First, the domain \mathbb{Z} of integer numbers is constructed from natural numbers \mathbb{N} , then rational numbers or fractions \mathbb{Q} from integers, and real numbers (also called decimal numbers in a special representation) from rational numbers. This process culminates in the building of complex numbers from real numbers. These examples are also very important for the understanding of subsequent chapters on algebra.

Basically, the construction of new number domains is motivated by the absence of many operations one would like to apply to numbers. For example, it is in general not possible to solve the domain of natural numbers. Moreover, we do not have any possibility to model most of the common geometric objects such as lines. Finally, non-linear equations such as $a^2 = b$ cannot be solved.

Mathematics must supply tools that help us understand the solution spaces for such problems. The set-theoretic approach has provided us with the construction of numbers, and we are now ready to find solutions to the aforementioned problems.

9.1 Integers \mathbb{Z}

We have seen that the equation $a + x = b$ for natural numbers has exactly one solution in the case $a \leq b$, we call it $b - a$. But for $a > b$, such a solution does not exist within the domain of natural numbers. Integer numbers, however, can solve this problem.

Lemma 72 Consider the relation R among pairs $(a, b), (p, q) \in \mathbb{N}^2$, defined by “ $(a, b)R(p, q)$ iff $a + q = b + p$ ”.

- (i) The relation R is an equivalence relation.
- (ii) A pair $(a, b) \in \mathbb{N}^2$ with $a \geq b$ is equivalent to $(a - b, 0)$, whereas a pair $(a, b) \in \mathbb{N}^2$ with $a < b$ is equivalent to $(0, b - a)$.
- (iii) Two pairs $(x, 0), (y, 0)$, or $(0, x), (0, y)$, respectively, are equivalent iff $x = y$. Two pairs $(x, 0), (0, y)$ are equivalent iff $x = y = 0$.

Proof The relation R is reflexive since $(a, b)R(a, b)$ iff $a + b = b + a$ by the commutativity of addition. It is symmetric for the same reason. And if $(a, b)R(p, q)$ and $(p, q)R(r, s)$, then $a + q = b + p$ and $p + s = q + r$, whence $a + q + p + s = b + p + q + r$, but then we may cancel $p + q$ and obtain the desired equality $a + s = b + r$. The other claims follow immediately from equivalence of ordered pairs. \square

Definition 39 The set \mathbb{N}^2/R is denoted by \mathbb{Z} , its elements $[a, b]$ are called integers. Each integer is uniquely represented by either $[a, 0]$ for $a \in \mathbb{N}$, or by $[0, b]$, $b \in \mathbb{N} - \{0\}$. We identify the former class $[a, 0]$ with its unique associated natural number a , and the latter class $[0, b]$ with the uniquely determined natural number b , together with a minus sign, i.e., by $-b$. The numbers $[a, 0]$ with $a \neq 0$ are called the positive integers, while the numbers $-b$, $b \neq 0$ are called the negative integers. The meeting point between these number types is the number $0 = -0$ (zero) which is neither positive nor negative.

The linear ordering on the natural numbers is extended to the integers by the rule that $[a, 0] < [c, 0]$ iff $a < c$, and $[0, a] < [0, c]$, i.e., $-a < -c$ iff $a > c$. Further, we set $[a, 0] > [0, b]$, i.e., $a > -b$ for all natural $b \neq 0$ and natural a . We finally set $|[a, 0]| = a$ and $|[0, b]| = b$ and call this the absolute value of the integer.

If $x = [a, b]$ is an integer, we further denote by $-x$ the integer $[b, a]$ and call it the additive inverse. This definition evidently generalizes the convention $-c = [0, c]$. We also write $a - b$ for $a + (-b)$.

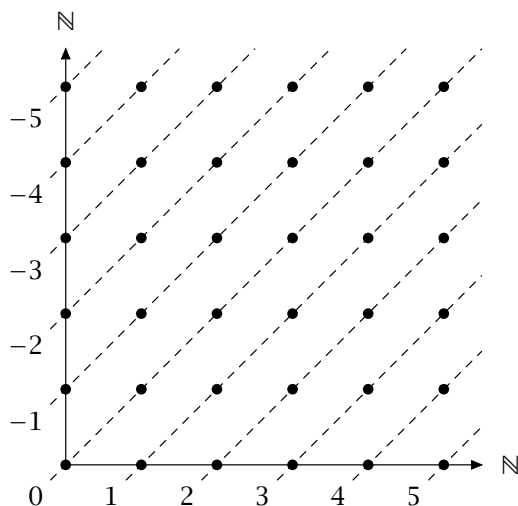


Fig. 9.1. If the relation R used for the definition of the integers is drawn as a subset of $\mathbb{N} \times \mathbb{N}$, each equivalence class (i.e., each integer) consists of the points lying on a dashed diagonal line.

This means that we have “embedded” the natural number set \mathbb{N} in the larger set \mathbb{Z} of integers as the non-negative integers, and that the ordering among natural numbers has been extended to a linear ordering on \mathbb{Z} . Observe however that the linear ordering on \mathbb{Z} is not a well-ordering since there is no smallest integer.

To calculate the “size” of \mathbb{Z} , observe that by lemma 72, $\mathbb{Z} = (\mathbb{N} - \{0\}) \sqcup -\mathbb{N}$. But we have two bijections $p : \mathbb{N} \xrightarrow{\sim} \mathbb{N} - \{0\} : n \mapsto n + 1$ and $q : \mathbb{N} \xrightarrow{\sim} -\mathbb{N} : n \mapsto -n$. Therefore $\mathbb{Z} \xrightarrow{\sim} \mathbb{N} \sqcup \mathbb{N}$, and $\mathbb{N} \xrightarrow{\sim} \mathbb{N} \sqcup \mathbb{N}$ by proposition 71, so $\mathbb{Z} \xrightarrow{\sim} \mathbb{N}$, i.e., \mathbb{Z} and \mathbb{N} have the same cardinality.

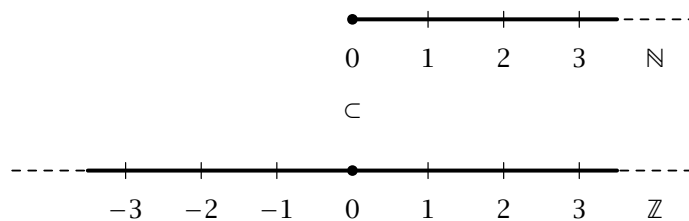


Fig. 9.2. The common representation of integers shows them as equidistant points on a straight line, with increasing values from left to right.

Next, we will create an arithmetic on \mathbb{Z} which extends the arithmetic on the natural numbers. The following technique for defining addition of integers is a prototype of the definition of functions on equivalence classes: One defines these functions on elements (representatives) of such equivalence classes and then shows that the result is in fact not a function of the representative, but only of the class as a such. If a definition on representatives works in this sense, one says that the function is *well defined*.

Definition 40 *Given two integers $[a, b]$ and $[c, d]$, their sum is defined by $[a, b] + [c, d] = [a + c, b + d]$, i.e., “factor-wise”. This function is well defined.*

In order to show that this function is well defined, we have to make sure that it is independent of the specific representatives of the equivalence classes. So assume that $[a, b] = [x, y]$, and $[c, d] = [r, s]$, i.e., $a + y = b + x$, and $c + s = d + r$, respectively. We have to show that $[a + c, b + d] = [x + r, y + s]$, i.e., $a + c + y + s = b + d + x + r$. But

$$\begin{aligned} a + c + y + s &= a + y + c + s \\ &= b + x + c + s \\ &= b + x + d + r \\ &= b + d + x + r, \end{aligned}$$

where the properties of commutativity and associativity of natural numbers have been used. Thus the addition of integers is indeed well defined.

Sorite 73 *Let \mathbb{Z} be provided with the addition $+: \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}$, and let a, b, c be any integers. Then we have these properties.*

- (i) (Associativity) $(a + b) + c = a + (b + c) = a + b + c$,
- (ii) (Commutativity) $a + b = b + a$,
- (iii) (Additive neutral element) $a + 0 = a$,
- (iv) (Additive inverse element) $a - a = 0$,
- (v) (Extension of natural arithmetic) *If $a, b \in \mathbb{N}$, then $[a + b, 0] = [a, 0] + [b, 0]$, i.e., it amounts to the same if we add two natural numbers a and b or the corresponding non-negative integers, also denoted by a and b .*
- (vi) (Solution of equations) *The equation $a + x = b$ in the “unknown” x has exactly one integer number solution x , i.e., $x = b - a$.*

Proof (i), (ii), (iii) Associativity, commutativity, and the neutrality of 0 immediately follows from associativity for natural numbers and the factor-wise definition of addition.

(iv) For $a = [u, v]$, we have $a - a = a + (-a) = [u, v] + [v, u] = [u + v, u + v] = 0$. The rest is immediate from the definitions. \square

Definition 41 Let $(a_i)_{i=0, \dots, n}$ be a sequence of integers. Then the sum of this sequence is defined by

$$\begin{aligned} n = 0 & : \sum_{i=0, \dots, n} a_i = a_0, \\ n > 0 & : \sum_{i=0, \dots, n} a_i = \left(\sum_{i=0, \dots, n-1} a_i \right) + a_n. \end{aligned}$$

It is also possible to extend the multiplication operation defined on \mathbb{N} to the integers. The definition is again one by representatives of equivalence classes $[a, b]$. To understand the definition, we first observe that a class $[a, b]$ is equal to the difference $a - b$ of natural numbers with the above identification. In fact, $[a, b] = [a, 0] + [0, b] = a + (-b) = a - b$. So, if we want to extend the arithmetic on the natural numbers, we should try to observe the hoped for and given rules, and thereby get the extension. So we should have $[a, b] \cdot [c, d] = (a - b) \cdot (c - d) = ac + bd - ad - bc = [ac + bd, ad + bc]$. This motivates the following definition:

Definition 42 Given two integers $[a, b]$ and $[c, d]$, their product is defined by $[a, b] \cdot [c, d] = [ac + bd, ad + bc]$. This function is well defined.

Sorite 74 Let a, b, c be three integers. We have these rules for their multiplication.

- (i) (Associativity) $(a \cdot b) \cdot c = a \cdot (b \cdot c) = a \cdot b \cdot c$,
- (ii) (Commutativity) $a \cdot b = b \cdot a$,
- (iii) (Multiplicative neutral element) the element $1 = [1, 0]$ is neutral for multiplication, $a \cdot 1 = a$,
- (iv) (Zero and negative multiplication) $a \cdot 0 = 0$ and $a \cdot (-b) = -(a \cdot b)$,
- (v) (Distributivity) $a \cdot (b + c) = a \cdot b + a \cdot c$,
- (vi) (Integrity) If $a, b \neq 0$, then $a \cdot b \neq 0$,
- (vii) (Additive monotony) if $a < b$, then $a + c < b + c$,
- (viii) (Multiplicative monotony) if $a < b$ and $0 < c$, then $a \cdot c < b \cdot c$,

(ix) (*Extension of natural arithmetic*) For two natural numbers a and b , we have $[a \cdot b, 0] = [a, 0] \cdot [b, 0]$. This allows complete identification of the natural numbers as a subdomain of the integers, with respect to addition and multiplication.

Proof Statements (i) through (v) and (ix) are straightforward and yield good exercises for the reader.

(vi) If $a = [r, s]$ and $b = [u, v]$, then the hypothesis means $r \neq s$ and $u \neq v$. Suppose that $r > s$ and $u > v$. Then $a = [r - s, 0]$ and $b = [u - v, 0]$, with the notation of differences of natural numbers as defined at the beginning of this section. But then $a \cdot b = [(r - s) \cdot (u - v), 0] \neq [0, 0]$. The other cases $r < s, u < v$, or $r < s, u > v$, or $r > s, u < v$ are similar.

(vii) First suppose that $r \geq s$ and $u \geq v$, and let $a = [r, s] = [r - s, 0]$ and $b = [u, v] = [u - v, 0]$. Then $a < b$ means $e = r - s < f = u - v$. So for $c = [g, h]$, we have $a + c = [e + g, h]$ and $b + c = [f + g, h]$. We may suppose that either h or g is zero. If $h = 0$, then $e + g < f + g$ implies $a + c = [e + g, 0] < [f + g, 0] = b + c$. Else we have $a + c = [e, h]$ and $b + c = [f, h]$. Suppose that $h \leq e$. Then $e - h < f - h$, whence $a + c = [e, h] = [e - h, 0] < [f - h, 0] = b + c$. If $e < h \leq f$, then $a + c = [e, h] = [0, h - e] < [f - h, 0] = b + c$. If $e, f < h$, then $h - e > h - f$, and then $a + c = [e, h] = [0, h - e] < [0, h - f]$. The other cases $r < s, u < v$, or $r < s, u > v$, or $r > s, u < v$ are similar.

(viii) If $0 \leq a < b$, we are done since this is the case for natural numbers, already proven in sorite 66. Else if $a < b < 0$, then we have $0 < (-b) < (-a)$, and then by the previous case, $(-b) \cdot c < (-a) \cdot c$, whence, $-(-a) \cdot c < -(-b) \cdot c$, but $-(-a) \cdot c = a \cdot c$, $-(-b) \cdot c = b \cdot c$, whence the statement in this case. Else if $a \leq 0 < b$, then $a \cdot c \leq 0 < b \cdot c$. \square

We are now capable of adding and subtracting any two integers, and of solving an equation of the type $a + x = b$. But equations of the type $a \cdot x = b$ have no integer solution in general, for example, there is no integer x such that $2 \cdot x = 3$.

Definition 43 If a and b are two integers, we say that a divides b iff there is an integer c with $a \cdot c = b$. We then write $a|b$.

Exercise 30 For any integer b , we have $b|b$, $-b|b$, $1|b$, and $-1|b$.

Definition 44 If $b \neq \pm 1$ is such that it is divided only by $\pm b$, 1 and -1 , then we call b a prime integer.

We shall deal with the prime numbers in chapter 16. For the moment, we have the following exercise concerning prime decomposition of integers. For this we also need the product of a finite sequence of integers, i.e.,

Definition 45 Let $(a_i)_{i=0,\dots,n}$ be a sequence of integers. Then the product of this sequence is defined by

$$\begin{aligned} n = 0 & : \prod_{i=0,\dots,n} a_i = a_0, \\ n > 0 & : \prod_{i=0,\dots,n} a_i = \left(\prod_{i=0,\dots,n-1} a_i \right) \cdot a_n. \end{aligned}$$

Exercise 31 Show that every non-zero integer $a \neq \pm 1$ has a multiplicative decomposition $a = \sigma \cdot \prod_i p_i$ where $\prod_i p_i$ is a product of positive primes p_i and $\sigma = \pm 1$.

Notation 8 We shall henceforth also write ab instead of $a \cdot b$ if no confusion is possible.

Proposition 75 (Triangle Inequality) If a and b are two integers, then we have the triangle inequality:

$$|a + b| \leq |a| + |b|.$$

Exercise 32 Give a proof of proposition 75 by distinction of all possible cases for non-negative or negative a, b .

9.2 Rationals \mathbb{Q}

The construction of the rational numbers is very similar to the procedure we have used for the construction of the integers. The main difference is that the underlying building principle is multiplication instead of addition. We denote by \mathbb{Z}^* the set $\mathbb{Z} - \{0\}$.

Lemma 76 On the set $\mathbb{Z} \times \mathbb{Z}^*$, the relation R , defined by “ $(a, b)R(c, d)$ iff $ad = bc$ ”, is an equivalence relation.

Proof This is an exercise for the reader. □

Definition 46 The set $(\mathbb{Z} \times \mathbb{Z}^*)/R$ of equivalence classes for the relation R defined in lemma 76 is denoted by \mathbb{Q} . Its elements, the classes $[a, b]$, are called rational numbers and are denoted by $\frac{a}{b}$ or a/b . The (non uniquely determined) number a is called the numerator, whereas the (non uniquely determined) number b is called the denominator of the rational number

$\frac{a}{b}$. Numerator and denominator are only defined relative to a selected representative of the rational number.

Before we develop the arithmetic operations on \mathbb{Q} , let us verify that again, the integers are embedded in the rationals. In fact, we may identify an integer a with its fractional representation $\frac{a}{1}$, and easily verify that $\frac{a}{1} = \frac{b}{1}$ iff $a = b$.

Here is the arithmetic construction:

Definition 47 Let $\frac{a}{b}$ and $\frac{c}{d}$ be two rationals. Then we define

$$\begin{aligned}\frac{a}{b} + \frac{c}{d} &= \frac{ad+bc}{bd} \\ \frac{a}{b} \cdot \frac{c}{d} &= \frac{ac}{bd}.\end{aligned}$$

We further set $-\frac{a}{b} = \frac{-a}{b}$ for the additive inverse of $\frac{a}{b}$. If $a \neq 0$, we set $(\frac{a}{b})^{-1} = \frac{b}{a}$, the latter being the multiplicative inverse of $\frac{a}{b}$. These operations are all well defined.

In order to manage comparison between rational numbers, we need the following exercise.

Exercise 33 If $\frac{a}{b}$ and $\frac{c}{d}$ are rational numbers, show that one may always find numerators and denominators such that $b = d$ (common denominator) and $0 < b$.

Definition 48 If $\frac{a}{b}$ and $\frac{c}{d}$ are rational numbers, we have the (well defined) relation $\frac{a}{b} < \frac{c}{d}$ iff $a < c$, where we suppose that we have a common positive denominator $0 < b = d$.

Sorte 77 Let $\frac{a}{b}, \frac{c}{d}, \frac{e}{f}$ be rational numbers. Then these rules hold.

- (i) (Additive associativity) $(\frac{a}{b} + \frac{c}{d}) + \frac{e}{f} = \frac{a}{b} + (\frac{c}{d} + \frac{e}{f}) = \frac{a}{b} + \frac{c}{d} + \frac{e}{f}$
- (ii) (Additive commutativity) $\frac{a}{b} + \frac{c}{d} = \frac{c}{d} + \frac{a}{b}$
- (iii) (Additive neutral element) $\frac{a}{b} + \frac{0}{1} = \frac{a}{b}$
- (iv) (Additive inverse element) $\frac{a}{b} + \frac{-a}{b} = \frac{0}{1}$
- (v) (Multiplicative associativity) $(\frac{a}{b} \cdot \frac{c}{d}) \cdot \frac{e}{f} = \frac{a}{b} \cdot (\frac{c}{d} \cdot \frac{e}{f}) = \frac{a}{b} \cdot \frac{c}{d} \cdot \frac{e}{f}$
- (vi) (Multiplicative commutativity) $\frac{a}{b} \cdot \frac{c}{d} = \frac{c}{d} \cdot \frac{a}{b}$
- (vii) (Multiplicative neutral element) $\frac{a}{b} \cdot \frac{1}{1} = \frac{a}{b}$
- (viii) (Multiplicative inverse element) If $a, b \neq 0$, then $\frac{a}{b} \cdot \frac{b}{a} = \frac{1}{1}$

- (ix) (Distributivity) $\frac{a}{b} \cdot (\frac{c}{d} + \frac{e}{f}) = \frac{a}{b} \cdot \frac{c}{d} + \frac{a}{b} \cdot \frac{e}{f}$
- (x) (Linear ordering) The relation $<$ among rational numbers is a linear ordering. Its restriction to the integers $\frac{a}{1}$ induces the given linear ordering among integers.
- (xi) (Additive monotony) If $\frac{a}{b} < \frac{c}{d}$, then $\frac{a}{b} + \frac{e}{f} < \frac{c}{d} + \frac{e}{f}$.
- (xii) (Multiplicative monotony) If $\frac{a}{b} < \frac{c}{d}$ and $\frac{0}{1} < \frac{e}{f}$, then $\frac{a}{b} \cdot \frac{e}{f} < \frac{c}{d} \cdot \frac{e}{f}$.
- (xiii) (Archimedean ordering) For any two positive rational numbers $\frac{a}{b}$ and $\frac{c}{d}$ there is a natural number n such that $\frac{n}{1} \cdot \frac{a}{b} > \frac{c}{d}$.

Proof Everything here is straightforward once one has shown that in fact $\frac{a}{b} < \frac{c}{d}$ is well defined. For then we may calculate everything on (common) positive denominators and thereby reduce the problem to the integers, where we have already established these properties. So let $\frac{a}{b} < \frac{c}{d}$, and b be positive. Suppose that $\frac{a}{b} = \frac{a'}{b'}$ and $\frac{c}{d} = \frac{c'}{d'}$. We know $a < c$. Then $ab' = ba'$ and $cb' = c'b$, and therefore $a'b = b'a < b'c = c'b$, whence $a' < c'$, since b is positive. \square

Given the complete compatibility of original natural numbers within integers, and the complete compatibility of integers within rational numbers, we also “abuse” the injections

$$\mathbb{N} \rightarrow \mathbb{Z} \rightarrow \mathbb{Q}$$

and treat them as inclusions, in the sense that

- a natural number is denoted by n rather than by the equivalence class $[n, 0]$ in \mathbb{Z} ,
- an integer is denoted by z rather than by the equivalence class $\frac{z}{1}$ in \mathbb{Q} .

Definition 49 If a rational number x is represented by $x = \frac{a}{b}$, then we define its absolute value $|x|$ by $|x| = \frac{|a|}{|b|}$. This is a well-defined definition.

Exercise 34 Prove that $|x|$ is well defined. An alternative definition is this: $|x| = x$ if $0 \leq x$, otherwise it is $|x| = -x$. Give a proof of this.

Proposition 78 (Triangle Inequality) If a and b are rational numbers, then we have the triangle inequality:

$$|a + b| \leq |a| + |b|.$$

Exercise 35 Use proposition 75 to give a proof of proposition 78.

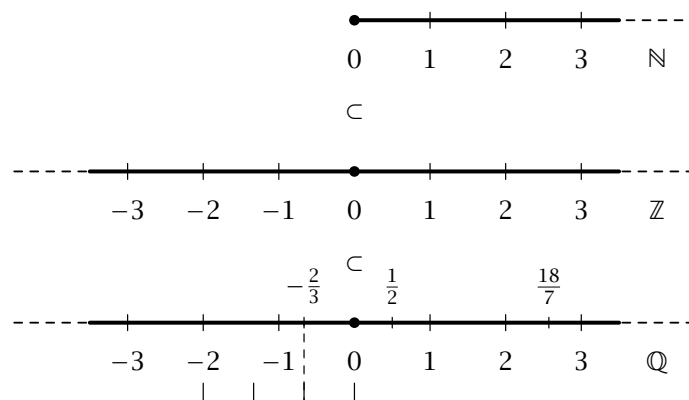


Fig. 9.3. The common representation of \mathbb{N} , \mathbb{Z} and \mathbb{Q} shows their sets as point-sets on a straight line, where the rational number $\frac{q}{p}$ is drawn as the equally divided line between the integer points 0 and q (supposing $p > 0$).

Since, in contrast to natural numbers and integers, there is a rational number between any two others, are there many more rationals than integers? The answer is negative. In fact, it can be shown that \mathbb{Z} and \mathbb{Q} have the same cardinality by an argument similar to the one used to prove that \mathbb{N} and \mathbb{Z} have the same cardinality.

9.3 Real Numbers \mathbb{R}

We are now able to solve equations of the type $ax + b = c$ for any rational coefficients a, b, c provided that $a \neq 0$. We also have a linear ordering on \mathbb{Q} and we are provided with arbitrary near numbers in the sense that for any two different rationals $x < y$, there is a rational number in-between, namely $x < \frac{1}{2}(x + y) < y$. But there are still many problems which cannot be solved in \mathbb{Q} . Intuitively, with rational numbers, there are now many points on the straight line, but there are still some gaps to be filled.

Some of the gaps on the rational number line are identified when we ask whether a so-called algebraic equation $x^2 = x \cdot x = 2$ can be solved in \mathbb{Q} . It will be shown later in chapter 16 that there is no such solution in \mathbb{Q} .

We recognize once again that in fact all our efforts in the construction of more numbers originate in the problem of solving special equations in a given number domain.

The real numbers are constructed from the rational numbers by an idea which results from the observation that the search for a solution of $x^2 = 2$ yields rational numbers y and z such that $y^2 < 2 < z^2$ and $|y - z| < 1/n$ for any non-zero natural number n , in other words, y and z are arbitrary near to each other. The point is that there is no rational number which gives us a precise solution, nonetheless, we would like to invent a number which solves the problem and in some sense is the “infinite approximation” suggested by the rational approximations.

In order to extend the rationals to the reals, we need to formalize what we vaguely described by “infinite approximation”. This definition goes back to the French mathematician Augustin Cauchy (1789–1857).

Definition 50 A Cauchy sequence of rational numbers is a sequence $(a_i)_{i \in \mathbb{N}}$ of rational numbers a_i such that for any positive natural number L , there is an index number N such that whenever indexes n, m satisfy $N < n, m$, then $|a_n - a_m| < \frac{1}{L}$.

The set of all Cauchy sequences is denoted by C .

Exercise 36 Every rational number r gives rise to the constant Cauchy sequence $e(r) = (a_i)_i$ which has $a_i = r$ for all indexes $0 \leq i$.

The sequence $(1/(i + 1))_{i \in \mathbb{N}}$ is a Cauchy sequence. More generally, any sequence $(a + a_i)_i$ is a Cauchy sequence if a is a rational number and if $(a_i)_i$ is a Cauchy sequence.

A sequence of rational numbers $(a_i)_i$ is said to *converge* to a rational number a if for any positive natural number L , there is an index N such that $n > N$ implies $|a_n - a| < \frac{1}{L}$.

Show that a convergent sequence is a Cauchy sequence, and that the rational number to which it converges is uniquely determined. We then write $\lim_{i \rightarrow \infty} a_i = a$. The sign ∞ means “infinity”, but it has no precise meaning when used without a determined context, such as $\lim_{n \rightarrow \infty}$.

Definition 51 A zero sequence is a Cauchy sequence $(a_i)_i$ which converges to 0:

$$\lim_{i \rightarrow \infty} a_i = 0.$$

The zero sequences are those sequences which we would like to forget about. To develop this idea, we begin with the following definition.

Definition 52 If $(a_i)_i$ and $(b_i)_i$ are sequences of rational numbers, their sum is defined by

$$(a_i)_i + (b_i)_i = (a_i + b_i)_i,$$

and their product is defined by

$$(a_i)_i \cdot (b_i)_i = (a_i \cdot b_i)_i.$$

Proposition 79 The following properties hold for sequences of rational numbers:

- (i) If $(a_i)_i$ and $(b_i)_i$ are Cauchy sequences, then so are their sum and product.
- (ii) If $(a_i)_i$ and $(b_i)_i$ are zero sequences, then so is their sum.
- (iii) If $(a_i)_i$ is a zero sequence and if $(b_i)_i$ is a Cauchy sequence, then their product is a zero sequence.

Proof Let $(a_i)_i$ and $(b_i)_i$ be Cauchy sequences. Given any positive natural number L , there is a common natural number N such that $n, m > N$ implies $|a_n - a_m| < \frac{1}{2L}$ and $|b_n - b_m| < \frac{1}{2L}$ if $n, m > N$. Then

$$\begin{aligned} |(a_n + b_n) - (a_m + b_m)| &= |(a_n - a_m) + (b_n - b_m)| \\ &\leq |a_n - a_m| + |b_n - b_m| \\ &< \frac{1}{2L} + \frac{1}{2L} \\ &= \frac{1}{L} \end{aligned}$$

by the triangle inequality. Further

$$\begin{aligned} |(a_n \cdot b_n) - (a_m \cdot b_m)| &= |(a_n \cdot b_n) - (a_n \cdot b_m) + (a_n \cdot b_m) - (a_m \cdot b_m)| \\ &= |(a_n \cdot (b_n - b_m) + (a_n - a_m) \cdot b_m)| \\ &\leq |a_n \cdot (b_n - b_m)| + |(a_n - a_m) \cdot b_m| \\ &= |a_n| \cdot |b_n - b_m| + |a_n - a_m| \cdot |b_m| \\ &< (|a_n| + |b_m|) \frac{1}{L} \end{aligned}$$

if $n, m > N$. Now, $|a_n| = |(a_n - a_{N+1}) + a_{N+1}| \leq \frac{1}{L} + |a_{N+1}|$ if $n > N$. Also $|b_m| \leq \frac{1}{L} + |b_{N+1}|$ if $m > N$. So, $|(a_n \cdot b_n) - (a_m \cdot b_m)| \leq k_N \cdot \frac{1}{L}$ if $n, m > N$, where k_N is a positive constant which is a function of N . Now, select N' such that $|a_n - a_m| < \frac{1}{k_N L}$ for $n, m > N'$ then we have $|(a_n \cdot b_n) - (a_m \cdot b_m)| \leq k_N \cdot \frac{1}{k_N L} = \frac{1}{L}$ for $n, m > N'$.

If $(a_i)_i$ and $(b_i)_i$ converge to 0, let N be such that $|a_n|, |b_n| < \frac{1}{2L}$ for $n > N$. Then $|a_n + b_n| \leq |a_n| + |b_n| < \frac{1}{2L} + \frac{1}{2L} = \frac{1}{L}$ for $n > N$.

Let $(a_i)_i$ and $(b_i)_i$ be two Cauchy sequences such that $(a_i)_i$ converges to 0. From the previous discussion we know that there is a positive constant k such that $|b_n| < k$ for all n . Now, let N be such that $|a_n| < \frac{1}{kL}$ for all $n > N$. Then $|a_n \cdot b_n| = |a_n| \cdot |b_n| < k \cdot \frac{1}{kL} = \frac{1}{L}$ for $n > N$. \square

The properties (ii) and (iii) of the set \mathcal{O} of zero sequences make \mathcal{O} a so-called ideal. This is an important structure in algebra, we come back to its systematic discussion in chapter 15. We are now ready to define real numbers.

Lemma 80 *The binary relation R on C defined by “ $(a_i)_i R (b_i)_i$ iff $(a_i)_i - (b_i)_i = (a_i - b_i)_i$ is a zero sequence” is an equivalence relation.*

Proof Clearly, R is reflexive and symmetric. Let $(a_i)_i, (b_i)_i, (c_i)_i$ be Cauchy sequences such that $(a_i)_i R (b_i)_i$ and $(b_i)_i R (c_i)_i$. Then $|a_n - c_n| = |a_n - b_n + b_n - c_n| \leq |a_n - b_n| + |b_n - c_n| < \frac{1}{L}$ for $n > N$ if N is such that $|a_n - b_n|, |b_n - c_n| < \frac{1}{2L}$ for $n > N$. \square

Definition 53 *The set C/R of equivalence classes under the relation R defined in lemma 80 is denoted by \mathbb{R} . Its elements are called real numbers.*

Lemma 81 *The equivalence class (the real number) of a Cauchy sequence $(a_i)_i$ is given by the “coset” of the ideal \mathcal{O} , i.e., $[(a_i)_i] = \{(a_i)_i + (c_i)_i \mid (c_i)_i \in \mathcal{O}\} = (a_i)_i + \mathcal{O}$.*

Proof If $(a_i)_i$ and $(b_i)_i$ are equivalent, then by definition $(a_i)_i = (b_i)_i + ((a_i)_i - (b_i)_i)$, and $(a_i)_i - (b_i)_i \in \mathcal{O}$. Conversely, if $(a_i)_i = (b_i)_i + (o_i)_i$, $(o_i)_i \in \mathcal{O}$, then $(a_i)_i R (b_i)_i$ by the definition of R . \square

Lemma 82 *We have an injection $e : \mathbb{Q} \rightarrow \mathbb{R}$ defined by $e(a) = (a)_i + \mathcal{O}$.*

Exercise 37 Give a proof of lemma 82.

We now want to develop the arithmetics on \mathbb{R} , and we want to show that the purpose of this construction is effectively achieved.

Lemma 83 *Let $(a_i)_i, (b_i)_i, (c_i)_i$ be Cauchy sequences of rational numbers.*

- (i) *If $(a_i)_i, (b_i)_i$ are equivalent, then so are $(a_i)_i + (c_i)_i, (b_i)_i + (c_i)_i$.*
- (ii) *If $(a_i)_i, (b_i)_i$ are equivalent, then so are $(a_i)_i \cdot (c_i)_i, (b_i)_i \cdot (c_i)_i$.*

Proof (i) In fact, $((a_i)_i + (c_i)_i) - ((b_i)_i + (c_i)_i) = (a_i - b_i)_i$, which is a zero sequence.

(ii) Similarly, $((a_i)_i \cdot (c_i)_i) - ((b_i)_i \cdot (c_i)_i) = ((a_i - b_i)_i)(c_i)_i$, but by proposition 79 this is a zero sequence. \square

This enables the definition of addition and multiplication of real numbers:

Definition 54 *If $(a_i)_i + \mathcal{O}$ and $(b_i)_i + \mathcal{O}$ are two real numbers, then we define*

$$\begin{aligned} ((a_i)_i + \mathcal{O}) + ((b_i)_i + \mathcal{O}) &= (a_i + b_i)_i + \mathcal{O} \\ ((a_i)_i + \mathcal{O}) \cdot ((b_i)_i + \mathcal{O}) &= (a_i \cdot b_i)_i + \mathcal{O}. \end{aligned}$$

By lemma 83, this definition is independent of the representative Cauchy sequences, i.e., it is well-defined.

Evidently, these operations, when restricted to the rational numbers, embedded via $e(r)$, $r \in \mathbb{Q}$, as above, yield exactly the operations on the rationals, i.e., $e(r + s) = e(r) + e(s)$ and $e(r \cdot s) = e(r) \cdot e(s)$. We therefore also use the rational numbers r instead of $e(r)$ when working in \mathbb{R} . If $x = (a_i)_i + \mathcal{O}$, we write $-x$ for $(-a_i)_i + \mathcal{O}$ and call it the *additive inverse* or *negative* of x .

The arithmetic properties of these operations on \mathbb{R} are collected in the following sorite:

Sorite 84 *Let x, y, z be real numbers.*

- (i) *(Additive associativity)* $(x + y) + z = x + (y + z) = x + y + z$
- (ii) *(Additive commutativity)* $x + y = y + x$
- (iii) *(Additive neutral element)* *The rational zero 0 is also the additive neutral element of the reals, i.e., $x + 0 = x$.*
- (iv) *(Additive inverse element)* $x + (-x) = 0$
- (v) *(Multiplicative associativity)* $(x \cdot y) \cdot z = x \cdot (y \cdot z) = x \cdot y \cdot z$
- (vi) *(Multiplicative commutativity)* $x \cdot y = y \cdot x$
- (vii) *(Multiplicative neutral element)* *The rational unity 1 is also the multiplicative neutral element of the reals, i.e., $x \cdot 1 = x$.*
- (viii) *(Multiplicative inverse element)* *If $x \neq 0$, then there is exactly one multiplicative inverse x^{-1} , i.e., $x \cdot x^{-1} = 1$, more precisely, there exists in this case a Cauchy sequence $(a_i)_i$ representing x and such*

that $a_i \neq 0$ for all i , and we may represent x^{-1} by the Cauchy sequence $(a_i^{-1})_i$.

(ix) (Distributivity) $x \cdot (y + z) = x \cdot y + x \cdot z$

Proof (i) through (vii) as well as (ix) are straightforward, because all the relevant operations are defined factor-wise on the sequence members (definition 54).

As to (viii), since $(a_i)_i$ does not converge to zero, there is a positive natural L such that for every N there is $n > N$ with $|a_n| \geq \frac{1}{L}$. Choose N such that $|a_n - a_m| < \frac{1}{2L}$ for all $n, m > N$. Fix $n > N$ such that $|a_n| \geq \frac{1}{L}$ as above. Then $|a_m| \geq |a_n| - |a_n - a_m| \geq \frac{1}{L} - \frac{1}{2L} = \frac{1}{2L} > 0$ for $n, m > N$. Therefore $(a_i)_i$ is equivalent to a sequence $(a'_i)_i$ without zero members, more precisely: there is I such that $a'_i = a_i$ for $i > I$. Then evidently the sequence $(1/a'_i)_i$ is the inverse of $(a'_i)_i$. The uniqueness of the inverse follows from the purely formal fact that $x \cdot y = x \cdot y' = 1$ implies $y = 1 \cdot y = (y \cdot x) \cdot y = y \cdot (x \cdot y) = y \cdot (x \cdot y') = (y \cdot x) \cdot y' = 1 \cdot y' = y'$. \square

Corollary 85 *If a, b, c are real numbers such that $a \neq 0$, then the equation $ax + b = c$ has exactly one solution x .*

This means that we have “saved” the algebraic properties of \mathbb{Q} to \mathbb{R} . But we wanted more than that. Let us first look for the linear ordering structure on \mathbb{R} .

Definition 55 *A real number $x = (a_i)_i + \mathcal{O}$ is called positive iff there is a rational number $\varepsilon > 0$ such that $a_i > \varepsilon$ for all but a finite set of indexes. This property is well defined. We set $x < y$ for two real numbers x and y iff $y - x$ is positive. In particular, x is positive iff $x > 0$.*

Proposition 86 *The relation $<$ on \mathbb{R} from definition 55 defines a linear ordering. The set \mathbb{R} is the disjoint union of the subset \mathbb{R}_+ of positive real numbers, the subset $\mathbb{R}_- = -\mathbb{R}_+ = \{-x \mid x \in \mathbb{R}_+\}$ of negative real numbers, and the singleton set $\{0\}$. We have*

- (i) $\mathbb{R}_+ + \mathbb{R}_+ = \{x + y \mid x, y \in \mathbb{R}_+\} = \mathbb{R}_+$,
- (ii) $\mathbb{R}_+ \cdot \mathbb{R}_+ = \{x \cdot y \mid x, y \in \mathbb{R}_+\} = \mathbb{R}_+$,
- (iii) $\mathbb{R}_- + \mathbb{R}_- = \{x + y \mid x, y \in \mathbb{R}_-\} = \mathbb{R}_-$,
- (iv) $\mathbb{R}_- \cdot \mathbb{R}_- = \{(-x) \cdot (-y) \mid x, y \in \mathbb{R}_+\} = \mathbb{R}_+$,
- (v) $\mathbb{R}_+ + \mathbb{R}_- = \{x - y \mid x, y \in \mathbb{R}_+\} = \mathbb{R}$,
- (vi) $\mathbb{R}_+ \cdot \mathbb{R}_- = \{x \cdot (-y) \mid x, y \in \mathbb{R}_+\} = \mathbb{R}_-$,
- (vii) (Monotony of addition) *if x, y, z are real numbers with $x < y$, then $x + z < y + z$,*

- (viii) (*Monotony of multiplication*) if x, y, z are real numbers with $x < y$ and $0 < z$, then $xz < yz$,
- (ix) (*Archimedean property*) if x and y are positive real numbers, there is a natural number N such that $y < Nx$,
- (x) (*Density of rationals in reals*) if $\varepsilon > 0$ is a positive real number, then there is a rational number ρ with $0 < \rho < \varepsilon$.

Proof Let us first show that $<$ is antisymmetric. If $x < y$, then $y - x$ is represented by a sequence $(a_i)_i$ with $a_i > \varepsilon$ for a positive rational ε . Then $x - y$ is represented by $(-a_i)_i$, and $-a_i < \varepsilon < 0$ for all but a finite set of indexes. If this were equivalent to a sequence $(b_i)_i$ with $b_i > \varepsilon' > 0$ except for a finite number of indexes, then $b_i - (-a_i) = b_i + a_i > \varepsilon + \varepsilon'$ could not be a zero sequence. Whence antisymmetry.

Also, if $y - x$ is represented by $(a_i)_i$ with $a_i > \varepsilon$ and $z - y$ is represented by $(b_i)_i$ with $b_i > \varepsilon'$ for all but a finite number of indexes, then $z - x = z - y + y - x$ is represented by $(a_i)_i + (b_i)_i = (a_i + b_i)_i$, and $a_i + b_i > \varepsilon + \varepsilon' > 0$ for all but a finite number of indexes, whence transitivity. Finally, if $x \neq y$, then $x - y \neq 0$. By the same argument as used in the previous proof, if $(d_i)_i$ represents $x - y$, then there is a positive rational ε and N , such that $|d_n| > \varepsilon$ for $n > N$. But since $(d_i)_i$ is a Cauchy sequence, too, the differences $|d_n - d_m|$ become arbitrary small for large n and m . So either d_n is positive or negative, but not both, for large n , and therefore $x - y$ is either positive or negative. This immediately entails statements (i) through (viii).

(ix) If x is represented by $(a_i)_i$ and y is represented by $(b_i)_i$, then there is a positive rational ε and an index M such that $a_n, b_n > \varepsilon$ for $n > M$. But since $(b_i)_i$ is a Cauchy sequence, there is also a positive δ and index M' such that $b_n < \delta$ for $n > M'$, and we may take the larger of M and M' and then suppose that $M = M'$ for our two conditions. Then, since \mathbb{Q} has the Archimedean ordering property by sorite 77, there is a natural N such that $N \cdot \varepsilon > 2 \cdot \delta$. Then we have $N \cdot a_n > N \cdot \varepsilon > 2 \cdot \delta > \delta > b_n$ for $n > M$, whence $N \cdot a_n - b_n > \delta > 0$, whence the claim.

(x) If the real number $\varepsilon > 0$ is represented by a Cauchy sequence $(e_i)_i$, then, by the very definition of positivity, there is a positive rational number δ such that $e_i > \delta$ for all but a finite number of indexes. But then $e_i - \frac{\delta}{2} > \frac{\delta}{2}$ for all but a finite number of indexes, and $\rho = \frac{\delta}{2}$ is sought-after rational number. \square

Definition 56 *The absolute value $|a|$ of a real number a is a if it is non-negative, and $-a$ else.*

Proposition 87 (Triangle Inequality) *If a and b are two real numbers, then we have the triangle inequality:*

$$|a + b| \leq |a| + |b|.$$

Proof Observe that, if a is represented by a Cauchy sequence $(a_i)_i$, then $|a|$ is represented by $(|a_i|)_i$. Therefore, if b is represented by $(b_i)_i$, then $|a + b|$ is represented by $(|a_i + b_i|)_i$, but by the triangle inequality for rationals, we have $|a_i| + |b_i| \geq |a_i + b_i|$, i.e., $|a| + |b| - |a + b|$ is represented by $(|a_i| + |b_i| - |a_i + b_i|)_i$, so it is not negative, thus $|a| + |b| - |a + b| \geq 0$, i.e., $|a| + |b| \geq |a + b|$. \square

We now have a completely general criterion for convergence in \mathbb{R} . Convergence is to be defined entirely along the lines of the definition of convergence for rational sequences.

Definition 57 A sequence $(a_i)_i$ of real numbers is said to converge to a real number a iff for every real $\varepsilon > 0$, there is an index N such that $n > N$ implies $|a_n - a| < \varepsilon$. Clearly, if such an a exists, then it is unique, and we denote convergence to a by $\lim_{i \rightarrow \infty} a_i = a$.

The sequence $(a_i)_i$ is Cauchy, iff for every real number $\varepsilon > 0$, there is a natural number N such that $n, m > N$ implies that $|a_n - a_m| < \varepsilon$.

Proposition 88 (Convergence on \mathbb{R}) A sequence $(a_i)_i$ of real numbers converges iff it is Cauchy.

Proof We omit the detailed proof since it is quite technical. However, the idea of the proof is easily described: Let $(a_i)_i$ be a Cauchy sequence of real numbers. For each $i > 0$, there is a rational number r_i such that $|a_i - r_i| < \frac{1}{i}$. This rational number can be found as follows: Represent a_i by a rational Cauchy sequence $(a_{ij})_j$. Then there is an index I_i such that $r, s > I_i$ implies $|a_{ir} - a_{is}| < \frac{1}{i}$. Then take $r_i = a_{ir}$ for any $r > I_i$. One then shows that $(r_i)_i$ is Cauchy and that $(a_i)_i$ converges to $(r_i)_i$. \square

This result entails a huge number of theorems about the existence of special numbers. We just mention one particularly important concept.

Definition 58 (Upper Bound) A real number b is an upper bound of a set $A \subset \mathbb{R}$, if $b \geq a$ for all $a \in A$, in short $b \geq A$.

Corollary 89 (Existence of Suprema) If A is a bounded, non-empty set, i.e., if there is $b \in \mathbb{R}$ such that $b \geq A$, then there is a uniquely determined supremum or least upper bound $s = \sup(A)$, i.e., an upper bound $s \geq A$ such that for all $t < s$, there is $a \in A$ with $a > t$.

Proof One first constructs a Cauchy sequence $(u_i)_i$ of upper bounds u_i of A as follows: Let u_0 be an existing upper bound and take $a_0 \in A$. Then consider the

middle $v_1 = (a_0 + u_0)/2$. If v_1 is an upper bound, set $u_1 = v_1$, otherwise set $u_1 = u_0$. In the first case, set $a_1 = a_0$, in the second case, there is $a_1 > v_1$; then consider the pair a_1, u_1 . In either case, their distance is half the first distance $|a_0 - u_0|$. We now again take the middle of this interval, i.e., $v_2 = (a_1 + u_1)/2$ and go on in the same way, i.e., defining a sequence of u_i by induction, and always taking the smallest possible alternative. This is a Cauchy sequence since the intervals are divided by 2 in each step. Further, the limit $u = \lim_{i \rightarrow \infty} u_i$, which exists according to proposition 88, is an upper bound, and there is no smaller one, as is easily seen from the construction. The details are left to the reader. \square

This corollary entails the following crucial fact:

Corollary 90 (Existence of n -th roots) *Let $a \geq 0$ be a non-negative real number and $n \geq 1$ a positive natural number, then there is exactly one number $b \geq 0$ such that $b^n = a$. This number is denoted by $\sqrt[n]{a}$ or by $a^{\frac{1}{n}}$, in the case of $n = 2$ one simply writes $\sqrt[2]{a} = \sqrt{a}$.*

Proof The cases $n = 1$ or $a = 0$ are clear, so suppose $n > 1$ and $a > 0$. Let A be the set of all q , such that $q^n < a$. This set is bounded since $a^n > a$. Take $b = \sup(A)$. We claim that $b^n = a$. Clearly $b^n \leq a$. Suppose that $b^n < a$. Then consider the following construction of a contradiction:

First we claim that for any $d > 0$ there is a natural number m such that $(1 + \frac{1}{m})^n < 1 + d$. We show this by induction on n . In the case $n = 1$ we have $(1 + \frac{1}{m}) < 1 + d$, we choose m such that $m > \frac{1}{d}$, which exists since $d > 0$.

Suppose that, for n , we have found M such that $(1 + \frac{1}{m})^n < 1 + \frac{d}{2}$ for $m \geq M$. Then, for $n + 1$, $(1 + \frac{1}{m})^{n+1} = (1 + \frac{1}{m})(1 + \frac{1}{m})^n < (1 + \frac{1}{m})(1 + \frac{d}{2}) = 1 + \frac{1}{m} + \frac{d}{2} + \frac{d}{2m}$. We require $1 + \frac{1}{m} + \frac{d}{2} + \frac{d}{2m} < 1 + d$, but this is true if $m > 1 + \frac{2}{d}$.

Next, replace b by $b(1 + \frac{1}{m})$ and then we ask for an m such that $(b(1 + \frac{1}{m}))^n = b^n(1 + \frac{1}{m})^n < a$. But we find such an m by the fact that for any positive d , we can find m such that $(1 + \frac{1}{m})^n < 1 + d$, and it suffices to take $d < a/b^n - 1$, this contradicts the supremum property of b , and we are done. \square

Exercise 38 Show that for two $a, b \geq 0$, we have $\sqrt[n]{a \cdot b} = \sqrt[n]{a} \cdot \sqrt[n]{b}$.

Definition 59 *One can now introduce more general rational powers x^q , $q \in \mathbb{Q}$ of a positive real number x as follows. First, one defines $x^0 = 1$, then for a negative integer $-n$, one puts $x^{-n} = 1/x^n$. For $q = n/m \in \mathbb{Q}$ with positive denominator m , one defines $x^{n/m} = (x^{1/m})^n$. One easily checks that this definition does not depend on the fractional representation of q .*

Exercise 39 Prove that for two rational numbers p and q and for two positive real numbers x and y , one has $x^{p+q} = x^p x^q$, $x^{pq} = (x^p)^q$, and $(xy)^p = x^p y^p$.

Exercise 40 Let $a > 1$ and $x > 0$ be real numbers. Show that the set

$$\{q \in \mathbb{Q} \mid \text{there are two integers } m, n, 0 < n, \\ \text{such that } a^m \leq x^n \text{ and } q = m/n\}$$

is non-empty and bounded from above. Its supremum is called the *logarithm of x for basis a* , and is denoted by $\log_a(x)$. It is the fundament of the construction of the exponential function and of the sine and cosine functions.

The general shape of real numbers as equivalence classes of rational Cauchy sequences is quite abstract and cannot, as such, be handled by humans when calculating concrete cases, and a fortiori cannot be handled by computers in hard- and software contexts. Therefore, one looks for more explicit and concrete representations of real numbers. Here is such a construction, which generalizes the b -adic representation of natural numbers discussed in chapter 7. We consider a natural base number $b > 1$ and for an integer n the index set $n] = \{i \mid i \in \mathbb{Z} \text{ and } i \leq n\}$. Let $(a_i)_{i \in n]}$ be a sequence of natural numbers with $0 \leq a_i < b$ and $a_n \neq 0$. Consider the partial sums $\Sigma_j = \sum_{i=n, n-1, n-2, \dots, j} a_i b^i$ for $j \in n]$.

Lemma 91 *The sequence $(\Sigma_j)_{n]}$ of partial sums as defined previously converges to a real number which we denote by $\sum_{i \in n]} a_i b^i$, and, when the base b is clear, by*

$$a_n a_{n-1} \dots a_0 . a_{-1} a_{-2} a_{-3} \dots \quad (9.1)$$

for non-negative n , or, if $n < 0$, by

$$0.00 \dots a_n a_{n-1} \dots, \quad (9.2)$$

i.e., we put zeros on positions $-1, \dots, n+1$ until the first non-zero coefficient a_n appears on position n . The number zero is simply denoted by 0.0 or even 0. The dots are meant to represent the given coefficients. If the coefficients vanish after a finite number of indexes, we can either stop the representation at the last non-vanishing coefficient a_m : $a_n a_{n-1} \dots a_0 . a_{-1} \dots a_{m+1} a_m$, or append any number of zeros, such as $a_n a_{n-1} \dots a_{m+1} a_m 000$.

Proof Let us show that the differences $|\Sigma_k - \Sigma_j|$ of the rational partial sums become arbitrarily small for large k and j . We may suppose that $j > k$ and then have $\Sigma_k - \Sigma_j = a_{j-1}b^{j-1} + \dots + a_k b^k < b^j + \dots + b^{k+1} = b^j(1 + b^{-1} + \dots + b^{k+1-j})$. But we have this quite general formula which is immediately checked by evaluation: $1 + b^{-1} + \dots + b^{-r} = (1 - b^{-r-1})/(1 - b) < 1/(1 - b)$. And therefore $0 \leq \Sigma_k - \Sigma_j < b^j \cdot 1/(1 - b)$. But the right side converges to zero as $j \rightarrow -\infty$, and we are done. \square

Definition 60 *The representation of a real number in the forms (9.1) or (9.2) is called the b -adic representation; in computer science, the synonymous term b -ary is more common. For $b = 2$ it is called the binary representation, for $b = 10$, the decimal representation, and for $b = 16$, it is called the hexadecimal representation. The extension of the b -adic representation to negative real numbers is defined by prepending the sign $-$. This extended representation is also called b -adic representation.*

Proposition 92 (Adic Representation) *Given a base number $b \in \mathbb{N}$, $b > 1$, every real number can be represented in the b -adic representation. The representation is unique up to the following cases: If the coefficients a_i are smaller than $b - 1$ until index m , and equal to $b - 1$ from $m - 1$ until infinity, then this number is equal to the number whose coefficients are the old ones for $i > m$, while the coefficient at index m is $a_m + 1$, and all lower coefficients vanish.*

Proof We may evidently suppose that the real number to be represented is positive, the other cases can be deduced from this. We construct the representation by induction as follows: Observe that we have $b^i > b^j > 0$ whenever $i > j$, i and j being integers. Moreover, b^i converges to 0 as $i \rightarrow -\infty$, and b^i becomes arbitrarily large if $i \rightarrow \infty$. Therefore, there is a unique j such that $b^j \leq x < b^{j+1}$. Within this interval of b powers there is a unique natural a_j with $0 < a_j < b$ such that $a_j b^j \leq x < (a_j + 1)b^j$. Consider the difference $x' = x - a_j b^j$, then we have $0 \leq x' < b^j$ and there is a unique natural $0 \leq a_{j-1} < b$ such that $a_{j-1} b^{j-1} \leq x' < (a_{j-1} + 1)b^{j-1}$. Then $x'' = x' - a_{j-1} b^{j-1} = x - a_j b^j - a_{j-1} b^{j-1}$ has $0 \leq x'' < b^{j-1}$, and we may go on in this way, defining a b -adic number. It is immediate that this number converges to x . The failure of uniqueness in the case where one has $b - 1$ until infinity is left as an exercise to the reader. \square

Example 22 In the decadic representation, the number $0.999\dots$ with a non-terminating sequence of 9s is equal to 1.0. Often, one writes $\dots a_{m+1} \bar{a}_m$ in order to indicate that the coefficient is constant and equal to a_m for all indexes $m, m - 1, m - 2, \dots$. Thus, in the binary representation, $0.\bar{1}$ equals 1.0.

Example 23 Show that for every real number, there is exactly one representation (without the exceptional ambiguity, i.e., avoiding the typical decimal $0.999\dots$ representation) of a real number r in the form

$$r = \pm(a_0.a_{-1}a_{-2}\dots) \cdot b^e$$

with $a_0 \neq 0$ for $r \neq 0$.

This is the so-called *floating point representation*, which for $b = 2$ has been standardized by the IEEE society to computerized representations. See chapter 14 on the first advanced topic for this subject.

At this point we are able to tackle the question whether the cardinalities of \mathbb{N} and \mathbb{R} are equal. This is not the case, in fact, there are many more reals than natural numbers. More precisely, there is an injection $\mathbb{N} \rightarrow \mathbb{R}$, which is induced by the chain $\mathbb{N} \rightarrow \mathbb{Z} \rightarrow \mathbb{Q} \rightarrow \mathbb{R}$ of injections, but there is no bijection $\mathbb{N} \xrightarrow{\sim} \mathbb{R}$. To show this, let us represent the reals by decimal numbers $x = n(x) + 0.x_0x_1\dots$, where $n(x) \in \mathbb{Z}$ and $0 \leq 0.x_0x_1\dots < 1$. Suppose that we take the unique representation $x = n(x) + 0.x_0x_1\dots x_t\bar{0}$ instead of $x = n(x) + 0.x_0x_1\dots (x_t - 1)\bar{9}$, for the case of an ambiguous representation. Now, suppose we are given a bijection $f : \mathbb{N} \xrightarrow{\sim} \mathbb{R}$. The f -image of $m \in \mathbb{N}$ is then $f(m) = n(f(m)) + 0.f(m)_0f(m)_1\dots$. Let $a \in \mathbb{R}$ be the following “antidiagonal” element $a = 0.a_0a_1\dots a_m\dots$. We define $a_m = 2$ if $f(m)_m = 1$ and $a_m = 1$ else. This is a decimal representation of a number a which must occur in our bijection. Suppose that $a = f(m_0)$. Then by construction of a , the digit of $f(m_0)$ at position m_0 after the dot is different from the digit of a at position m_0 after the dot, so a cannot occur, which is a contradiction to the claimed bijection f .

$f(0)$	92736.282109927835...
$f(1)$	2.8 <u>1</u> 4189264762...
$f(2)$	1623.109473637637...
\vdots	\vdots
$f(m_0) = a ?$	0.121...

Fig. 9.4. A tentative “bijection” $f : \mathbb{N} \rightarrow \mathbb{R}$.

9.4 Complex Numbers \mathbb{C}

The last number domain which we need in the general mathematical environment are the complex numbers. The theory that we have developed so far enables us to solve equations such as $ax + b = c$, and we have convergence of all Cauchy sequences, including the standard adic representations. But there is still a strong deficiency: General equations cannot be solved in \mathbb{R} . More precisely, one can easily show that an equation of the form $ax^3 + bx^2 + cx + d = 0$ with real coefficients a, b, c, d and $a \neq 0$ always has a solution (a “root”) in \mathbb{R} . But an equation with an even maximal power of the unknown, such as $ax^4 + bx^3 + cx^2 + dx + e = 0$, cannot be solved in general. Two hundred years ago, mathematicians were searching for a domain of numbers where the special equation $x^2 + 1 = 0$ has a solution. Evidently, such a solution does not exist in \mathbb{R} since in \mathbb{R} any square is non-negative, and therefore $x^2 + 1 \geq 1$.

The solution was rigorously conceptualized by Carl Friedrich Gauss. Instead of working in \mathbb{R} , he considers the plane \mathbb{R}^2 of pairs of real numbers. His trick is to give this set an arithmetic structure, i.e., addition and multiplication, such that the existing \mathbb{R} arithmetic and the entire Cauchy sequence structure are embedded and such that we can effectively solve the critical equation $x^2 + 1 = 0$. But Gauss’ invention is much deeper: It can be shown that in his construction, every equation $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 = 0$ has a solution, this is the fundamental theorem of algebra, see remark 22 in chapter 16. Therefore, the solvability of that single equation $x^2 + 1 = 0$ implies that we are done once for all with this type of equation solving. The type of equation is called *polynomial equation*, we shall come back to this structure in the second part of the book (chapter 15).

Gauss’ complex numbers are defined as follows. We consider the Cartesian product \mathbb{R}^2 , i.e., the set of ordered pairs (x, y) of real numbers. When thinking of the arithmetical structure (addition, multiplication) on \mathbb{R}^2 , we denote this domain by \mathbb{C} and call it the domain of *complex numbers*. Here are the two fundamental operations, addition and multiplication of complex numbers:

Definition 61 *Given two complex numbers (x, y) and $(u, v) \in \mathbb{C}$, we define the sum*

$$(x, y) + (u, v) = (x + u, y + v),$$

while the product is defined by

$$(x, y) \cdot (u, v) = (xu - yv, xv + yu).$$

Here is the sorite for this arithmetic structure:

Sorite 93 Let x, y, z be complex numbers, and denote $0 = (0, 0)$ and $1 = (1, 0)$. Then:

- (i) (Additive associativity) We have $(x + y) + z = x + (y + z)$ and denote this number by $x + y + z$.
- (ii) (Multiplicative associativity) We have $(x \cdot y) \cdot z = x \cdot (y \cdot z)$ and denote this number by $x \cdot y \cdot z$, or also xyz , if no confusion is likely.
- (iii) (Commutativity) We have $x + y = y + x$ and $x \cdot y = y \cdot x$.
- (iv) (Distributivity) We have $x \cdot (y + z) = x \cdot y + x \cdot z$.
- (v) (Additive and multiplicative neutral elements) We have $0 + x = x$ and $1 \cdot x = x$.
- (vi) If $a \neq 0$, then every equation $a \cdot x = b$ has a unique solution; in particular, the solution of $a \cdot x = 1$, the multiplicative inverse of a , is denoted by a^{-1} . The solution of $a + x = 0$, the additive inverse (or negative) of a , is denoted by $-a$.

Proof The statements (i) through (v) follow from the arithmetic properties of reals.

(vi) Let $a = (x, y) \neq (0, 0)$, Then $x^2 + y^2 > 0$. But then $(\frac{1}{x^2+y^2}, 0) \cdot (x, -y) \cdot (x, y) = (1, 0)$, so $a^{-1} = (\frac{1}{x^2+y^2}, 0) \cdot (x, -y)$ is an inverse of a . It is unique by an argument already used in corresponding situations. If z and z' are two inverses of a , then $z = z \cdot 1 = z \cdot (a \cdot z') = (z \cdot a) \cdot z' = 1 \cdot z' = z'$. \square

The geometric view of Gauss is this: We have an injection $\mathbb{R} \rightarrow \mathbb{C}$ which sends a real number a to the complex number $(a, 0)$. Similarly to the embedding $\mathbb{Q} \rightarrow \mathbb{R}$ discussed above, all arithmetic operations, addition and multiplication, “commute” with this embedding, i.e., $(a + b, 0) = (a, 0) + (b, 0)$ and $(a \cdot b, 0) = (a, 0) \cdot (b, 0)$. We therefore identify the real number a with its image $(a, 0)$ in \mathbb{C} . With this convention, denote the complex number $(0, 1)$ by i , and call it the *imaginary unit*. Evidently, $i^2 = -1$. This means that in \mathbb{C} , the equation $x^2 + 1 = 0$ now has a solution, namely $x = i$.

Further, for a complex number $x = (a, b)$, we write $Re(x) = a$ and call it the *real part* of x , similarly we write $Im(x) = b$ and call it the *imaginary*

part of x ; complex numbers of the shape $(0, b)$ are called *imaginary*. Clearly, x is uniquely determined by its real and imaginary parts, in fact:

$$x = (\operatorname{Re}(x), \operatorname{Im}(x)).$$

We then have this crucial result, justifying the geometric point of view:

Proposition 94 For any complex number x , we have a unique representation

$$x = \operatorname{Re}(x) + i \cdot \operatorname{Im}(x)$$

as the sum of a real number (i.e., $\operatorname{Re}(x)$) and an imaginary number (i.e., $i \cdot \operatorname{Im}(x)$).

Proof This is obvious. □

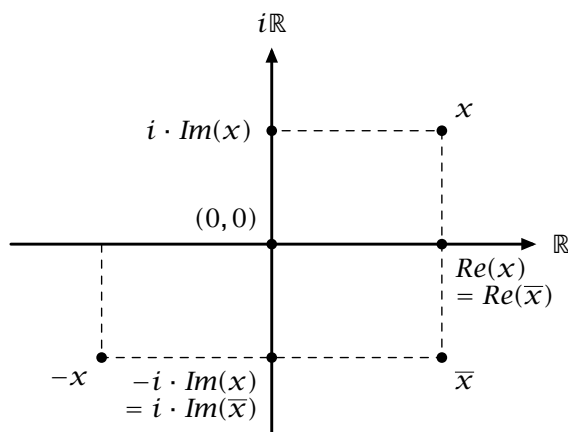


Fig. 9.5. The representation of complex numbers in the plane as introduced by the German mathematician Carl Friedrich Gauss (1777-1855). In the Gauss-plane, the conjugate \bar{x} of x is the point obtained by reflecting x at the abscissa, while $-x$ is the point obtained from x by a rotation of 180° around the origin $(0, 0)$ of the Gauss-plane.

Exercise 41 Using the representation introduced in proposition 94, show the validity of these arithmetical rules:

1. $(a + i \cdot b) + (c + i \cdot d) = (a + c) + i \cdot (b + d),$
2. $(a + i \cdot b) \cdot (c + i \cdot d) = (ac - bd) + i \cdot (ad + bc).$

The complex numbers have a rich inner structure which is related to the so-called conjugation.

Definition 62 The conjugation is a map $\mathbb{C} \rightarrow \mathbb{C} : x \mapsto \bar{x}$ defined by $\bar{x} = \operatorname{Re}(x) - i \cdot \operatorname{Im}(x)$, i.e., $\operatorname{Re}(\bar{x}) = \operatorname{Re}(x)$ and $\operatorname{Im}(\bar{x}) = -\operatorname{Im}(x)$.

The norm of a complex number x is defined by $|x| = \sqrt{x \cdot \bar{x}}$, which is a non-negative real, since $x \cdot \bar{x} = \operatorname{Re}(x)^2 + \operatorname{Im}(x)^2 \geq 0$.

Observe that the norm of a complex number $x = a + i \cdot b$ is the Euclidean length of the vector $(a, b) \in \mathbb{R}^2$ known from high school.

Sorite 95 Let $x, y \in \mathbb{C}$. Then

- (i) $x = \bar{x}$ iff $x \in \mathbb{R}$, and $\bar{x} = -x$ iff x is imaginary,
- (ii) $|x| = 0$ iff $x = 0$,
- (iii) $\operatorname{Re}(x) = \frac{x + \bar{x}}{2}$ and $\operatorname{Im}(x) = \frac{x - \bar{x}}{2i}$,
- (iv) if $x \neq 0$, then the multiplicative inverse of x is $x^{-1} = |x|^{-2} \cdot \bar{x}$,
- (v) $\overline{\bar{x}} = x$; in particular, conjugation is a bijection,
- (vi) $\overline{x + y} = \bar{x} + \bar{y}$,
- (vii) $\overline{x \cdot y} = \bar{x} \cdot \bar{y}$,
- (viii) if x is real, then $|x|$ in the sense of real numbers coincides with $|x|$ in the sense of complex numbers, which justifies the common notation,
- (ix) $|x \cdot y| = |x| \cdot |y|$,
- (x) (Triangle inequality) $|x + y| \leq |x| + |y|$.

Proof The only non-trivial statement is the triangle inequality. It suffices to show that $|x + y|^2 \leq (|x| + |y|)^2$. This gives us the inequality $y\bar{x} + x\bar{y} \leq 2|x||y|$, and then, by putting $a = x\bar{y}$, we get inequality $a + \bar{a} \leq |a| + |\bar{a}|$ which is obvious by simple explication of the coordinates of the complex number a . \square

Categories of Graphs

In this chapter, we introduce the concept of a graph. Note that, this is homonymous with but really different from the already known concept of a graph *relation*. Please do observe this historically grown ambiguity. Of course, both concepts are related by the fact that they allude to something being drawn: The graph of a function is just what in nice cases will be drawn as a graphical representation of that function, whereas the other meaning is related to the graphical representation of assignments between nodes of a processual assembly—the concrete situations are completely different.

As a preliminary construction we need this setup: Given a set V , always finite in this context, we have the Cartesian product $V^2 = V \times V$. In addition we define the *edge set* as ${}^2V = \{a \subset V \mid 1 \leq \text{card}(a) \leq 2\}$. It has this name because it parametrizes the set of all undirected lines, i.e., edges, including single points (“loop at x ”), between any two elements of V . We have the evident surjection $|\cdot| : V^2 \rightarrow {}^2V : (x, y) \mapsto \{x, y\}$, which has a number of sections which we (somewhat ambiguously) denote by $\vec{\cdot} : {}^2V \rightarrow V^2$, i.e., $|\cdot| \circ \vec{\cdot} = \text{Id}_{{}^2V}$.

Exercise 42 Give reasons for the existence of sections $\vec{\cdot}$ of $|\cdot|$.

We further denote by $\ast : V^2 \rightarrow V^2 : (x, y) \mapsto (y, x)$ the exchange bijection, and note that $(x, y)^{\ast\ast} = (x, y)$.

10.1 Directed and Undirected Graphs

Definition 63 A directed graph or digraph is a map $\Gamma : A \rightarrow V^2$ between finite sets. The elements of the set A are called arrows, the elements of V are called vertexes of the directed graph. By the universal property of the Cartesian product (see proposition 57), these data are equivalent to the data of two maps, $\text{head}_\Gamma : A \rightarrow V$ and $\text{tail}_\Gamma : A \rightarrow V$; more precisely, we set $\text{tail}_\Gamma = \text{pr}_1 \circ \Gamma$ and $\text{head}_\Gamma = \text{pr}_2 \circ \Gamma$. For $a \in A$ and $h = \text{head}_\Gamma(a)$ and $t = \text{tail}_\Gamma(a)$, we also write $a : t \rightarrow h$ or $t \xrightarrow{a} h$.

The intuitive meaning of a directed graph is that we are given a set of objects some of which are connected by arrows, and that there may exist several “parallel” arrows between a pair of given tail and head objects.

Notation 9 An intuitive two-dimensional notation of digraphs involves drawing them in the plane: vertexes are arbitrarily placed as dots with a label attached (the label denotes the element from V). Arrows are drawn as curves from the the dot representing the tail vertex to the dot representing the head vertex, with an arrow head attached. A label is attached to the curve denoting the corresponding element from A . For convenience, dots may be omitted and the labels of the vertexes put in their place. Both vertex and arrow labels can be omitted, if the particular labeling is immaterial to the specific situation.

Example 24 The graph Γ consists of the set of vertexes $V = \{B, C, D, F\}$ and the set of arrows $A = \{a, b, c, d, e, g\}$ (figure 10.1).

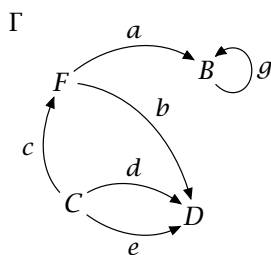


Fig. 10.1. The graph $\Gamma : A \rightarrow V^2$.

If the simplified notation is used, the map Γ is given by:

$$F \xrightarrow{a} B, F \xrightarrow{b} D, C \xrightarrow{c} F, C \xrightarrow{d} D, C \xrightarrow{e} D, B \xrightarrow{g} B$$

Taking the arrow a as an example, we have

$$\text{tail}_\Gamma(a) = F \text{ and } \text{head}_\Gamma(a) = B.$$

Example 25 For every directed graph $\Gamma : A \rightarrow V^2$, we have the *dual graph* $\Gamma^* = ?^* \circ \Gamma$, evidently $\Gamma^{**} = \Gamma$.

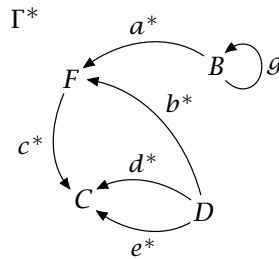


Fig. 10.2. The dual graph Γ^* of the graph Γ from example 24 (figure 10.1).

Example 26 Given a binary relation A in a set V , we have the associated directed graph defined by $\Gamma = A \subset V^2$, the inclusion of A in V^2 . So the arrows here identify with the associated ordered pairs. In particular, the *complete* directed graph $\text{CompDi}(V)$ over a set V is defined by the identity on the Cartesian product $A = V^2$ (figure 10.3)). Clearly, $\text{CompDi}(V)^* \cong \text{CompDi}(V)$. The *discrete* directed graph $\text{DiDi}(V)$ over the set V is the one defined by the empty set of arrows (figure 10.4).

Example 27 In process theory, a labeled transition system (LTS) is a subset $T \subset S \times \text{Act} \times S$ of the Cartesian product of a *state space* S , a set Act of *labels*, together with a selected start state $s_0 \in S$. For each state $s \in S$, there is a number of transitions, e.g., the triples $(s, l, t) \in T$, parameterizing “transitions from state s to state t via the transition type l ”. Defining the directed graph $\Gamma = T \rightarrow S^2$ via $\text{head}_\Gamma = \text{pr}_3$ and $\text{tail}_\Gamma = \text{pr}_1$, we associate a directed graph with the LTS, together with a distinct vertex s_0 . Show that conversely, every directed graph, together with a distinct vertex defines an LTS. How are these two constructions related?

Example 28 If for a directed graph $\Gamma = A \subset V^2$, the vertex set is a disjoint union $V = V_1 \cup V_2$ of subsets, and if for all arrows a , we have $\Gamma(a) \in$

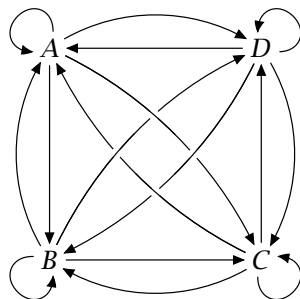


Fig. 10.3. The complete digraph $CompDi(V)$ over the vertex set $V = \{A, B, C, D\}$.

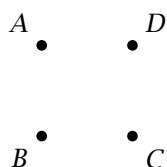


Fig. 10.4. The discrete digraph $DiDi(V)$ over the vertex set $V = \{A, B, C, D\}$.

$V_1 \times V_2 \cup V_2 \times V_1$, then the graph is called *bipartite* (with respect to V_1 and V_2). For any partition $V = V_1 \cup V_2$ of a finite set V , one has the *complete bipartite digraph* $BipDi(V_1, V_2)$, defined by the inclusion $V_1 \times V_2 \cup V_2 \times V_1 \subset V^2$.

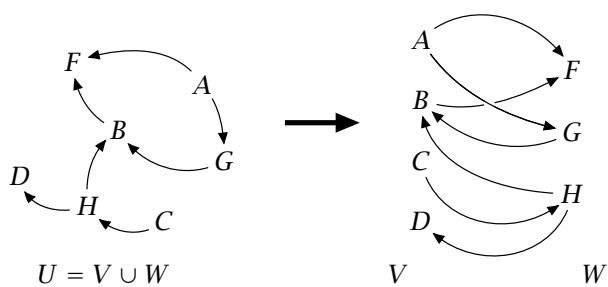


Fig. 10.5. A bipartite graph with vertexes $U = V \cup W$.

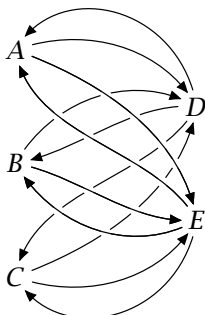


Fig. 10.6. A complete bipartite graph with vertexes $U = \{A, B, C\} \cup \{D, E\}$.

Example 29 In automata theory, a special type of directed graphs is called *Petri net*, which was introduced by Carl Adam Petri in 1962. A Petri net is specified by two sets: the set P of *places*, and the set Tr of *transitions*. It is supposed that some places are related “by input arcs” to transitions as *input places*, whereas some places are related “by output arcs” to transitions as *output places*. It is also assumed that every transition has at least one input and one output place, and that an input place cannot be an output place of the same transition. This means that we can view a Petri net as an LTS (except that no initial state s_0 is specified) where the labels are the transitions, and where the triples $(p, t, q) \in P \times Tr \times P$ are the elements of the ternary state space relation T in example 27. The axiom eliminating the possibility “input = output” means that the directed graph of the Petri net has no *loop*, i.e., no arrow of type $p \xrightarrow{a} p$.

The next subject relates set theory to graph theory.

Definition 64 For a set x and for $n \in \mathbb{N}$, we define inductively $\bigcup^0 x = x$ and $\bigcup^{n+1} x = \bigcup(\bigcup^n x)$. The set x is called *totally finite* iff there is a natural number m such that $\bigcup^m x = \emptyset$. Call the minimal such m the *level* $lev(x)$ of x .

Example 30 Let $x = \{\{\{\{\emptyset\}, \emptyset\}, \emptyset, \{\emptyset\}\}, \{\{\emptyset\}\}\}$. Then $x_0 = \bigcup^0 x = x$, $x_1 = \bigcup^1 x = \bigcup x_0 = \{\{\{\emptyset\}, \emptyset\}, \emptyset, \{\emptyset\}\}$, $x_2 = \bigcup^2 x = \bigcup x_1 = \{\{\emptyset\}, \emptyset\}$, $x_3 = \bigcup^3 x = \bigcup x_2 = \{\emptyset\}$, and, finally, $x_4 = \bigcup^4 x = \bigcup x_3 = \emptyset$. Thus x is totally finite and $lev(x) = 4$.

The set \mathbb{N} of natural numbers is not totally finite. The set $a = \{a\}$ is finite, but not totally finite, as can be easily verified.

Clearly, if x is totally finite, and if $y \in x$, then y is also totally finite and $lev(y) < lev(x)$. We now associate the notation $Fi(x)$ to any totally finite set as follows (Paul Finsler (1894–1970) was a mathematician at the University of Zurich).

Definition 65 *The vertex set V of the Finsler digraph $Fi(x)$ of a totally finite set x is the union $V = \{x\} \cup \bigcup_{i=0, \dots, lev(x)-1} \bigcup^i x$. Observe that all pairs $\{x\}, \bigcup^i x$ of sets are mutually disjoint, otherwise, x would not be totally finite! The arrow set is the set $\{(r, s) \mid s \in r\} \subset V^2$, i.e., the arrows $r \xrightarrow{a} s$ of $Fi(x)$ correspond to the element (\in) relation.*

Example 31 With x defined as in example 30, the vertex set of $Fi(x)$ is $V = \{x\} \cup \bigcup_{i=0,1,2,3} \bigcup^i x = \{x\} \cup \bigcup_{i=0,1,2,3} x_i = \{x\} \cup x_0 \cup x_1 \cup x_2 \cup x_3$. Denoting $a = \{\{\{\emptyset\}, \emptyset\}, \emptyset, \{\emptyset\}\}$, $b = \{\{\emptyset\}\}$, $c = \{\{\emptyset\}, \emptyset\}$, and $d = \{\emptyset\}$, then $V = \{x\} \cup \{a, b\} \cup \{c, d, \emptyset\} \cup \{d, \emptyset\} \cup \{\emptyset\} = \{x, a, b, c, d, \emptyset\}$. The arrow set is $\{(x, a), (x, b), (a, c), (a, d), (a, \emptyset), (b, d), (c, d), (c, \emptyset), (d, \emptyset)\}$. The resulting Finsler digraph is shown in figure 10.7.

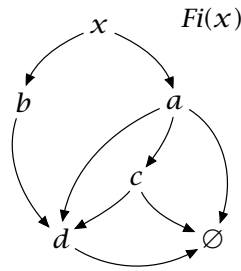


Fig. 10.7. The Finsler digraph $Fi(x)$ of the set x from example 31.

The Finsler digraphs characterize totally finite sets, in other words, we may redefine such sets starting from graphs. This is a new branch of theoretical computer science used in parallel computing theory, formal ontologies, and artificial intelligence. Important contributions to this branch of computer science, mathematics, and formal logic have been made by Jon Barwise, Lawrence Moss [3], and Peter Aczel [1].

Example 32 In an object-oriented language, for example Java, we consider a class library which we suppose being given as a set L , the elements

C of which represent the library’s classes, including (for simplicity) the primitive type classes of integers, floating point numbers, strings, and booleans. For each class C we have a number of fields (instance variables) defined by their names F and class types $T \in L$. This defines a digraph Λ_L , the vertex set being L , and the arrows being either the triples (C, F, T) where F is the name of a field of class C , and where T is the type class of F , or the pairs (C, S) , where S is the direct superclass of C ; we set $head_\Lambda(C, F, T) = T$ and $tail_\Lambda(C, F, T) = C$, or, for superclass arrows, $head_\Lambda(C, S) = S$ and $tail_\Lambda(C, S) = C$.

If one forgets about the direction in a digraph, the remaining structure is that of an “undirected” graph, or simply “graph”. We shall henceforth always write *digraph* for a directed graph, and *graph* for an undirected graph, if confusion is unlikely.

Definition 66 An undirected graph (or simply graph) is a map $\Gamma : A \rightarrow {}^2V$ between finite sets. The elements of the set A are called edges, the elements of V are called vertexes of the graph. For $a \in A$ and $\Gamma(a) = \{x, y\}$, we also write $x \xrightarrow{a} y$, which is the same as $y \xrightarrow{a} x$.

The two-dimensional notation for graphs is similar to that of digraphs, except that arrow heads are omitted.

Example 33 For each directed graph Γ , one generates the *associated graph* $|\Gamma| = |\cdot| \circ \Gamma$, and for any given (undirected) graph Γ , one generates an *associated directed graph* $\vec{\Gamma} = \vec{\cdot} \circ \Gamma$, the latter construction supposing that a section $\vec{\cdot}$ for the graph’s vertex set is given.

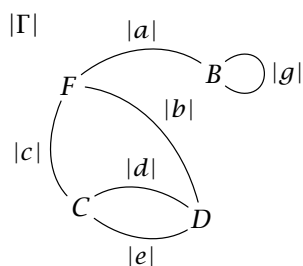


Fig. 10.8. The graph $|\Gamma|$ associated to the graph from example 24 (figure 10.1).

Example 34 A graph $\Gamma : A \rightarrow {}^2V$ is *complete* iff Γ is a bijection onto the subset ${}^2V - \{\{x\} \mid x \in V\} \subset {}^2V$. The complete graph $Comp(V)$ of a set V is the inclusion ${}^2V - \{\{x\} \mid x \in V\} \subset {}^2V$, i.e., intuitively the set of all edges between any two different points in V . A graph $\Gamma : A \rightarrow {}^2V$ is *bipartite*, iff there is a partition $V = V_1 \cup V_2$ such that for all edges a , $\Gamma(a) = \{x, y\}$ with $x \in V_1$ and $y \in V_2$. The complete bipartite graph $Bip(V_1, V_2)$ for two disjoint sets V_1 and V_2 is the embedding $\{\{x, y\} \mid x \in V_1, y \in V_2\} \subset {}^2V$. The discrete (undirected) graph over the vertex set V is denoted by $Di(V)$.

10.2 Morphisms of Digraphs and Graphs

Evidently, there are many (directed or undirected) graphs which look essentially the same, similarly to sets which are essentially the same, in the sense that they are equipollent, i.e., they have same cardinality. In order to control this phenomenon, we need a means to compare graphs in the same way as we had to learn how to compare sets by use of functions.

Definition 67 Let $\Gamma : A \rightarrow V^2$ and $\Delta : B \rightarrow W^2$ be two digraphs. A morphism $f : \Gamma \rightarrow \Delta$ of digraphs is a pair $f = (u, v)$ of maps $u : A \rightarrow B$ and $v : V \rightarrow W$ such that $v^2 \circ \Gamma = \Delta \circ u$, in other words, for any arrow $t \xrightarrow{a} h$ in Γ , we have $v(t) \xrightarrow{u(a)} v(h)$. This means that we have the following commutative diagram:

$$\begin{array}{ccc} A & \xrightarrow{\Gamma} & V^2 \\ \downarrow u & & \downarrow v^2 \\ B & \xrightarrow{\Delta} & W^2 \end{array}$$

In particular, the identity $Id_\Gamma = (Id_A, Id_V)$ is a morphism, and, if $f = (u, v) : \Gamma \rightarrow \Delta$ and $g = (u', v') : \Delta \rightarrow \Theta$ are two morphisms, then their composition is a morphism $g \circ f = (u' \circ u, v' \circ v) : \Gamma \rightarrow \Theta$.

We denote the set of digraph morphisms $f : \Gamma \rightarrow \Delta$ by $Digraph(\Gamma, \Delta)$.

Example 35 Figure 10.9 shows two digraphs Γ and Δ and a morphism $f = (u, v)$. The map v on the vertexes is drawn with light gray arrows, the map u on edges is drawn with dark gray arrows.

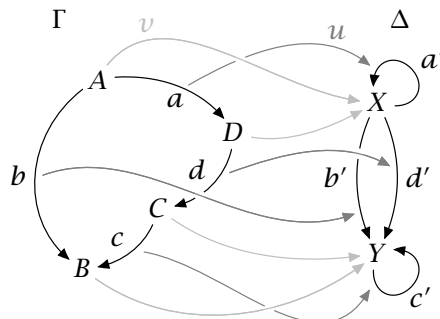


Fig. 10.9. A morphism of digraphs.

Sorite 96 Call the two maps u and v defining a digraph morphism $f = (u, v)$ its components.

- (i) If $f : \Gamma \rightarrow \Delta, g : \Delta \rightarrow \Theta, h : \Theta \rightarrow \Psi$ are morphisms of digraphs, then $h \circ (g \circ f) = (h \circ g) \circ f$, which we denote (as usual) by $h \circ g \circ f$.
- (ii) Given $f : \Gamma \rightarrow \Delta$, there is a morphism $k : \Delta \rightarrow \Gamma$ such that $k \circ f = Id_{\Gamma}$ and $f \circ k = Id_{\Delta}$ iff the components u and v of $f = (u, v)$ are bijections. In this case, f is called an isomorphism of digraphs.

Proof (i) Associativity of digraph morphisms follows directly from the associativity of set maps which define digraph morphisms.

(ii) Since the composition of two digraph morphisms is defined by the set-theoretic composition of their two components, the claim is immediate from the synonymous facts for set maps. □

Remark 13 Informally speaking, the system *Digraph* of digraphs Γ, Δ, \dots together with their sets *Digraph*(Γ, Δ) of morphisms, including their associative composition (sorite 96) is called the *category* of digraphs. A formal definition of a category will be given in volume II of this book. Evidently, we already have encountered the structure of category for sets and their functions. Categories constantly appear in all of mathematics, they are the most powerful unifying structure of modern mathematics and computer science.

Example 36 An immediate class of morphisms is defined by *directed subgraphs* (or *subdigraphs*). More precisely, given a digraph $\Gamma : A \rightarrow V^2$, if inclusions $u : A' \subset A$ and $v : V' \subset V$ of subsets A' and V' are such that $\Gamma(A') \subset (V')^2$, then we have an *induced digraph* $\Gamma' : A' \rightarrow (V')^2$, and a subdigraph inclusion morphism $(u, v) : \Gamma' \subset \Gamma$. In particular, if for a

subset $V' \subset V$, we take $A' = \Gamma^{-1}((V')^2)$, we obtain the subdigraph $\Gamma|_{V'}$ induced on V' .

The set cardinality classifies sets up to bijections. In particular, for finite sets, the natural number $\text{card}(a)$ is a classifying property. Similarly, for digraphs we also want to have prototypes of digraphs such that each digraph is isomorphic to such a prototype. To begin with we have this:

Exercise 43 Show that each digraph $\Gamma : A \rightarrow V^2$ is isomorphic to a digraph $\Gamma' : A \rightarrow \text{card}(V)^2$.

Exercise 44 Prove the equality $\text{BipDi}(V_1, V_2) = \text{BipDi}(V_2, V_1)$ and the isomorphism $\text{BipDi}(V_1, V_2)^* \simeq \text{BipDi}(V_1, V_2)$.

For a digraph $\Gamma : A \rightarrow V^2$ and $e, f \in V$, we consider the sets $A_{e,f} = \Gamma^{-1}(e, f) = \{a \mid a \in A, \text{tail}(a) = e, \text{head}(a) = f\}$.

Definition 68 Let $\Gamma : A \rightarrow V^2$ be a digraph, and fix a bijection $c : \text{card}(V) \xrightarrow{\sim} V$. The adjacency matrix of Γ (with respect to c) is the function $\text{Adj}_c(\Gamma) : \text{card}(V)^2 \rightarrow \mathbb{N}$ defined by $(i, j) \mapsto \text{card}(A_{c(i), c(j)})$.

We shall deal extensively with matrixes in chapter 21. But we will already now show the standard representation of a matrix, and in particular of the adjacency matrix. It is a function of a finite number of pairs (i, j) . Such a function is usually represented in a graphical form as a tabular field with n rows and n columns, and for each row number i and column number j , we have an entry showing the function value $\text{Adj}_c(i, j)$, i.e.,

$$\text{Adj}_c = \begin{pmatrix} \text{Adj}_c(0,0) & \text{Adj}_c(0,1) & \dots & \text{Adj}_c(0,n-1) \\ \text{Adj}_c(1,0) & \text{Adj}_c(1,1) & \dots & \text{Adj}_c(1,n-1) \\ \vdots & \vdots & & \vdots \\ \text{Adj}_c(n-1,0) & \text{Adj}_c(n-1,1) & \dots & \text{Adj}_c(n-1,n-1) \end{pmatrix}$$

Observe however that here, the row and column indexes start at 0, whereas in usual matrix theory, these indexes start at 1, i.e., for usual matrix notation, the entry at position (i, j) is our position $(i-1, j-1)$.

Example 37 For simplicity's sake, let the vertexes of a digraph Γ be the natural numbers less than 6. Thus, the bijection c is the identity. The adjacency matrix of Γ is shown below the digraph in figure 10.10.

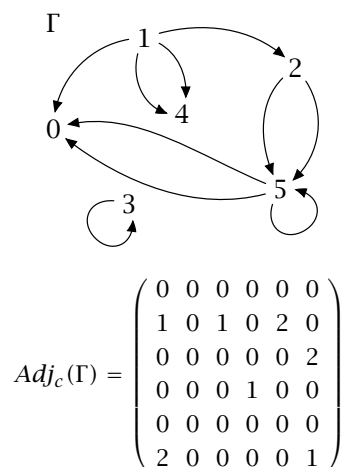


Fig. 10.10. Adjacency matrix of a digraph.

The following numerical criterion for isomorphic digraphs is very important for the computerized representations of digraphs:

Proposition 97 *If Γ and Δ are two digraphs such that $Adj(\Gamma) = Adj(\Delta)$, then they are isomorphic, i.e., there is an isomorphism $\Gamma \xrightarrow{\sim} \Delta$.*

Proof Suppose that for two digraphs $\Gamma : A \rightarrow V^2$ and $\Delta : B \rightarrow W^2$, $Adj(\Gamma) = Adj(\Delta)$. Then the number of rows of the two matrixes is the same, and therefore we have a bijection $\nu : V \rightarrow W$ such that the matrix index (i, j) in $Adj(\Gamma)$ corresponds to the same index of $Adj(\Delta)$. Moreover, we have $A = \bigsqcup_{x \in V^2} \Gamma^{-1}(x)$ and $B = \bigsqcup_{y \in W^2} \Delta^{-1}(y)$. But since by hypothesis $card(\Gamma^{-1}(x)) = card(\Delta^{-1}(\nu^2(x)))$ for all $x \in V$, setting $\nu^2 = \nu \times \nu$, we have bijections $u_x : \Gamma^{-1}(x) \rightarrow \Delta^{-1}(\nu^2(x))$ and their disjoint union yields a bijection $u : A \rightarrow B$ which defines the desired isomorphism (u, ν) . \square

So the adjacency matrix of a digraph describes the digraph “up to isomorphisms”. The converse is not true, but it can easily be said what is missing: Essentially, we have to take into account the bijection c , which labels the vertexes. In fact, if $\Gamma \xrightarrow{\sim} \Delta$, then their adjacency matrixes are related to each other by conjugation with a permutation matrix relabeling the vertexes.

Example 38 The bipartite digraphs can now be denoted by simple numbers, i.e., if n and m are natural numbers, we have the bipartite digraph $BipDi(n, m)$, and every bipartite digraph $BipDi(V_1, V_2)$ with $card(V_1) = n$

and $\text{card}(V_2) = m$ is isomorphic to $\text{BipDi}(n, m)$. This applies also to complete or discrete digraphs. Given a natural number n , the complete digraph $\text{CompDi}(n)$ is isomorphic to any $\text{CompDi}(V)$ such that $\text{card}(V) = n$. And the discrete digraph $\text{DiDi}(n)$ is isomorphic to any $\text{DiDi}(V)$ if $\text{card}(V) = n$.

An important type of morphisms with special digraphs, namely chains, as domains is defined as follows:

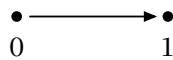
Definition 69 Given $n \in \mathbb{N}$, the directed chain $[n]$ of length n is the digraph on the vertex set $V = n + 1$ with the n arrows $(i, i + 1), i \in n$. In particular, if $n = 0$, then $[0] = \text{DiDi}(1)$ (discrete with one vertex), whereas in general, the leftmost arrow is $0 \longrightarrow 1$, followed by $1 \longrightarrow 2$, etc., up to $(n - 1) \longrightarrow n$.

Definition 70 Given a digraph Γ and $n \in \mathbb{N}$, a path of length n in Γ is a morphism $p : [n] \rightarrow \Gamma$, write $l(p) = n$ for the length. (Equivalently, p may be described as a sequence of arrows $(a_i)_{i=1, \dots, n}$ in Γ such that for every $i < n$, $\text{head}(a_i) = \text{tail}(a_{i+1})$.) If $p(0) = v$ and $p(n) = w$, one also says that p is a path from v to w . If there is a path from v to w , we say, that w is reachable from v . A path of length 0—just one vertex v in Γ —is called the lazy path at v , and also denoted by v . A non-lazy path $p : [n] \rightarrow \Gamma$ such that $p(0) = p(n)$ is called a cycle in Γ . A cycle of length 1 at a vertex v is called a loop at v .

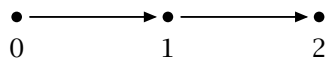
Directed chain $[0]$ of length 0



Directed chain $[1]$ of length 1



Directed chain $[2]$ of length 2



Directed chain $[4]$ of length 4

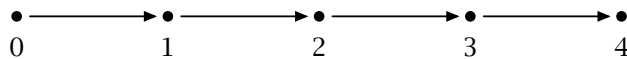


Fig. 10.11. Directed chains.

Example 39 In the digraph of figure 10.12, the light gray arrows form a path from A to B , the dark gray ones a cycle from A to A , and the dashed arrow is a loop at D .

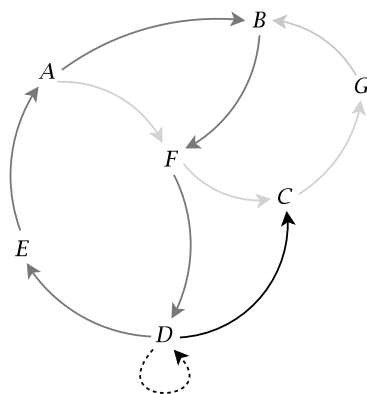
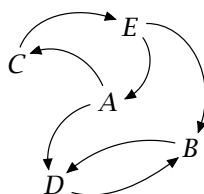


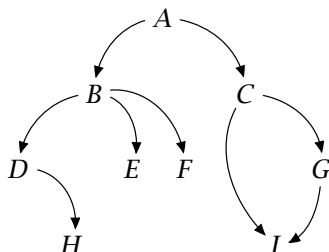
Fig. 10.12. Paths and cycles in a digraph.

Definition 71 A vertex v in a digraph Γ such that for every vertex w in Γ , there is a path from v to w , is called a root or source of Γ , a root in the dual digraph is called a co-root or sink of Γ . A digraph without directed cycles and with a (necessarily unique) root is called a directed tree. A vertex v in a directed tree, which is not the tail of an arrow, is called a leaf of the tree.

Example 40 The roots in the figure are $\{A, C, E\}$ and the co-roots are $\{D, B\}$. Note that if a root lies on a cycle, all vertexes on the cycle are roots. The analogue statement is valid for co-roots.



The following figure shows a tree with root A and no co-root, with leaves $\{D, E, F, H, I\}$.



Exercise 45 Given two paths p and q of lengths n and m , respectively, in a digraph Γ such that p ends where q begins, we have an evident *composition qp of paths* of length $l(qp) = n + m = l(q) + l(p)$, defined by joining the two arrow sequences in the connecting vertex. Composition of paths is associative, and the composition with a lazy path at v , if it is defined, yields $vp = p$, or $pv = p$.

This allows us to characterize totally finite sets by their Finsler digraphs (see definition 65). To this end, we consider the following subgraph construction: If Γ is a graph and v a vertex, we define $v\rangle$ to be the set of all vertexes w such that there is a path from v to w . The directed sub-

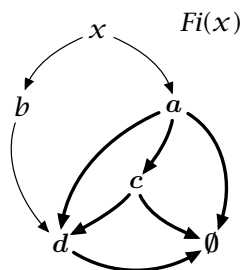


Fig. 10.13. The vertexes and edges of the subdigraph $a\rangle$ of $Fi(x)$ from figure 10.7 are shown in bold.

graph induced on $v\rangle$ is also denoted by $v\rangle$ and is called the *subdigraph generated by v* .

Proposition 98 A digraph Γ is isomorphic to a Finsler digraph $Fi(x)$ of a totally finite set x iff it is a directed tree such that for any two vertexes v_1 and v_2 , if $v_1\rangle$ is isomorphic to $v_2\rangle$, then $v_1 = v_2$.

Proof Let x be totally finite. Then in the Finsler digraph $Fi(x)$, every $y \in x$ is reached from the vertex x . And if $z \in \bigcup^{n+1} x = \bigcup \bigcup^n x$, then it is reached from a selected element in $\bigcup^n x$. So x is the root of $Fi(x)$. If y is a vertex in $Fi(x)$, then clearly $y \in Fi(y)$. Let us show by induction on $lev(y)$ that an isomorphism $Fi(y) \cong Fi(z)$ implies $y = z$. In fact, if $lev(y) = 0$, then $Fi(y)$ and $Fi(z)$ are both one-point digraphs, and both y and z must be the empty set. In general, if $Fi(y) \cong Fi(z)$, then the roots must also correspond under the given isomorphism, and therefore there is a bijection between the vertexes y_i reached from y by an arrow, and the vertexes z_i reached from z by an arrow. If z_i corresponds to y_i , then also $z_i \in Fi(y_i)$, and therefore, by recursion, $y_i = z_i$, which implies $y = z$. Conversely, if a directed tree $F : V \rightarrow A^2$ is such that any two vertexes v_1 and v_2 with $v_1 \in Fi(v_2)$ must be equal, then it follows by induction on the length of the maximal path from a given vertex that F is isomorphic to the Finsler digraph of a totally finite set. In fact, if a directed tree has maximal path length 0 from the root, it is the Finsler digraph of the empty set. In general, the vertexes $y_i, i = 1, \dots, k$ reached from the root y by an arrow have a shorter maximal path than the root and are also directed trees with the supposed conditions. So by recursion, they are Finsler digraphs $Fi(t_i)$ of totally finite, mutually different sets t_i . Then evidently, y is isomorphic to the Finsler digraph of the set $\{t_i \mid i = 1, \dots, k\}$. \square

We now turn to the subject of morphisms between *undirected* graphs. We need the undirected variant of the square map $v^2 : V^2 \rightarrow W^2$ associated with a map $v : V \rightarrow W$, i.e., $v^2 : V^2 \rightarrow W^2 : \{x, y\} \mapsto \{v(x), v(y)\}$.

Definition 72 Let $\Gamma : A \rightarrow V^2$ and $\Delta : B \rightarrow W^2$ be two graphs. A morphism $f : \Gamma \rightarrow \Delta$ of graphs is a pair $f = (u, v)$ of maps $u : A \rightarrow B$ and $v : V \rightarrow W$ such that $v^2 \circ \Gamma = \Delta \circ u$, in other words, for any edge $t \xrightarrow{a} h$ in Γ , we have $v(t) \xrightarrow{u(a)} v(h)$. This means we have the commutative diagram

$$\begin{array}{ccc}
 A & \xrightarrow{\Gamma} & V^2 \\
 u \downarrow & & \downarrow v^2 \\
 B & \xrightarrow{\Delta} & W^2
 \end{array}$$

In particular, the identity $Id_\Gamma = (Id_A, Id_V)$ is a morphism, and, if $f = (u, v) : \Gamma \rightarrow \Delta$ and $g = (u', v') : \Delta \rightarrow \Theta$ are two morphisms, then their composition is a morphism $g \circ f = (u' \circ u, v' \circ v) : \Gamma \rightarrow \Theta$.

Exercise 46 Show that the mapping $\Gamma \mapsto |\Gamma|$ can be extended to morphisms, i.e., if $f = (u, v) : \Gamma \rightarrow \Delta$, is a morphism, then so is $|f| : |\Gamma| \rightarrow |\Delta|$. Also, $|Id_\Gamma| = Id_{|\Gamma|}$, and $|f \circ g| = |f| \circ |g|$.

The following sorite looks exactly like its corresponding version for digraphs:

Sorite 99 *Call the two maps u and v defining a graph morphism $f = (u, v)$ its components.*

- (i) *If $f : \Gamma \rightarrow \Delta$, $g : \Delta \rightarrow \Theta$, and $h : \Theta \rightarrow \Psi$ are morphisms of graphs, then $h \circ (g \circ f) = (h \circ g) \circ f$, which we (as usual) denote by $g \circ h \circ f$.*
- (ii) *Given $f : \Gamma \rightarrow \Delta$, there is a morphism $k : \Delta \rightarrow \Gamma$ such that $k \circ f = Id_\Gamma$ and $f \circ k = Id_\Delta$ iff the components u and v of $f = (u, v)$ are bijections. In this case, f is called an isomorphism of graphs.*

Proof This results from the corresponding set-theoretic facts much as it did for the digraph sorite 96. \square

Example 41 In complete analogy with example 36 on directed subgraphs, we have subgraphs of undirected graphs, i.e., we require that for a graph $\Gamma : A \rightarrow {}^2V$ and two subsets $u : A' \subset A$ and $v : V' \subset V$, Γ restricts to $\Gamma' : A' \rightarrow {}^2V'$, and in particular, if $A' = \Gamma^{-1}({}^2(V'))$, we get the subgraph $\Gamma|_{V'}$ induced on V' .

As for digraphs, clearly every graph with vertex set V is isomorphic to a graph the vertexes of which are elements of the natural number $n = \text{card}(V)$. If $\{e, f\} \in {}^2V$, we denote $A_{\{e, f\}} = \Gamma^{-1}(\{e, f\})$.

Definition 73 *Let $\Gamma : A \rightarrow {}^2V$ be a graph, and fix a bijection $c : \text{card}(V) \rightarrow V$. The adjacency matrix of Γ (with respect to c) is the function $Adj_c(\Gamma) : \text{card}(V)^2 \rightarrow \mathbb{N} : (i, j) \mapsto \text{card}(A_{\{c(i), c(j)\}})$.*

Example 42 The adjacency matrix for $|\Gamma|$ (where Γ is the same as in example 37) is:

$$Adj_c(|\Gamma|) = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 & 2 \\ 1 & 0 & 1 & 0 & 2 & 0 \\ 0 & 1 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 & 0 & 0 \\ 2 & 0 & 2 & 0 & 0 & 1 \end{pmatrix}$$

Obviously the adjacency matrix of an undirected graph is symmetrical, i.e., $Adj_c(\Gamma)(i, j) = Adj_c(\Gamma)(j, i)$ for all pairs i, j .

Proposition 100 *If Γ and Δ are two graphs such that $\text{Adj}(\Gamma) = \text{Adj}(\Delta)$, then they are isomorphic.*

Proof The proof of this proposition is completely analogous to the proof of the corresponding proposition 97 for digraphs. \square

Definition 74 *Given $n \in \mathbb{N}$, the chain $|n|$ of length n is the graph on the vertex set $V = n + 1$ such that we have the n edges $\{i, i + 1\}, i \in n$. In particular, if $n = 0$, then $|0| = \text{Di}(1)$ (discrete with one vertex), whereas in general, the leftmost edge is $0 \text{ --- } 1$, followed by $1 \text{ --- } 2$, etc., up to $(n - 1) \text{ --- } n$.*

Example 43 For any natural number n , we have the complete graph $\text{Comp}(n)$, which in literature is often denoted by K_n , it is isomorphic to any complete graph $\text{Comp}(V)$ with $\text{card}(V) = n$. For two natural numbers n and m , we have the complete bipartite graph $\text{Bip}(n, m)$, often denoted by $K_{n,m}$, which is isomorphic to any complete bipartite graph $\text{Bip}(V_1, V_2)$ such that $\text{card}(V_1) = n$ and $\text{card}(V_2) = m$.

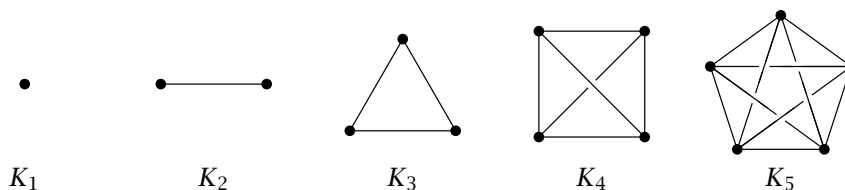


Fig. 10.14. The complete graphs K_n for $n = 1 \dots 5$.

Definition 75 *Given a graph Γ and $n \in \mathbb{N}$, a walk of length n in Γ is a morphism $p : |n| \rightarrow \Gamma$, we write $l(p) = n$ for its length. If $p(0) = v$ and $p(n) = w$, one also says that p is a walk from v to w . If there is a walk from v to w , we say that w is reachable from v . A walk of length 0 (just one vertex v in Γ) is called the lazy walk at v , and also denoted by v . A non-lazy walk $p : |n| \rightarrow \Gamma$ such that $p(0) = p(n)$ is called a cycle in Γ . A cycle of length 1 at a vertex v is called a loop at v .*

A graph is said to be connected if any two vertexes can be joined by a walk. A connected graph without (undirected) cycles is called a tree.

A directed graph Γ is called connected if $|\Gamma|$ is so.

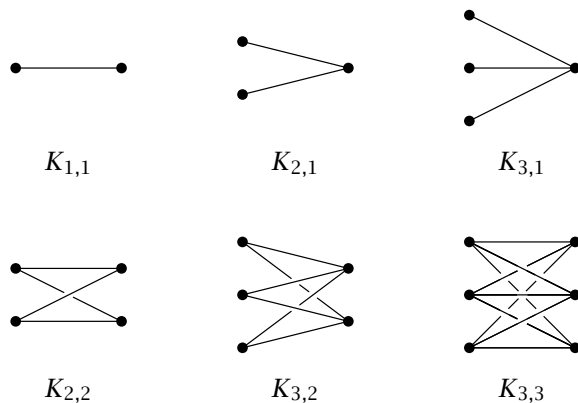


Fig. 10.15. The complete bipartite graphs $K_{n,m}$, for $1 \leq n \leq 3$ and $1 \leq m \leq 3$.

As with directed graphs, we may also compose walks, more precisely, if p is a walk from v to w , and q is one from w to z , then we have an evident walk qp from v to z of length $l(qp) = l(q) + l(p)$.

Remark 14 Alternatively we can define a walk from v to w in a graph Γ as a path from v to w in any digraph $\vec{\Gamma}$ such that $\Gamma = |\vec{\Gamma}|$. We shall also use this variant if it is more appropriate to the concrete situation.

Lemma 101 *If Γ is a graph, the binary relation \sim defined by “ $v \sim w$ iff there is a walk from v to w ” is an equivalence relation. The subgraph induced on an equivalence class is called a connected component of Γ .*

If Γ is a directed graph, its connected components are the directed subgraphs induced on the equivalence classes of vertexes defined by the associated graph $|\Gamma|$. Since there are no paths or walks between any two distinct connected components, two (di)graphs Γ and Δ are isomorphic iff there is an enumeration $\Gamma_i, \Delta_i, i = 1, \dots, k$ of their connected components such that component Γ_i is isomorphic to component Δ_i .

Proof The relation \sim is evidently reflexive, just take the lazy walk. It is symmetric since, if $p : |n| \rightarrow \Gamma$ is a walk from v to w , then the “reverse” walk $r : |n| \rightarrow |n|$ with $r(i) = n - i$ turns p into the walk $p \circ r$ from w to v . The rest is clear. \square

Remark 15 If Γ is a tree with vertex set V , and if v is a vertex, then the graph induced on $V - \{v\}$ is a disjoint union of connected components Γ_i which are also trees. Therefore, the tree Γ is determined by these

subtrees Γ_i , together with the vertexes being joined by an edge from v to determined vertexes v_i in Γ_i (in fact, if there were more than one such vertex, we would have cycles, and Γ would not be a tree). This allows us to define the tree concept recursively by the total vertex set V , the selected vertex v , the edges $v \text{ --- } v_i$, and the subtrees Γ_i . This is the definition of a tree given by Donald Knuth in his famous book, "The Art of Computer Programming" [31]. It has however the shortcoming that it distinguishes a vertex v , which is not the intrinsic property of a tree. For computer science it has the advantage that it is constructive and recursive.

10.3 Cycles

In this section, we want to state a number of elementary facts concerning the existence or absence of cycles in directed and undirected graphs.

Definition 76 *An Euler cycle e in a digraph/graph Γ is a cycle such that every vertex and every arrow/edge of Γ lies on the cycle (i.e., e is surjective on the vertexes and on the arrows/edges), but every arrow/edge appears only once (i.e., e is bijective on the arrows/edges).*

A Hamilton cycle h in a digraph/graph Γ is a cycle which contains every vertex exactly (i.e., h is surjective on the vertexes), except for the start and end vertex which it hits twice (i.e., is a bijection on the vertexes $0, 1, 2, \dots, l(h) - 1$).

The condition for the existence of Euler cycles in digraphs uses the degree of a vertex:

Definition 77 *If v is a vertex of a digraph $\Gamma : A \rightarrow V^2$, the head degree of v is the number*

$$\text{deg}^-(v) = \text{card}(\{h \mid h \in A, \text{head}(h) = v\}),$$

the tail degree of v is the number

$$\text{deg}^+(v) = \text{card}(\{h \mid h \in A, \text{tail}(h) = v\}),$$

and the degree of v is the number

$$\text{deg}(v) = \text{deg}^-(v) + \text{deg}^+(v).$$

For a graph Γ , the degree $\text{deg}(v)$ of a vertex v is defined as the degree of v in any of the digraphs $\vec{\Gamma}$ such that $|\vec{\Gamma}| = \Gamma$; observe that this number is independent of the choice of the digraph $\vec{\Gamma}$.

Here are two classical results by Leonhard Euler (1707–1783):

Proposition 102 *Let Γ be a digraph. Then it has an Euler cycle iff it is connected and for every vertex v , we have $\text{deg}^-(v) = \text{deg}^+(v)$.*

Proof In both implications we may give the proof for the subgraph of Γ obtained by omitting the loops. We may then put the loops back again without changing the validity of the proof.

If Γ has an Euler cycle $c : [n] \rightarrow \Gamma$, any two vertexes are connected by a sub-path associated to this cycle, so Γ is connected. If v is a vertex of Γ , and if $w \xrightarrow{a} v, w \neq v$ is an arrow of Γ , then a appears right before a subsequent arrow $v \xrightarrow{a'} w'$ in the cycle c . But these two arrows appear exactly once each in c , and therefore $\text{deg}^-(v) = \text{deg}^+(v)$.

Conversely, if Γ is connected and for every vertex v , $\text{deg}^-(v) = \text{deg}^+(v)$, then we first construct a covering of Γ by cycles c_i . Here we need only the hypothesis that the degrees are all even. Let a be an arrow of Γ , then there must exist an arrow a' starting from $\text{head}(a)$. The head of a' must also have an arrow a'' the tail of which is the head of a' . After a finite number of such steps, we end up at a vertex which we had already encountered before, and this defines a first cycle. Now omit all these cycle's arrows, and the hypothesis about the even number of degrees still holds, but for the smaller digraph obtained after elimination of the cycle's arrows. By induction we have a covering of this digraph by cycles, and, together with our first cycle obtain the desired covering. Clearly, a union of cycles with disjoint arrows covering a connected digraph is again a cycle. To see this observe that (1) a cycle may start at any of its vertexes; (2) if two cycles c and d in Γ have the vertex v in common, but have disjoint arrow sets, then there is also a cycle containing the union of these arrow sets; (3) if Γ is connected, any given cycle in Γ can be extended by adding to it another cycle of the given covering, having disjoint arrows. \square

Proposition 103 *Let Γ be a graph. Then it has an Euler cycle iff it is connected and for every vertex v , $\text{deg}(v)$ is an even number (a multiple of 2).*

Proof The proof is similar to the case of digraphs, except that the existence of an Euler cycle for an even number $\text{deg}(v)$ in each vertex v has to be shown. Here, we may couple edges at v in pairs of edges and associate with each pair one incoming and one outgoing arrow. This procedure yields a digraph over the given graph, and we may apply proposition 102. \square

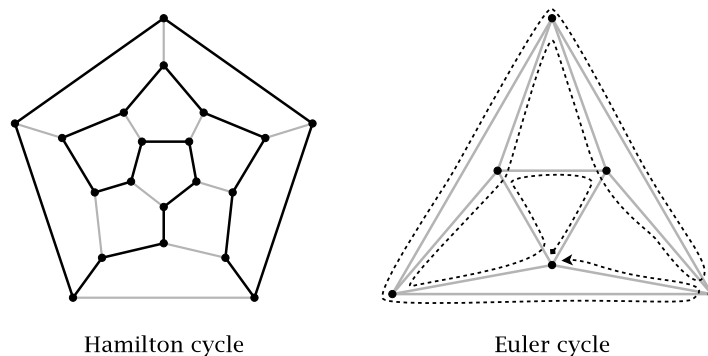


Fig. 10.16. A Hamilton cycle (heavy lines) on the flattened dodecahedron and a Euler cycle (dashed line) on the flattened octahedron. The dodecahedron has no Euler cycle, since there are vertexes with odd degree.

Exercise 47 A political group wants to make a demonstration in Zurich and asks for official permission. The defined street portions of the demonstration path are the following edges, connecting these places in Zurich: Z = Zweierplatz, S = Sihlporte, HB = Hauptbahnhof, C = Central, HW = Hauptwache, Pr = Predigerplatz, M = Marktgasse, Pa = Paradeplatz, B = Bellevue.

$$\begin{aligned} Z & \text{ --- } S, Z \text{ --- } Pa, S \text{ --- } HB, S \text{ --- } Pa, HB \text{ --- } C, \\ HB & \text{ --- } HW, Pa \text{ --- } HW, Pa \text{ --- } M, Pa \xrightarrow{1} B, Pa \xrightarrow{2} B, \\ HW & \text{ --- } Pr, Pr \text{ --- } M, M \text{ --- } B, B \text{ --- } C. \end{aligned}$$

The permission is given if a walk can be defined such that each connection is passed not more than once. Will the permission be given?

A *spanning sub(di)graph* of a (di)graph is a sub(di)graph which contains all vertexes.

Proposition 104 *Every connected graph has a spanning tree, i.e., a spanning subgraph which is a tree.*

Proof The proof is by induction on the number of edges. If there is only one vertex, we are done. Else, there are at least two different vertexes v and w which are connected by an edge a . Discard that edge. Then, if the remaining graph is still connected, we are done. Else, every vertex must be reachable from either v or w . Assume that, before eliminating a , every vertex x was reachable from v . So

if x is still reachable from v it pertains to the connected component of v , else, there walk from x to v must traverse w via a . But then, x is in the connected component of w . So the graph without a has exactly the two connected components C_v of v and C_w of w . By induction, each component C_v, C_w has a spanning tree T_v, T_w , respectively. Adding the edge a to the union of the disjoint spanning trees T_v and T_w still yields a tree, which is the required spanning tree. \square

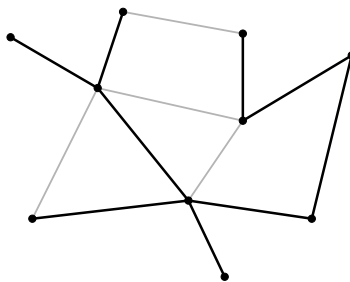


Fig. 10.17. Spanning tree of an undirected graph.

Construction of Graphs

We already know about some constructions of new (di)graphs from given ones: The dual digraph is such a construction. The spanning tree is a second one. Here more of these constructions are presented.

We have seen in set theory that there are universal constructions of new sets from given ones, such as the Cartesian product $a \times b$, the coproduct $a \sqcup b$, and the function set a^b . In graph theory, one also has such constructions which we discuss now. They are of very practical use, especially in computer science where the systematic construction of objects is a core business of software engineering.

Given two digraphs $\Gamma : A \rightarrow V^2$ and $\Delta : B \rightarrow W^2$, we have the Cartesian product $\Gamma \times \Delta : A \times B \rightarrow (V \times W)^2$, defined by the canonical¹ isomorphism of sets $t : V^2 \times W^2 \rightarrow (V \times W)^2 : ((e_1, e_2), (f_1, f_2)) \mapsto ((e_1, f_1), (e_2, f_2))$. In other words, we have

$$\begin{aligned} \text{head}(a, b) &= (\text{head}(a), \text{head}(b)) \\ \text{tail}(a, b) &= (\text{tail}(a), \text{tail}(b)). \end{aligned}$$

Example 44 Figure 11.1 shows the Cartesian product of two digraphs Γ and Δ .

¹ In mathematics, a construction is called “canonical” if no particular trick is necessary for its elaboration, it is realized by the given “surface structures”. Attention: the attribute “natural”, which we would also like to use instead, is reserved for a technical term in category theory.

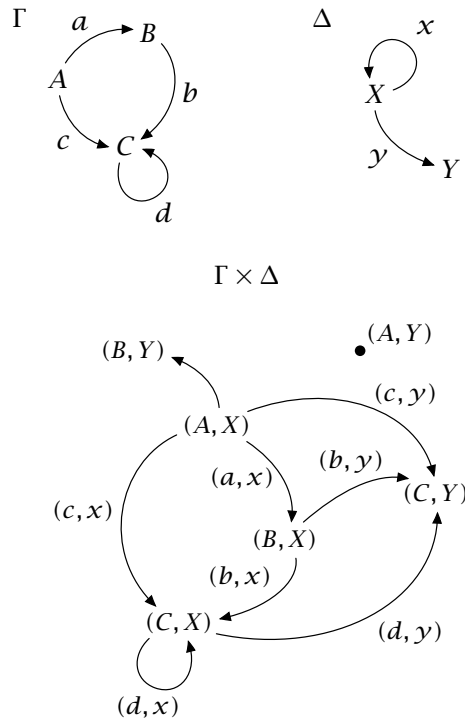


Fig. 11.1. The Cartesian product of two digraphs Γ and Δ .

Exercise 48 Show that the Cartesian product shares formal properties which are completely analogous to those stated for sets. Use the commutative diagram introduced in that context.

We also have a coproduct $\Gamma \sqcup \Delta : A \sqcup B \rightarrow (V \sqcup W)^2$, defined by the coproduct $A \sqcup B \rightarrow V^2 \sqcup W^2$ of the given maps, followed by the obvious injection $V^2 \sqcup W^2 \rightarrow (V \sqcup W)^2$.

Exercise 49 Show that the coproduct shares formal properties which are completely analogous to those stated for sets. Use the commutative diagram introduced in that context.

The coproduct of undirected graphs is constructed in the same way as the coproduct of digraphs, except that we have to replace the right exponent X^2 by the left one 2X and do the analogous mappings. The coproduct of (di)graphs is also called the *disjoint union of (di)graphs*.

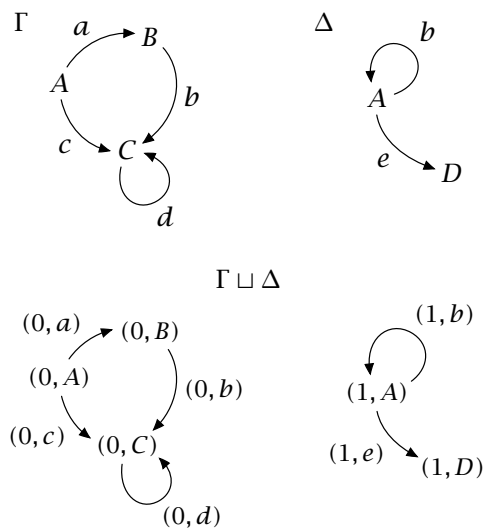


Fig. 11.2. The coproduct of the same two digraphs Γ and Δ as in figure 11.1. The construction of the coproduct of sets (definition 28) ensures that homonymous vertices and edges (e.g., A) are disambiguated (i.e., $(0, A)$ resulting from Γ and $(1, A)$ from Δ).

The Cartesian product of undirected graphs Γ and Δ cannot be constructed analogously to the Cartesian product of directed graphs. In this case, however, motivated by exercise 46, we can construct a graph, which can be considered the best approximation to a product: Take the Cartesian product of two digraphs $\vec{\Gamma}$ and $\vec{\Delta}$, together with the two projections $\vec{\Gamma} \times \vec{\Delta} \rightarrow \vec{\Gamma}$ and $\vec{\Gamma} \times \vec{\Delta} \rightarrow \vec{\Delta}$, and then the associated undirected projections $|\vec{\Gamma} \times \vec{\Delta}| \rightarrow \Gamma$ and $|\vec{\Gamma} \times \vec{\Delta}| \rightarrow \Delta$. Denote the product $|\vec{\Gamma} \times \vec{\Delta}|$ by $\Gamma \vec{\times} \Delta$ to stress that this object is *not* well defined.

Example 45 Consider the case for $\Gamma = \Delta = K_3$. The directed graphs associated to the undirected complete graph K_3 fall into two equivalence classes, where equivalence is defined by isomorphy. One contains those \vec{K}_3 , that are directed cycles, the other those that are not. Both types are shown at the top of figure 11.4. The product $|\vec{\Gamma} \times \vec{\Delta}|$ depends strongly on the particular type of the directed graphs chosen, as can be seen in figures 11.3, 11.4, and 11.5.

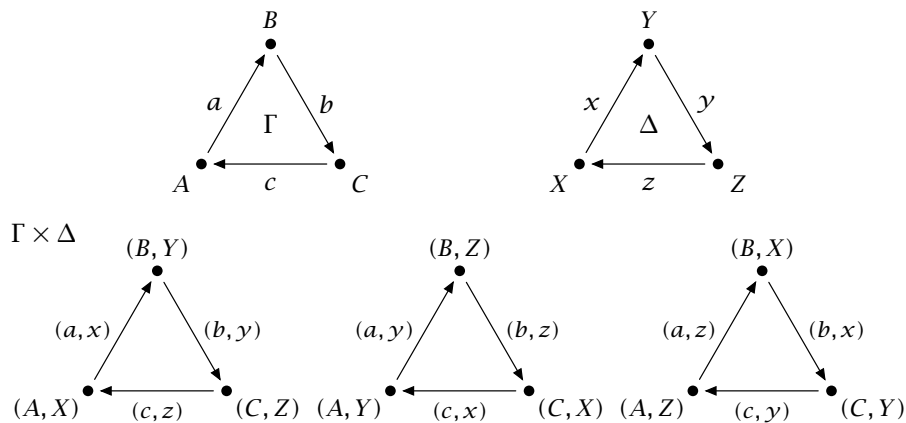


Fig. 11.3. The product of two directed versions of K_3 , both of which are cyclic. The product consists of three connected components.

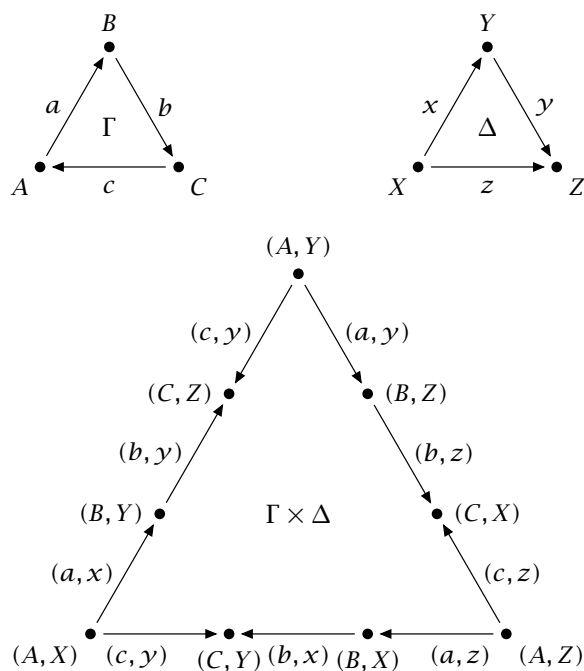


Fig. 11.4. The product of two directed versions of K_3 , where one is cyclic and the other one is not. The product is a connected graph.

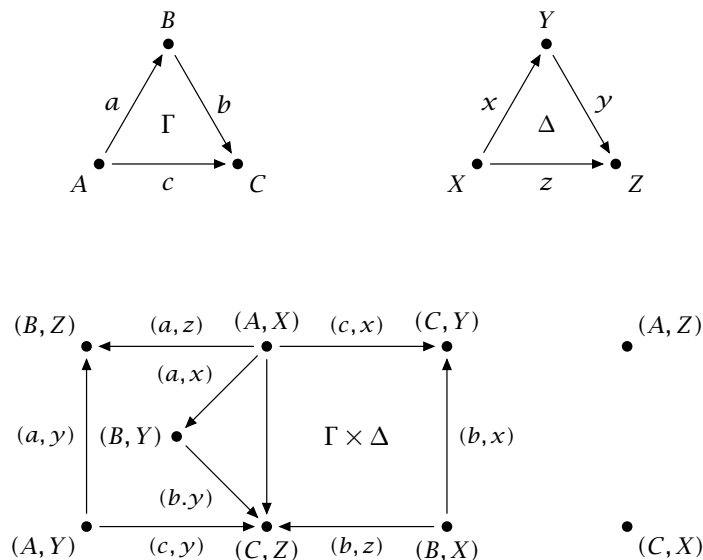


Fig. 11.5. The product of two directed versions of K_3 none of which is cyclic. The product consists of a connected component and two isolated vertexes.

Given two digraphs Γ and Δ , their *join BipDi*(Γ, Δ) is the digraph obtained from $\Gamma \sqcup \Delta$ by adding two arrows $v \rightarrow w$ and $w \rightarrow v$ between any pair of vertexes v of Γ and w of Δ (both directions). This generalizes the *BipDi*-construction from example 28. A similar construction works for two graphs Γ and Δ , their *join Bip*(Γ, Δ) adds one edge $v \text{ --- } w$ between any two vertexes v in Γ and w in Δ .

Graphs may also be constructed from non-graphical data, one of which is defined by a covering of a set:

Definition 78 A covering of a non-empty finite set X is a subset V of 2^X , consisting of non-empty sets, such that $\bigcup V = X$.

A covering gives rise to a graph as follows:

Definition 79 Let V be a covering. Then the line skeleton $LSK(V)$ of V is the graph the vertex set of which is V , while the edge set is the set of two-element sets $\{v, w\} \subset V$ such that $v \cap w \neq \emptyset$.

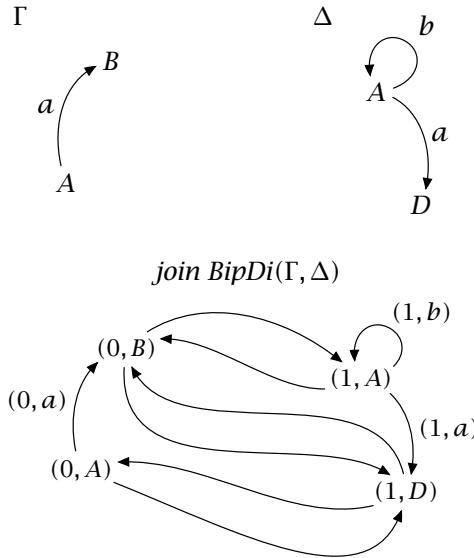


Fig. 11.6. The join bipartite digraph of two digraphs Γ and Δ .

A large class of graphs is indeed derived from coverings:

Proposition 105 Every graph $\Gamma : A \rightarrow {}^2V$ without loops and multiple edges (i.e., Γ has no values of form $\{v\}$, and is injective) is isomorphic to the line skeleton of a covering.

Proof In fact, let $\Gamma : A \rightarrow {}^2V$ be such a graph. Then for each vertex x , we set $A_x = \{a \mid a \in A, x \in \Gamma(a)\}$ for the set of lines joining x to another vertex. Then the subsets $A_x \subset A$ define a covering C of A , and for two different vertexes x and y , $A_x \cap A_y$ is a singleton set containing exactly the line joining x to y . This means $LSK(C) \cong \Gamma$. □

Example 46 Figure 11.7 shows a covering $V = \{A, B, C, D, E, F, G\}$ of the set of letters from a to q , and, on the right, its line skeleton $LSK(V)$.

Another example is taken from music theory. Consider the C major scale $C = \{c, d, e, f, g, h\}$ and denote the seven triadic degrees by $I = \{c, e, g\}$, $II = \{d, f, a\}$, $III = \{e, g, h\}$, $IV = \{f, a, c\}$, $V = \{g, h, d\}$, $VI = \{a, c, e\}$ and $VII = \{h, d, f\}$. These triads obviously form a covering of C . The line skeleton $LSK(\{I, II, III, IV, V, VI, VII\})$ is illustrated in figure 11.8.

We have chosen a 3-dimensional representation, in order to emphasize the geometric structure induced by the triangles formed by three adja-

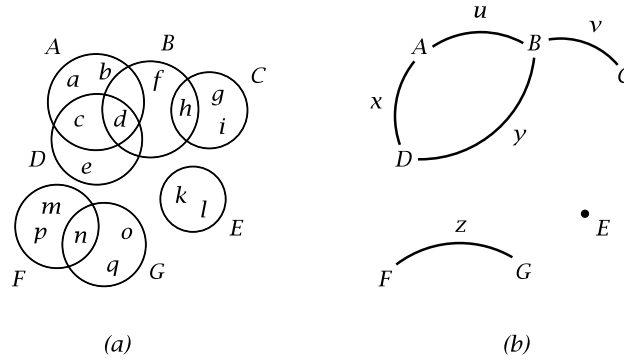


Fig. 11.7. A covering (a), and its LSK(b).

cent triads. These triangles have as their vertexes three triads sharing exactly one element. Such a geometric structure is called a *Möbius strip*.

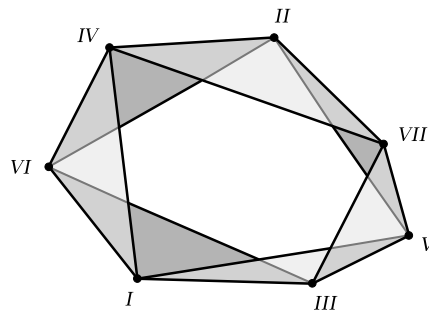


Fig. 11.8. The line skeleton of the covering of a major scale by triads.

Some Special Graphs

We shall discuss two types of special graphs: n -ary trees and Moore graphs.

12.1 n -ary Trees

Binary trees and, more generally, n -ary trees are very frequent in computer algorithms. Intuitively, they formalize a decision hierarchy, where at each step, there is a limited number of alternatives. To formalize these alternatives, we first need the digraph of n -ary alternatives for natural $n \geq 2$. This is the *loop digraph* $Loop(n) : n \rightarrow 1$, consisting of n loops $0, 1, \dots, n - 1$ and one vertex 0 (figure 12.1). More generally, the loop digraph of a set L is the unique digraph $Loop(L) : L \rightarrow 1$.

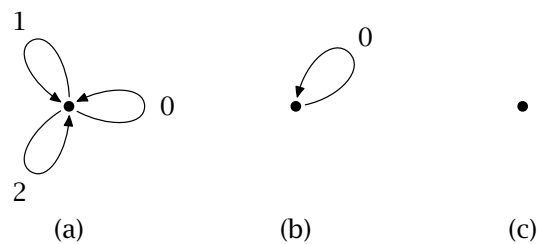


Fig. 12.1. Loop digraphs: (a) $Loop(3)$, (b) $Loop(1)$, (c) $Loop(0)$. In all three cases, the single vertex carries the label 0.

Definition 80 For a natural number $n \geq 2$, an n -ary tree is a morphism of digraphs $N : \Gamma \rightarrow \text{Loop}(n)$, such that

- (i) $|\Gamma|$ is an undirected tree,
- (ii) Γ has a root,
- (iii) each vertex v of Γ has $\text{deg}^+(v) \leq n$,
- (iv) for any two arrows a and b with common tail v , $N(a) \neq N(b)$.

For an arrow $v \xrightarrow{a} w$ in an n -ary tree, w is called a child of v , whereas the necessarily unique v for a given w is called the parent of w .

For $n = 2$, i.e., for binary trees, the labeling has values 0 and 1, but one often calls the value 0 the *left* and 1 the *right alternative*.

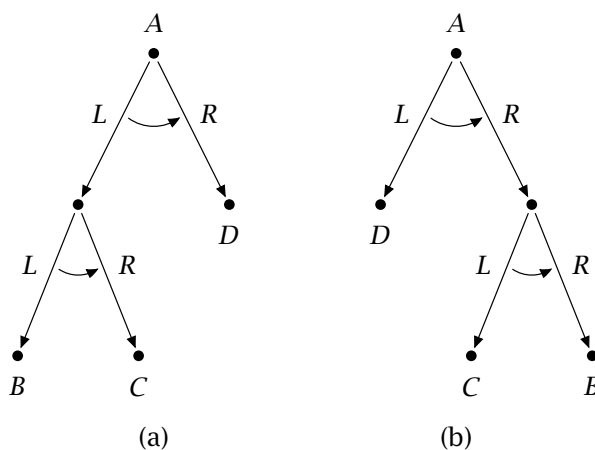


Fig. 12.2. A binary tree where the left alternative is labeled L , and the right alternative R (a). The binary tree in (b) is isomorphic to (a) as a directed graph, but not as a binary tree. The order of the branches, indicated by the angle arrows, is crucial.

Often, the value $N(a)$ of an arrow in an n -ary tree is denoted as a label of a , but it is more than just a labeling convention. In fact, the comparison of n -ary trees is defined by special morphisms:

Definition 81 Given two n -ary trees $N : \Gamma \rightarrow \text{Loop}(n)$ and $R : \Delta \rightarrow \text{Loop}(n)$, a morphism of n -ary trees is a triple $(f : \Gamma \rightarrow \Delta, N, R)$ denoted by $f : N \rightarrow R$, where $f : \Gamma \rightarrow \Delta$ is a morphism of digraphs such that $N = R \circ f$.

In particular, for binary trees, this means that a left/right arrow of Γ must be mapped to a left/right arrow of Δ . So it may happen that Γ and Δ are isomorphic digraphs without being isomorphic with the added n -labels of alternatives.

12.2 Moore Graphs

A Moore graph is a special type of a process digraph as already presented in a generic form in example 27. Moore graphs arise in the theory of sequential machines and automata, which we introduce here since their concepts are ideally related to the theory of digraphs thus far developed.

To begin with, we denote by $Path(\Gamma)$ the set of paths in a digraph Γ , including the composition qp of paths p and q if possible (see exercise 45 for this construction). Also denote by $Path_v(\Gamma)$ the set of paths starting at vertex v . For a loop digraph $Loop(A)$, the composition of any two paths is possible and associative, moreover, the lazy path e at vertex 0 is a “neutral element” for this composition, i.e., $e \cdot p = p \cdot e = p$ for any path $p \in Path(Loop(A))$. This structure is also denoted by $Word(A)$ and called the *word monoid over A* (a justification for this terminology is given in chapter 15). Any path p in $Word(A)$ is defined by the (possibly empty) sequence a_1, \dots, a_k of its arrows (the letters of the word) from A since the vertex 0 is uniquely determined. Further, if $p = a_1 \dots a_k$ and $q = b_1 \dots b_l$, the composition is exactly the concatenation of the letters of the two words: $q \cdot p = b_1 \dots b_l a_1 \dots a_k$.

In automata theory, if n is a positive natural number, one considers as “input set” the n -cube $Q^n = 2^n$, the elements of which are the “ n -bit words” such as $w = (0, 0, 1, 0, 1, 1, \dots, 1)$ (see also figure 13.3 on page 146). For $n = 1$ the two words (0) and (1) are called *bits*, thus elements of Q^n are also defined as sequences of n bits. An 8-bit word is called a *byte*. Evidently $card(Q^n) = 2^n$. The word monoid $Word(Q^n)$ is called the *input monoid*, its elements are called *tapes*, so tapes in $Word(Q^n)$ are just sequences consisting of n -bit words.

Definition 82 *An automaton of n variables is a set function*

$$A : Word(Q^n) \rightarrow Q.$$

All automata can be constructed by a standard procedure which is defined by sequential machines:

Definition 83 A sequential machine of n variables is a map $M : S \times Q^n \rightarrow S$, where S is called the state space of the machine M . If M is clear from the context, we also write $s \cdot q$ instead of $M(s, q)$.

The Moore graph of a sequential machine M is the digraph $\text{Moore}(M) : S \times Q^n \rightarrow S^2$ defined by $\text{Moore}(M)(s, q) = (s, M(s, q))$.

Example 47 Consider a sequential machine with $Q = 2$ and $n = 2$, i.e., $Q^n = \{00, 01, 10, 11\}$, and the set of states $S = \{s_0, s_1, s_2\}$. Let the map M be as follows:

$$\begin{array}{lll} M(s_0, 00) = s_0, & M(s_1, 00) = s_0, & M(s_2, 00) = s_2, \\ M(s_0, 01) = s_0, & M(s_1, 01) = s_1, & M(s_2, 01) = s_1, \\ M(s_0, 10) = s_1, & M(s_1, 10) = s_2, & M(s_2, 10) = s_2, \\ M(s_0, 11) = s_1, & M(s_1, 11) = s_2, & M(s_2, 11) = s_0. \end{array}$$

The Moore graph $\text{Moore}(M)$ is defined by the set of vertexes $V = S = \{s_0, s_1, s_2\}$ and arrows $A = S \times \{0, 1\}^2$ and is illustrated in figure 12.3.

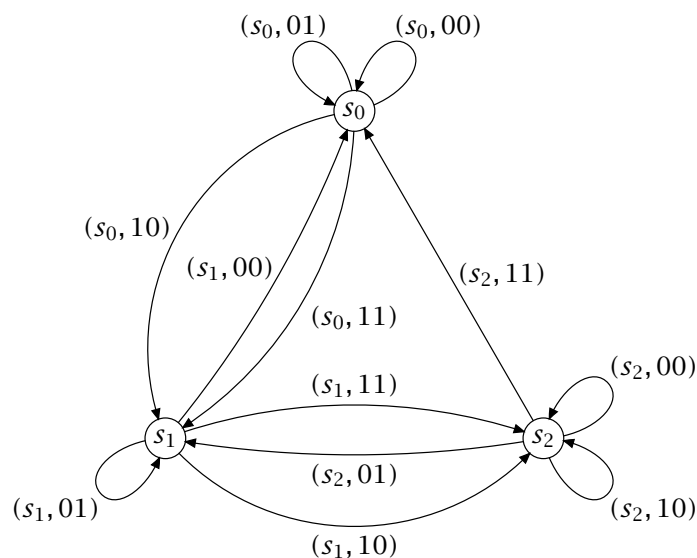


Fig. 12.3. The Moore graph $\text{Moore}(M)$.

Here is the description of paths in the Moore graph of a sequential machine:

Proposition 106 For a sequential machine $M : S \times Q^n \rightarrow S$, a canonical bijection

$$PW : \text{Path}(\text{Moore}(M)) \rightarrow S \times \text{Word}(Q^n)$$

is given as follows:

If

$$p = s_1 \xrightarrow{(s_1, q_1)} s_2 \xrightarrow{(s_2, q_2)} s_3 \cdots \xrightarrow{(s_{m-1}, q_{m-1})} s_m,$$

then $PW(p) = (s_1, q_1 q_2 \dots q_{m-1})$.

Under this bijection, for a given state $s \in S$, the set $\text{Path}_s(\text{Moore}(M))$ corresponds to the set $\{s\} \times \text{Word}(Q^n)$.

Proof The map PW is injective since from the word $q_1 q_2 \dots q_{m-1}$ and the state s_1 we can read all the letters q_1, q_2, \dots, q_{m-1} , and then reach the other states by $s_2 = s_1 q_1, s_3 = s_2 q_2, \dots, s_m = s_{m-1} q_{m-1}$. It is surjective, since for any pair $(s_1, q_1 q_2 \dots q_{m-1})$ the above reconstruction yields a preimage. The last statement is immediate from the definition of PW . \square

We are now ready to define automata associated with sequential machines. To this end, we fix a sequential machine M in n variables over the state space S , an “initial” state s , and a so-called *output function* $O : S \rightarrow Q$. The automaton $\text{Automaton}(M, s, O) : \text{Word}(Q^n) \rightarrow Q$ is defined as follows. Denote by $\text{head} : \text{Path}_s(\text{Moore}(M)) \rightarrow S : p \mapsto \text{head}(p)$ the map associating with each path the head of its last arrow. Also denote $(s, ?) : \text{Word}(Q^n) \rightarrow \{s\} \times \text{Word}(Q^n) : w \mapsto (s, w)$. Then we define

$$\text{Automaton}(M, s, O) = O \circ \text{head} \circ PW^{-1} \circ (s, ?) : \text{Word}(Q^n) \rightarrow Q.$$

leads to the following result:

Proposition 107 For every automaton A , there is a sequential machine M , an initial state s , and an output function O such that

$$A = \text{Automaton}(M, s, O).$$

Proof In fact, given the automaton $A : \text{Word}(Q^n) \rightarrow Q$, take the state space $S = \text{Word}(Q^n)$, the output function $O = A$ and the sequential machine $M : S \times Q^n \rightarrow Q$ defined by $M(s, q) = sq$. With the initial state e (the empty word) we have the automaton $\text{Automaton}(M, e, A)$, and this is the given automaton O . \square

Example 48 We revisit the sequential machine from example 47 and define the output function $O : S \rightarrow Q$ as:

$$O(s_0) = 0,$$

$$O(s_1) = 1,$$

$$O(s_2) = 1.$$

This yields an automaton $A = \text{Automaton}(M, s_0, O)$, where s_0 has been chosen as the “start” state. The result of A applied to the input sequence $10\ 10\ 00\ 01\ 11 \in \text{Word}(Q^n)$ is computed as the value of the expression $O(\text{head}(PW^{-1}((s_0, ?)(10\ 10\ 00\ 01\ 11))))$. Let us do this step by step:

1. $(s_0, ?)(10\ 00\ 01\ 01\ 11) = (s_0, 10\ 10\ 00\ 01\ 11)$;
2. $PW^{-1}(s_0, 10\ 10\ 00\ 01\ 11)$ is the path traced in the Moore graph of figure 12.3 starting at s_0 ; it is:

$$p = s_0 \xrightarrow{(s_0,10)} s_1 \xrightarrow{(s_1,10)} s_2 \xrightarrow{(s_2,00)} s_2 \xrightarrow{(s_2,01)} s_1 \xrightarrow{(s_1,11)} s_2$$

3. $\text{head}(p) = s_2$;
4. $O(s_2) = 1$.

CHAPTER 13

Planarity

Planarity deals with the problem of how graphs can be drawn on a particular surface, such as the plane \mathbb{R}^2 , the sphere $S^2 = \{(x, y, z) \mid x^2 + y^2 + z^2 = 1\} \subset \mathbb{R}^3$, or the torus, which intuitively looks like the surface of a doughnut. We shall use here several concepts which will only be explained rigorously in the chapter on topology in the second volume of this book. However, the elementary character of the results and the important problem of drawing graphs suggests a preliminary treatment of the subject in this first part of the course.

13.1 Euler's Formula for Polyhedra

To begin with, this chapter only deals with *undirected graphs which have no loops and no multiple edges*. In fact, drawing such graphs immediately implies drawing of any more general graphs. In view of proposition 105, we shall call such graphs *skeletal graphs*. So skeletal graphs are characterized by their set V of vertexes, together with a subset $A \subset {}^2V$ containing only edges with exactly two elements.

The ad hoc definition we use now (and explain in a more general way in the chapter on limits) is that of continuity:

Definition 84 *A continuous curve in \mathbb{R}^2 (or on the sphere S^2) is an injective map $c : [0, 1] \rightarrow \mathbb{R}^2$ (or $[0, 1] \rightarrow S^2$) defined on the unit interval $[0, 1] = \{x \mid 0 \leq x \leq 1\} \subset \mathbb{R}$ such that for any $\varepsilon > 0$ and any $s \in [0, 1]$, there is a $\delta > 0$ such that for any $t \in [0, 1]$, if $|s - t| < \delta$, then $d(c(t), c(s)) < \varepsilon$,*

where the common Euclidean distance function d is defined by

$$d((x_1, y_1), (x_2, y_2)) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}.$$

Intuitively, continuity of such a curve means that you can draw the curve from the beginning (curve parameter 0) to the end (curve parameter 1) without lifting the pencil. We shall consider only continuous curves here and omit this adjective in the following discussion. Denote by $]0, 1[$ the unit interval without the two endpoints 0 and 1, the so-called *interior of the unit interval*. Here is the definition of a drawing of a skeletal graph:

Definition 85 A drawing D of a skeletal graph $\Gamma : A \rightarrow {}^2V$ in X (where X is commonly the plane \mathbb{R}^2 , but can also be the sphere S^2 or any more general space, where continuity is reasonably defined) is given by $D = (r, c = (c_a)_{a \in A})$, where

- (i) $r : V \rightarrow X$ is an injection,
- (ii) for each edge $v \xrightarrow{a} w$ in A , there is a curve $c_a : [0, 1] \rightarrow X$ such that

$$\{c_a(0), c_a(1)\} = r(\Gamma(a)),$$

- (iii) for any two different edges a and b , the image $c_a(]0, 1[)$ of the interior of the first curve is disjoint from the image $c_b([0, 1])$ of the entire second curve.

A skeletal graph is planar iff it has a drawing in \mathbb{R}^2 .

The image $\bigcup_{a \in A} \text{Im}(c_a)$ is denoted by $D(\Gamma)$ and is called the drawn graph.

Remark 16 It is easy to prove this proposition: A graph has a drawing in \mathbb{R}^2 iff it has a drawing in S^2 . The proof uses the stereographic projection, known from geography, in fact a bijection $S^2 - \text{NorthPole} \rightarrow \mathbb{R}^2$, which will be introduced in the chapter on topology. This implies that planarity is equivalent to the existence of a drawing on the sphere S^2 .

A drawing of a connected skeletal graph on S^2 is also called a *polyhedron*. This is justified by the fact that if you are positioned within a polyhedron (such as a cube or a tetrahedron), you see the edges of the polyhedron on your visual sphere as if they were edges of a graph drawing on S^2 .

If D is a drawing of a skeletal graph Γ , then the complement $\mathbb{R}^2 - D(\Gamma)$ of the drawn graph is the disjoint union of a number of regions which

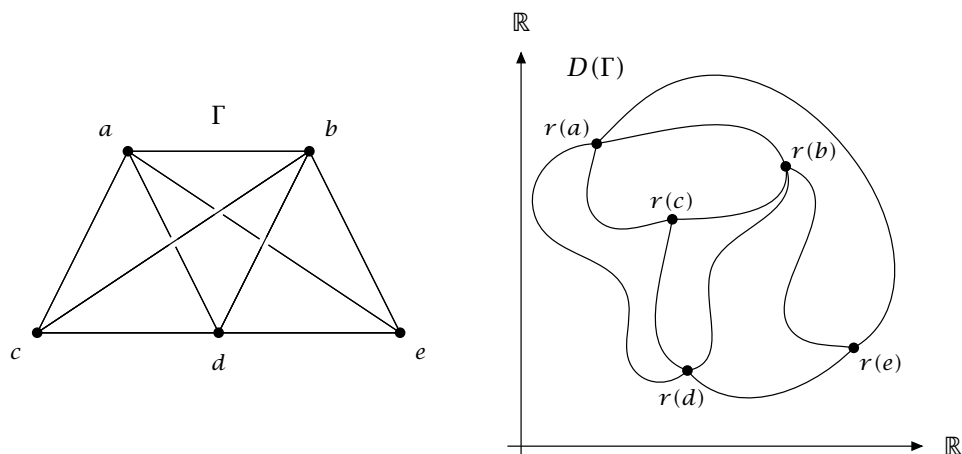


Fig. 13.1. A graph Γ and its drawing $D(\Gamma)$. Intuitively speaking, in a drawing, no two lines intersect except at their endpoints.

are separated by the drawn graph. They are defined as follows: On any subset $X \subset \mathbb{R}^2$, we consider the relation $x \sim y$ iff there is any curve c in X (attention: these curves are not the curves used in drawings of graphs) such that $c(0) = x$ and $c(1) = y$.

Exercise 50 Show that \sim is an equivalence relation.

The equivalence classes of \sim are called the *connected components* of X . We are now able to write down the famous Euler formula for polyhedra:

Proposition 108 For a skeletal graph $\Gamma : A \rightarrow {}^2V$, let $D(\Gamma)$ be a polyhedron, and C the set of connected components of the drawing's complement $\mathbb{R}^2 - D(\Gamma)$, and set

1. $\varepsilon = \text{card}(V)$,
2. $\varphi = \text{card}(A)$,
3. $\sigma = \text{card}(C)$.

Then we have

$$\varepsilon - \varphi + \sigma = 2.$$

Proof Postponed to the chapter on topology in volume II. □

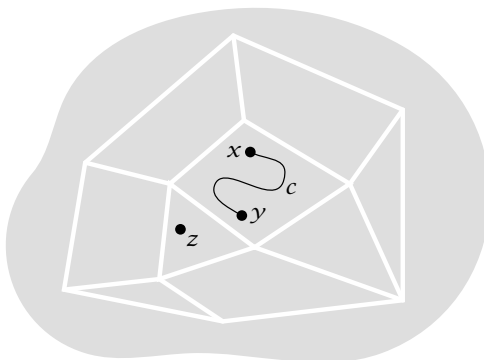


Fig. 13.2. The gray regions are the connected components of a graph. Here x and y are in the same equivalence class, whereas x and z are not.

Exercise 51 The n -cube Q^n introduced in section 12.2 gives rise to the n -cube graph, also denoted by Q^n . It is a skeletal graph with vertex set Q^n and edges of the form $\{x, y\}$, where x and y differ exactly by one bit. Find a drawing D of Q^3 . Show that in this case, the numbers in the Euler formula are $\varepsilon = 8$, $\varphi = 12$, $\sigma = 6$.

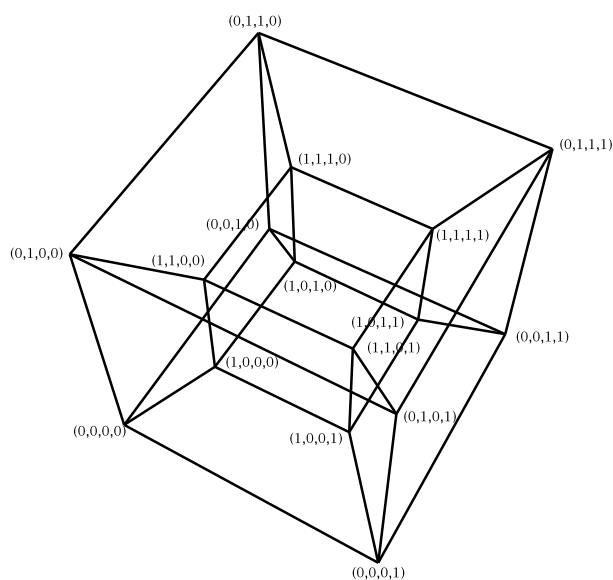


Fig. 13.3. Hypercube Q^4 .

Example 49 In figure 13.4 a dodecahedron has been flattened to show the number of vertexes (circled numbers), edges (small-sized numbers) and faces (large-sized numbers). Euler's formula can be verified using $\varepsilon = 20$, $\varphi = 30$ and $\sigma = 12$, i.e., $20 - 30 + 12 = 2$. This shows that the flattened dodecahedron can be drawn with no intersecting edges.

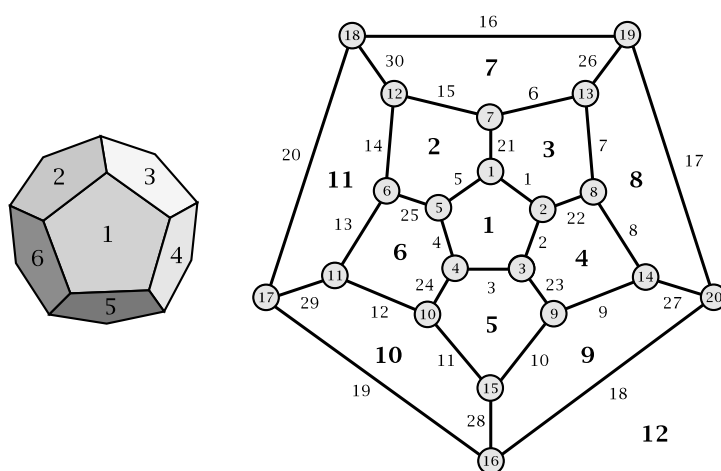


Fig. 13.4. The dodecahedron and its flattened and stretched representation as a graph.

Observe that in Euler's formula there is a rather deep insight into drawings: In fact, the numbers ε and φ are defined by the "abstract" graph, whereas the number σ is a function of the specific drawing. Therefore, all drawings must show the same number of connected components of the drawing's complement.

13.2 Kuratowski's Planarity Theorem

The previous formula is applicable only if we are sure that the graph is planar. Evidently, planarity of a graph Γ is a property which remains conserved if we pass to an isomorphic graph. In other words, there should be an abstract criterion which tells us when a skeletal graph is planar. We shall present the criterion proved by Kazimierz Kuratowski (1896–1980).

To this end we first need the following construction:

Definition 86 If $\Gamma \rightarrow {}^2V$ is a skeletal graph, and if $\Gamma(a) = \{x, y\} \in V$ is any two-element set defined by an edge a , we denote by $\Gamma_a : A' \rightarrow {}^2V'$ the skeletal graph with $A' = A - \{a\}$ and $V' = (V - \{x, y\}) \sqcup \{a\}$, and we have

1. if $\Gamma(b) \cap \{x, y\} = \emptyset$, then $\Gamma_a(b) = \Gamma(b)$,
2. else $\Gamma_a(b) = (\Gamma(b) - \{x, y\}) \cup \{a\}$.

Γ_a is called an elementary contraction of Γ . A contraction of a skeletal graph Γ is a graph Δ which is isomorphic to a finite succession

$$(\dots ((\Gamma_{a_1})_{a_2}) \dots)_{a_m}$$

of elementary contractions of Γ .

Intuitively, Γ_a results from the removal of the edge a in Γ , the insertion of a new point, and the connection of all open-ended edges from the removal to the new point.

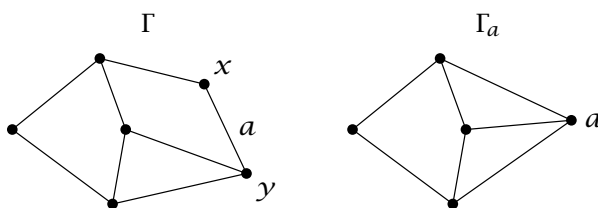


Fig. 13.5. A graph Γ and its elementary contraction Γ_a .

Proposition 109 If Δ is a contraction of Γ , then, if Γ is planar, so is Δ .

Exercise 52 Give a proof of proposition 109. Draw this for an elementary contraction.

This is the main theorem:

Proposition 110 A skeletal graph is planar iff it contains no subgraph that has a contraction which is isomorphic to one of the graphs $K_{3,3}$ or K_5 .

Proof Postponed to the chapter on topology in volume II. \square

Exercise 53 Can you find a drawing of Q^4 (figure 13.3)? Hint: Search for a K_5 or $K_{3,3}$.

First Advanced Topic

14.1 Floating Point Arithmetic

In this section, we give an overview of computer-oriented arithmetic based on the adic representation of real numbers as described in proposition 92. Evidently, representation of numbers and arithmetic calculations on a computer are bounded by finite memory and time. Since real numbers are based upon infinite information, computer arithmetic must be limited to finite truncations of real arithmetic. Floating point arithmetic is one approach to this problem. It is based on the well-known *approximative* representation of real numbers in scientific contexts, such as Avogadro's number $N \approx 6.02214 \times 10^{23}$, Planck's constant $h \approx 6.6261 \times 10^{-34} \text{J s}$ or the circle circumference number $\pi \approx 3.14159$. Here we consent that any number can be written as a small positive or negative number f (in fact $1 \leq |f| < 10$ or $f = 0$ in the decimal system) times a power 10^e of the basis (here 10) for a positive or negative integer exponent e . The term "floating point" stems from the difference to the "fixed point" representation given in proposition 92 of chapter 9, where the dot is relative to the 0-th power of the base, whereas here, the dot is a relative information related to the variable power of the base.

Remark 17 Observe that the limitation to a finite number of digits in the adic representation of real numbers implies that we are limited to special rational numbers (not even $1/3 = 0.333\dots$ is permitted in the decimal system!). So why not just take the fractional representation which is a precise one? The point is that, evidently, numerator and denominator of

a fraction would also grow to unlimited size when doing arithmetic. So the problem is not the fractional representation (though it would make arithmetic precise in some cases), but the general program for calculating with numbers within an absolutely limited memory space.

The IEEE (Institute for Electrical and Electronics Engineers) #754 standard defines a specific way of floating point representation of real numbers. It is specified for base 2 by a triple of (s, e, f) , where $s \in \{0, 1\}$ is the bit for the sign, i.e., $s = 0$ iff the number is not negative, $s = 1$ iff it is negative. The number e is the exponent and takes integer values $e \in [-127, 128]$, corresponding to a range of 8-bits ($2^8 = 256$). The exponent is encoded by a bias integer $bias = 127$. This means that we have to consider $e - bias$ instead of e in the computerized representation. The extra value -127 is set to 0. This exponent is reserved to represent number zero. We will make this more precise in a moment. The maximal value $e = 128$, shifted to 255 by the bias, is used to represent different variants of infinity and “not-a-number” symbols. We therefore may vary the exponents in the integer interval $[-126, 127]$. In the binary representation using eight bits, and with regard to the bias, this amounts to considering the zero number exponent symbol 00000000 whereas the infinities and not-a-number symbols are represented by the maximum 11111111 ($= 2^8 - 1 = 255$). So the exponent for common numbers takes shifted values in the binary interval $[00000001, 11111110]$, corresponding to the unshifted interval $[-126, 127]$.

The third number f is called the *mantissa* (an official, but mathematically ill-chosen terminology) or *fraction* (non-official, but correct terminology, also adapted by Knuth). We stick to the latter. The fraction f is encoded in binary representation $f = f_1 f_2 f_3 \dots f_{23} \in [0, 2^{23} - 1]$, i.e., in binary representation

$$000000000000000000000000 \leq f_1 f_2 \dots f_{23} \leq 111111111111111111111111$$

and stands for the fraction $1.f = 1.f_1 f_2 \dots f_{23}$ in binary representation.

Given these three ingredients, the number $\langle s, e, f \rangle$ denoted by the triple (s, e, f) is the following real number:

$$\langle s, e, f \rangle = (-1)^s \times 2^{e-bias} \times 1.f$$

This is the *normalized floating point representation* of the IEEE standard #754. It is also called *single precision* representation since a finer representation using two 32-bit words exists, but we shall not discuss this

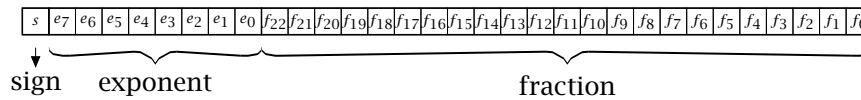


Fig. 14.1. IEEE 32-bit representation.

refinement here. Figure 14.1 shows a schematic representation of the configuration of s , e and f in a 32-bit word. The values $s = e = f = 0$ represent zero:

$$0 = (0, 00000000, 000000000000000000000000).$$

Non-zero numbers are given by these 32-bit words:

- $s \in \{0, 1\}$, one bit,
- $e \in [00000001, 11111110]$, eight bits,
- $f \in [000000000000000000000000, 111111111111111111111111]$, 23 bits.

The largest positive number that can be stored is

$$1.111111111111111111111111 \times 2^{127} = 3.402823 \dots \times 10^{38},$$

whereas the smallest positive number is

$$1.000000000000000000000000 \times 2^{-126} = 1.175494 \dots \times 10^{-38}.$$

The special infinities and not-a-number (NaN) values are the following 32-bit words:

- $(0, 11111111, 000000000000000000000000) = \text{Inf}$, positive infinity,
- $(1, 11111111, 000000000000000000000000) = -\text{Inf}$, negative infinity,
- $(0, 11111111, 011111111111111111111111) = \text{NaNs}$, NaN generating a “trap” for the compiler, while
- $(0, 11111111, 100000000000000000000000) = \text{NaNQ}$, where Q means “quiet”, calculation proceeds without generating a trap.

Given this standardized normal representation, we shall now discuss the arithmetical routines. The general procedure is a very simple two-step

algorithm: first, one of the operations of addition, subtraction, multiplication, or division is performed, and then, in a second step, the result is recast to the normalized representation shown above.

Regarding the first step, we shall only discuss addition, since the other operations run in a completely similar way. To make the algorithm more transparent, we use another (also Knuth's, but not IEEE-standardized) normalization by replacing the representation $1.f \times 2^{e-bias}$ by $0.1f \times 2^{e-bias+1}$. Denote this representation by $\langle\langle s, e, g \rangle\rangle = (-1)^s \times 2^{e-bias} \times g$ with $0 \leq g < 1$, i.e., we have $\langle s, e, f \rangle = \langle\langle s, e + 1, 0.1f \rangle\rangle$. Observe that we now need 24 bits after the dot in order to represent this second normalized representation. Observe also that the third coordinate g of the second representation is the real number, including zero 0.0, and not just the 23-bit word to the right of the dot as with the IEEE standard.

Given two normalized representations $\langle\langle s_u, e_u, g_u \rangle\rangle, \langle\langle s_v, e_v, g_v \rangle\rangle$, the computer calculation does not yield the exact sum, but an approximation $\langle\langle s_w, e_w, g_w \rangle\rangle$. In general, it is not normalized anymore. This will be corrected by the subsequent normalization algorithm. However, the latter will perform a further rounding error.

Here is the algorithm **A** for **addition**:

- A.1:** We check whether $e_u \geq e_v$. If not, we exchange the summands and proceed. As addition is commutative, this is a reasonable step, and it will also be the reason why floating point addition is in fact commutative.
- A.2:** Set $e_w = e_u$.
- A.3:** If $e_u - e_v \geq 24 + 2 = 26$, we have a large difference of exponents, i.e., the smaller summand has its highest digit below the 24 digits admitted in the normalized representation of $\langle\langle s_u, e_u, g_u \rangle\rangle$. We therefore set $g_w = g_u$. In this case, go to **A.6**. Actually, in this case, the result is already normalized, and we could terminate the entire addition here.
- A.4:** Divide g_v by $2^{e_u - e_v}$, i.e., shift the binary representation by up to 9 places to the right. Attention: This procedure requires the computer memory to hold up to $24 + 9 = 33$ places temporarily.

- A.5:** Set $g_w = g_u + g_v$. This last step gives us the correct sum $\langle\langle s_w, e_w, g_w \rangle\rangle$, but we have a g_w which might not be normalized, i.e., we could have $g_w \geq 1$.
- A.6:** Normalize, i.e., apply the normalization algorithm **N** to $\langle\langle s_w, e_w, g_w \rangle\rangle$.

The **normalization** algorithm **N** runs as follows, it converts a “raw” exponent e and a “raw” fraction $0 \leq g$ into a normalized representation.

- N.1:** If $g \geq 1$ (fractional overflow), then go to step **N.4**. If $g = 0$, set e to the lowest possible exponent in the normalized representation (in fact $e = -127$ or 00000000 in the biased binary IEEE representation and $e = -126$ in the second normalized representation).
- N.2:** (Normalization of g , i.e., $g < 1$ but large enough) If $g \geq 1/2$, go to step **N.5**.
- N.3:** (g is too small, but does not vanish) Multiply g by 2, decrease e by 1 and return to step **N.2**.
- N.4:** Divide g by 2 and increase e by 1 and return to step **N.1**.
- N.5:** (Round g to 24 places) This means that we want to change g to the nearest multiple of 2^{-24} . One looks at $2^{24} \times g = \dots g_{-24}.h$ and checks the part h after the dot. According to whether h is less than $1/2$ or not, this part is omitted and $2^{24} \times g$ is replaced by $\dots g_{-24}.0 + 1$ or $\dots g_{-24}.0$, see the following remark 18 for a comment. If the rounded g is 1, return to step **N.1**.
- N.6:** (check e) If e is too large (more than 127), an *exponent overflow* condition is sensed. If e is too small (less than -126 in the IEEE standard, or less than -125 in the second normalized form $\langle\langle \rangle\rangle$ used in this algorithm), then an *exponent underflow* condition is sensed.

This concludes the normalization algorithm (except for the actions to be taken for the over- and underflow situations).

Remark 18 There is an axiomatic version of this theory, where the rounding $round(x)$ of a number x is given axiomatically with conditions $round(-x) = -round(x)$ and $x \leq y$ implies $round(x) \leq round(y)$. We then define rounded arithmetic operations by $u \oplus v = round(u + v)$, $u \otimes v = round(u \times v)$, $u \ominus v = round(u - v)$, $u \oslash v = round(u/v)$.

Attention: The floating point operations lose virtually all of the nice properties of the original operations. In particular, associativity and distributivity are lost. However, all floating point operations remain commutative if the originals were so, in particular addition and multiplication with floating point numbers are commutative. See [32].

14.2 Example for an Addition

Let $u = 235.5$ and $v = 22.6$ as an example for the addition algorithm. We expect the result to be $w = 258.1$.

Binary representation

$$u = 1.110101111000000000000000_2 \times 2^7$$

and

$$v = 1.0110100110011001100110\dots_2 \times 2^4$$

Note that v cannot be exactly represented as a binary number with finitely many non-vanishing digits: it is an infinite fraction with periodically recurring digits.

IEEE notation

$$u = \langle 0, 1000\ 0110, 1101\ 0111\ 0000\ 0000\ 0000\ 0000 \rangle$$

and

$$v = \langle 0, 1000\ 0011, 0110\ 1001\ 1001\ 1001\ 1001\ 101 \rangle$$

Note that in this representation the last digit of v has been rounded up.

Knuth's notation

$$u = \langle \langle 0, 1000\ 0101, 0.1110\ 1011\ 1000\ 0000\ 0000\ 0000 \rangle \rangle$$

and

$$v = \langle \langle 0, 1000\ 0010, 0.1011\ 0100\ 1100\ 1100\ 1100\ 1101 \rangle \rangle$$

In particular

$$g_u = 0.11101011110000000000000000$$

and

$$g_v = 0.101101001100110011001101$$

Adding u and v

A.1: nothing to do: u is already greater than v

A.2: $e_w = e_u = 1000\ 0101$

A.3: nothing to do: the exponents differ only by 3

A.4: the division results in $g'_v = 0.000101101001100110011001101$, we now need 27 places to work with this number

A.5: performing the addition $g_w = g_u + g'_v$ we have

$$\begin{array}{r} 0.111010111000000000000000 \\ + 0.00010110100110011001101 \\ \hline g_w = 1.00000010000110011001101 \end{array}$$

A.6: normalization is necessary, because $g_w > 1$.

Normalization

N.1: $g_w \geq 1$, so we continue at N.4.

N.4: $g'_w = g_w/2 = 0.1000000100001100110011001101$, and
 $e'_w = e_w + 1 = 1000\ 0110$
do N.1 again.

N.1: Now, $g'_w < 1$.

N.2: Since $g'_w \geq 1/2$, we continue at N.5.

N.5: $2^{24} \cdot g'_w = 100000010000110011001100.1101$.

Since the first place after the decimal point is 1, so we need to increase this product by 1, divide it by 2^{24} and truncate it after 24 digits:

$$g''_w = 0.100000010000110011001101$$

N.6: Nothing to do, since the exponent is small.

Knuth's notation

Assembling the various parts (s_w , e'_w , g''_w) we get:

$$w = \langle \langle 0, 1000\ 0110, 0.1000\ 0001\ 0000\ 1100\ 1100\ 1101 \rangle \rangle$$

IEEE notation

$$w = \langle 0, 1000\ 0111, 0000\ 0010\ 0001\ 1001\ 1001\ 101 \rangle$$

Binary representation

$$w = 1.00000010000110011001101_2 \times 2^8$$

which translates to

$$1.00820314884185791015625 \times 256 = 258.100006103515625$$

Remark 19 The result of the addition algorithm is quite close, but not exactly equal, to our expected result 258.1. This fact shows that calculations with floating point numbers are always only approximations.

PART II

**Algebra, Formal Logic, and
Linear Geometry**

Monoids, Groups, Rings, and Fields

This chapter introduces the indispensable minimal algebraic structures needed for any further discussion, be it for formal logic and data structures, solving of linear equations, geometry, or differential calculus. The students are asked to study this omnipresent material with particular care.

We shall also see that many of the following structures have been encountered implicitly in the previous theory. We have encountered recurrent “laws” such as associativity, or identity properties. Therefore, the following theory is also an exercise in abstraction, a essential activity in computer science.

15.1 Monoids

Definition 87 A monoid is a pair $(M, *)$ where M is a set, and where $* : M \times M \rightarrow M$ is a “composition” map satisfying these properties:

(i) (Associativity) For any triple $k, l, m \in M$, we have

$$(k * l) * m = k * (l * m),$$

which we also write as $k * l * m$.

(ii) (Neutral element) There is an element $e \in M$ such that

$$m * e = e * m = m$$

for all $m \in M$. Evidently, this element is uniquely determined by this property. It is called the neutral element of the monoid.

A monoid $(M, *)$ is called commutative iff, for all $m, n \in M$,

$$m * n = n * m.$$

Usually, we identify a monoid $(M, *)$ with its set M if the composition $*$ is clear. The notation of the product $m * n$ may vary according to the concrete situation, sometimes even without any notation, such as mn for $m * n$. For commutative monoids, one often uses the symbol $+$ instead of $*$ and says that the monoid is *additive*.

Example 50 Denoting by $*$ the multiplication on $\mathbb{N}, \mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$, these sets define monoids, all of them commutative, with the number 1 being the neutral element in each of them. They are usually called the *multiplicative monoids* of these number domains.

The subset $U = S(\mathbb{C}) = \{z \mid |z| = 1\}$ of \mathbb{C} is a monoid under ordinary multiplication since $|1| = 1$ and $|z \cdot w| = |z| \cdot |w|$, whence $z \cdot w \in U$ if $z, w \in U$. The monoid U is also called the *unit circle*.

Example 51 Given a set X , the set $End(X)$ of set maps $f : X \rightarrow X$, together with the usual composition of maps, is a monoid with neutral element $e = Id_X$. If we restrict the situation to the bijective maps in $End(X)$, we obtain a monoid $Sym(X)$ (same neutral element). This monoid will play a crucial role: We shall discuss it extensively in the following section about groups.

Example 52 If $\Gamma : A \rightarrow V^2$ is a directed graph, the set $End(\Gamma)$ of endomorphisms of Γ , i.e., of morphisms $f = (u, v) : \Gamma \rightarrow \Gamma$, together with the composition \circ of digraph morphisms, is a monoid. In general, this is not a commutative monoid. We have a number of important special cases of this construction.

To begin with, if the arrow set A of Γ is empty, the endomorphism monoid identifies with the monoid $End(V)$.

In section 12.2 on Moore graphs, we introduced the word monoid $Word(A)$ of a set A . A path $p \in Word(A)$ is just a sequence of elements $p = (a_1, a_2, \dots, a_k)$, and also the composition $p = a_1 \cdot a_2 \cdot \dots \cdot a_k$ of the paths a_i of length one. This is why this monoid is also called the *word monoid*

over the alphabet A . A word is then a synonym for “path”, and the letters of the word are the elements of A . The lazy path is called the *empty word*.

As with sets, where we introduced functions or set maps, we need the formalism for comparing different instances of the monoid concept. Here is the evident definition:

Definition 88 Given two monoids $(M, *_M)$ and $(N, *_N)$, a homomorphism $f : (M, *_M) \rightarrow (N, *_N)$ is a set map $f : M \rightarrow N$ such that

- (i) $f(m *_M n) = f(m) *_N f(n)$ for all $m, n \in M$,
- (ii) $f(e_M) = e_N$ for the neutral elements $e_M \in M$ and $e_N \in N$.

Again, if the multiplications are clear, then we just write $f : M \rightarrow N$ to denote a monoid homomorphism. The set of monoid homomorphisms $f : M \rightarrow N$ is denoted by $\text{Monoid}(M, N)$.

If we are given three monoids M, N, O and two monoid homomorphisms $f : M \rightarrow N$ and $g : N \rightarrow O$, their composition $g \circ f : M \rightarrow O$ is defined by the underlying set map composition, and this is also a monoid homomorphism.

Exercise 54 Show that the composition of three monoid homomorphisms, if defined, is associative, and that the identity map $\text{Id}_M : M \rightarrow M$ of any monoid M is a monoid homomorphism, the *identity homomorphism of M* .

Show that for a monoid homomorphism $f : M \rightarrow N$ the following statements are equivalent:

- (i) There is a homomorphism $g : N \rightarrow M$ such that

$$g \circ f = 1_M \text{ and } f \circ g = 1_N,$$

- (ii) f is a bijection of sets.

A homomorphism satisfying these properties is called an *isomorphism* of monoids. One says that two monoids M and N are isomorphic iff there is an isomorphism $f : M \rightarrow N$. The homomorphism g is uniquely determined and is called the inverse of f . It is denoted by $g = f^{-1}$. If $M = N$ (i.e., they are same as monoids, not only as sets), a homomorphism is called an *endomorphism*. The set of monoid endomorphisms, together with the composition of monoid homomorphisms and the identity on M , is itself a monoid and is denoted by $\text{End}(M)$. An isomorphism which is

also an endomorphism is called *automorphism*. The subset of automorphisms in $End(M)$ is also a monoid and is denoted by $Aut(M)$.

Here is the “universal property” of the word monoid construction:

Proposition 111 (Universal Property of the Word Monoid) *Let A be a set, and N a monoid. Then the following map of sets is a bijection:*

$$r : Monoid(Word(A), N) \rightarrow Set(A, N) : f \mapsto f|_A$$

Proof The map r is injective since, if $f(a)$ is given for all letters $a \in A$, then $f(a_1 \cdot a_2 \cdot \dots \cdot a_k) = f(a_1) \cdot f(a_2) \cdot \dots \cdot f(a_k)$, for any word $a_1 \cdot a_2 \cdot \dots \cdot a_k \in Word(A)$, the empty path v must be mapped to the neutral element e_N , and we know f from its action on letters. Conversely, if $g : A \rightarrow N$ is a set map, define $f(a_1 \cdot a_2 \cdot \dots \cdot a_k) = g(a_1) \cdot g(a_2) \cdot \dots \cdot g(a_k)$, which is well defined since the letters of the word $a_1 \cdot a_2 \cdot \dots \cdot a_k$ are uniquely determined. Also, set $f(v) = e_N$. Then this yields a monoid homomorphism, since the multiplication of words is defined as their concatenation. So r is surjective. \square

The word monoid is therefore a kind of “free” object in the sense that any “wild” map on its letters extends uniquely to a monoid homomorphism.

The so-called submonoids constitute a special type of homomorphisms:

Definition 89 *If $(M, *)$ is a monoid with neutral element e , a submonoid $(M', *')$ of M is a subset $M' \subset M$ such that for all $m, n \in M'$, $m * n \in M'$ and $e \in M'$, while the multiplication $*$ ' is the restriction of $*$ to M' . A submonoid therefore gives rise to the evident embedding homomorphism $i_M : M' \hookrightarrow M$.*

Exercise 55 Given a monoid $(M, *)$ and a (possibly empty) subset $S \subset M$, there is a unique minimal submonoid M' of M such that $S \subset M'$. It is denoted by $\langle S \rangle$ and is called the *submonoid generated by S* . Show that $\langle S \rangle$ consists of all finite products $s_1 * s_2 * \dots * s_n$, $s_i \in S$, and of the neutral element e .

Example 53 We have plenty of submonoids among our previous examples (with the ordinary multiplications): $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$, $U \subset \mathbb{C}$. If Γ is a digraph, we have $Aut(\Gamma) \subset End(\Gamma)$.

Exercise 56 The additive monoids, i.e., where $*$ is $+$, also define a chain of submonoids $\mathbb{N} \subset \mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$. If A is any set, the map $l : Word(A) \rightarrow (\mathbb{N}, +)$ defined by the length of a word is a monoid homomorphism, i.e., $l(pq) = l(p) + l(q)$ and $l(e) = 0$.

15.2 Groups

Definition 90 A monoid $(G, *)$ is called a group if every $g \in G$ is invertible, i.e.,

$$\text{there is } h \in G \text{ such that } g * h = h * g = e.$$

The element h is uniquely determined by g and is called the inverse of g and denoted by g^{-1} . A commutative or abelian group is a group which is a commutative monoid.

If G and H are two groups, a monoid homomorphism $f : G \rightarrow H$ is called a group homomorphism. The set of group homomorphisms $f : G \rightarrow H$ is denoted by $\text{Group}(G, H)$. A group isomorphism is a monoid isomorphism among groups. Accordingly, the set of monoid endomorphisms of a group G is denoted by $\text{End}(G)$, while the monoid of automorphisms is denoted by $\text{Aut}(G)$. A subgroup $G \subset H$ is a submonoid $G \subset H$ where the involved monoids G and H are groups.

Example 54 The symmetries on the square form a non-commutative group. The eight elements of the group are: the identity i , the three clockwise rotations r_1 , r_2 and r_3 by angles 90° , 180° and 270° , respectively; further, the four reflections about the horizontal axis h , the vertical axis v , the first diagonal d_1 and the second diagonal d_2 . Figure 15.1 shows a graphical representation of those eight operations. The product of two

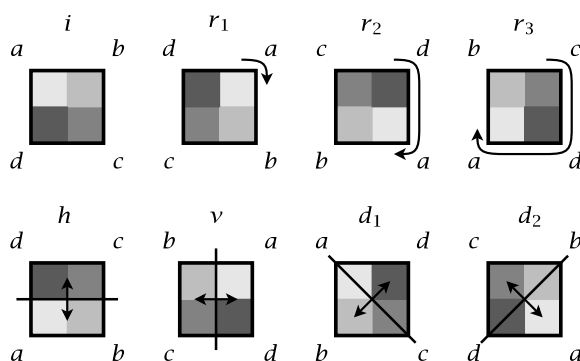


Fig. 15.1. The symmetries on the square.

elements x and y , i.e., $x \cdot y$, is defined as applying the operation y first,

then x . The multiplication table for this group is shown in figure 15.2. Although this group is not commutative, it has commutative subgroups, for example $\{i, r_1, r_2, r_3\}$.

\cdot	i	r_1	r_2	r_3	h	v	d_1	d_2
i	i	r_1	r_2	r_3	h	v	d_1	d_2
r_1	r_1	r_2	r_3	i	d_1	d_2	v	h
r_2	r_2	r_3	i	r_1	v	h	d_2	d_1
r_3	r_3	i	r_1	r_2	d_2	d_1	h	v
h	h	d_2	v	d_1	i	r_2	r_3	r_1
v	v	d_1	h	d_2	r_2	i	r_1	r_3
d_1	d_1	h	d_2	v	r_1	r_3	i	r_2
d_2	d_2	v	d_1	h	r_3	r_1	r_2	i

Fig. 15.2. The multiplication table of the group of symmetries on the square. The entry in row x and column y is the product $x \cdot y$

Example 55 The inclusion chain of additive monoids $\mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$ are all commutative groups. The set $\mathbb{Z}^* = \{1, -1\}$ is a multiplicative group in the multiplicative monoid \mathbb{Z} . More generally, if $(M, *)$ is a monoid, the subset M^* of invertible elements in the sense of definition 90 is a subgroup of M . It is called the *group of invertible elements of M* . Then we have this inclusion chain of groups of invertible elements within the multiplicative monoids of numbers: $\mathbb{Z}^* \subset \mathbb{Q}^* \subset \mathbb{R}^* \subset \mathbb{C}^*$. Observe that $\mathbb{Z}^* = \{-1, 1\}$, $\mathbb{Q}^* = \mathbb{Q} - \{0\}$, $\mathbb{R}^* = \mathbb{R} - \{0\}$, and $\mathbb{C}^* = \mathbb{C} - \{0\}$.

Exercise 57 Show that for a monoid M the submonoid $\text{Aut}(M) \subset \text{End}(M)$ is a group. $\text{Aut}(M)$ is called the *automorphism group of M* .

The submonoid $\text{Sym}(X) \subset \text{End}(X)$ of bijections of a set X is a very important group. It is called the *symmetric group of X* , its elements are called the *permutations of X* . The permutation group of the interval $[1, n] = \{1, 2, 3, \dots, n\}$ of natural numbers is called the *symmetric group of rank n* and denoted by S_n .

Exercise 58 Show that, if $\text{card}(X) = n$ is finite, then there is an isomorphism of groups $\text{Sym}(X) \cong S_n$.

A permutation $p \in S_n$ is best described by cycles. A *cycle* for p is a sequence $C = (c_1, c_2, \dots, c_k)$ of pairwise different elements of $[1, n]$ such that $p(c_1) = c_2, p(c_2) = c_3, \dots, p(c_{k-1}) = c_k, p(c_k) = c_1$, where k is called the length of C and denoted by $l(C)$. The underlying set $\{c_1, c_2, \dots, c_k\}$ is denoted by $|C|$, i.e., $l(C) = \text{card}(|C|)$. Conversely, each sequence $C = (c_1, c_2, \dots, c_k)$ of pairwise different elements of $[1, n]$ denotes by definition a permutation p such that $p(c_1) = c_2, p(c_2) = c_3, \dots, p(c_{k-1}) = c_k, p(c_k) = c_1$, while the other elements of $[1, n]$ are left fixed under p . We also denote this p by C . Given such cycles C_1, C_2, \dots, C_r , one denotes by $C_1 C_2 \dots C_r$ the permutation $C_1 \circ C_2 \circ \dots \circ C_r$. If a cycle C has length 2, i.e., $C = (x, y)$, then it is called a *transposition*, it just exchanges the two elements x and y , and $(x, y)(x, y) = \text{Id}$.

Exercise 59 Let $n = 4$, take $C_1 = (124)$ and $C_2 = (23)$. Calculate the permutation $C_1 C_2$.

Proposition 112 Let $p \in S_n$. Then there is a sequence C_1, C_2, \dots, C_r of cycles of S_n such that

- (i) the underlying sets $|C_i|$ are mutually disjoint,
- (ii) $[1, n] = \bigcup_i |C_i|$,
- (iii) $p = C_1 C_2 \dots C_r$.

The sets $|C_i|$ are uniquely determined by p , and for any two such representations C_1, C_2, \dots, C_r and C'_1, C'_2, \dots, C'_r of p , if $|C_i| = |C'_j|$, then there is an index $1 \leq t \leq l = l(C_i) = l(C'_j)$ such that $C_i = (c_1, c_2, \dots, c_l)$ and $C'_j = (c_t, c_{t+1}, \dots, c_l, c_1, c_2, \dots, c_{t-1})$.

This representation of p is called the cycle representation.

Proof If such a cycle representation of p exists, then if $i \in C_j$, then $C_j = \{p^k(i) \mid k = 0, 1, 2, \dots\}$, and conversely, every such so-called *orbit* set $\{p^k(x) \mid k = 0, 1, 2, \dots\}$ defines the cycle containing x . So the cycles are identified by these orbits which are defined uniquely by p . Such sets define a partition of $[1, n]$. In fact, if $\{p^k(i) \mid k = 0, 1, 2, \dots\} \cap \{p^k(i') \mid k = 0, 1, 2, \dots\} \neq \emptyset$, then there are k and k' such that $p^k(i) = p^{k'}(i')$. Suppose $k \geq k'$. Then multiplying by $(p^{-1})^{k'}$, we obtain $p^{k-k'}(i) = i'$, i.e., the orbit of i' is contained in that of i . But since S_n is finite, not all powers p, p^2, p^3, \dots can be different. If $p^t = p^{l+t}, t \geq 1$, then by multiplication with $(p^{-1})^l$, we have $p^t = \text{Id}$, i.e., $p^{-1} = p^{t-1}$. Therefore $p^{k-k'}(i') = (p^{-1})^{k-k'}(i') = (p^{t-1})^{k-k'}(i') = i$, and both orbits coincide. So the cycle representation is the uniquely determined orbit representation, which is a partition of $[1, n]$, thus (i) and (ii) follow.

Part (iii) follows since the cycles act like p on their elements. The last statement is clear. \square

Exercise 60 Beside the cycle representation of permutations, a tabular representation can be used, where an element on the top line is mapped to the element below it. Find the cycle representation for the following permutations:

$$\begin{array}{cccccccc} 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 6 & 5 & 8 & 7 & 2 & 3 & 1 \end{array} \in S_8 \quad \begin{array}{cccccc} 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 1 & 4 & 2 & 5 & 3 \end{array} \in S_6$$

Definition 91 The cardinality of a group G is called the order of G , it is denoted by $\text{ord}(G)$. A group with finite order is called a finite group. A group G with $\text{ord}(G) = 1$ is called trivial.

Exercise 61 Show that any two trivial groups are isomorphic.

Definition 92 If $n \in \mathbb{N}$, we define $n! = 1$ if $n = 0$, and $n! = 1 \cdot 2 \cdot 3 \cdot 4 \cdot \dots \cdot n$ else. The number $n!$ is called n -factorial.

Exercise 62 Give a rigorous inductive definition of $n!$.

Proposition 113 If $n \in \mathbb{N}$, then

$$\text{ord}(S_n) = n!$$

Proof The image $i = p(n)$ of n under a permutation $p \in S_n$ can be any of the n elements in $[1, n]$. The other elements in $[1, n - 1]$ are sent to the complement $[1, n] - \{i\}$ having $n - 1$ elements. By induction, this gives $(n - 1)!$ possibilities, and we have $n \cdot ((n - 1)!) = n!$, as required, and since the induction start $n = 1$ is evident, we are done. \square

The following result is a fundamental tool for constructing subgroups of a given group:

Proposition 114 Let $X \subset G$ be any subset of a group G . Then there is a unique minimal subgroup H of G such that $X \subset H$. The group H consists of the neutral element e of G and of all finite products $x_1 * x_2 * \dots * x_k$, where either $x_i \in X$ or $x_i = y_i^{-1}$, $y_i \in X$. The subgroup H is denoted by $\langle X \rangle$ and is called the subgroup generated by (the elements of) X .

If there is a finite set $X \subset G$ such that G is generated by X , i.e., $G = \langle X \rangle$, then G is called a finitely generated group. In particular, if $X = \{x\}$, one writes $G = \langle x \rangle$ and calls G a cyclic group.

Proof The uniqueness of the minimal subgroup H results from the fact that for any non-empty family $(G_i)_i$ of subgroups of G , their intersection $\bigcap_i G_i$ is a subgroup of G . The family of all subgroups G_i containing X is not empty (G is in the family), so the intersection is this unique minimal subgroup H . Clearly, H contains all finite products of the described type. Moreover, this set is a subgroup, the inverse of a product $x_1 * x_2 * \cdots * x_k$ being $x_k^{-1} * x_{k-1}^{-1} * \cdots * x_1^{-1}$, which is of the required type. \square

Exercise 63 Find some 2-element sets that generate the group of symmetries on the square (see figures 15.1 and 15.2). Can you state a necessary condition for a 2-element set to generate the group?

Definition 93 For an element x of a group G , the order of $\langle x \rangle$ is called the order of x .

Proposition 115 If $n \geq 2$ is a natural number, then S_n is generated by the set $\{(1, k) \mid k = 2, 3, \dots, n\}$ of transpositions. Since a transposition has order 2, any $x \in S_n$ can be written as a product $x = (1, k_1)(1, k_2) \dots (1, k_r)$.

The subset A_n of the elements $x \in S_n$ which can be written as a product of an even number (a multiple of 2) of transpositions is a subgroup of S_n of order $n!/2$. It is called the alternating group of rank n .

Proof For $n = 2$, S_n is obviously generated by $(1, 2)$. For the general case $n > 2$, let $p(n) = i$ for a given permutation $p \in S_n$. Then $(i, n)p$ fixes n and is an element of S_{n-1} . Therefore, by induction, it is a product of transpositions $(1, r)$, $r < n$. But $(i, n) = (1, i)(1, n)(1, i)$, so $p = (i, n)(i, n)p = (1, i)(1, n)(1, i)((i, n)p)$, and we are done.

Clearly, A_n is a subgroup. If $(1, i)$ is any transposition, then $A_n \rightarrow S_n : p \mapsto (1, i)p$ is a bijective map from A_n to the set B_n of permutations which are products of an odd number of transpositions. If we can show that $A_n \cap B_n = \emptyset$, then S_n is the disjoint union of A_n and B_n , and therefore $S_n = A_n \sqcup B_n$, whence $\text{card}(A_n) = n!/2$. To show this, given a permutation p , denote by $i(p) = (i_1, i_2, \dots, i_n)$ the sequence with $j = p(i_j)$. Let $s(p)$ be 1 if the number of pairs $u < v$ such that $i_u > i_v$ in $i(p)$ is even, and -1 if it is odd. Clearly, $s(\text{Id}) = 1$. If $(i, i + 1)$ is a transposition of neighboring numbers, then evidently $s((i, i + 1) \cdot p) = -s(p)$. Moreover, a general transposition (i, j) is the product of an odd number of transpositions $(k, k + 1)$ of neighboring numbers. In fact, if $j = i + 1$, we are done, and for $j > i + 1$, we have $(i, j) = (i, k)(k, j)(i, k)$ for $i < k < j$, so our claim follows by induction. Therefore $s((i, j)) = -1$. If p is a product of r transpositions, we have $s(p) = -1^r$, and r cannot be even and odd at the same time. So $A_n \cap B_n = \emptyset$. \square

The alternating group A_n is a typical example of a group construction which has far-reaching consequences for all mathematical theories which involve groups, and this is virtually the entire mathematical science. The observation is that A_n is related to a particular group homomorphism $sig : S_n \rightarrow \mathbb{Z}^*$ defined by $sig(x) = 1$ if $x \in A_n$, and $sig(x) = -1$ else (verify that sig is indeed a group homomorphism). The point is that, by definition, $A_n = \{x \in S_n \mid sig(x) = 1\}$, i.e., A_n is the subgroup of elements being sent to the neutral element 1 of the codomain group. The systematic context is this:

Definition 94 The kernel of a group homomorphism $f : G \rightarrow H$ is the subgroup $Ker(f) = \{x \in G \mid f(x) = e_H\}$.

The image of f is the set-theoretic image $Im(f)$, a subgroup of H .

Proposition 116 A group homomorphism f is an injective map iff its kernel is trivial.

Proof If f is injective, then $Ker(f) = f^{-1}(e)$ must be a singleton, so it is trivial. Conversely, if $Ker(f) = e$, then $f(x) = f(y)$ implies $e = f(x)f(y)^{-1} = f(x)f(y^{-1}) = f(xy^{-1}) = e$, whence $xy^{-1} = e$, there $x = y$. \square

Therefore we have $A_n = Ker(sig)$. But there is more: We have two equipollent complementary subsets A_n and $S_n - A_n$ of the full group S_n , and the image group \mathbb{Z}^* is of order 2. This suggests that we may try to reconstruct the codomain group from the domain group and the kernel group. This can effectively be done, but we need a small preliminary discussion.

Definition 95 Given a subgroup $H \subset G$ and an element $g \in G$, the left H -coset of g is the set $gH = \{gh \mid h \in H\}$. The set of left H -cosets of G is denoted by G/H . The right H -coset is the set $Hg = \{hg \mid h \in H\}$. The set of right H -cosets of G is denoted by $H \setminus G$.

Sorite 117 Given a subgroup $H \subset G$, we have these facts:

- (i) The relation " $x \sim y$ iff $x \in yH$ " is an equivalence relation, where the equivalence classes are the left cosets, i.e., two cosets are either disjoint or equal, and $G/H = G/\sim$. Each left coset gH is equipollent with H by means of $h \mapsto gh$. This means that we have a set bijection $G/H \times H \xrightarrow{\sim} G$.
- (ii) The relation " $x \sim y$ iff $x \in Hy$ " is an equivalence relation, the equivalence classes are the right cosets, i.e., two cosets are either

disjoint or equal, and $H \setminus G = G / \sim$. Each right coset Hg is equipollent with H by means of $h \mapsto hg$. We have a set bijection $H \setminus G \xrightarrow{\sim} G/H$, and therefore also $H \setminus G \times H \xrightarrow{\sim} G$.

The common cardinality of G/H or $H \setminus G$ is denoted by $(G : H)$ and is called the index of H in G .

(iii) Let G be a finite group, then we have the Lagrange equation

$$\text{card}(G) = \text{card}(H) \cdot (G : H).$$

In particular, the order of any subgroup H of a finite group G divides the order of G . More specifically, if $x \in G$ is any element and if G is finite, then the order of x divides the order of G .

Proof (i) The relation $x \sim y$ means that $x = yh$, for some $h \in H$. Taking $h = e$, we obtain $x \sim x$, and from $x = yh$ one deduces $y = xh^{-1}$, i.e., $y \sim x$. Finally, $x = yh$ and $y = zk$, $k \in H$, implies $x = zkh$, whence $x \sim z$.

The surjection $f_g : H \rightarrow gH : h \mapsto gh$, $g \in G$, is an injection because $f_g^{-1}(gh) = f_{g^{-1}}(gh) = g^{-1}(gh) = h$. So $G/H \times H \xrightarrow{\sim} G$. The proof of (ii) works similarly with "left" and "right" being exchanged.

The Lagrange equation (iii) now follows from (ii) since $\text{card}(G) = \text{card}(G/H \times H) = \text{card}(G/H) \cdot \text{card}(H) = (G : H) \cdot \text{card}(H)$. \square

The kernel of a homomorphism is more than a subgroup, it is exactly such that the set of cosets G/H can be made into a group in a canonical way.

Proposition 118 Let H be a subgroup of a group G . Then the following properties are equivalent:

- (i) There is a group K and a group homomorphism $f : G \rightarrow K$ such that $H = \text{Ker}(f)$.
- (ii) Left and right cosets coincide, i.e., for all $x \in G$, $xH = Hx$. The composition $xH \cdot yH = xyH$ then defines a group structure, the quotient group, on G/H . The group H is the kernel of the group homomorphism $G \rightarrow G/H : x \mapsto xH$.

Proof (i) implies (ii): If $H = \text{Ker}(f)$, then for every $x \in G$ and every $h \in H$, $f(xhx^{-1}) = f(x)f(h)f(x^{-1}) = f(x)f(x^{-1}) = f(x)f(x)^{-1} = e$, so $xHx^{-1} \subset H$, but also $x^{-1}Hx \subset H$, whence $xHx^{-1} = H$, i.e., $xH = Hx$, for all x .

(ii) implies (i): The composition $xH \cdot yH = xyH$ is well defined since if $xH = x'H$, then $xyH = xHy = x'H y = x'yH$, and if $yH = y'H$, then $xyH = xy'H$. It is a group composition having H as neutral element and $x^{-1}H$ as the inverse

of xH . The map $f : G \rightarrow G/H : g \mapsto gH$ is a surjective homomorphism, and $gH = H$ iff $g \in H$. So $H = \text{Ker}(f)$. \square

Definition 96 A subgroup $H \subset G$ with the equivalent properties of proposition 118 is called a normal subgroup of G ; the group G/H is called the quotient group of G modulo H .

Taking proposition 118 (i) with $G = S_n$, $H = A_n$, $K = \mathbb{Z}^*$ and $f = \text{sig}$, we see that the alternating group A_n is a normal subgroup of S_n .

Exercise 64 Show that every subgroup of a commutative group is normal. For example, given $n \in \mathbb{N}$, we consider the additive subgroup $\langle n \rangle \subset \mathbb{Z}$. Show that this is the group consisting of all multiples $z \cdot n$, $z \in \mathbb{Z}$ of n ; therefore, we also write $\mathbb{Z} \cdot n$ for $\langle n \rangle$.

Definition 97 The quotient group $\mathbb{Z}/(\mathbb{Z} \cdot n)$ is denoted by \mathbb{Z}_n and is called the cyclic group of order n . Its order is indeed n . In fact, by the Euclidean division theorem, every integer w is written uniquely in the form $w = q \cdot n + r$, $0 \leq r < n$. Therefore every coset is of the form $r + \mathbb{Z} \cdot n$, $0 \leq r < n$, and if $r + \mathbb{Z} \cdot n = r' + \mathbb{Z} \cdot n$, then $r = r' + q \cdot n$, therefore $q = 0$, and $r' = r$.

If we deal with elements of \mathbb{Z}_n , they are represented by cosets $r + \mathbb{Z}$. Often, one makes calculations on \mathbb{Z}_n but works with such representatives. In order to tell that two representatives $r, s \in \mathbb{Z}$ represent the same coset, one writes $r \equiv s \pmod{n}$ or $r \equiv s \pmod{(n)}$ or else $r \equiv s \pmod{(n)}$ or even $r = s \pmod{(n)}$.

We are now ready for the reconstruction of the image $\text{Im}(f)$ of a group homomorphism f by means of the domain group and the kernel $\text{Ker}(f)$:

Proposition 119 If $f : G \rightarrow H$ is a group homomorphism, there is a canonical isomorphism of groups

$$p : G/\text{Ker}(f) \xrightarrow{\sim} \text{Im}(f)$$

given by $p(g\text{Ker}(f)) = f(g)$ and satisfying the following commutative diagram:

$$\begin{array}{ccc} G & \xrightarrow{f} & H \\ \downarrow & & \uparrow \\ G/\text{Ker}(f) & \xrightarrow{\sim} & \text{Im}(f) \end{array}$$

Proof The map $p : G/\text{Ker}(f) \rightarrow \text{Im}(f)$ is well defined since $g\text{Ker}(f) = h\text{Ker}(f)$ iff $g^{-1}h \in \text{Ker}(f)$, so $e = f(g^{-1}h) = f(g^{-1})f(h)$, i.e., $f(g) = f(h)$. It is surjective by construction and also a group homomorphism by the definition of the quotient group. Its kernel is the set of those cosets $g\text{Ker}(f)$ such that $f(g) = e$, i.e., $g \in \text{Ker}(f)$, i.e., $g\text{Ker}(f) = \text{Ker}(f)$, the neutral element of the quotient group $G/\text{Ker}(f)$. So by proposition 116 it is injective. \square

Example 56 Reconsidering the homomorphism $\text{sig} : S_n \rightarrow \mathbb{Z}^*$, we know that $\text{Ker}(\text{sig}) = A_n$, and $\text{Im}(\text{sig}) = \mathbb{Z}^*$. Therefore, by proposition 119, $S_n/A_n \xrightarrow{\sim} \mathbb{Z}^*$.

15.3 Rings

Although groups are fundamental to mathematics, their structure is somewhat too poor for realistic applications. We have seen in chapter 9 that the important number domains $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ share a simultaneous presence of two kinds of operations: addition and multiplication. We have even learned that the very construction of the real numbers needs the auxiliary space of Cauchy sequences, which also shares addition and multiplication with the other spaces. Here is the precise statement about the common properties of these combined operations:

Definition 98 A ring is a triple $(R, +, *)$ such that $(R, +)$ is an additive abelian group with neutral element 0_R (or 0 if the context is clear), and where $(R, *)$ is a multiplicative monoid with neutral element 1_R (or 1 if the context is clear). These two structures are coupled by the distributivity laws:

$$\begin{aligned} \text{for all } x, y, z \in R, \quad & x * (y + z) = x * y + x * z \\ \text{and} \quad & (x + y) * z = x * z + y * z. \end{aligned}$$

One refers to $(R, +)$ when referring to the additive group of a ring and to $(R, *)$ when referring to the multiplicative monoid of the ring. Usually, one simply writes R for the ring $(R, +, *)$ if addition and multiplication are clear from the context. The group of multiplicatively invertible elements of a ring is denoted by R^* . If the context is clear, then one often writes ab or $a \cdot b$ or $a.b$ instead of $a * b$.

A ring is commutative iff its multiplicative monoid is so.

A subring of a ring is a ring which is simultaneously an additive subgroup and a multiplicative submonoid.

Example 57 As already announced, we have a number of prominent rings from previous theory. If we denote by $+$ and $*$ the ordinary addition and multiplication, respectively, in the number domains $\mathbb{Z}, \mathbb{Q}, \mathbb{R}$ and \mathbb{C} , then each of $(\mathbb{Z}, +, *)$, $(\mathbb{Q}, +, *)$, $(\mathbb{R}, +, *)$, and $(\mathbb{C}, +, *)$ is a commutative ring. Moreover, the set of Cauchy sequences C , together with the sum and product of Cauchy sequences, is a commutative ring.

The additive cyclic groups $(\mathbb{Z}_n, +)$ can also be turned into commutative rings by defining the multiplication $(r + n \cdot \mathbb{Z}) * (s + n \cdot \mathbb{Z}) = rs + n \cdot \mathbb{Z}$. Evidently, this multiplication is well defined, since two representations $r \equiv r' \pmod{n}$ have their difference in $n \cdot \mathbb{Z}$.

Exercise 65 Verify that multiplication in \mathbb{Z}_n is well defined. What is the multiplicative neutral element $1_{\mathbb{Z}_n}$?

Rings are also related to each other by ring homomorphisms:

Definition 99 *A set map $f : R \rightarrow S$ of rings R and S is a ring homomorphism if f is a group homomorphism of the additive groups of R and S and if f is a monoid homomorphism of the multiplicative monoids of R and S . The set of ring homomorphisms from R to S is denoted by $\text{Ring}(R, S)$.*

The composition of two ring homomorphisms $f : R \rightarrow S$ and $g : S \rightarrow T$ is the set-theoretic composition $g \circ f$. The properties of group and monoid homomorphisms imply that composition is also a ring homomorphism.

An isomorphism f of rings is a homomorphism of rings which is an isomorphism of additive groups and of multiplicative monoids (which is equivalent to the condition that f is a bijective set map). An endomorphism or rings is a homomorphisms of one and the same ring. A ring automorphism is an endomorphism which is an isomorphism of rings.

Example 58 The inclusions $\mathbb{Z} \subset \mathbb{Q} \subset \mathbb{R} \subset \mathbb{C}$ are ring homomorphisms. The conjugation $\bar{} : \mathbb{C} \xrightarrow{\sim} \mathbb{C}$ is an automorphism of \mathbb{C} . The canonical maps $\mathbb{Z} \rightarrow \mathbb{Z}_n$ are ring homomorphisms.

Example 59 A crucial construction of rings which are not commutative in general is the so-called “monoid algebra”. A very important special case

is the omnipresent polynomial ring, so the following construction is far from an academic exercise.

To construct a monoid algebra, we need a commutative ring $(R, +, \cdot)$ and a monoid $(M, *)$. The monoid algebra of R and M is the following ring: Its set is the subset $R\langle M \rangle$ of the set R^M consisting of all functions $h : M \rightarrow R$ such that $h(z) = 0_R$ for all but a finite number of arguments $z \in M$. Given a pair $(r, m) \in R \times M$, the special function $f : M \rightarrow R$ with $f(m) = r$ and $f(n) = 0$ for all $n \neq m$ is denoted by $r \cdot m$. The addition on $R\langle M \rangle$ is defined by $(f + g)(m) = f(m) + g(m)$. Clearly this function evaluates to 0_R for all but a finite number of arguments if f and g do so. The product is defined by $(f \cdot g)(m) = \sum_{n,t} g(n) \cdot f(t)$, where the sum is taken over the finite number of pairs (n, t) such that $n * t = m$ and either $g(n)$ or $f(t)$ differs from 0_R . If there is no such non-zero value, the product is defined to be the zero function $0_R \cdot e_M$.

Exercise 66 Given a ring R and a monoid M as above, show that the monoid algebra $R\langle M \rangle$ defined in example 59 is a ring with additive neutral element $0_R \cdot e_M$ and multiplicative neutral element $1_R \cdot e_M$. It is commutative iff M is so. Every element $f \in R\langle M \rangle$ can be written as a finite sum $f = \sum_{i=1, \dots, k} f_i \cdot m_i = f_1 \cdot m_1 + f_2 \cdot m_2 + \dots + f_k \cdot m_k$, with $f_i = f(m_i)$. If $f \neq 0$, the the summands $f_i \cdot m_i$ are uniquely determined up to permutations if one only adds up those with $f_i \neq 0$. One therefore also uses the notation $f = \sum_m f_m \cdot m$ where it is tacitly supposed that only a finite number of summands is considered, of course comprising only those m where $f(m) \neq 0$.

There is an injective ring homomorphism $R \rightarrow R\langle M \rangle : r \mapsto r \cdot e_M$, and an injective homomorphism $M \rightarrow R\langle M \rangle : m \mapsto 1_R \cdot m$ of multiplicative monoids. One therefore also identifies elements of R and M their respective images in the monoid algebra. We observe that under this identification, any two elements $r \in R$ and $f \in R\langle M \rangle$ commute, i.e., $r \cdot f = f \cdot r$. This is the theoretical reason why the word “ R -algebra” comes into this construction.

Example 60 If in particular $M = \text{Word}(A)$ is the word monoid of an alphabet A , then the monoid algebra is called the *R -algebra of non-commutative polynomials in the indeterminates from A* . So every element is a sum of so-called *monomials* $r \cdot X_1 X_2 \dots X_k$ in the indeterminates $X_i \in A$. In particular, if $A = \{X_1, X_2, \dots, X_n\}$ is a finite set of

so-called “indeterminates” X_i , then the monoid algebra is denoted by $R\langle X_1, X_2, \dots, X_n \rangle$.

The most prominent such algebra is the case for $A = \{X\}$. We then get the words $X^k = XXX \dots X$, the k -fold juxtaposition of the unique letter X . A polynomial in the indeterminate X is then written in the form

$$f = r_k \cdot X^k + r_{k-1} \cdot X^{k-1} + r_{k-2} \cdot X^{k-2} + \dots + r_2 \cdot X^2 + r_1 \cdot X + r_0,$$

or else $f(X)$ in order to put the indeterminate X in evidence. The monoid algebra is then commonly denoted by $R[X]$ instead of $R\langle X \rangle$. Compare addition and multiplication of such polynomials in X to what you know from high school mathematics.

For a non-zero polynomial in X , the maximal power X^k of X such that the coefficient r_k is different from zero is called the degree of f , in symbols $\deg(f)$, and r_k is called the *leading coefficient*.

In particular, if the degree of f is 3 the polynomial $f(X)$ is called *cubic*, if it is 2, then $f(X)$ is called *quadratic*, if it is 1, then $f(X)$ is called *linear*, and if $f(X) = r_0 \in R$, then $f(X)$ is called a *constant polynomial*.

Remark 20 Consider the monoid $\text{ComWord}(A)$ whose elements are equivalence classes of words over A in the sense that $w \sim w'$ iff the words' letters are just permutations of each other. Then the product of words from $\text{Word}(A)$ is well defined on equivalence classes, which are also called commutative words. Taking the monoid $\text{ComWord}(A)$, the monoid algebra is commutative and is denoted by $R[A] = R\langle \text{ComWord}(A) \rangle$. It is called the *R -algebra of commutative polynomials in the indeterminates from A* .

The power of the monoid algebra construction is shown in this proposition:

Proposition 120 (Universal Property of Monoid Algebras) *Let R be a commutative ring, M a monoid, and S a (not necessarily commutative) ring. Suppose that $f : R \rightarrow S$ is a ring homomorphism such that for all $r \in R$ and $s \in S$, $f(r) \cdot s = s \cdot f(r)$. Then for any monoid homomorphism $\mu : M \rightarrow S$ into the multiplicative monoid of S , there is exactly one ring homomorphism $f\langle \mu \rangle : R\langle M \rangle \rightarrow S$ which extends f and μ , i.e., $f\langle \mu \rangle|_R = f$ and $f\langle \mu \rangle|_M = \mu$.*

Proof On an argument $\sum_m g_m \cdot m$, any ring homomorphism $f\langle\mu\rangle$ which extends f and μ as required, must be defined by $f\langle\mu\rangle(\sum_m g_m \cdot m) = \sum_m f\langle\mu\rangle(g_m) \cdot f\langle\mu\rangle(m) = \sum_m f(g_m) \cdot \mu(m)$. But this is a well defined map since the sum representation of the argument is unique. It is now straightforward to check that this map is indeed a ring homomorphism. \square

The most important consequence of this proposition is this corollary:

Corollary 121 *If R is a commutative ring, and if $\{X_1, X_2, \dots, X_n\}$ is a finite set, then every set map $v : X_i \mapsto x_i \in R$ extends to a ring homomorphism $R\langle v \rangle : R\langle X_1, X_2, \dots, X_n \rangle \rightarrow R$, whose value on a monomial $r \cdot X_{i_1} X_{i_2} \dots X_{i_k}$ is $r \cdot x_{i_1} x_{i_2} \dots x_{i_k}$. This homomorphism is called the evaluation of polynomials with respect to v .*

Proof This follows at once from proposition 111 and proposition 120, applying the universal property of the word monoid over the symbols X_i to obtain μ , and the universal property of a monoid algebra for $f = Id_R$ and μ . \square

Example 61 If $f = f(X) = r_k \cdot X^k + r_{k-1} \cdot X^{k-1} + r_{k-2} \cdot X^{k-2} + \dots + r_2 \cdot X^2 + r_1 \cdot X + r_0$ is a polynomial in $\mathbb{C}[X]$, then the map $X \mapsto x \in \mathbb{C}$ defines the *evaluation* $f(x) = r_k \cdot x^k + r_{k-1} \cdot x^{k-1} + r_{k-2} \cdot x^{k-2} + \dots + r_2 \cdot x^2 + r_1 \cdot x + r_0$ of $f(X)$ at x . This means that a polynomial $f(X)$ defines a function $f(?) : \mathbb{C} \rightarrow \mathbb{C}$, which is called a *polynomial function*. Generally speaking, a polynomial function is simply a function defined by a polynomial and an evaluation of its indeterminates, as guaranteed by the above proposition 120.

In analogy to normal subgroups, which are the kernels of group homomorphisms, it is possible to define structures which are the kernels of ring homomorphisms. We shall only need the theory for commutative rings. Here is the characterization:

Proposition 122 *Let J be an additive subgroup of a commutative ring R . Then the following properties are equivalent:*

- (i) *There is a homomorphism of commutative rings $f : R \rightarrow S$ such that $J = \text{Ker}(f)$ for the underlying additive groups.*
- (ii) *J is a subgroup of the additive group of R , and for every $r \in R$, if $x \in J$, then $r \cdot x \in J$. The multiplication $(x + J) \cdot (y + J) = xy + J$ defines the multiplication of a ring structure on the quotient group R/J , the quotient ring. The group J is the kernel of the ring homomorphism $R \rightarrow R/J : x \mapsto x + J$.*

Proof The proof is completely analogous to that regarding normal subgroups and kernels of group homomorphisms. We leave it as an exercise to the reader. \square

Definition 100 A subgroup $J \subset R$ with the equivalent properties of proposition 122 is called an ideal of R .

Example 62 We are now capable to better understand the construction of the ring of real numbers \mathbb{R} from Cauchy sequences. The set C of Cauchy sequences as defined in definition 50 is a commutative ring, this is statement (i) in proposition 79, together with the constant sequence $(1)_i$ as multiplicative neutral element, while statements (i) and (ii) in proposition 79 tell us that the zero sequences \mathcal{O} define an ideal. Finally, by lemma 81, the equivalence relation in lemma 80 is precisely the relation defined by the ideal \mathcal{O} , and the quotient structure C/\mathcal{O} is the same as the structure defined in definition 54.

Example 63 Reconsidering the construction of the cyclic groups \mathbb{Z}_n , we recognize that the subgroup $\mathbb{Z} \cdot n$ of \mathbb{Z} is in fact an ideal. The construction of the ring multiplication on the quotient group $\mathbb{Z}/(\mathbb{Z} \cdot n)$ is exactly the one we just defined for the quotient ring.

The type of ideal in the previous example is of major importance for the theory of prime numbers and follows from the Euclidean algorithm:

Definition 101 An ideal J of a commutative ring R is called a principal ideal if there is an element $x \in J$ such that $J = R \cdot x = \{r \cdot x \mid r \in R\}$. Such an ideal is also denoted by $J = (x)$. A ring the ideals of which are all principal is called a principal ideal ring.

Proposition 123 The ring \mathbb{Z} is a principal ideal ring. If J is an ideal of \mathbb{Z} , then either $J = (0)$ or $J = (n)$, where $n \in \mathbb{N}$ is the smallest positive integer in J .

Proof If J is a non-zero ideal in \mathbb{Z} , then there is a smallest positive element n in J . For any $j \in J$, we have $j = an + b$, $0 \leq b < n$, by elementary natural arithmetic. But then $b = j - an \in J$, whence $b = 0$, therefore $J = (n)$. \square

We also have the analogous ring-theoretic proposition analogous to proposition 119 for groups:

Proposition 124 If $f : R \rightarrow S$ is a homomorphism of commutative rings, there is a canonical isomorphism of rings

$$R/\text{Ker}(f) \xrightarrow{\sim} \text{Im}(f).$$

It is given by the map $g + \text{Ker}(f) \mapsto f(g)$ and satisfies the commutative diagram

$$\begin{array}{ccc} R & \xrightarrow{f} & S \\ \downarrow & & \uparrow \\ R/\text{Ker}(f) & \xrightarrow{\sim} & \text{Im}(f) \end{array}$$

Proof The proof of this proposition follows the same line as that of proposition 119, we leave it as an exercise. \square

We shall see in the following section that this proposition entails the remarkable isomorphism of rings $\mathbb{C} \xrightarrow{\sim} \mathbb{R}[X]/(X^2 + 1)$. To this end we need some extra properties specific to rings such as \mathbb{R} .

15.4 Fields

Definition 102 A ring $K \neq 0$ is called a skew field if every non-zero element is invertible. A commutative skew field is also called a field.

Example 64 The rings \mathbb{Q} , \mathbb{R} and \mathbb{C} are fields. This follows right from the properties of these rings which we have exhibited in chapter 9.

The main fact about polynomial rings $K[X]$ over fields K is that they are principal ideal rings. To see this, we need the following lemma:

Lemma 125 If K is a field, and if f and g are non-zero polynomials in $K[X]$, then

$$\text{deg}(f \cdot g) = \text{deg}(f) + \text{deg}(g).$$

Proof If $f = a_n X^n + a_{n-1} X^{n-1} + \dots + a_0$ and $g = b_m X^m + b_{m-1} X^{m-1} + \dots + b_0$ with $a_n, b_m \neq 0$, then $f \cdot g = a_n b_m X^{n+m} + \dots + a_0 b_0$, the highest coefficient $a_n b_m \neq 0$ because we are in a field, so $\text{deg}(f \cdot g) = \text{deg}(f) + \text{deg}(g)$. \square

This entails Euclid's division theorem for polynomial rings over fields:

Proposition 126 (Division Theorem) If f and g are non-zero polynomials in the polynomial algebra $K[X]$ over a field K , then there is a polynomial h such that either $f = h \cdot g$ or $f = h \cdot g + r$, where $\text{deg}(r) < \text{deg}(g)$. The polynomials h and r are uniquely determined.

Proof Let $\deg(f) < \deg(g)$, then $f = 0 \cdot g + f$ is a solution. If $\deg(f) \geq \deg(g)$, then for $f = a_n X^n + a_{n-1} X^{n-1} + \dots + a_0$ and $g = b_m X^m + b_{m-1} X^{m-1} + \dots + b_0$, we consider $f' = f - \frac{a_n}{b_m} X^{n-m} \cdot g$. The polynomial f' has smaller degree than f , and we may proceed by induction to find a solution $f' = h' \cdot g + r'$. Then $f = (\frac{a_n}{b_m} X^{n-m} + h') \cdot g + r'$ solves the problem. As to uniqueness, if we have two decompositions $f = h_1 \cdot g + r_1 = h_2 \cdot g + r_2$, then $(h_1 - h_2) \cdot g = r_2 - r_1$, which, for reasons of degree, only works if $h_1 - h_2 = r_2 - r_1 = 0$. \square

Example 65 To compute the quotient and remainder of the division of $2X^4 + 4X^3 + 17X^2 + 13X + 23$ by $2X^2 + 5$, the tabular method can be used:

$$\begin{array}{r}
 2x^4 + 4x^3 + 17x^2 + 13x + 23 \quad : \quad 2x^2 + 5 = \quad x^2 + 2x + 6 \\
 \underline{2x^4 \qquad + 5x^2} \\
 4x^3 + 12x^2 + 13x \\
 \underline{4x^3 \qquad + 10x} \\
 12x^2 + 3x + 23 \\
 \underline{12x^2 \qquad + 30} \\
 3x + -7
 \end{array}$$

Thus, the quotient is $x^2 + 2x + 6$ and the remainder is $3x - 7$.

The Division Theorem implies the announced result:

Proposition 127 *If K is a field, then the polynomial ring $K[X]$ is a principal ideal ring. If J is an ideal of $K[X]$, then either $J = (0)$ or $J = (g)$, where g is a polynomial in J of minimal degree.*

Proof In fact, the proof works like for integers. Take a minimal polynomial f of positive degree in an ideal J , then by proposition 126, $J = (f)$. \square

We may now demonstrate the announced isomorphism between \mathbb{C} and $\mathbb{R}[X]/(X^2 + 1)$. Consider the ring homomorphism $v : \mathbb{R}[X] \rightarrow \mathbb{C}$ defined by the natural embedding $\mathbb{R} \subset \mathbb{C}$ and the evaluation $X \mapsto i = \sqrt{-1}$. Clearly, v is surjective, i.e., $Im(v) = \mathbb{C}$. The kernel of v is a principal ideal ring, $Ker(v) = (t)$. But no non-zero linear polynomial $a \cdot X + b$ is in this kernel, since $a \cdot i + b = 0$ iff $a = b = 0$. On the other hand, $X^2 + 1 \in Ker(v)$, therefore by proposition 127, $Ker(v) = (X^2 + 1)$. By proposition 124, this implies $\mathbb{R}[X]/(X^2 + 1) \cong \mathbb{C}$.

Exercise 67 Using the isomorphism $\mathbb{R}[X]/(X^2 + 1) \cong \mathbb{C}$, try to describe the conjugation on \mathbb{C} by means of the quotient ring $\mathbb{R}[X]/(X^2 + 1)$.

For both constructions, that of \mathbb{R} from Cauchy sequences, and that of \mathbb{C} from polynomials over \mathbb{R} , we are left with the question: Why do these quotient rings yield fields? Here is the answer:

Definition 103 A proper ideal J of a commutative ring R is called maximal if there is no ideal I such that $J \subsetneq I \subsetneq R$.

Proposition 128 An ideal J is maximal in R iff the quotient R/J is a field.

Proof If J is not maximal, then a strictly larger ideal I such that $J \subsetneq I \subsetneq R$ has an image I' in R/J which is a proper ideal such that $0 \subsetneq I' \subsetneq R/J$, so R/J cannot be a field. In fact, suppose that R/J is a field and take $x \neq 0 \in I'$. Take its inverse x^{-1} . Then, since I' is an ideal, $x^{-1}x = 1$ is an element of I' . Therefore $I' = R/J$. If J is maximal, then any $x \notin J$ must generate the improper ideal R together with J , in particular there are $u \in R$ and $v \in J$ such that $1 = ux + v$. Then the image $1 + J = ux + J = (u + J) \cdot (x + J)$, i.e., every non-zero element $x + J$ in R/J is invertible, and we have a field. \square

Now, the missing links are easily inserted:

Example 66 The ideal (X^2+1) in $\mathbb{R}[X]$ is maximal. In fact, a strictly larger ideal must be of shape $(X+a)$, but then we must have the factorization $X^2+1 = (X+a) \cdot (X+b)$. Evaluating (X^2+1) at $x = -a$ yields $a^2+1 = 0$, an impossibility in \mathbb{R} . Turning to Cauchy sequences, any non-zero sequence $(a_i)_i$ has a sequence $(b_i)_i$ such that $(a_i)_i \cdot (b_i)_i$ converges to 1, this was shown in statement (viii) of sorite 84. So every ideal which is strictly larger than \mathcal{O} must be the principal ideal (1) , i.e., the entire ring, and we are done.

CHAPTER 16

Primes

This chapter is devoted to prime numbers and prime polynomials, a type of objects which play an important role in cryptography. Every computer scientist should therefore have some knowledge about prime factorization in the ring of integers \mathbb{Z} and the polynomial algebras $K[X]$ over one indeterminate X with coefficients in a field K .

16.1 Prime Factorization

We first need a small preliminary discussion about the construction of ideals by generators, a generalization of the concept of a principal ideal introduced in the last chapter.

Definition 104 *If $G \subset R$ is a non-empty subset of a commutative ring R , the ideal generated by G is defined as the set of all sums $r_1 \cdot g_1 + r_2 \cdot g_2 + \dots + r_k \cdot g_k$ where $g_i \in G$ and $r_i \in R$. It is denoted by (G) or (h_1, h_2, \dots, h_t) if $G = \{h_1, h_2, \dots, h_t\}$ is finite.*

Exercise 68 Show that that the set (G) is indeed an ideal.

Here is the definition of a prime element in a commutative ring:

Definition 105 *A non-zero element $p \in R$ in a commutative ring R is prime if it is not invertible and if $p = q \cdot r$ implies that either q or r is invertible. For polynomial algebras, prime polynomials are also called irreducible.*

To begin with, we make sure that in the most interesting rings, every non-invertible element has a factorization into prime elements:

Lemma 129 *If $R = \mathbb{Z}$ or $R = K[X]$, where K is a field, then every non-invertible element $x \neq 0$ has a factorization $x = p_1 \cdot p_2 \cdot \dots \cdot p_k$ as a product of primes p_i .*

Proof For a non-invertible, non-zero $p \in \mathbb{Z}$, we proceed by induction on $|p| \geq 2$. A factorization $p = q \cdot r$ implies $|p| = |q| \cdot |r|$. So if $|p| = 2$, then $p = \pm 2$ is already prime, since one of its factors in $2 = q \cdot r$ must be 1. In the general case, a factorization $p = q \cdot r$ with non-invertible factors implies $|q|, |r| < |p|$, so by induction on these factors, we are done. For a non-invertible, non-zero $p \in K[X]$, we proceed by induction on $\deg(p) \geq 1$. A factorization $p = q \cdot r$ implies $\deg(p) = \deg(q) + \deg(r)$. If $\deg(p) = 1$, then either q or r is a constant (degree zero), and therefore invertible. If $\deg(p) > 1$, then if all factorizations have either q or r of degree zero, then these factors are invertible and p is prime. Else, degrees decrease and induction applies to the factors. \square

One could imagine that there is only a finite number of primes. But we have the classical result about prime numbers in \mathbb{Z} , Euclid's theorem:

Proposition 130 *The set of primes in \mathbb{Z} is infinite.*

Proof Suppose that $p_1 = 2, p_2 = 3, \dots, p_k$ is the finite set all positive primes in \mathbb{Z} . Then the prime factorization of $q = 1 + \prod_{i=1,2,\dots,k} p_i$ must contain one of these primes, say p_t . Then we have $q = p_t \cdot u = 1 + \prod_{i=1,2,\dots,k} p_i$, and therefore¹ $1 = p_t \cdot (u - \prod_{i=1,2,\dots,\hat{t},\dots,k} p_i)$, which is impossible since p_t is not invertible by hypothesis. So there are infinitely many primes in \mathbb{Z} . \square

Example 67 In particular 0_R is not prime since $0_R = 0_R \cdot 0_R$, and $1_R, -1_R$ are not prime since they are invertible. The numbers $\pm 2, \pm 3, \pm 5, \pm 7, \pm 11, \pm 13, \pm 17, \pm 19$ in \mathbb{Z} are primes, while $12, -24, 15$ are not. In $\mathbb{R}[X]$ all linear polynomials $a \cdot X + b, a \neq 0$, and all quadratic polynomials $a \cdot X^2 + b$ with $a, b > 0$ are prime. (Use the argumentation from example 66 to prove that $X^2 + 1 = (u \cdot X + v)(r \cdot X + s)$ is impossible in $\mathbb{R}[X]$.)

Clearly, prime factorization is not unique because of the commutativity of the ring and the existence of invertibles, e.g., $12 = 2 \cdot 2 \cdot 3 =$

¹ The convention \hat{x}_i means that in an indexed sequence, the object x_i with index i is omitted, and only the earlier and later terms are considered. For example, $x_1, x_2, x_3, \dots, \hat{x}_9, \dots, x_{20}$ means that we take the sequence $x_1, x_2, x_3, \dots, x_8, x_{10}, \dots, x_{20}$.

$3 \cdot (-2) \cdot (-2)$. However, this is the only source of ambiguities in prime factorization for our prominent rings.

Definition 106 If x and y are two elements in a commutative ring such that there is an element z with $y = x \cdot z$, then we say that x divides y or that x is a divisor of y and write $x|y$. If x does not divide y , we write $x \nmid y$. Clearly, $x|y$ is equivalent to the inclusion $(y) \subset (x)$ of ideals.

If $0 = x \cdot y$ for $x, y \neq 0$, then x is called a zero divisor. A commutative ring without zero divisors is called an integral domain.

Lemma 131 In an integral domain R , the generator of a principal ideal a is unique up to invertible elements, i.e., $(a) = (b)$ iff there is $c \in R^*$ such that $b = c \cdot a$.

Proof Clearly the existence of such an $c \in R^*$ is sufficient. Conversely, $(a) = (b)$ implies $a = xb$ and $b = ya$, whence $a = xya$, i.e., $xy = 1$ if $a \neq 0$. But if $a = 0$, then also $b = 0$ and we have discussed all cases. \square

Notation 10 If R is a principal integral domain, the ideal generated by two elements a and b is principal, i.e., $(a, b) = (d)$. This is equivalent to the two facts that (1) $d|a$ and $d|b$, (2) there are two elements u and v such that $d = u \cdot a + v \cdot b$. Such a d is called the greatest common divisor of a and b , in symbols $d = \gcd(a, b)$. The ideal $(a) \cap (b)$ is also principal, $(a) \cap (b) = (l)$. This means that (1) $a|l$ and $b|l$, (2) whenever $a|x$ and $b|x$, then $l|x$. Such an l is called the least common multiple of a and b , in symbols $l = \text{lcm}(a, b)$. Observe that \gcd and lcm are only determined up to invertible elements.

Proposition 132 (Euclidean Algorithm) For two integers a and b the $\gcd(a, b)$ is calculated by this recursive algorithm:

$$\gcd(a, b) = \begin{cases} a & \text{if } b = 0 \\ \gcd(b, r) & \text{otherwise} \end{cases}$$

where r is determined by the Division Theorem: $a = q \cdot b + r$ (proposition 68).

Proof Suppose that we have this chain of successive divisions:

$$\begin{aligned}
a &= q_1 \cdot b + r_1, \\
b &= q_2 \cdot r_1 + r_2, \\
r_1 &= q_3 \cdot r_2 + r_3, \\
r_2 &= q_4 \cdot r_3 + r_4, \\
&\vdots \\
r_k &= q_{k+2} \cdot r_{k+1}.
\end{aligned}$$

Then the claim is that $r_{k+1} = \gcd(a, b)$. In fact, clearly $r_{k+1} | a$ and $r_{k+1} | b$, by successive replacement in these formulas from the end. Moreover, each remainder r_1, r_2 , etc. is a combination $u \cdot a + v \cdot b$, and so is in particular the last, i.e., r_{k+1} , whence the proposition. \square

Example 68 Using the Euclidean Algorithm, the greatest common divisor of 17640 and 462 is calculated as follows:

$$\begin{array}{ll}
17640 = 38 \cdot 462 + 84 & a = q_1 \cdot b + r_1 \\
462 = 5 \cdot 84 + 42 & b = q_2 \cdot r_1 + r_2 \\
84 = 2 \cdot 42 & r_1 = q_3 \cdot r_2
\end{array}$$

Therefore, $\gcd(17640, 462) = 42$.

Example 69 Fields and all subrings of fields are integral domains. In particular, all the rings $\mathbb{Z}, \mathbb{Q}, \mathbb{R}, \mathbb{C}$ are integral domains, and so are the polynomial algebras $K[X]$ over a field K , because formula $\deg(f \cdot g) = \deg(f) + \deg(g)$ guarantees that $f \cdot g \neq 0$ for non-zero polynomials f and g .

Remark 21 The converse is also true: Any integral domain is a subring of a field.

Lemma 133 *If the principal ideal ring R is an integral domain, then an ideal I is maximal iff there is a prime p with $I = (p)$. If q is another prime with $I = (q)$, then there is an invertible element e such that $q = e \cdot p$.*

Proof If $I = (p)$ is maximal, then $p = q \cdot r$ implies $(p) \subset (q)$, so either $(q) = R$, and $q \in R^*$, or else $(p) = (q)$, whence $r \in R^*$, therefore p is prime. If I is not maximal, there is q such that $I \subsetneq (q) \subsetneq R$, i.e., $p = q \cdot r$, and $q, r \notin R^*$, therefore p is not prime. \square

Proposition 134 *For a positive prime number $p \in \mathbb{Z}$, the finite ring \mathbb{Z}_p is a field. In particular, the multiplicative group \mathbb{Z}_p^* has $p - 1$ elements, and*

therefore for every integer x , we have $x^p \equiv x \pmod{p}$, and $x^{p-1} \equiv 1 \pmod{p}$ if $p \nmid x$. The latter statement is called Fermat's little theorem.

Proof The finite ring \mathbb{Z}_p is a field by lemma 133. By the Lagrange equation from sorite 117, the order of an element $x \in G$ divides the order of G . Since the order of \mathbb{Z}_p^* is $p - 1$, we have $x^{p-1} = 1$. \square

The next result leads us towards the uniqueness property of prime factorization:

Proposition 135 *Let R be a principal integral domain and $p \in R$ prime. Then if $p \mid a_1 \cdot a_2 \cdot \dots \cdot a_n$, then there is an index i such that $p \mid a_i$.*

Proof In fact, by lemma 133, $R/(p)$ is a field, and therefore, the fact that $p \mid a_1 \cdot a_2 \cdot \dots \cdot a_n$ yields $0 = \bar{a}_1 \cdot \bar{a}_2 \cdot \dots \cdot \bar{a}_n$ in the field $R/(p)$, but this implies that one of these factors vanishes, i.e., p divides one of the factors a_i . \square

Proposition 136 *Let R be a principal integral domain. Then, if an element a has two prime factorizations $a = p_1 \cdot p_2 \cdot \dots \cdot p_k = q_1 \cdot q_2 \cdot \dots \cdot q_l$, then $k = l$, and there is a permutation $\pi \in S_k$ and a sequence e_1, e_2, \dots, e_k of invertible elements such that $q_i = e_i \cdot p_{\pi(i)}$, $i = 1, 2, \dots, k$.*

Proof This is clear from the preceding proposition 135 if we take the factorization $q_1 \cdot q_2 \cdot \dots \cdot q_l$ and a divisor p_j of this product. This means $p_j = e_i \cdot q_i$ for an invertible e_i . Then, dividing everything by p_j reduces to a shorter product, and we proceed by induction. \square

Proposition 137 *In \mathbb{Z} , every non-invertible element $a \neq 0$ has a unique factorization by positive primes $p_1 < p_2 < \dots < p_r$ and positive powers t_1, t_2, \dots, t_r of the form*

$$a = \pm p_1^{t_1} \cdot p_2^{t_2} \cdot \dots \cdot p_r^{t_r}.$$

Proof This follows directly from the above proposition 136, when ordering the prime numbers by size. \square

Corollary 138 *In the polynomial algebra $K[X]$ over a field K , every polynomial of positive degree is a product of irreducible polynomials, and this factorization is unique in the sense of proposition 136. In particular, every linear polynomial is irreducible.*

Proof This follows from the existence of a prime factorization after lemma 129 and the uniqueness statement in proposition 136. \square

Remark 22 It can be shown that in $\mathbb{C}[X]$, the irreducible polynomials are exactly the linear polynomials. This means that every polynomial $f(X)$ of positive degree is a product $f(X) = a(X - b_1)(X - b_2) \dots (X - b_k)$, $a \neq 0$. This is a quite profound theorem, the so-called *fundamental theorem of algebra*.

Exercise 69 Show that $\sqrt{2}$ is not a rational number. Use the prime factorization of numerator a and denominator b in a fictitious representation $\sqrt{2} = \frac{a}{b}$.

Exercise 70 Use exercise 69 to show that the set $\mathbb{Q}(\sqrt{2})$ consisting of the real numbers of form $z = a\sqrt{2} + b$, with $a, b \in \mathbb{Q}$, is a subfield of \mathbb{R} . Show that $\mathbb{Q}(\sqrt{2}) \cong \mathbb{Q}[X]/(X^2 - 2)$.

16.2 Roots of Polynomials and Interpolation

In this section, let K be a field. Let $x \in K$. Then we know from example 61 that for a polynomial $f(X) \in K[X]$, there is an evaluation $f(x) \in K$. We now want to discuss the relation between the polynomial $f(X)$ and the polynomial function $f(?) : K \rightarrow K : x \mapsto f(x)$.

Definition 107 If $f(X) \in K[X]$, then an element $x \in K$ is a root of $f(X)$ if $f(x) = 0$.

Lemma 139 If $x \in K$ is a root of a polynomial $f(X) \in K[X]$, then $(X - x) \mid f(X)$.

Proof We have the division with remainder $f(X) = Q(X) \cdot (X - x) + c$, $c \in K$. The evaluation of f at x yields $c = 0$, whence the claim. \square

Proposition 140 If x_1, x_2, \dots, x_r are r different roots of $f(X) \in K[X]$, then

$$(X - x_1)(X - x_2) \dots (X - x_r) \mid f(X).$$

In particular, a non-zero polynomial $f(X)$ has at most $\deg(f)$ different roots.

Proof The proof uses induction on r . By the above lemma 139, the claim is true for $r = 1$. But since each $X - x_r$ is prime and differs from all $X - x_j$, $j \neq i$, we have a factorization $f(X) = g(X) \cdot (X - x_r)$. So all x_i , $i < r$ are roots of $g(X)$, and by the induction hypothesis, we are done. \square

Corollary 141 *If two polynomials $f, g \in K[X]$ of degrees $\deg(f) < n$ and $\deg(g) < n$ agree on n different argument values $x_i, i = 1, 2, \dots, n$, i.e., $f(x_i) = g(x_i)$, then $f = g$.*

Proof This is an easy exercise using proposition 140. \square

Corollary 142 *If for $K = \mathbb{Q}, \mathbb{R}, \mathbb{C}$, two polynomial functions $f(?)$ and $g(?)$ coincide, then the polynomials $f(X), g(X) \in K[X]$ are equal.*

Proof This follows from the fact that we have infinitely many arguments, where the functions coincide. \square

This allows us to identify polynomials and their associated functions, but this is only a special situation, which does not generalize to arbitrary polynomial algebras.

As an example that the fact stated in corollary 142 is not valid for $K = \mathbb{Z}_2$, consider the polynomials $f(X) = X^2 + 1$ and $g(X) = X + 1$ in $\mathbb{Z}_2[X]$. Evidently, $f(X)$ and $g(X)$ are different regarded as polynomials, but $f(0) = g(0) = 1$, and $f(1) = g(1) = 0$, and thus are equal regarded as polynomial functions.

We have not yet determined if there is always a polynomial $f(X)$ of degree strictly less than n such that its values $f(x_i) = y_i$ can be prescribed for n different arguments x_1, x_2, \dots, x_n . This is indeed guaranteed by various so-called *interpolation formulas*, the best known being those by Lagrange and Newton. Since the result must be unique by corollary 141, we may pick one such formula.

Proposition 143 (Newton Interpolation Formula) *Suppose that we are given a sequence $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ of couples $(x_i, y_i) \in K^2$ for a field K , where $x_i \neq x_j$ for $i \neq j$. Then there is a (necessarily unique) polynomial $f(X) \in K[X]$ of degree $\deg(f) < n$ such that $f(x_i) = y_i, i = 1, 2, \dots, n$. It is given by the Newton interpolation formula*

$$f(X) = a_0 + a_1(X - x_1) + a_2(X - x_1)(X - x_2) + \dots \\ a_{n-1}(X - x_1)(X - x_2) \dots (X - x_{n-1}).$$

Exercise 71 Give a proof of proposition 143. Start with the evaluation at x_1 and calculate the coefficient a_0 . Then proceed successively with the calculation of all coefficients a_1, a_2, \dots, a_{n-1} .

Why are such formulas called “interpolation formulas”? The point is that we are often given a series of “values” y_i for arguments x_i , but we do not know which function $f : K \rightarrow K$ takes these values, $y_i = f(x_i)$. In most cases there is a large number of solutions for this problem. Any solution, such as the polynomial solution given in proposition 143, will also give us values for all other $x \in K$. For example, if $K = \mathbb{R}$, we get all the evaluations $f(x)$ for the intervals $x \in [x_i, x_{i+1}]$. This means that f can also be evaluated on values ‘between’ the given arguments, which is the very meaning of the word ‘interpolation’.

Example 70 Given are the four points $p_1 = (-2, 3)$, $p_2 = (-\frac{1}{2}, -\frac{1}{2})$, $p_3 = (1, \frac{1}{2})$ and $p_4 = (2, -1)$ in \mathbb{R}^2 . The goal is to construct the interpolation polynomial $f(X) \in \mathbb{R}[X]$ through these points. Proposition 143 ensures that $f(X)$ is of the form:

$$f(X) = a_0 + a_1(X - x_1) + a_2(X - x_1)(X - x_2) + a_3(X - x_1)(X - x_2)(X - x_3)$$

Now, setting $X = x_i$ and $f(X) = y_i$ for $i = 1, 2, 3, 4$, the a_j for $j = 0, 1, 2, 3$ are calculated as follows:

For p_1 , every term but the first, a_0 , vanishes, thus

$$a_0 = 3$$

For p_2 :

$$-\frac{1}{2} = a_0 + a_1(-\frac{1}{2} - (-2))$$

thus, after substituting the known value for a_0 , and solving for a_1 :

$$a_1 = -\frac{7}{3}$$

For p_3 :

$$\frac{1}{2} = a_0 + a_1(1 - (-2)) + a_2(1 - (-2))(1 - (-\frac{1}{2}))$$

which yields, using the previously calculated values for a_0 and a_1 :

$$a_2 = 1$$

And finally, for p_4 :

$$-1 = a_0 + a_1(2 - (-2)) + a_2(2 - (-2))(2 - (-\frac{1}{2})) + a_3(2 - (-2))(2 - (-\frac{1}{2}))(2 - 1)$$

produces

$$a_3 = -\frac{7}{15}$$

Putting everything together, and expanding the polynomial:

$$\begin{aligned} f(X) &= 3 - \frac{7}{3}(X+2) + (X+2)\left(X + \frac{1}{2}\right) - \frac{7}{15}(X+2)\left(X + \frac{1}{2}\right)(X-1) \\ &= -\frac{7}{15}X^3 + \frac{3}{10}X^2 + \frac{13}{15}X - \frac{1}{5} \end{aligned}$$

The polynomial $f(X)$ is drawn in figure 16.1.

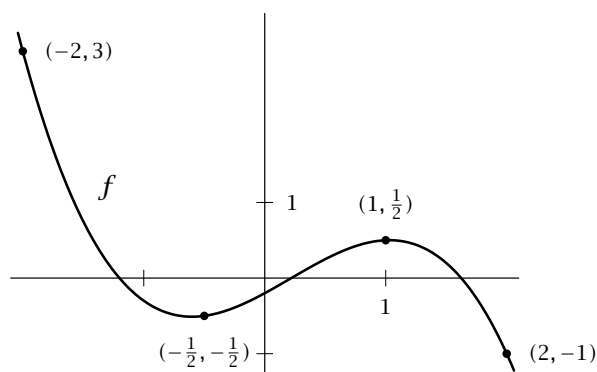


Fig. 16.1. The polynomial $f(X) = -\frac{7}{15}X^3 + \frac{3}{10}X^2 + \frac{13}{15}X - \frac{1}{5}$.

Observe that the interpolation polynomial may not necessarily satisfy specific conditions on the “smoothness” of the curve. Indeed, if n points are specified, the resulting polynomial will have degree $n - 1$, its shape becoming increasingly “bumpy” as the degree rises, with interpolated values straying widely off the mark.

Therefore, for practical purposes, more flexible interpolation techniques, like splines, are used (see [6] and second volume of this book for examples).

Formal Propositional Logic

Until now, we have been using logic in its role as a guiding principle of human thought—as such it has been described under the title of “propositional logic” in the introductory section 1.1. We are not going to replace this fundamental human activity, but we are interested in the problem of mimicking logical reasoning on a purely formal level. What does this mean? The idea is that one would like to incorporate logic in the mathematical theory, which means that we want to simulate the process of logical evaluation of propositions by a mathematical setup. Essentially, this encompasses three tasks:

1. The first task is to give a mathematical construction which describes the way in which logically reasonable propositions should be built. For example, as we have learned from section 1.1, if \mathcal{A}, \mathcal{B} are propositions, one may want to build “NOT \mathcal{A} ”, “ \mathcal{A} IMPLIES \mathcal{B} ”, and so forth. But the construction should not deal with contents, it should just describe the combinatorial way a new expression is constructed. In other words, this is a problem of syntax (relating to how expressions of a sign system are combined independent of their specific meaning). The question addressed by the first task is: *How can such a syntactical building scheme be described?* This task is evidently required if we want to delegate some logical work to machines such as computers, where content cannot be achieved without a very precise and mechanical control of form.
2. Suppose that the first task of formalization has been achieved. One then needs to rebuild the truth values which propositions should ex-

press. This refers to the *meaning* or *semantics* of such propositions. This is a radical point of view: The meaning of a proposition is not its specific contents, but one of the two truth values: “true” or “false”, or, equivalently, “is the case” or “is not the case”. For example, the meaning of the proposition “if it is raining, then I am tired” has nothing to do with rain or fatigue, it is only the truth value of the implication. It is this semantical issue which is mimicked by the formal theory of propositional logic. So one needs a domain of truth values, which is addressed by a formal propositional expression in order to evaluate its ‘meaning’. Such domains will be called *logical algebras*. *Therefore, the second task is to give a rigorous mathematical description of what a logical algebra is which manages the semantic issue of propositional logic.*

3. The third task deals with the connection of the first two tasks, i.e., given a syntactical description of how to build correct propositions, as well as a logical algebra which manages the semantics of truth values, we need to specify how these two levels are related to each other. *This third task is that of “signification”, i.e., the mathematically rigorous definition of an association of meaning with a given propositional expression.*

We should, however, pay attention to the power of such a formalization procedure, since often a naive understanding thereof claims to be a replacement of our human reasoning and its contents by purely formal devices. There is a fundamental error, which is easily explained: A formal, i.e., mathematical theory of propositional logic as sketched above, must rely on given mathematical structures and results, the most important one being set theory, the theory of natural numbers, and—above all—the recursion theorem, which is a fundamental tool for the construction of formal expressions and for the proof of properties which are shared by formal expressions. In order to keep the level of human thought and the formal logical level separate, we never use symbols of formal logic in formulas which are meant as a human thought. In a famous 1931 paper *On Formally Undecidable Propositions of Principia Mathematica and Related Systems*, the mathematician Kurt Gödel demonstrated that within any given branch of mathematics, there would always be some propositions that could not be proved either true or false using the rules and axioms of that mathematical branch itself. Gödel himself alluded to machine-supported logical reasoning. But subsequently, many philosophers, the-

ologists and would-be scientists concluded that human thought, as well, is tainted by these incompleteness properties, actually derived for formal systems only. We hope that the reader may understand in the following that these formalizations of human thought are just low-level simulations of its mechanical flattening, and not of proper thought. An example will be presented arguing that there are propositions which are undecidable in a given formal system, but can be decided using unformalized thought.

17.1 Syntactics: The Language of Formal Propositional Logic

The language of formal propositional logic aims to create the variety of possible propositions. The concept of proposition is elementary. Therefore, the only possibility to build new propositions is to start with given ones and to combine them by logical operations. Here is the formal setup of the repertory of expressions with which we shall work in order to build logically reasonable propositions:

Definition 108 A propositional alphabet is a triple $A = (V, B, C)$ of mutually disjoint sets, where

- (i) the set V is supposed to be denumerable, i.e., equipollent to \mathbb{N} , and the elements $v_0, v_1, \dots \in V$ are called propositional variables;
- (ii) the set $B = \{ (,) \}$ of $($, the left bracket and $)$, the right bracket;
- (iii) the set $C = \{ !, \&, |, \rightarrow \}$ consists of these four logical connective symbols, the negation symbol $!$, the conjunction symbol $\&$, the disjunction symbol $|$, and the implication symbol \rightarrow .

The monoid $EX = EX(A) = \text{Word}(V \sqcup B \sqcup C)$ is called the monoid of expressions over A . Within EX , the subset $S(EX)$ of sentences over A is the smallest subset $S(EX) \subset EX$ with the following properties:

- (i) $V \subset S(EX)$;
- (ii) if $\alpha \in S(EX)$, then $(!\alpha) \in S(EX)$;
- (iii) if $\alpha, \beta \in S(EX)$, then $(\alpha \& \beta) \in S(EX)$;
- (iv) if $\alpha, \beta \in S(EX)$, then $(\alpha | \beta) \in S(EX)$;
- (v) if $\alpha, \beta \in S(EX)$, then $(\alpha \rightarrow \beta) \in S(EX)$.

The set $S(EX)$ is also called the propositional language defined by A .

Observe that the symbols of A have been chosen for the sake of the logical context, but strictly speaking, they are simply sets with no special properties whatsoever.

As it is defined, the language $S(EX(A))$ is not effectively described—we can only tell that, for example, expressions such as v_0 , $(!v_3)$, and $(v_2 \& v_6)$ are sentences. But there is no general control of what is possible and what is not.

Exercise 72 Show that the expression $!(!v_1)$ is not a sentence.

Is $(v_3 \rightarrow (!v_2))$ a sentence?

We now give a recursive construction of the subset $S(EX) \subset EX$ for a given alphabet A . To this end, we use the fact that we have the union of disjoint sets $S(EX) = \bigcup_{n \geq 0} S_n(EX)$, where $S_n(EX) = S(EX) \cap EX_n$, with $EX_n = \{w \in EX \mid l(w) = n\}$ the set of expressions of word length n . Here is the structure of $S(EX)$ given in these terms:

Definition 109 Given a propositional alphabet A , let S_n be the following subsets of EX_n , which we define by recursion on n :

- (i) For $n = 0, 2, 3$, we set $S_n = \emptyset$;
- (ii) we set $S_1 = V$;
- (iii) we set $S_4 = \{(!v_i) \mid v_i \in V\}$;
- (iv) we set S_5 to the set of words of one of these three types: $(v_i \& v_j)$, $(v_i \mid v_j)$, $(v_i \rightarrow v_j)$, where $v_i, v_j \in V$ are any two propositional variables;
- (v) for $n > 5$, we set $S_n = S_n^! \cup S_n^{\&} \cup S_n^{\mid} \cup S_n^{\rightarrow}$, where $S_n^* = \{(\alpha * \beta) \mid \alpha, \beta \in \bigcup_{i < n} S_i, l(\alpha) + l(\beta) = n - 3\}$ for the symbols $*$ $\in \{\&, \mid, \rightarrow\}$, and where $S_n^! = \{(!\alpha) \mid \alpha \in S_{n-3}\}$.

We then set $S = \bigcup_{n \geq 0} S_n$.

Example 71 To illustrate the construction of the sets S_i , we give them for the set of variables $V = \{v\}$ up to S_8 . The sets grow quickly with increasing i even with only one variable. The reader should check that the sets conform to the rules of definition 109, in particular that each S_i only contains expressions of exact length i .

$$\begin{aligned}
S_0 &= \{\}, \\
S_1 &= \{v\}, \\
S_2 &= \{\}, \\
S_3 &= \{\}, \\
S_4 &= \{(!v)\} \\
S_5 &= \{(v \& v), (v | v), (v \rightarrow v)\} \\
S_6 &= \{\} \\
S_7 &= \{(!(!v))\} \\
S_8 &= \{(v \& (!v)), ((!v) \& v), \\
&\quad (v | (!v)), ((!v) | v), \\
&\quad (v \rightarrow (!v)), ((!v) \rightarrow v), \\
&\quad (! (v \& v)), (! (v | v)), (! (v \rightarrow v))\}
\end{aligned}$$

Proposition 144 *Given a propositional alphabet A , we have*

$$S(EX) = S,$$

where S is the set defined in definition 109.

Proof Clearly, S is contained in $S(EX)$. Let us check that it fulfills the five axioms (i) through (v). By construction, $V = S_1$, whence (i). If $\alpha \in S$, then for $l(\alpha) \leq 3$, we only have $\alpha \in V$, and S_4 covers the case $(! \alpha) \in S$, else $l(\alpha) > 3$, and this case is covered by construction v of the definition of S , whence (ii). The cases (iii)–(v) are similar: for $l(\alpha) + l(\beta) \leq 3$, we must have $\alpha, \beta \in V$, this is covered by (iv) in the definition of S . For $l(\alpha) + l(\beta) > 3$, construction (v) in the definition does the job. \square

We now know how the construction of the sentences over A works, but we still do not know how many sentences α and β could give rise to one and the same sentence, say $w = (\alpha \& \beta)$.

Lemma 145 *If $left(w)$ and $right(w)$ denote the numbers of left and right brackets, respectively, in a sentence $w \in S(EX)$, then we have $left(w) = right(w)$.*

Proof The construction of $S(EX)$ by S guaranteed by proposition 144 yields a straightforward inductive proof by the length of a sentence. \square

Exercise 73 Give a proof of lemma 145 by recursion on the length of w .

Lemma 146 *Let $w \in S(EX)$, and suppose that we have a left bracket $($ in w , i.e., $w = u(x$, with $l(u) > 0$. Then if the number of left brackets in $(x$ is l and the number of right brackets in $(x$ is r , we have $l < r$.*

Proof Induction on $l(w)$: For $l(w) \leq 5$, it is clear by the explicit words from rules (i)–(iv) in the definition of S . For general w , if $w = u(x = (!\alpha)$, then $u = (!$ or $u = (!u'$, where $u'(v' = \alpha$. By induction, the number of left brackets in $(v'$ is smaller than the number of right brackets. So the same is true for $(x = (v')$. A similar argument is used for the other connectives. \square

Proposition 147 *Let $w \in S(EX)$. Then exactly one of the following decompositions is the case: $w \in V$, or $w = (!\alpha)$, or $w = (\alpha \& \beta)$, or $w = (\alpha | \beta)$, or $w = (\alpha \rightarrow \beta)$, where α and β are uniquely determined sentences. They are called the components of w .*

Proof First, suppose $(!\alpha) = (!\alpha')$, then clearly the inner words α and α' must be equal. Second, suppose $(!\alpha) = (\beta * \gamma)$, then the letter $!$ must be the first letter of β , which is impossible for any sentence. Then suppose $(\alpha * \beta) = (\gamma * \delta)$. If $l(\alpha) = l(\gamma)$, then $\alpha = \gamma$, therefore also $\beta = \delta$. So suppose wlog (without loss of generality) $l(\alpha) < l(\gamma)$. Then $\beta = (\gamma' * \delta)$, where $\gamma = x(\gamma'$. So by lemma 146, $(\gamma'$ has fewer left than right brackets. But this contradicts the fact that β and δ have the same number of left and right brackets. \square

This proposition has deep implications. In fact, it allows the definition of functions on $S(EX)$ by recursion on the length of sentences and in function of the unique logical connectives defining compound sentences. But let us first discuss the announced “logical algebras”.

17.2 Semantics: Logical Algebras

On the syntactic level of a formal propositional language $S(EX)$ over a propositional alphabet A , the sentences look like meaningful expressions. However, they really only *look* meaningful. In order to load such expressions with logical meaning, we need to provide the system with logical values. In order to have a first orientation of what a logical algebra should be, we refer to the “Boolean algebra” of subsets $L = 2^a$ of a set a as discussed in chapter 3, and especially in proposition 7. More specifically, in the special case of $a = 1 = \{0\}$, we have two subsets, $\perp = \emptyset = 0$ and $\top = 1$. Following a classical idea of the great mathematician and philosopher Gottfried Wilhelm Leibniz (1646–1716) and its elaboration by the mathematician George Boole (1815–1864), one can mimic truth values on

the Boolean algebra $L = 2 = 2^1$ as described in proposition 7. The value “true” is represented by \top , whereas the value “false” is represented by \perp . The truth table of the conjunction \mathcal{A} AND \mathcal{B} is given by the Boolean operation of intersection of the truth values assigned to the components, i.e., if \mathcal{A} is true and \mathcal{B} is false, the truth value of the conjunction is $\top \cap \perp = \perp$, and so on with all other combinations. In other words, we are combining the values $value(\mathcal{A}), value(\mathcal{B}) \in 2$ under Boolean operations. We see immediately that the Boolean operation “ \cap ” stands for conjunction, “ \cup ” for disjunction, and complementation “ $-$ ” for negation, whereas the truth value of \mathcal{A} IMPLIES \mathcal{B} is the value of (NOT \mathcal{A}) OR \mathcal{B} , i.e., $(-value(\mathcal{A})) \cup value(\mathcal{B})$. However, it is not always reasonable to deduce logical implication from negation and disjunction, we rather would like to leave this operation as an autonomous operation. This is very important in order to cope with non-classical logical approaches, such as fuzzy logic. This approach was formalized by the mathematician Arend Heyting (1898-1980):

Definition 110 A Heyting algebra (HA) is a partially ordered set (L, \leq) , together with

- three binary operations: the join $a \vee b$, the meet $a \wedge b$, and the implication $a \Rightarrow b$ for $a, b \in L$, and
- two distinguished elements \perp and \top , called “False” and “True”, respectively.

These data are subjected to the following properties:

- (i) \perp and \top are the minimal and maximal element with respect to the given partial ordering, i.e., $\perp \leq x \leq \top$ for all $x \in L$.
- (ii) The operations join and meet are commutative.
- (iii) Join and meet are mutually distributive, i.e., $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$ and $x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z)$, for all $x, y, z \in L$.
- (iv) The join $x \vee y$ is a least upper bound (l.u.b) of x and y , i.e., $x, y \leq x \vee y$ and for any z with $x, y \leq z$, we have $x \vee y \leq z$.
- (v) The meet $x \wedge y$ is a greatest lower bound (g.l.b.) of x and y , i.e., $x \wedge y \leq x, y$ and for any z with $z \leq x, y$, we have $z \leq x \wedge y$.
- (vi) (Adjunction) For any $z \in L$, $z \leq (x \Rightarrow y)$ iff $z \wedge x \leq y$.

For an element x of a Heyting algebra L , we define the negation of x by $\neg x = (x \Rightarrow \perp)$.

Exercise 74 Show that in a Heyting algebra, we always have $y \leq \neg x$ iff $y \wedge x = \perp$. Deduce from this that always

$$\neg x \wedge x = \perp.$$

Use the adjunction characterization (vi) of definition 110 to prove that

$$((x \vee y) \Rightarrow z) = ((x \Rightarrow z) \wedge (y \Rightarrow z)).$$

More specifically, prove *De Morgan's first law*:

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

Definition 111 In a Heyting algebra L , a complement of an element x is an element a such that $x \wedge a = \perp$ and $x \vee a = \top$.

Lemma 148 The complement of an element x in a Heyting algebra L , if it exists, is uniquely determined.

Proof If b is another complement of x , we have $b = b \wedge \top = b \wedge (x \vee a) = (b \wedge x) \vee (b \wedge a) = (x \wedge a) \vee (b \wedge a) = (x \vee b) \wedge a = \top \wedge a = a$. \square

Exercise 75 Show that $y \leq \neg x \Leftrightarrow y \wedge x = \perp$.

An important and classical type of special Heyting algebras is defined by these properties:

Lemma 149 For a Heyting algebra L , the following properties are equivalent: For all $x \in L$,

- (i) $x \vee \neg x = \top$,
- (ii) we have $\neg\neg x = x$.

Proof Observe that always $\neg\top = \perp$ and $\neg\perp = \top$.

(i) implies (ii): If $x \vee \neg x = \top$, then by De Morgan's first law (exercise 74), $\neg x \wedge \neg\neg x = \perp$. So $\neg\neg x$ is a complement of $\neg x$, but x is also a complement of $\neg x$, therefore, by uniqueness of complements (lemma 148), $\neg\neg x = x$.

Conversely, if $\neg\neg x = x$, then by De Morgan's first law, $x \vee \neg x = \neg(\neg x \wedge \neg\neg x) = \neg(\neg x \wedge x) = \neg\perp = \top$. \square

Definition 112 A Heyting algebra which has the equivalent properties of lemma 149 is called a Boolean algebra (BA).

Example 72 The classical example is the “Boolean algebra” of subsets $L = 2^a$ of a given set a , as discussed in chapter 3. Here, the partial ordering is the set inclusion, i.e., $x \leq y$ iff $x \subset y$. We define $\neg x = a - x$, and have $\perp = \emptyset, \top = a, \vee = \cup, \wedge = \cap$, and $x \Rightarrow y = \neg x \cup y$. Evidently, $\neg \neg x = (\neg x) \cup \emptyset = \neg x$, whence $x \Rightarrow y = (\neg x) \vee y$, i.e., implication is deduced from the other operations. It is really a BA since the double complementation is the identity: $\neg(\neg(x)) = x$. This is the a posteriori justification for the name “Boolean algebra”, which we attributed to the powerset algebra in chapter 3. But observe that for a general set a , many truth values are possible besides the classical \perp and \top .

Example 73 A less classic example is the set $L = \text{Fuzzy}(0, 1)$ of all intervals $I_x = [0, x[\subset [0, 1[$ of the “half open” real unit interval $[0, 1[= \{x \mid 0 \leq x < 1\} \subset \mathbb{R}$. Its elements are so-called *fuzzy* truth values, meaning that something may be true to $x \cdot 100\%$, i.e., not entirely true, and not entirely false. The percentage is given by the upper number x of the interval I_x . The partial ordering is again that of subset inclusion, and the extremal values are $\perp = I_0 = \emptyset$ and $\top = I_1$. We also take $\vee = \cup$ and $\wedge = \cap$. This means $I_x \vee I_y = I_{\max(x,y)}$ and $I_x \wedge I_y = I_{\min(x,y)}$.

The implication is a little more tricky. We must have $I_z \leq (I_x \Rightarrow I_y)$ iff $I_z \cap I_x \subset I_y$. The solution therefore must be $I_x \Rightarrow I_y = \bigcup_{z, I_z \cap I_x \subset I_y} I_z$. Therefore $I_x \Rightarrow I_y = I_w$, with $w = \sup\{z \mid I_z \cap I_x \subset I_y\}$. This gives the following implications: $I_x \Rightarrow I_y = \top$ if $x \leq y$, and $I_x \Rightarrow I_y = I_y$ if $x > y$. In particular, $\neg I_x = \perp$ if $x \neq 0$, $\neg \perp = \top$. And finally, $\neg \neg I_x = \top$ if $x \neq 0$, $\neg \neg \perp = \perp$. This also shows that this Heyting algebra $\text{Fuzzy}(0, 1)$ is not Boolean. For the general theory of fuzzy systems, see [34] or [48].

We shall henceforth use the term *logical algebra* for a Heyting algebra or, more specifically, for a Boolean algebra. Intuitively, a logical algebra is a structure which provides us with those operations “negation”, “conjunction”, “disjunction”, and “implication” which are defined by the formal logical setup and should simulate the “reality of logical contents”.

Here are the general properties of logical algebras:

Sorite 150 *Let L be a Heyting algebra and $x, y \in L$. Then*

- (i) $x \leq \neg \neg x$,
- (ii) $x \leq y$ implies $\neg y \leq \neg x$,
- (iii) $\neg x = \neg \neg \neg x$,

- (iv) (De Morgan's first law) $\neg(x \vee y) = \neg x \wedge \neg y$.
- (v) $\neg\neg(x \wedge y) = \neg\neg x \wedge \neg\neg y$,
- (vi) $x \wedge y = \perp$ iff $y \leq \neg x$,
- (vii) $x \wedge \neg x = \perp$,
- (viii) $(x \Rightarrow x) = \top$,
- (ix) $x \wedge (x \Rightarrow y) = x \wedge y$,
- (x) $y \wedge (x \Rightarrow y) = y$,
- (xi) $x \Rightarrow (y \wedge z) = (x \Rightarrow y) \wedge (x \Rightarrow z)$.

In particular, if L is a Boolean algebra, then

- (xii) $x = \neg\neg x$,
- (xiii) $x \vee \neg x = \top$,
- (xiv) $x \Rightarrow y = \neg x \vee y$,
- (xv) $x \leq y$ iff $\neg y \leq \neg x$,
- (xvi) (De Morgan's second law) $\neg(x \wedge y) = \neg x \vee \neg y$.

Proof By exercise 74, $x \leq \neg\neg x$ iff $x \wedge \neg x = \perp$, but the latter is always true. If $x \leq y$, then also $x \wedge \neg y \leq y \wedge \neg y = \perp$, so $\neg y \wedge x = \perp$, therefore $\neg y \leq \neg x$.

Statement (iii) follows immediately from (i) and (ii).

(iv) is De Morgan's first law (see exercise 74).

The proof of (v) is quite technical and is omitted, see [36].

Statements (vi) and (vii) follow from exercise 74.

Statements (viii) to (xi) immediately follow from the characterization of the g.l.b. and adjointness for implication.

(xii) and (xiii) is the characterization from lemma 149.

For (xiv), we show $z \leq (\neg x \vee y)$ iff $z \wedge x \leq y$. If $z \wedge x \leq y$, then $z = z \wedge 1 = z \wedge (\neg x \vee x) = (z \wedge \neg x) \vee (z \wedge x) \leq \neg x \vee y$. Conversely, if $z \leq (\neg x \vee y)$, then $z \wedge y \leq (\neg x \vee y) \wedge y \leq (\neg x \wedge y) \vee (y \wedge y) \leq (\neg x \wedge y) \vee y$.

(xv) follows from (ii) and (xii). □

17.3 Signification: Valuations

We may now turn to the third component of a semiotic (sign-theoretic) system: the signification process from the expressive surface down to the semantic depth. Our modeling of such a system by use of a propositional alphabet A and a logical algebra L must provide us with a function

$value : S(EX) \rightarrow L$ such that each sentence $w \in S(EX)$ is assigned a truth value $value(w) \in L$. Such a valuation should however keep track with the requirement that logical evaluation is related to the logical composition of these sentences. Here is the precise setup:

Definition 113 *Given a propositional alphabet A and a logical algebra L , a valuation is a map $value : S(EX) \rightarrow L$ such that for all sentences $\alpha, \beta \in S(EX)$, we have*

- (i) $value(!\alpha) = \neg(value(\alpha))$,
- (ii) $value((\alpha \mid \beta)) = value(\alpha) \vee value(\beta)$,
- (iii) $value((\alpha \& \beta)) = value(\alpha) \wedge value(\beta)$,
- (iv) $value((\alpha \rightarrow \beta)) = value(\alpha) \Rightarrow value(\beta)$.

The set of valuations $value : S(EX) \rightarrow L$ over the propositional alphabet A with values in the logical algebra L is denoted by $\mathcal{V}(A, L)$.

And here is the existence theorem for valuations:

Proposition 151 *Given a propositional alphabet A , a set of variables V and a logical algebra L , the functional restriction map*

$$var : \mathcal{V}(A, L) \rightarrow Set(V, L) : value \mapsto value|_V$$

is a bijection, i.e., for each set map $v : V \rightarrow L$ defined on the propositional variables, there exists exactly one valuation $value : S(EX) \rightarrow L$ such that $value|_V = v$. In other words, once we know the values of the variables, we know the values of all expressions.

Proof Injectivity is proved by induction on the length of a sentence w . For length 1, we have values from V , and this is what we want. For $l(w) > 1$, we have one of the forms $w = (!\alpha), (\alpha \mid \beta), (\alpha \& \beta), (\alpha \rightarrow \beta)$, and the induction hypothesis on α, β , together with the axiomatic properties (i)-(iv) of valuations solve the problem. Conversely, if a map $v_V : V \rightarrow L$ is given, its extension to any sentence $w \in S(EX)$ may be defined by recursion following the axiomatic properties (i)-(iv) of valuations. This is indeed well defined, because by proposition 147, the form and the components of a compound sentence are uniquely determined by w . \square

Exercise 76 Given a propositional alphabet A , let $a = 3 = \{0, 1, 2\}$ and consider the powerset Boolean algebra $L = 2^3$. Let a value map $value \in \mathcal{V}(A, L)$ be defined by $value(v_0) = \perp$, $value(v_1) = \{0, 2\}$, $value(v_2) = \{1\}, \dots$ Calculate the value

$$\text{value}(((!(v_2 \& !(v_0))) \mid ((!v_1) \rightarrow v_2))).$$

Notation 11 For long expressions, such as the still rather short example in the previous exercise 76, the number of brackets becomes cumbersome to keep under control. To avoid this effect, one uses the same trick as for ordinary arithmetic: strength of binding, i.e., the formal rules are replaced by some implicit bracket constructions which enable us to omit brackets in the notation. The rules of binding strength are these: $!$ binds stronger than $\&$ and \mid , and these bind stronger than \rightarrow . Under this system of rules, the above example

$$(((!(v_2 \& !(v_0))) \mid ((!v_1) \rightarrow v_2))$$

would be shortened to

$$!(v_2 \& !v_0) \mid (!v_1 \rightarrow v_2)$$

The main problem of formal propositional logic is to understand which kind of sentences $w \in S(EX)$ have the value $\text{value}(w) = \top$ under certain valuations. The most prominent of these questions is the problem of so-called tautologies, i.e., sentences, which have the value \top under all possible valuations of a given class. Here is the formalism:

Definition 114 Given a propositional alphabet A , a logical algebra L , a sentence $s \in S(EX)$, and a valuation $v \in V(A, L)$, one says that s is v -valid if $v(s) = \top$, in signs: $v \models s$. If $v \models s$ for all $v \in V(A, L)$, one says that s is L -valid, in signs $L \models s$. In particular, if $2 \models s$, one says that s is classically valid or that s is a tautology and writes $\models s$. If s is valid for all Boolean algebras, one says that s is Boolean valid and writes $BA \models s$. If s is valid for all Heyting algebras, one says that s is Heyting valid and writes $HA \models s$.

Exercise 77 Show that for any sentence $s \in S(EX)$, we have $HA \models s \rightarrow !!s$ and $BA \models !!s \rightarrow s$. Give an example of a HA L such that $L \models !!s \rightarrow s$ is false.

In order to control what kinds of sentences are valid for given valuation classes, one needs a more constructive approach, such as the axiomatic method.

17.4 Axiomatics

Axiomatics is about the construction of a set of new sentences from given ones by use of a predetermined system of inference rules. This setup is a particular case of so-called production grammars, which will be discussed in chapter 19. Here, we stay rather intuitive as to the general nature of such an inference rule system and will only concentrate on a special type, the one defined by the classical inference rule *modus ponens* (the Latin name is a marker for the Medieval tradition of formal logic). It is the rule which we apply constantly: If \mathcal{A} is the case (i.e., true), and if the implication \mathcal{A} IMPLIES \mathcal{B} is the case, then also \mathcal{B} is the case. In fact, otherwise, by absolute logic, \mathcal{B} would not be the case (false), but then \mathcal{A} IMPLIES \mathcal{B} cannot be the case by the very definition of implication. The formal restatement of this inference rule defines classical axiomatics as follows.

Definition 115 *One is given a set $AX \subset S(EX)$ of sentences, called axioms. A proof sequence (with respect to AX) is a finite sequence $p = (s_i)_{i=1,2,\dots,n} \in S(EX)^n$ of positive length $l(p) = n$ such that $s_1 \in AX$, and for $i > 1$, either $s_i \in AX$, or there are two sentences s_k and $s_l = (s_k \rightarrow s_i)$ where $k, l < i$. A terminal sentence s_n in a proof chain is called a theorem with respect to AX . The set of theorems is denoted by $S_{AX}(EX)$. If $S_{AX}(EX)$ is clear, the fact that $s \in S_{AX}(EX)$ is also denoted by $\frac{}{AX} s$.*

Intuitively, the role of axioms is this: One accepts axioms as being true a priori, i.e., one only looks for classes of valuations which map all axioms to \top . One then wants to look for all sentences which are also true if the axioms are so. The only rule is the formalized modus ponens: Given two theorems s and $(s \rightarrow t)$, then t is also a theorem. Evidently, this process can be managed by a machine. The point of axiomatics is that proof sequences starting from a particular set of axioms yield the tautologies.

Definition 116 *The axioms of classical logic (CL) are the sentences which can be built from any given three sentences $\alpha, \beta, \gamma \in S(EX)$ by one of the following constructions:*

- (i) $(\alpha \rightarrow (\alpha \& \alpha))$
- (ii) $((\alpha \& \beta) \rightarrow (\beta \& \alpha))$
- (iii) $((\alpha \rightarrow \beta) \rightarrow ((\alpha \& \gamma) \rightarrow (\beta \& \gamma)))$

- (iv) $((\alpha \rightarrow \beta) \& (\beta \rightarrow \gamma)) \rightarrow (\alpha \rightarrow \gamma)$
- (v) $(\beta \rightarrow (\alpha \rightarrow \beta))$
- (vi) $((\alpha \& (\alpha \rightarrow \beta)) \rightarrow \beta)$
- (vii) $(\alpha \rightarrow (\alpha \mid \beta))$
- (viii) $((\alpha \mid \beta) \rightarrow (\beta \mid \alpha))$
- (ix) $((\alpha \rightarrow \beta) \& (\beta \rightarrow \gamma)) \rightarrow ((\alpha \mid \beta) \rightarrow \gamma)$
- (x) $((\neg\alpha) \rightarrow (\alpha \rightarrow \beta))$
- (xi) $((\alpha \rightarrow \beta) \& (\alpha \rightarrow (\neg\beta))) \rightarrow (\neg\alpha)$
- (xii) $(\alpha \mid (\neg\alpha))$

The axioms of intuitionistic logic (IL) are those sentences in CL built from all constructions except for the last, $(\alpha \mid (\neg\alpha))$.

The intuitionistic axiom system IL contains those axioms which we need to produce sentences which are Heyting valid. Recall that we have in fact Heyting algebras L , for example $L = \text{Fuzzy}(0, 1)$, where $x \vee (\neg x) \neq \top$ in general. The crucial proposition is this:

Proposition 152 Given a propositional alphabet A and a sentence $s \in S(EX)$, the following statements are equivalent:

For classical logic:

- (i) The sentence s is a tautology, i.e., $\models s$.
- (ii) The sentence s is Boolean valid, i.e., $BA \models s$.
- (iii) s is a theorem with respect to CL, i.e., $\frac{}{CL} s$

And for intuitionistic logic:

- (i) The sentence s is Heyting valid, i.e., $HA \models s$.
- (ii) s is a theorem with respect to IL, i.e., $\frac{}{IL} s$

Proof (ii) implies (i): This is similar to proposition 153 which deals with (iii) implies (i), i.e., the special case where BA is the set 2 .

(i) implies (iii): This part is proposition 154.

The equivalence of (i) and (ii) in the intuitionistic case follows these lines: Soundness, i.e., (i) implies (ii) follows as easily as soundness for Boolean algebras. Completeness is proved as follows: From $\frac{}{IL} s$, one constructs a special Heyting algebra, the so-called Lindenbaum algebra H_{IL} . Then one shows that the validity for this algebra implies $\frac{}{IL} s$. So finally, as $HA \models s$ implies $H_{IL} \models s$, we are

done. The details are beyond the scope of this introductory book, but see [38] for details. \square

The equivalence of (i) and (iii) of the Boolean part of this theorem is of particular significance. Historically, each direction has been introduced as a distinct theorem. We now state both theorems separately.

The first theorem, corresponding to the direction from (iii) to one deals with the requirement that only tautologies can be proved from the axiom system CL of classical logic, only tautologies are generated. This is the so-called *soundness theorem*:

Proposition 153 (Soundness) *If $\vdash_{CL} s$, then $\models s$.*

This is just an exercise:

Exercise 78 Prove proposition 153 as follows: First show that all axioms are classically valid. Then show by induction on the proof chain length that any theorem is classically valid.

The more involved part is the converse, the so-called *completeness theorem*:

Proposition 154 (Completeness) *If $\models s$, then $\vdash_{CL} s$.*

We shall not give a proof of the completeness theorem, which is quite involved. The original proof of such a theorem was given in 1921 by Emil Post (see [45]).

Example 74 The sentences $(\alpha \rightarrow (\beta \mid \alpha))$ for $\alpha, \beta \in S(EX)$ seem to be obvious theorems of CL, especially since $(\alpha \rightarrow (\alpha \mid \beta))$ is an axiom schema. But we have to provide a proper proof sequence in order to establish this fact. On the right of each line of the proof sequence we indicate whether we have used an axiom (ax.) or applied modus ponens (m.p.) to two of the previous lines. Circled numbers ① and ② are used as abbreviations to refer to the formulas in line 1 and line 2, respectively. (To be absolutely accurate, we should state that the following proof sequence is really a schema for generating proof sequences of $(\alpha \rightarrow (\beta \mid \alpha))$ for all $\alpha, \beta \in S(EX)$).

1. $(\alpha \rightarrow (\alpha \mid \beta))$	ax. (vii)
2. $((\alpha \mid \beta) \rightarrow (\beta \mid \alpha))$	ax. (viii)
3. $(\textcircled{1} \rightarrow (\textcircled{2} \rightarrow \textcircled{1}))$	ax. (v)
4. $(\textcircled{2} \rightarrow \textcircled{1})$	m.p. 1, 3
5. $((\textcircled{2} \rightarrow \textcircled{1}) \rightarrow ((\textcircled{2} \& \textcircled{2}) \rightarrow (\textcircled{1} \& \textcircled{2})))$	ax. (iii)
6. $((\textcircled{2} \& \textcircled{2}) \rightarrow (\textcircled{1} \& \textcircled{2}))$	m.p. 4, 5
7. $(\textcircled{2} \rightarrow (\textcircled{2} \& \textcircled{2}))$	ax. (i)
8. $(\textcircled{2} \& \textcircled{2})$	m.p. 2, 7
9. $(\textcircled{1} \& \textcircled{2})$	m.p. 6, 8
10. $((\alpha \rightarrow (\alpha \mid \beta)) \& ((\alpha \mid \beta) \rightarrow (\beta \mid \alpha))) \rightarrow (\alpha \rightarrow (\beta \mid \alpha))$	ax. (iv)
11. $(\alpha \rightarrow (\beta \mid \alpha))$	m.p. 9, 10

Hence $\frac{}{CL} (\alpha \rightarrow (\beta \mid \alpha))$ for all $\alpha, \beta \in S(EX)$.

This proof was rather easy and short. However, proof sequences for theorems of even a little more complexity tend to become long and intricate. Therefore, whenever one has established the theoremhood of a sentence, one may use it in subsequent proof sequences just as if it were an axiom. If asked, one could then always recursively expand the theorems to their proof sequences to get a sequence in the originally required form.

Sorite 155 *Abbreviating $((\alpha \rightarrow \beta) \& (\beta \rightarrow \alpha))$ by $(\alpha \leftrightarrow \beta)$, the following sentences are tautologies:*

1. (Associativity)

$$((\alpha \mid \beta) \mid \gamma \leftrightarrow \alpha \mid (\beta \mid \gamma)) \text{ and } ((\alpha \& \beta) \& \gamma \leftrightarrow \alpha \& (\beta \& \gamma))$$

2. (Commutativity) $(\alpha \mid \beta \leftrightarrow \beta \mid \alpha)$ and $(\alpha \& \beta \leftrightarrow \beta \& \alpha)$

3. (De Morgan's Laws)

$$(!(\alpha \mid \beta)) \leftrightarrow (!\alpha \& !\beta) \text{ and } (!(\alpha \& \beta)) \leftrightarrow (!\alpha \mid !\beta)$$

Proof The sentences can be proved along the lines of example 74. However the proofs quickly become very unwieldy, and to handle them at all, a number of tools have to be developed, such as the Deduction Theorem. For more details about proof theory, see any book on mathematical logic, such as [17]. \square

One also calls sentences s and t *equivalent* iff $(s \leftrightarrow t)$ is a tautology. Notice that from the associativity, we may group conjunctions or disjunctions in any admissible way and obtain sentences which are equivalent to each other. We therefore also omit brackets in multiple conjunctions or disjunctions, respectively.

Exercise 79 A sentence s is in *disjunctive normal form* iff $s = s_1 | s_2 | \dots | s_k$, where each s_i is of the form $s_i = s_{i1} \& s_{i2} \& \dots \& s_{ik(i)}$ with s_{ij} being a propositional variable $v \in V$ or its negation $(!v)$.

A sentence s is in *conjunctive normal form* iff $s = s_1 \& s_2 \& \dots \& s_k$, where each s_i is of the form $s_i = s_{i1} | s_{i2} | \dots | s_{ik(i)}$ with s_{ij} being a propositional variable $v \in V$ or its negation $(!v)$.

Use the axioms and the sorite 155 to show that every sentence is equivalent to a sentence in disjunctive normal form and also to a sentence in conjunctive normal form.

Exercise 80 Define the *Sheffer stroke operator* by $(\alpha || \beta) = !(\alpha \& \beta)$. Show that every sentence is equivalent to a sentence, where only the Sheffer operator occurs. In electrical engineering the stroke operator is also known as NAND.

Formal Predicate Logic

Up to now, we have succeeded in formalizing basic logic as controlled by truth values produced by the propositional connectives \neg , $\&$, \vee , and \rightarrow of negation, conjunction, disjunction, and implication. However, nothing has been done to mimic the ‘anatomy of propositions’, in fact, we had just offered an ‘amorphous’ set of propositional variables v_0, v_1, \dots with no further differentiation. So the truth value of sentences was based on the completely arbitrary valuation of propositional variables.

Now, mathematics needs more refined descriptions of how truth and falsity are generated. For example, the simple set-theoretic definition $a \cap b = \{x \mid x \in a \text{ and } x \in b\}$ uses composed **predicates**: (1) $P(x)$ is true iff $x \in a$, (2) $Q(x)$ is true iff $x \in b$, and the combination thereof $(P \& Q)(x)$ is true iff $P(x)$ and $Q(x)$ are both true. So, first of all we need to *formalize the concept of a predicate*.

Next, let us look at the pair axiom: “If a and b are two sets, then there is the pair set $\{a, b\}$.” This statement uses the predicates (1) $S(x)$ is true iff x is a set, (2) $E(x, y)$ is true iff both $S(x)$ and $S(y)$ are true, and $x \in y$. We implicitly also need the predicate $I(x, y)$ which is true iff both $S(x)$ and $S(y)$ are true, and $x = y$. This setup transforms the axiom to the shape “If a and b are such that $S(a)$ and $S(b)$, then there is c with $E(a, c)$ and $E(b, c)$, and if x is such that $E(x, c)$, then $I(x, a)$ or $I(x, b)$.” Besides the predicative formalization, we here encounter two more specifications: the first part “If a and b are such that $S(a)$ and $S(b)$...”, which means “Whenever we take a, b ...”, in other words, we suppose a given universe of objects from where we may select instances a and b and then ask them to comply with certain predicates $S(a)$ and $S(b)$, i.e., to be sets. This is

expressed by the so-called *universal quantifier*: “**For all** a, b, \dots ”. Further, we also recognize an *existence quantifier*: “...there is c with ...”, which is expressed by “... **there exists** c with ...”.

In order to cope with the common mathematical constructions, one usually is a bit more specific in the formalization of predicates. As modeled from set theory, there are two basic types of predicates: relations and functions. This means that we are considering predicates defined by relations and functions in the following sense: Given n sets A_1, \dots, A_n and an n -ary relation, i.e., a subset $R \subset A_1 \times \dots \times A_n$ of their Cartesian product, one defines the associated n -ary predicate by $R(x_1, \dots, x_n)$ which is true iff $(x_1, \dots, x_n) \in R$. Observe that n -ary relations generalize the more restrictive concept of an n -ary relation $R \subset a^n$ introduced in definition 33 by varying each of the n factors of a^n . Similarly, if $f : A_1 \times \dots \times A_n \rightarrow A_{n+1}$ is a set function, one defines the predicate $f(x_1, \dots, x_n, y)$ which is true iff $f(x_1, \dots, x_n) = y$. This includes the two special cases where $n = 0$. For 0-ary relations, this means that we consider the ‘empty Cartesian product’ (check the universal property of Cartesian products to understand the following definition). We are given a subset of the final set 1, in other words, one of the classical truth values \perp, \top of the Boolean algebra 2. Thus, we also include the extremal truth values as basic predicates. As to functions of 0 variables, this means that the domain is again the empty Cartesian product 1. So a 0-ary function $f : 1 \rightarrow A_1$ is identified with the image $y = f(0)$ of the unique argument $0 \in 1$, in other words, 0-ary functions are just ‘constant’ elements in A_1 .

A last remark must be made concerning variables. We have constantly used some symbols a, b, x , etc. to feed the predicates. The nature of these variables has not been discussed. Since predicates are generated by relations or functions, we may interpret variables to refer to values in the corresponding domain sets A_i . However, variables may not refer to relations or functions or even higher order objects, such as relations of relations, etc. Together with this last restriction we have what is called *first order (formal) predicate logic*.

We are now ready to set up the formal framework. The methodology is quite the same as for formal propositional logic: One first defines the syntactical structures, then the objects of semantics, and finally the formalization of signification.

18.1 Syntactics: First-order Language

The basis of the formalized language is again a set of alphabetic symbols which we then combine to obtain reasonable words. Let us first formalize the relations and functions, together with the corresponding variables. To this end, if S is a set, define by $Sequ(S)$ the set of finite sequences $s = (A_1, \dots, A_n), A_i \in S$, where for $n = 0$ we take by definition the unique sequence which is defined on the empty index set 0 .

Exercise 81 Show that the set $Sequ(S)$ always exists, and that $Sequ(S) = Sequ(T)$ iff $S = T$.

Definition 117 For a finite set S , a signature is a triple

$$\Sigma = (FunType : Fun \rightarrow Sequ(S), RelType : Rel \rightarrow Sequ(S), (V_A)_{A \in S})$$

of two set maps and a family of denumerable sets V_A . In the uniquely determined S , finite by hypothesis, the elements $A \in S$ are called sorts, the elements $f \in Fun$ (in the uniquely determined set Fun) are called function symbols, and the elements $R \in Rel$ (in the uniquely determined set Rel) are called relation symbols. One supposes that all the sets Fun , Rel , and V_A are mutually disjoint. The elements $x \in V_A$ are called variables of sort A . The values of $FunType(f)$ and $RelType(R)$ are called the types of f and R , respectively. The length $n \geq 0$ of the type (A_1, \dots, A_n) of a relation symbol is called its arity; the number $n \geq 0$ in the type $(A_1, A_2, \dots, A_{n+1})$ of a function symbol is called its arity; so by definition, function symbols always have at least one sort in their type. In particular, 0-ary functions are called constants, whereas 0-ary relations are called atomic propositions.

Given a signature Σ , a function symbol f , together with its type $(A_1 \dots A_n, A_{n+1})$, is denoted by

$$f : A_1 \dots A_n \rightarrow A_{n+1},$$

where the last sort is denoted by A since its semantic role to be defined later is that of a codomain, but it is a sort much as the others are. A relation symbol R , together with its type $(A_1 \dots A_n)$, is denoted by

$$R \rightarrow A_1 \dots A_n.$$

In order to denote the sort A of a variable $x \in V_A$ or the $(n + 1)^{st}$ sort A of a function $f : A_1 \dots A_n \rightarrow A$ one also writes $x : A$, or $f : A$, respectively.

Since we are interested in a vocabulary of general usage, we shall moreover suppose that the following special symbols are part of our signature:

- For each sort A , the relational equality symbol $\stackrel{A}{=}$ with $\text{RelType}(\stackrel{A}{=}) = (A, A)$ is an element of Rel , and we usually use the infix notation $(a \stackrel{A}{=} b)$ instead of $\stackrel{A}{=} (a, b)$, and, if the sort A is clear, we just write $a = b$, but be warned: this is by no means equality in the sense of set theory, it is just an abbreviation of $\stackrel{A}{=}$ and has no content whatsoever on the present syntactical level of the theory.
- Among the atomic proposition symbols we have the falsity atom \perp and the truth atom \top . We shall not invent new symbols for these entities in order to distinguish them from the synonymous entities in logical algebras since there is no danger of confusion.

Example 75 As a prototypical example, we shall develop the predicate logic which describes Peano's construction of natural arithmetic, a setup, which we have modeled on the set theory of finite ordinal numbers, and which has been described in terms of Peano's five axioms (see propositions 45 and 47).

For Peano's axioms we need this repertory of symbols and operations:

- A symbol for the constant 0;
- a set of variables x, y, \dots to designate natural numbers;
- a predicate symbol of equality $x = y$ between natural numbers x, y ;
- a function symbol for the sum $x + y$ of two natural numbers x, y ;
- a function symbol for the product $x \cdot y$ of two natural numbers x, y ;
- a function symbol for the successor x^+ of a natural number x .

This requirement analysis yields the following signature: We have a single sort (the natural numbers) A , so the set of sorts is $S = \{A\}$. Accordingly, we have one (denumerable) set of variables V_A , from which we choose the variables x, y, \dots . To fix ideas, take the set of words $V_A = \{x_0, x_1, x_2, \dots\}$ with indexes being natural numbers in their decimal representation. The set of relation symbols is $\text{Rel} = \{\stackrel{A}{=} \mapsto AA, \perp \mapsto 0, \top \mapsto 1\}$, the set of function symbols is $\text{Fun} = \{\overset{A}{+} : AA \rightarrow A, \overset{A}{\cdot} : AA \rightarrow A, \overset{A}{+} : A \rightarrow A, \overset{A}{0} : 1 \rightarrow A\}$, the superscripts being added to indicate that we are only setting up a symbol set, and not real arithmetic operations. The type maps have

been given by the arrow notation within Fun . This means, that actually, $FunType(\overset{A}{+}) = (A, A, A)$, and $FunType(\overset{A}{0}) = (A)$, the latter being the constant symbol for zero.

With these symbols, one now defines the alphabet of a predicate language as follows:

Definition 118 A predicative alphabet is a triple $P = (\Sigma, B, C)$ of these sets:

- (i) The set Σ is a signature, with the defined set S of sorts, the set Fun of function symbols, the set Rel of relation symbols, and the family $(V_A)_S$ of sets of variables.
- (ii) The set $B = \{ (,) , , \}$ of left and right brackets, and the comma.
- (iii) The set $C = \{ !, \&, |, \rightarrow, \forall, \exists \}$ of connectives, where \forall is called the universal quantifier, and \exists is called the existence quantifier.

One again supposes that the sets V_A, Fun, Rel, B, C are mutually disjoint.

We again have this monoid of predicative expressions:

Definition 119 Given a predicative alphabet P as explicited in definition 118, the monoid $EX = EX(P)$ of expressions over P is the word monoid $Word(V_S \sqcup Fun \sqcup Rel \sqcup B \sqcup C)$, with $V_S = \bigsqcup_{A \in S} V_A$.

Like with sentences, we want to construct reasonable predicative expressions, which this time we call *formulas* instead of sentences. The construction needs an intermediate step.

Definition 120 Given a predicative alphabet P , the set $Term(P)$, the elements of which are called terms, is defined as the (uniquely determined) minimal subset $Term(P) \subset EX(P)$ such that the following two conditions hold. Simultaneously we add the recursive definition of the sort of a term.

- (i) $V_S \subset Term(P)$, and a variable $x \in V_A$ regarded as a term has the same sort $x : A$ as the variable as such. Attention: the expressions $x : A$ and $f : A$ are not words of $Term(P)$ or of any other formula, they are normal mathematical formulas.
- (ii) If, for $0 \leq n$, $t_1 : A_1, \dots, t_n : A_n$ are terms with the respective sort sequence of sorts (type) A_1, \dots, A_n , and if $f : (A_1 \dots A_n) \rightarrow A$ is a function symbol, then the expression $f(t_1, \dots, t_n)$ is a term, and we

define $f(t_1, \dots, t_n) : A$. In particular, the constants $f() : A$ are terms (the case of $n = 0$); for practical reasons we also include the words f (and the notation $f : A$ for $f() : A$) together with the constants $f()$.

Example 76 Taking up our prototypical example 75 of Peano arithmetic, we have these terms:

- the variables x_n with their (unique) sort $x_n : A$,
- the constant symbol $\overset{A}{0} : A$,
- the function symbols with terms in their argument places, e.g., ${}^{+A}(\overset{A}{0})$, which we presently also abbreviate by $\overset{A}{1}$, etc., $n \overset{A}{+} 1$ for ${}^{+A}(\overset{A}{n})$ (attention: this is only an informal convention to save space, but not strictly conforming to the definition)
- the function expressions $(\overset{A}{0} \overset{A}{+} \overset{A}{0})$, $(\overset{A}{0} \overset{A}{+} \overset{A}{1})$, $(\overset{A}{0} \overset{A}{+} \overset{A}{1})$, $((\overset{A}{0} \overset{A}{+} \overset{A}{0}) \overset{A}{+} \overset{A}{1})$, and so forth.

Similarly to the recursive construction of sentences in definition 109 and proposition 144, one may describe $Term(P)$ recursively.

Exercise 82 Give a recursive definition of $Term(P)$ in terms of its intersections $Term(P)_n = Term(P) \cap EX(P)_n$, starting from $Term(P)_0 = \emptyset$ and $Term(P)_1 = Fun_0 \sqcup V_S$, where $Fun_0 = \{f \mid f \in Fun, FunType(f) = (A), \text{ i.e., } 0\text{-ary}\}$ is the set of constant symbols.

We may now define general formulas for a predicative language by induction:

Definition 121 Given a predicative alphabet P , the set $F(EX)$ over the predicative alphabet P is the smallest subset $F(EX) \subset EX = EX(P)$ containing all these words which are called its formulas:

- the relational formulas $R(t_1, \dots, t_n)$ for $R \in Rel$, and terms $t_i \in Term(P)$ with $t_i : A_i$ for the type $RelType(R) = (A_1, \dots, A_n)$, including the 0-ary relation words $R()$, which we again shorten to R ; this includes in particular the equality formulas $(t \overset{A}{=} s)$ for terms $s : A$ and $t : A$;
- the truth formula \top and the falsity formula \perp ;

- (iii) negation: if ϕ is a formula, then so is $(\neg\phi)$;
- (iv) disjunction: if ϕ and ψ are formulas, then so is $(\phi \vee \psi)$;
- (v) conjunction: if ϕ and ψ are formulas, then so is $(\phi \wedge \psi)$;
- (vi) implication: if ϕ and ψ are formulas, then so is $(\phi \rightarrow \psi)$;
- (vii) universal quantification: if ϕ is a formula, and if x is any variable, then $(\forall x)\phi$ is a formula;
- (viii) existential quantification: if ϕ is a formula, and if x is any variable, then $(\exists x)\phi$ is a formula.

The set $F(EX)$ of formulas over the predicative alphabet P is called the predicative language defined by P .

Example 77 Continuing example 76 of Peano arithmetic, we have these formulas: In addition to the falsity and truth formulas \perp and \top , we have the equality relation formulas $t \stackrel{A}{=} s$, for terms s and t , e.g., $\overset{A}{0} \stackrel{A}{=} (\overset{A}{0} + \overset{A}{0})$. We then have the formulas obtained by logical connectives which are recursively applied to formulas, e.g., $(\neg\perp)$, $(\overset{A}{0} \stackrel{A}{=} (\overset{A}{0} + x_7))$, $(\top \wedge (\overset{A}{0} \stackrel{A}{=} \overset{A}{0}))$. Finally we have the formulas obtained from the universal quantifier, e.g., $(\forall x_3)(\overset{A}{0} \stackrel{A}{=} (x_3 + x_7))$, or from the existence quantifier, e.g., $(\exists x_1)(\overset{A}{0} \stackrel{A}{=} +^A(x_1))$.

Exercise 83 Give a recursive definition of $F(EX)$ which is based on the formulas of given word length n , namely, $F(EX)_n = F(EX) \cap EX_n$, where EX_n is the set of expression with word length n .

Within this vast vocabulary, many formulas are just meaningless ‘forms’. For example, the formula $(f(x) = 3)$ has no meaning if the variable x is not specified, even if we know a priori that the sorts $f : B$ and $3 : B$ coincide. However, if we prepend the existence quantifier, i.e., $(\exists x)(f(x) = 3)$, then the formula may become meaningful, i.e., loaded with a semantic truth value. Therefore one is interested in defining which variables in a formula are “bound” by a quantifier, and which are not, i.e., “free”. Here is the precise definition of the set $Free(\phi) \subset V_S$ of free variables of ϕ . It is, however, convenient to start this definition with the set of free variables of terms. Attention: We use the fact that the components of compound formulas, such as $(\forall x)\phi$ or $(\phi \rightarrow \psi)$, are uniquely determined. We have shown such a uniqueness theorem in proposition 147,

and it is recommended to meditate over this fact in the present context, too! In particular, the *scope of a variable* x is the uniquely determined formula ϕ following $(\forall x)$ or $(\exists x)$ in a formula $\dots (\forall x)\phi \dots$ or $\dots (\exists x)\phi \dots$, respectively.

- $Free(\top) = Free(\perp) = \emptyset$;
- for a constant term $t = f()$, we set $Free(t) = \emptyset$;
- for a variable x , we set $Free(x) = \{x\}$;
- for a term $t = f(t_1, \dots, t_n)$, $n > 0$, we set $Free(t) = \bigcup_i Free(t_i)$;
- if $\phi = R(t_1, \dots, t_n)$ is a relational formula, then $Free(\phi) = \bigcup_i Free(t_i)$, i.e., the set of all variables appearing in the terms t_i ;
- we set $Free(!\phi) = Free(\phi)$;
- if σ is one of the formulas $(\phi \mid \psi)$, $(\phi \& \psi)$, $(\phi \rightarrow \psi)$, then $Free(\sigma) = Free(\phi) \cup Free(\psi)$;
- we set $Free((\forall x)\phi) = Free((\exists x)\phi) = Free(\phi) - \{x\}$.

The concept of free variables is quite delicate. For example, the variable x is free in the formula $((\forall x)(f(x) = g(y)) \& (x = x))$ because it is free in the second formula, whereas it is not free in the first formula. Intuitively, the role of x in these two component formulas is completely different: In the first one, x could be replaced by any other variable of the same sort without changing the formula's meaning. In the second component, we could not do so because the usage of this one changes radically if we embed it in a larger context of formulas and variables.

Definition 122 A formula ϕ without free variables, i.e., $Free(\phi) = \emptyset$, is called a (predicative) sentence.

The only way to produce non-trivial sentences without free variables is to apply quantifiers. Suppose that the set V_S of variables is linearly ordered (we know that a finite disjoint union of n denumerable sets can be linearly ordered, for example by interpreting the j^{th} element of the i^{th} set, $i = 0, 1, 2 \dots n - 1$, as the natural number $j \cdot n + i$). We write this ordering as $x < y$ for variables x and y .

Definition 123 Given a formula $\phi \in F(EX)$, let $x_1 < x_2 < \dots x_r$ be the ordered sequence of the elements of $Free(\phi)$. Then we denote by

$(\forall)\phi$ the sentence $(\forall x_1)(\forall x_2)\dots(\forall x_r)\phi$, and by $(\exists)\phi$ the sentence $(\exists x_1)(\exists x_2)\dots(\exists x_r)\phi$ and call these sentences the universal or existential closures, respectively, of ϕ .

18.2 Semantics: Σ -Structures

Semantics has to provide us with logical algebras, where the truth values can be calculated from the formal data. Here is the framework for this calculation. Given an alphabet, the invariant data are the sets B of brackets and comma, and the set C of connectives. The set which can vary is the signature Σ . The semantic structure is tied to the signature. We need these objects:

Definition 124 Given a signature Σ , a (set-theoretic) Σ -structure \mathfrak{M} is defined by these sets:

- (i) For each sort $A \in S$, we are given a set \mathfrak{M}_A .
- (ii) For each n -ary relational symbol $R \rightarrow A_1 \dots A_n$, we are given a subset $\mathfrak{M}_R \subset \mathfrak{M}_{A_1} \times \dots \times \mathfrak{M}_{A_n}$, called a relation (recall that we had defined a relation by a subset of the second power X^2 of a set X , and a graph as a subset of a Cartesian product $X \times Y$ of sets X and Y , the present one is a generalization of those concepts). In particular, for atomic propositions with $n = 0$, e.g., \perp and \top , we are given the subsets of the final set 1 , i.e., the truth values $\perp = 0$ and $\top = 1$, elements in the Boolean algebra 2 .
- (iii) For each n -ary function symbol $f : A_1 \dots A_n \rightarrow A$, we are given a set function $\mathfrak{M}_f : \mathfrak{M}_{A_1} \times \dots \times \mathfrak{M}_{A_n} \rightarrow \mathfrak{M}_A$. In particular, for $n = 0$ we are given a ‘‘constant’’, i.e., an element $\mathfrak{M}_f \in \mathfrak{M}_A$.
- (iv) For each equality symbol $\underline{=}$, we are given the diagonal relation $\mathfrak{M}_{\underline{=}} = \Delta_A \subset \mathfrak{M}_A^2$.

We shall now be able to define truth values in the Boolean algebras of the sets $\mathfrak{M}_{A_1} \times \dots \times \mathfrak{M}_{A_n}$ and to define signification of formulas with respect to these logical algebras.

Of course, if we would take a more general Heyting algebra on the powerset of such a Cartesian product, or even on Cartesian products of digraphs and still more ‘exotic’ objects, we would obtain a more general

predicate logic. This can be done in the so-called topos theory (see for example [21]), but for our modest needs, we stick to the classical situation of Boolean powerset algebras. This is what we mean when talking about “set-theoretic” Σ -structures.

Example 78 Following up our prototypical example 77 of Peano arithmetic, we may define a Σ -structure \mathfrak{N} which everybody would expect: For the sort A , take $\mathfrak{N}_A = \mathbb{N}$. For \perp and \top , we have no choice by definition, i.e., $\perp = 0 \subset 1$ and $\top = 1 \subset 1$. For equality, we have to take $\mathfrak{N}_{=} = \Delta_{\mathbb{N}}$. For the function $+$ we take the ordinary addition $\mathfrak{N}_{+} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} : (x, y) \mapsto x + y$, for \cdot we take the ordinary multiplication $\mathfrak{N}_{\cdot} : \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N} : (x, y) \mapsto x \cdot y$, and for $^+$ we take the ordinary successor $\mathfrak{N}_{^+} : \mathbb{N} \rightarrow \mathbb{N} : x \mapsto x^+$. Finally, set $\mathfrak{N}_{0} = 0 \in \mathbb{N}$.

But we could also take any other structure \mathfrak{N}' , for example exchanging addition and multiplication in the above \mathfrak{N} , i.e., $\mathfrak{N}'_{+} = \mathfrak{N}_{\cdot}$ and $\mathfrak{N}'_{\cdot} = \mathfrak{N}_{+}$, and setting $\mathfrak{N}'_{0} = 23 \in \mathbb{N}$.

18.3 Signification: Models

The truth values of formulas are constructed as follows.

Definition 125 Let \mathfrak{N} be a Σ -structure. If t is a term and ϕ is a formula, and if $x_1 < \dots < x_n$ are their free variables with corresponding sorts A_1, \dots, A_n , we denote by \mathfrak{N}_t and \mathfrak{N}_{ϕ} , the Cartesian product $\mathfrak{N}_{A_1} \times \dots \times \mathfrak{N}_{A_n}$, including the special case $n = 0$, where we set $\mathfrak{N}_t = 1$ or $\mathfrak{N}_{\phi} = 1$. This set is called the free range of t and ϕ .

Next, we need to define evaluation of a term for specific values under a given Σ -structure. Let t be a term, and $x \in \mathfrak{N}_t$. Then the evaluation $s[x] \in \mathfrak{N}_s$ at x of a term s with $Free(s) \subset Free(t)$ is recursively defined by (1) the component at position s , $s[x] = x_s \in \mathfrak{N}_A$ if $s : A$ is a variable of sort A ; (2) the value $s[x] = \mathfrak{N}_f(t_1[x], \dots, t_m[x]) \in \mathfrak{N}_A$ for $s = f(t_1, \dots, t_m)$ and we have $f : A$.

We shall now attribute to each formula $\phi \in F(EX)$ a truth value $\mathsf{T}(\phi)$ in the Boolean algebra $2^{\mathfrak{N}_{\phi}}$, i.e., a subset $\mathsf{T}(\phi) \subset \mathfrak{N}_{\phi}$.

Definition 126 If $\phi \in F(EX)$ is a formula, and if \mathfrak{M} is a Σ -structure, one defines $\mathsf{T}(\phi)$ according to these cases:

- (i) If $\phi = R$ is an atomic proposition, one sets $\mathsf{T}(\phi) = \mathfrak{M}_R \in 2$, in particular, $\mathsf{T}(\top) = \top$ and $\mathsf{T}(\perp) = \perp$.
- (ii) If $\phi = R(t_1, \dots, t_m)$, $m > 0$, then

$$\mathsf{T}(\phi) = \{x \mid x \in \mathfrak{M}_\phi, (t_1[x], \dots, t_m[x]) \in \mathfrak{M}_R\}$$

- (iii) If $\phi = (!\psi)$, then $\text{Free}(\phi) = \text{Free}(\psi)$, and we set $\mathsf{T}(\phi) = \mathfrak{M}_\phi - \mathsf{T}(\psi)$.

- (iv) For the three cases $\phi = (\psi * \rho)$ where $*$ = $\{\&, |, \rightarrow\}$, one has $\text{Free}(\phi) = \text{Free}(\psi) \cup \text{Free}(\rho)$, and therefore canonical projections $p_\psi : \mathfrak{M}_\phi \rightarrow \mathfrak{M}_\psi$ and $p_\rho : \mathfrak{M}_\phi \rightarrow \mathfrak{M}_\rho$. We then use the Boolean connectives and define

- $\mathsf{T}((\psi \& \rho)) = p_\psi^{-1}(\mathsf{T}(\psi)) \cap p_\rho^{-1}(\mathsf{T}(\rho))$,
- $\mathsf{T}((\psi | \rho)) = p_\psi^{-1}(\mathsf{T}(\psi)) \cup p_\rho^{-1}(\mathsf{T}(\rho))$,
- $\mathsf{T}((\psi \rightarrow \rho)) = \mathsf{T}((!\psi) | \rho)$,

- (v) For $\phi = (\forall x)\psi$ or $\phi = (\exists x)\psi$, one has $\text{Free}(\phi) = \text{Free}(\psi) - \{x\}$ and therefore the projection $p : \mathfrak{M}_\psi \rightarrow \mathfrak{M}_\phi$. Then one sets

- $\mathsf{T}((\forall x)\psi) = \{y \mid y \in \mathfrak{M}_\phi, p^{-1}(y) \subset \mathsf{T}(\psi)\}$,
- $\mathsf{T}((\exists x)\psi) = \{y \mid y \in \mathfrak{M}_\phi, p^{-1}(y) \cap \mathsf{T}(\psi) \neq \emptyset\}$,

including the special case where $x \notin \text{Free}(\psi)$. In this case the projection is the identity, and we have $\mathsf{T}((\forall x)\psi) = \mathsf{T}(\psi)$ and $\mathsf{T}((\exists x)\psi) = \mathsf{T}(\psi)$.

Given these truth evaluations of formulas, one can state validity of formulas similarly to propositional validity. If $\text{Free}(\phi) = x_1 < x_2 < \dots < x_m$ defines a subsequence of a sequence $y_1 < y_2 < \dots < y_n$ of variables with sorts $y_i : B_i$, then, if $y \in \mathfrak{M}_{B_1} \times \dots \times \mathfrak{M}_{B_n}$, we have the projection y_ϕ of y to the coordinate sequence from \mathfrak{M}_ϕ . We then define that ϕ is valid in y , $\mathfrak{M} \models \phi[y]$ iff $y_\phi \in \mathsf{T}(\phi)$. If ϕ is a sentence, we write $\mathfrak{M} \models \phi$ for this fact (which is now independent of y), and this means that $\mathsf{T}(\phi) = \top$. One then says that the Σ -structure \mathfrak{M} is a model for the sentence ϕ . If ϕ is not a sentence, one considers its universal closure sentence $(\forall)\phi$, see definition 123, and defines validity of ϕ by “ $\mathfrak{M} \models \phi$ iff $\mathfrak{M} \models (\forall)\phi$ ”, which means that $\mathfrak{M} \models \phi[y]$ for all y as above.

Example 79 To conclude example 78 of Peano arithmetic, we want to model the formulas which are given by Peano’s five axioms. Here are

these formulas, including those defining addition and multiplication, all of which are in fact sentences (observe that one could omit the quantifiers in the following sentences and then use the universal closure to model the Peano axioms):

(i) (Zero is not a successor) $(\forall x_1)(!(0 \stackrel{A}{=} +^A(x_1)))$,

(ii) (Equal successors have equal predecessors)

$$(\forall x_1)(\forall x_2)((+^A(x_1) \stackrel{A}{=} +^A(x_2)) \rightarrow (x_1 \stackrel{A}{=} x_2)),$$

(iii) (Zero is additive neutral element) $(\forall x_1)((x_1 \stackrel{A}{+} 0) \stackrel{A}{=} x_1)$,

(iv) (Recursive definition of addition)

$$(\forall x_1)(\forall x_2)((x_1 \stackrel{A}{+} +^A(x_2)) \stackrel{A}{=} (+^A(x_1 \stackrel{A}{+} x_2))),$$

(v) (Zero is multiplicative “neutralizer”) $(\forall x_1)((x_1 \stackrel{A}{\cdot} 0) \stackrel{A}{=} 0)$,

(vi) (Recursive definition of multiplication)

$$(\forall x_1)(\forall x_2)((x_1 \stackrel{A}{\cdot} +^A(x_2)) \stackrel{A}{=} ((x_1 \stackrel{A}{\cdot} x_2) \stackrel{A}{+} x_1)),$$

(vii) (Principle of induction) Denote by $\Phi(x_i)$ a formula, where x_i pertains to its free variables. By $\Phi(+^A(x_i))$ and $\Phi(0)$, we denote the formula after the replacement of each occurrence of x_i by $+^A(x_i)$ and 0 , respectively. Then, for each formula $\Phi(x_i)$, we have this formula:

$$((\Phi(0) \& (\forall x_i)(\Phi(x_i) \rightarrow \Phi(+^A(x_i)))) \rightarrow (\forall x_i)\Phi(x_i)).$$

The last item (vii) is not one formula, but one formula for each formula Φ . A *Peano structure* is a structure \mathfrak{M} such that for each of the formulas Ψ described in (i)–(vii), we have $\mathfrak{M} \models \Psi$.

Let us now check the validity of formula (i) for our structure \mathfrak{M} described in example 78. Formula (i) has the shape $\phi = (\forall x_1)\psi$, which is evaluated according to the projection $p : \mathfrak{M}_\psi \rightarrow \mathfrak{M}_\phi = 1$, since $\text{Free}(\phi) = \emptyset$. So let us check the fiber $\mathfrak{M}_\psi = p^{-1}(0)$ and test whether $\mathfrak{M}_\psi \subset \text{T}(\psi)$, i.e., $\mathfrak{M}_\psi = \text{T}(\psi)$. But $\text{T}(\psi) = \text{T}(! (0 \stackrel{A}{=} +^A(x_1))) = \mathbb{N} - \text{T}(0 \stackrel{A}{=} +^A(x_1))$, and $\text{T}(0 \stackrel{A}{=} +^A(x_1)) = \{x \mid x \in \mathfrak{M}_\psi = \mathbb{N}, 0 = 0[x] = +^A(x_1)[x] = x + 1\}$, which is the empty set, we are done, i.e., $\text{T}(\psi) = \mathbb{N}$ and $\mathfrak{M} \models \phi$.

Exercise 84 Check whether our structure \mathfrak{N} described in example 78 is a Peano structure. Do the same for the structure \mathfrak{N}' .

Many of the possible formulas are equivalent in the sense that they yield the same logical values. More precisely:

Definition 127 Given a Σ -structure \mathfrak{N} , two formulas ϕ and ψ are called equivalent iff we have $\mathfrak{N} \models (\phi \leftrightarrow \psi)$ with the usual bimplication $(\phi \leftrightarrow \psi)$ as an abbreviation for the conjunction $((\phi \rightarrow \psi) \& (\psi \rightarrow \phi))$.

So we are looking for equivalent formulas which look as simple as possible. One such simplified type is the *prenex normal form*:

Definition 128 A formula is in prenex normal form (or shorter: is prenex) iff it is an uninterrupted (possibly empty) sequence of universal or existential quantifiers, followed by a formula without quantifiers. It is in Skolem normal form if it is in prenex normal form such that all existential quantifiers precede all universal quantifiers.

Example 80 The formula $(\forall x)(\exists y)(\forall z)((x = y) \mid (z < w))$ is in prenex normal form.

Here is the crucial result with regard to Skolem normalization:

Proposition 156 Every formula ϕ of a Σ -structure \mathfrak{N} is equivalent to a Skolem formula ψ .

Proof The proof of this theorem is by induction on the length of the formula. It is not difficult, but uses a number of auxiliary lemmas which we have no place to deal with here. However, the principal ideas are these: To begin with, if a bound quantifier in $(\forall x)\phi$ is replaced by any other variable z except the free variables of ϕ different from x , the new formula is equivalent to the old one. One then shows that $!(\forall x)\phi$ is equivalent to $(\exists x)!(\phi)$ and that $!(\exists x)\phi$ is equivalent to $(\forall x)!(\phi)$. If $(\phi * (\exists x)\psi)$, where $* \in \{!, \&\}$, is a formula, we may suppose that x is not one of the variables in $Free(\psi)$, and then $(\phi * (\exists x)\psi)$ is equivalent to $(\exists x)(\phi * \psi)$, similarly for the existence quantifier. Formulas of shape $(\psi \rightarrow \phi)$ are equivalent to $((!\psi) \mid \phi)$ by the very definition of truth values for implications. This gives us the prenex normal form. To construct the Skolem normal form, one needs auxiliary formulas and free variables, which must be added to the given formula in order to enable the existence quantifiers to precede the universal quantifiers, see [27] for details. \square

Example 81 The mathematician Paul Finsler has proposed a problem in absolute vs. formal mathematical reasoning which has provoked violent reactions among mathematicians. His proposal regards statements which cannot be proved by formal reasoning, but nevertheless can be proved by non-formal reasoning, i.e., by non-formalized human thought.

Suppose that we are given a formal system and a finite (or even denumerable) alphabet which is used to write down formal proofs in the shape of finite chains of words (including the empty space to separate words), as described in the preceding chapters. Clearly, the set of these chains is denumerable. Now, we are only interested in such chains of words which are correct (formal) proofs of a very specific type of statement: We consider binary representations of numbers $d = 0.d_1d_2\dots$, and we only consider those chains $Ch \rightarrow d$ of words which are correct proofs of the fact that either a specific binary number d has or has not infinitely many zeros. We may order these proofs lexicographically and also order their binary numbers according to this ordering. We then obtain a sequence $d(1), d(2), \dots, d(n), \dots$ of all binary numbers which admit any proof chain of the given type. So observe that there are no other formal proofs in this formal framework which decide on the infinity or non-infinity of zeros in binary numbers. Now define a new binary number a by an antidiagonal procedure: $a_n = 1 - d(n)_n, n = 1, 2, \dots$. For this binary number, there is no formal proof in our repertory, since any proof $Ch \rightarrow a$ would place a at a position m , say. And if $a = d(m)$, then we have a contradiction $a_m = 1 - d(m)_m = d(m)_m$. So a has no formal proof.

But we may give an informal proof, and show that a has an infinity of zeros:

In fact, take the binary sequence $d = 0.1111\dots$ having only 1 as entries. By definition, this d has no zeros. A formal proof, say $Ch_0 \rightarrow 0.1111\dots$, of this is immediate. But then the formal proofs $Ch_0 \& Ch_0 \& Ch_0 \dots Ch_0$, n times, for $n = 1, 2, 3 \dots$ all also do the job. So the number $0.1111\dots$ appears an infinity of times (once for each such formal proof), and the antidiagonal therefore has an infinity of zeros.

The point is that by construction of the set of *all* proofs in the given formal system, this set cannot contain the proof just delivered. (The formal description uses the above letters which we may easily suppose to be part of our alphabet.) Finsler argues that we *must have left the formal system, because the proof is, by construction, not in the list of proofs.*

Languages, Grammars, and Automata

Until now, all formalizations of logic have been presented on a level which does not directly involve computers in the sense of machines which can execute commands and produce an output relating to formalized logical expressions. So our overall plan to incorporate logical reasoning in computerized operations still lacks the conceptual comprehension of how machines may (or may not) tackle such a task. We have learned in chapters 17 and 18 that formalized logic is built upon word monoids over adequate finite or infinite alphabets. We have also learned that the reasonable expressions of formal logic are words which can be given by recursion on the word length and a defined set of construction rules. More explicitly, this was formalized in the axiomatic setup of formal logic, where the deduction of theorems starts from a set of axioms and proceeds by the application of a set of deduction rules.

Now, it turns out that this setup of formal logic and its construction methods is not really bound to the logical context. In fact, computers don't care about what words they are dealing with, the only relevant point of view to them is that they are allowed to build new words from given ones, following certain rules, and specific alphabets. The semantical issue of logic is not a *conditio sine qua non* for the formal control of reasonable expressions (words). In what follows, we shall therefore develop the more comprising context of general formal languages, their generative description by use of so-called phrase structure grammars, and their machine-oriented restatements by automata. So we have a triple perspective: first

the “static” description of a (formal) language, second its more “dynamic” description by grammatical production systems, and third, the machine processes of automata, which encompass certain languages. *The point of this triple approach is that in fact, with respect to the languages they are generating, certain classes of grammars correspond to prominent classes of automata (such as stack automata or Turing machines) arranged in a four-fold hierarchy due to the linguist Noam Chomsky [14].*

19.1 Languages

This section is not more than a souped-up review of what we have already learned about word monoids in section 15.1. To deal with (formal) languages, we need an alphabet and then investigate the set of words generated by some determined procedure. However, it is worth extending the word concept to infinite words, *streams*, for the sake of completeness of conceptualization. *In this section, we shall always reserve the letter \mathcal{A} for a set which plays the role of an alphabet.* Attention: The alphabet may be a finite or infinite set, no restriction on the cardinality is assumed in general.

Definition 129 *Given an alphabet \mathcal{A} , a stream or infinite word (over \mathcal{A}) (in contrast to a common word, which is also termed finite stream) is an infinite sequence $s = (s_0, s_1, \dots) \in \mathcal{A}^{\mathbb{N}}$ of \mathcal{A} -letters. The length of a stream s is said to be infinity, in symbols: $l(s) = \infty$. The (evidently disjoint) union $Word(\mathcal{A}) \cup \mathcal{A}^{\mathbb{N}}$ is denoted by $Stream(\mathcal{A})$ and is called the stream monoid over \mathcal{A} . Its monoid product is defined as follows: If $x, y \in Word(\mathcal{A})$, we reuse the given product xy in $Word(\mathcal{A})$. If $x \in \mathcal{A}^{\mathbb{N}}$ is a stream and y is any element of $Stream(\mathcal{A})$, we set $xy = x$ and say that streams are right absorbing; if $x = a_1 a_2 \dots a_n \in Word(\mathcal{A})$ and $y = (y_0, y_1, \dots)$ is a stream, we define $xy = (a_1, a_2, \dots, a_n, y_0, y_1, \dots)$, i.e., x is prepended to the stream y . In particular, if $x = \varepsilon$ is the neutral element, we set $xy = y$. In theoretical computer science it is common to call a proper left (right) factor x of a word or stream z , i.e., $z = xy$ ($z = yx$) a prefix (suffix) of z ; if $z = uxv$ with $u, v \neq \varepsilon$, then x is called infix of z .*

Exercise 85 Verify that $Stream(\mathcal{A})$ is indeed a monoid, and that we have the submonoid $Word(\mathcal{A}) \subset Stream(\mathcal{A})$ of finite words.

Definition 130 Given an alphabet \mathcal{A} , a stream language (over \mathcal{A}) is a subset $L \subset \text{Stream}(\mathcal{A})$. If the stream language L is contained in the submonoid $\text{Word}(\mathcal{A})$, it is called a word language, or simply a language. The set of languages over \mathcal{A} identifies with the powerset $2^{\text{Word}(\mathcal{A})}$ and is denoted by $\text{Lang}(\mathcal{A})$.

We shall mostly deal with word languages, except in rare cases, where the exception is explicitly indicated. So languages are completely unstructured subsets of $\text{Word}(\mathcal{A})$. Therefore, the Boolean operations on the Boolean algebra $\text{Lang}(\mathcal{A})$ generate new languages from given ones, in particular, we have the union $L_1 \cup L_2$ and the intersection $L_1 \cap L_2$ of two languages L_1 and L_2 , as well as the complement $-L$ of a language L over \mathcal{A} . Moreover, if $L_1, L_2 \in \text{Lang}(\mathcal{A})$, we have the product language $L_1 L_2 = \{x\mathcal{Y} \mid x \in L_1, \mathcal{Y} \in L_2\}$. In particular, for $n \in \mathbb{N}$, we have the powers $L^n = LL \dots L$ (n times) of L , including $L^0 = \{\varepsilon\}$, the *unit language* over \mathcal{A} . We say that L is *closed under concatenation* iff $L^2 \subset L$.

Example 82 Let $\mathcal{A} = \{a, b\}$, and $L_1, L_2, L_3 \in \text{Lang}(\mathcal{A})$ be defined as follows: L_1 is the language of non-empty words of the form $abab \dots$ of finite or infinite length (we also write $L_1 = \{(ab)^n \mid n > 0\}$). L_2 is the language of words of length ≤ 4 . L_3 is the language of non-empty words of the form $baba \dots$ of finite or infinite length ($L_3 = \{(ba)^n \mid n > 0\}$). Then $L_1 \cap L_2$ is the set $\{ab, abab\}$. L_2^2 is the language of words of length ≤ 8 and L_1 is closed under concatenation. $L_1 \cup L_3$ is the language of all non-empty words with alternating letters a and b , finite or infinite, with first letter a or b . The complement $-L_2$ contains all words with length > 4 . Finally, $L_1 \cap L_3$ is empty.

Exercise 86 Show that for a given alphabet \mathcal{A} , $\text{Lang}(\mathcal{A})$, together with the product of languages is a monoid with neutral element $\{\varepsilon\}$.

Definition 131 Given an alphabet \mathcal{A} , the Kleene operator is the map

$$* : \text{Lang}(\mathcal{A}) \rightarrow \text{Lang}(\mathcal{A}) : L \mapsto L^* = \langle L \rangle$$

which associates with every language L the monoid L^* generated by L .

Example 83 Let $\mathcal{A} = \{a, b, c\}$ and $L = \{aa, ab, ac, ba, bb, bc, ca, cb, cc\}$. Then L^* is the language of all words of even length.

Exercise 87 Show that for a given alphabet \mathcal{A} , a language L is closed under concatenation iff $L^* = L \cup \{\varepsilon\}$. Further show that the Kleene operator is *idempotent*, i.e., $L^{**} = L^*$.

Remark 23 We avoid the common, but ill-chosen notation \mathcal{A}^* for the word monoid $Word(\mathcal{A})$ since it conflicts with the Kleene operator. In fact, $Word(L) \neq L^*$ in general. Verify this latter inequality.

We shall now give an important example of languages as related to automata, which will be dealt with more extensively in section 19.3. Recall from section 12.2 about Moore graphs that a sequential machine of n variables was a map $M : S \times Q^n \rightarrow S$ involving a state space S and the n -cube Q^n as an input set. More generally, we may define a *sequential machine over an alphabet \mathcal{A} and state space S* as being a map $M : S \times \mathcal{A} \rightarrow S$, and again we write $s \cdot a$ for $M(s, a)$ if M is clear. The *Moore graph of M* is defined as previously in the special case $\mathcal{A} = Q^n$ by $Moore(M) : S \times \mathcal{A} \rightarrow S^2 : (s, a) \mapsto (s, s \cdot a)$. The proposition 106 of section 12.2 discussed in the case of $\mathcal{A} = Q^n$ is also valid for general alphabets:

Proposition 157 For a sequential machine $M : S \times \mathcal{A} \rightarrow S$, a canonical bijection

$$PW : Path(Moore(M)) \rightarrow S \times Word(\mathcal{A})$$

is given as follows. If

$$p = s_1 \xrightarrow{(s_1, a_1)} s_2 \xrightarrow{(s_2, a_2)} s_3 \dots \xrightarrow{(s_{m-1}, a_{m-1})} s_m,$$

then $PW(p) = (s_1, a_1 a_2 \dots a_{m-1})$.

Under this bijection, for a given state $s \in S$, the set $Path_s(Moore(M))$ of paths starting at s corresponds to the set $\{s\} \times Word(\mathcal{A})$.

Proof The proof is completely analogous to the proof of proposition 106 in the case of $\mathcal{A} = Q^n$, we therefore refer to that text. \square

Under the previous bijection PW we can associate with each couple $(s, w) \in S \times Word(\mathcal{A})$ the state $W(M)(s, w)$ resulting from the successive application of the word's letters by this map:

$$W(M) : S \times Word(\mathcal{A}) \rightarrow S : (s, w) \mapsto head(PW^{-1}(s, w))$$

If M is clear, we shall also write $s \cdot w$ instead of $W(M)(s, w)$.

Example 84 Given a sequential machine M over an alphabet \mathcal{A} and state space S , one is often interested in a set $E \subset S$ of *final* states insofar as they may be reached by M from an *initial* state $i \in S$. This means by definition that we look for words $w \in \text{Word}(\mathcal{A})$ such that $i \cdot w \in E$. Denote the language (i.e., the set) of these words by $(i : M : E)$, or $(i : E)$ if M is clear from the context, and call these words *the words which are accepted by the sequential machine M* and this language *the language which is accepted by the sequential machine M* .

To give a concrete example, take $S = \{1, 2, 3, 4\}$, $\mathcal{A} = \{a, b, c\}$, while the machine M is defined by this table (state i is mapped by letter x to the state on the column below x on the row of state i , e.g., $3 \cdot c = 4$):

letter \rightarrow	a	b	c
state 1	2	2	1
state 2	1	2	3
state 3	1	2	4
state 4	1	3	1

The graph for this machine is shown in figure 19.1.

Take $i = 2, E = \{4, 2\}$ and calculate the language $(i : E)$.

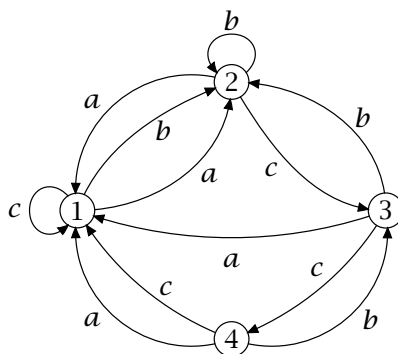


Fig. 19.1. The graph for the sequential machine of example 84.

Before leaving this generic subject, we should make a concluding remark about alphabets as they occur in the real life of computer scientists. Mathematically, the set \mathcal{A} comprising the “letters” $x \in \mathcal{A}$ is quite irrelevant,

and this is also generally true for computer science. Therefore standardization committees have agreed to create standard alphabets of natural numbers that represent known sets of letters. The most famous is the *American Standard Code for Information Interchange (ASCII)* character set codification, as made precise by the Standard ANSI X3.4-1986, “US-ASCII. Coded Character Set - 7-Bit American Standard Code for Information Interchange”. Figure 19.2 shows a sample of that encoding.¹

octal	decimal	hexadecimal	Name
⋮	⋮	⋮	⋮
060	48	0x30	0 (zero)
061	49	0x31	1
062	50	0x32	2
063	51	0x33	3
064	52	0x34	4
065	53	0x35	5
066	54	0x36	6
067	55	0x37	7
070	56	0x38	8
071	57	0x39	9
072	58	0x3a	: (colon)
073	59	0x3b	; (semicolon)
074	60	0x3c	< (less than)
075	61	0x3d	= (equals)
076	62	0x3e	> (greater than)
077	63	0x3f	? (question mark)
0100	64	0x40	@ (commercial at)
0101	65	0x41	A
0102	66	0x42	B
0103	67	0x43	C
0104	68	0x44	D
⋮	⋮	⋮	⋮

Fig. 19.2. Excerpt from ASCII encoding.

¹ See <http://www.asciitable.com> for the complete table of the $2^7 = 128$ characters encoded by ASCII.

Here, the octal representation refers to the 8-ary representation of numbers, whereas the hexadecimal refers to basis 16. One has the following prefix notations in order to distinguish different adic representations: binary and decimal numbers have no prefix, octal numbers have the prefix 0, and hexadecimal numbers start with 0x.

Thus ASCII sets up a bijection of the integer interval $[0, 127]$ and a set of relevant characters, predominantly used in the Angloamerican culture. However, as computers have spread over all cultures, more comprehensive character and sign types have been included in the standardization, ultimately leading to the *Unicode* standard. This is a 16-bit character set standard, designed and maintained by the non-profit consortium *Unicode Inc.* Parallel to the development of Unicode an ISO/IEC standard was worked on, putting a large emphasis on being compatible with existing character codes such as ASCII. Merging the ISO (International Organization for Standardization) standard effort and Unicode in 1992, the *Basic Multilingual Plane BMP* was created. But presently the BMP is half empty, although it covers all major languages, including Roman, Greek, Cyrillic, Chinese, Hiragana, Katakana, Devanagari, Easter Island “rongorongo”, and even Elvish (but leaves out Klingon).²

19.2 Grammars

We evidently need means to classify languages, since to date the Babylonian variety of languages is beyond control: just imagine all possible languages based on the traditional European alphabet. We are rightly confused by the world’s variety of dead or living languages, and by the ever growing variety of computer languages. A natural way to access languages is a rule system which directly produces language items, i.e., a grammatical construction which we also use in natural language to build new sentences from given ones and from phrase building schemes. Observe that this is a totally different approach to languages as compared to the language construction by a sequential machine introduced in the above example 84. We shall however relate these approaches in the following section 19.3 on automata.

² See <http://www.unicode.org> for more information about the Unicode standard.

Definition 132 Given an alphabet \mathcal{A} , a production grammar over \mathcal{A} is a map

$$f : \text{Lang}(\mathcal{A}) \rightarrow \text{Lang}(\mathcal{A})$$

which commutes with unions, i.e., for any family $(L_i)_{i \in I}$ of languages L_i over \mathcal{A} , we have

$$f\left(\bigcup_I L_i\right) = \bigcup_I f(L_i),$$

in particular $f(\emptyset) = \emptyset$. If $w \in \text{Word}(\mathcal{A})$ is a word, we set $f(w) = f(\{w\})$ and obtain a restricted map $f : \text{Word}(\mathcal{A}) \rightarrow \text{Lang}(\mathcal{A}) : x \mapsto f(x)$, which we denote by the same symbol. One then has

$$f(L) = \bigcup_{x \in L} f(x)$$

for any language $L \in \text{Lang}(\mathcal{A})$. Conversely, if we are given any map $g : \text{Word}(\mathcal{A}) \rightarrow \text{Lang}(\mathcal{A}) : x \mapsto f(x)$, we obtain a production grammar (again denoted by the same symbol) $f : \text{Lang}(\mathcal{A}) \rightarrow \text{Lang}(\mathcal{A})$ defined by the above formula $f(L) = \bigcup_{x \in L} f(x)$. In examples, we shall use either definition according to the concrete situation.

If a production grammar is such that $f(x)$ is always a singleton set, i.e., $f(x) = \{y\}$ for all $x \in \text{Word}(\mathcal{A})$, then one calls f deterministic, otherwise, it is called nondeterministic. For a deterministic f , we also write $f(x) = y$ instead of $f(x) = \{y\}$.

Given a production grammar $f : \text{Lang}(\mathcal{A}) \rightarrow \text{Lang}(\mathcal{A})$ and an initial language $I \in \text{Lang}(\mathcal{A})$, one has the language $f^\infty(I)$ generated by f starting from I , i.e., $f^\infty(I) = \bigcup_{0 \leq i} f^i(I)$ with $f^0 = \text{Id}$ and $f^i = f \circ f \circ \dots \circ f$, i times, for positive i . If we are also given a language of terminals T , the language generated starting from I and terminating in T is defined by $(I : f : T) = T \cap f^\infty(I)$. If the production grammar f is clear, one also writes $(I : T)$ instead of $(I : f : T)$. For a given alphabet \mathcal{A} , two production grammars f_1 and f_2 with initial and terminal languages I_1 and I_2 and T_1 and T_2 , respectively, are called equivalent iff $(I_1 : f_1 : T_1) = (I_2 : f_2 : T_2)$.

Example 85 L-systems are a type of production grammar proposed by biologist Aristid Lindenmayer in 1968 to provide an axiomatic description of plant growth. We give a simple example of a so-called *turtle graphics* production grammar t to illustrate the power of L-systems for the production of complex graphical objects associated with the language $t^\infty(I)$. One starts with a small alphabet $\mathcal{A} = \{F, G, +, -\}$, a grammar $t = t_{w_1, \dots, w_n}$ which is defined by a finite set $\{w_1, \dots, w_n\}$ of words

and the function $t = t_{w_1, \dots, w_n}(x) = \{x|w_1, \dots, x|w_n\}$ on words x , where $x|w$ denotes the word deduced from x by replacing each appearance of the letter F by w . For example, if $x = F-FG+F$ and $w = FG-F$, then $x|w = FG-F-FG-FG+FG-F$.

Let us consider the deterministic case $t_w, w = F-FG+F$ with one initial word $x_0 = FG$, i.e., $I = \{x_0 = FG\}$. Then the language $t_w^\infty(x_0)$ is the infinite set

$$\{FG, F-FG+FG, F-FG+F-F-FG+FG+F-FG+FG \dots\}.$$

The turtle language graphically interprets letters and words as follows: Read a word as a sequence of commands from left to right, so $F+FG$ means: First do F, then do +, then do F, then do G. The command associated with F is this: You are a turtle moving on a white paper surface. Whenever you move, you leave a trace of ink on the surface. Now, doing F in a word where we have k appearances of the letter x means that the turtle has a given direction and moves on a straight line of defined length $\frac{1}{k}$. Doing G means that the turtle draws a circle of diameter $\frac{1}{4k}$ around its center, but then recovers its position after drawing the circle. Doing + means that the turtle just turns clockwise by 90 degrees around its center, whereas - means a counter-clockwise turn by 90 degrees.

What is the graphical interpretation of the production rule $x \mapsto x|w$? It means that every straight line segment in the turtle drawing defined by x is replaced by the drawing defined by w placed in the direction of that line and shrunk by the factor k such that the total length of the drawing remains constant, i.e., 1 in our case. This is also why L-systems are labeled “rewriting systems”. Observe that in contrast to F, the action G is not rewritten, it is a kind of “terminal” entity.

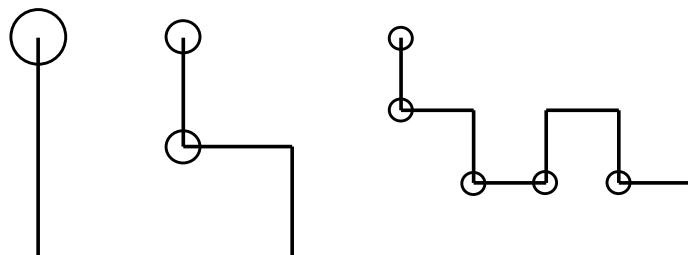


Fig. 19.3. The graphical interpretation of the first three words of the L-system featured in example 85.

An important class of production grammars is described by the following definition:

Definition 133 We are given a finite alphabet $\mathcal{A} = \mathcal{T} \cup \mathcal{N}$, which is the disjoint union of two subsets of letters: the (lower case) terminal symbols $t \in \mathcal{T}$ and the (upper case) nonterminal symbols $X \in \mathcal{N}$. We are also given a start symbol $S \in \mathcal{N}$ and a relation $R \subset (\text{Word}(\mathcal{A}) - \text{Word}(\mathcal{T})) \times \text{Word}(\mathcal{A})$. The production grammar $f_{\mathcal{T}, \mathcal{N}, S, R}$ is defined by the quadruple $(\mathcal{T}, \mathcal{N}, S, R)$ on words $x \in \text{Word}(\mathcal{A})$ as follows:

$$f_{\mathcal{T}, \mathcal{N}, S, R}(x) = \{x\}$$

if there is no $u \in \text{pr}_1(R), a, b \in \text{Word}(\mathcal{A})$ with $x = aub$

$$f_{\mathcal{T}, \mathcal{N}, S, R}(x) = \{y \mid \text{there are words } a, b \text{ and } (u, v) \in R$$

such that $x = aub$ and $y = avb\}$

otherwise.

The quadruple $(\mathcal{T}, \mathcal{N}, S, R)$, together with the production grammar it defines, is called a phrase structure grammar. In this context, the language $(S : \text{Word}(\mathcal{T}))$ is called the language generated by the phrase structure grammar $(\mathcal{T}, \mathcal{N}, S, R)$. If $y \in f_{\mathcal{T}, \mathcal{N}, S, R}(x)$, one also writes $x \rightarrow y$ and says that y is obtained from x by application of the rules R . This applies in particular if $(x, y) \in R$, then one says for $x \rightarrow y$ that x is the pattern for the replacement y . If $f(x)$ is a finite set $\{y_1, y_2, \dots, y_r\}$, then one also writes $x \rightarrow y_1 | y_2 | \dots | y_r$.

Remark 24 We should add a remark here concerning the question when two languages L and L' are identical. By definition, there are two alphabets \mathcal{A} and \mathcal{A}' such that $L \subset \text{Word}(\mathcal{A})$ and $L' \subset \text{Word}(\mathcal{A}')$. Saying that the sets L and L' are the same means that their elements coincide, i.e., the words in L and in L' are the same, and this means that they are the same sequences of letters from \mathcal{A} and \mathcal{A}' respectively. In other words, there is a common subset $\mathcal{A}'' \subset \mathcal{A} \cap \mathcal{A}'$ such that $L \subset \text{Word}(\mathcal{A}'')$ and $L' \subset \text{Word}(\mathcal{A}'')$ and that these subsets are equal. In the above definition of a language $(S : \text{Word}(\mathcal{T}))$ generated by a phrase grammar, this applies in the sense that neither the set of nonterminals nor the total set of terminals is relevant to the definition of $(S : \text{Word}(\mathcal{T}))$, it is only the set-theoretic identification which counts.

19.2.1 The Chomsky Hierarchy

We now discuss the four-fold hierarchy

$$\text{type 3} \subset \text{type 2} \subset \text{type 1} \subset \text{type 0}$$

of successively more comprising language types introduced by Noam Chomsky. In the following discussion of Chomsky types, we suppose that in a phrase structure grammar $(\mathcal{T}, \mathcal{N}, S, R)$, the sets \mathcal{T} , \mathcal{N} and R are all finite. To begin with, we look at the innermost set of languages, those of type 3.

Definition 134 *If for a phrase structure grammar $(\mathcal{T}, \mathcal{N}, S, R)$, every rule $x \rightarrow y$ in R has the shape $X \rightarrow Yt$ ($X \rightarrow tY$) or $X \rightarrow s$ for nonterminal letters X and Y and terminal letters s and t , the grammar is called left linear (right linear). If a rule $X \rightarrow \varepsilon$ is also admitted in a left linear (right linear) phrase structure grammar, it is called left (right) regular.*

Proposition 158 *For a language L the following four properties are equivalent:*

- (i) *There is a left linear phrase structure grammar which generates the language $L - \{\varepsilon\}$.*
- (ii) *There is a right linear phrase structure grammar which generates the language $L - \{\varepsilon\}$.*
- (iii) *There is a left regular phrase structure grammar which generates the language L .*
- (iv) *There is a right regular phrase structure grammar which generates the language L .*

Proof (i) implies (iii): If the left linear phrase structure grammar $G_l = (\mathcal{T}, \mathcal{N}, S, R)$ generates $L - \{\varepsilon\}$ (it cannot generate ε , by the nature of its rules). If $\varepsilon \in L$, then we add a new nonterminal element S_0 and the rules $S \rightarrow S_0$ and $S_0 \rightarrow \varepsilon$, and the new left regular phrase structure grammar does the job.

(iii) implies (i): If the left regular phrase structure grammar $(\mathcal{T}, \mathcal{N}, S, R)$ generates L , and $\varepsilon \in L$, then we successively reduce the number of nonterminals X which have the rule $X \rightarrow \varepsilon$ until they have disappeared. The crucial case is $X \rightarrow \varepsilon$, when this is the only rule for X on the left side. If we omit this rule, we not only prevent ε from being generated, but all the words stemming from a rule $Y \rightarrow xX$ followed by $X \rightarrow \varepsilon$ are in danger. Therefore, we have to add the rule $Y \rightarrow x$ for each $Y \rightarrow xX$, then we can omit $X \rightarrow \varepsilon$, and are done.

The proof for the right linear cases ((ii) iff (iv)) works the same way, therefore we omit it.

We are left with the equivalence of left linear and right linear generation of languages, i.e., (i) iff (ii). We show that (i) implies (ii), the converse follows by exchanging left and right. To begin with, we may choose a new start symbol S' and add to every rule $S \rightarrow ?$ a rule $S' \rightarrow ?$. Then the new grammar never has its start symbol on the right hand side of any rule and is of course equivalent to the original. So we may suppose wlog that S never appears on the right side of a rule. We then construct a right linear grammar $G_r = G_l^*$ which is equivalent to G_l . But we construct more: The rules of G_r are such that the same operator $*$, when applied to G_r , with left and right exchanged, yields $*G_r = G_l$.

This is the new rule set R^* , the alphabet being unaltered:

1. The rules $S \rightarrow t$ are left unchanged.
2. A rule $S \rightarrow Xs$ is replaced by the rule $X \rightarrow s$.
3. A rule $X \rightarrow s$ with $X \neq S$, is replaced by the rule $S \rightarrow sX$.
4. A rule $X \rightarrow Yt$ with $X \neq S$, is replaced by the rule $Y \rightarrow tX$.

We now show that $(S : R : \mathcal{T}) \subset (S : R^* : \mathcal{T})$. The converse is true by exchanging the roles of left and right and by the remark that the star operator, when applied to G_r gives us back G_l . The proof is by induction on the length n of a path $S \rightarrow Xs \rightarrow \dots w$ with $w \in (S : R : \mathcal{T})$. For $n = 1$ this is rule 1. If $S \rightarrow Xs \rightarrow \dots vu = w$ in G_l has length $n + 1$, where the path $Xs \rightarrow \dots vu$ has length n and stems from the length n path $X \rightarrow \dots v$, then we show that we have a path $S \rightarrow \dots vX \rightarrow vu$ in G_r . We show by induction on path length, that if $X \rightarrow \dots v$ in G_l has length m , then there is a path $S \rightarrow vX$ of length m in G_r . For $m = 1$, this is rule 3. In general, we have $X \rightarrow Yx \rightarrow \dots yx = v$, where the rule $X \rightarrow Yx$ in R is converted into the rule $Y \rightarrow xX$ in R^* according to rule 4 above. By induction hypothesis we now have this new path: $S \rightarrow \dots yY \rightarrow yxX = vX$, the first part being implied from $Y \rightarrow \dots y$ to the right of $X \rightarrow Yx \rightarrow \dots yx$, and we are done. \square

Definition 135 *A language which shares the equivalent properties of proposition 158 is called regular or of type 3.*

The crucial fact about type 3 languages is this:

Proposition 159 *If $L, L' \in \text{Lang}(\mathcal{A})$ are of type 3 (i.e., regular), then so are*

$$L \cup L', L \cap L', L^*, LL', \text{Word}(\mathcal{A}) - L.$$

Languages of type 3 are closed under all boolean operations as well as the Kleene operator and the product of languages.

Proof The proof idea is exemplified for the statement of $L \cup L'$ being of type 3 if L and L' are so. Take two phrase structure grammars $G = (\mathcal{T}, \mathcal{N}, S, R)$ and $G' = (\mathcal{T}', \mathcal{N}', S', R')$ which generate L and L' , respectively. It is clear that one may suppose that the nonterminal sets \mathcal{N} and \mathcal{N}' are disjoint. From the proof of proposition 158, we may also assume that the start symbols S and S' are never on the right side of a rule. But then we create a new set $\mathcal{N}^* = \mathcal{N} \cup \mathcal{N}' \cup \{S^*\}$ with a new start symbol S^* not in \mathcal{N} and \mathcal{N}' , while the old rules are inherited, except for the rules $S \rightarrow w, S' \rightarrow w'$, which we replace by the rules $S^* \rightarrow w, S^* \rightarrow w'$, and we are done. We refer to [43] for a complete proof. \square

Example 86 Let $\mathcal{T} = \{a, b, c\}$ and $\mathcal{N} = \{S, A, B, C\}$ and consider the language $L_1 = \{a^l b^m c^n \mid l > 0, m > 0, n > 0\}$. A right linear grammar for this language consists of the rules $R_1 = \{S \rightarrow aA, A \rightarrow aA, A \rightarrow bB, B \rightarrow bB, B \rightarrow cC, C \rightarrow c, C \rightarrow cC, C \rightarrow c\}$. A right *regular* grammar can be expressed with the (simpler) set of rules $R_1^R = \{S \rightarrow aA, A \rightarrow aA, A \rightarrow bB, B \rightarrow bB, B \rightarrow cC, C \rightarrow cC, C \rightarrow \varepsilon\}$. Note the different handling of the end of words. A regular grammar can also be found for the language $L_2 = \{a^l b^m c^n \mid l \geq 0, m \geq 0, n \geq 0\}$, with the rules $R_2 = \{S \rightarrow A, A \rightarrow aA, A \rightarrow B, B \rightarrow bB, B \rightarrow C, C \rightarrow cC, C \rightarrow \varepsilon\}$. For L_2 no *linear* grammar exists.

A *left* linear grammar for L_1 is given by the rules $R_1^L = \{S \rightarrow Bc, S \rightarrow Cc, C \rightarrow Cc, C \rightarrow Bc, B \rightarrow Bb, B \rightarrow Ab, A \rightarrow Aa, A \rightarrow a\}$. Using this grammar a derivation for the word $abcc$ is given as follows:

$$\begin{aligned} S &\rightarrow Cc \\ &\rightarrow Bcc \\ &\rightarrow Abcc \\ &\rightarrow abcc \end{aligned}$$

A useful property of languages of type 3 is embodied by the following lemma:

Lemma 160 (Type 3 Pumping Lemma) *Let $G = (\mathcal{T}, \mathcal{N}, S, R)$ be a linear grammar and L the language generated by G . Let $x \in L$ where $l(x) > \text{card}(\mathcal{N})$. Then there exist words $x, z, w \in \text{Lang}(\mathcal{T})$, $z \neq \varepsilon$, such that $x = yzw$ and $yz^k w \in L$ for $k = 0, 1, \dots$.*

Proof Consider a word x of length $|x| > |\mathcal{N}|$. Then a derivation consists of $|x|$ steps. Since the number of nonterminals is less than $|x|$, there must be a subderivation of length at most $|\mathcal{N}|$ that begins with a nonterminal, say A , and

ends with the same non-terminal A , e.g. $S \rightarrow \dots yA \rightarrow \dots yzA \rightarrow \dots yzw$. But the subderivation $A \rightarrow \dots zA$, can be substituted for the second A , thus yielding $yzzw$, and again, $yzzzw$, and so on. The subderivation can be left out entirely, yielding yw . \square

The languages of type 2 are the context free languages which may be used to describe programming languages, mainly in its wide-spread Backus-Naur form (BNF), and more standardized as augmented BNF (ABNF) or the extended BNF (EBNF) (standard ISO 14977); see 19.2.2 below for this type of grammars.

Definition 136 A phrase structure grammar $(\mathcal{T}, \mathcal{N}, S, R)$, with alphabet $\mathcal{A} = \mathcal{T} \cup \mathcal{N}$ is said to be

- (i) context free if its rules are all of shape $X \rightarrow x$ with $X \in \mathcal{N}$ and $x \in \text{Word}(\mathcal{A})$;
- (ii) reduced if it is context free and for each nonterminal A different from the start symbol S , there is a rule $A \rightarrow t$ with $t \in \mathcal{T}$ terminal, and for each nonterminal $A \in \mathcal{N}$, there is a rule $S \rightarrow vAw$ with $v, w \in \text{Word}(\mathcal{A})$;
- (iii) in Chomsky normal form if its rules are of shape $X \rightarrow t$ or $X \rightarrow AB$ for $X, A, B \in \mathcal{N}$ and $t \in \mathcal{T}$;
- (iv) in Greibach normal form if its rules are of shape $X \rightarrow xw$ with $X \in \mathcal{N}, x \in \mathcal{T}$, and $w \in \text{Word}(\mathcal{N})$.

Proposition 161 For a language L the following four properties are equivalent:

- (i) There is a context free phrase structure grammar which generates the language L .
- (ii) There is a reduced context free phrase structure grammar which generates the language L .
- (iii) There is a phrase structure grammar in Chomsky normal form which generates the language $L - \{\varepsilon\}$.
- (iv) There is a phrase structure grammar in Greibach normal form which generates the language $L - \{\varepsilon\}$.

Proof We have given a proof of proposition 158. The proof of this proposition is however too long for our context, therefore we refer to [43]. \square

Definition 137 A language which shares the equivalent properties of proposition 161 is called context free or of type 2.

By virtue of the first criterion in definition 136, a language of type 3 is evidently of type 2. The crucial fact about type 2 languages is this:

Proposition 162 *If $L, L' \in \text{Lang}(\mathcal{A})$ are of type 2 (i.e., context free), then so are*

$$L \cup L', L^* \text{ and } LL'.$$

Proof Again, we give an idea of the proof for the union $L \cup L'$. Take the definition 136, (i), for two type 2 languages L and L' . Let two phrase structure grammars $G = (\mathcal{T}, \mathcal{N}, S, R)$ and $G' = (\mathcal{T}', \mathcal{N}', S', R')$ generate languages L and L' , respectively. One may again assume that the sets \mathcal{N} and \mathcal{N}' of nonterminals are disjoint. Then just take the union of \mathcal{N} and \mathcal{N}' and add a new start symbol S^* , together with the two rules $S^* \rightarrow S$ and $S^* \rightarrow S'$, which solves the problem. We refer to [43] for a complete proof. \square

There exists also a pumping lemma for languages of type 2:

Lemma 163 (Type 2 Pumping Lemma) *Let L be context free. Then there exists n such that for every $x \in L$ with $l(x) \geq n$ there exist words u, v, y, z, w where $v \neq \varepsilon$ or $z \neq \varepsilon$ such that $x = uvyzw$ and $uv^k yz^k w \in L$ for $k = 0, 1, \dots$*

Proof See [28] for a proof. \square

Example 87 We can use a context free grammar to describe expressions of elementary arithmetic. Expressions of this type are common in the syntax of programming languages. Let $\mathcal{N} = \{S, E, F, T\}$ and take $\mathcal{T} = \{+, *, (,), x, y, z\}$ where the letters x, y and z denote variables in the programming language. The rules are given by $R = \{E \rightarrow T + E, E \rightarrow T, T \rightarrow F * T, T \rightarrow F, F \rightarrow (E), F \rightarrow x | y | z\}$.

A derivation of the expression $x + y * (z + y)$ is given in figure 19.4.

Note how the rules model the usual precedence rules of the operators $+$ and $*$. This is of great practical value when implementing a parser for an actual programming language.

Figure 19.5 shows the derivation in form of a *syntax tree*. Each node of the tree is an application of a rule, the resulting expression can be read off the leaves of the tree in left-to-right order.

Exercise 88 Not every language of type 2 is of type 3, i.e., the inclusion type 3 \subset type 2 is proper. Construct a context free grammar for the language $L = \{a^n b^n \mid n \geq 1\}$ over $\mathcal{A} = \{a, b\}$, then use the pumping lemma for type 3 languages to show that there is no regular grammar for L .

$$\begin{aligned}
E &\rightarrow T + E \\
&\rightarrow F + E \\
&\rightarrow x + E \\
&\rightarrow x + T \\
&\rightarrow x + F * T \\
&\rightarrow x + y * T \\
&\rightarrow x + y * F \\
&\rightarrow x + y * (E) \\
&\rightarrow x + y * (T + E) \\
&\rightarrow x + y * (F + E) \\
&\rightarrow x + y * (z + E) \\
&\rightarrow x + y * (z + T) \\
&\rightarrow x + y * (z + F) \\
&\rightarrow x + y * (z + y)
\end{aligned}$$

Fig. 19.4. A derivation of the expression $x + y * (z + y)$

Definition 138 A phrase structure grammar $(\mathcal{T}, \mathcal{N}, S, R)$, with alphabet $\mathcal{A} = \mathcal{T} \cup \mathcal{N}$ is said to be context sensitive iff for every rule $x \rightarrow y$ in R , we have $l(x) \leq l(y)$. A language is called context sensitive or of type 1 iff it is generated by a context sensitive phrase structure grammar.

Evidently, the characterization of context free languages by Chomsky normal form grammars implies that type 2 is a subset of type 1. The crucial fact about type 1 languages is this:

Proposition 164 If $L \in \text{Lang}(\mathcal{A})$ are of type 1 (i.e., context sensitive), then so is its complement language $\text{Word}(\mathcal{A}) - L$.

Proof We refer to [43] for a proof. □

Example 88 Let $\mathcal{T} = \{a, b, c\}$ and $\mathcal{N} = \{S, A, B, C, D, E\}$. The language $L = \{a^k b^k c^k \mid k > 0\}$ is of type 1. The rather complicated grammar is given by the following set of rules: $R = \{S \rightarrow Abc, A \rightarrow a, A \rightarrow aB, B \rightarrow aC, Cb \rightarrow bC, Cc \rightarrow Dc, D \rightarrow bc, D \rightarrow Ebc, bE \rightarrow Eb, aE \rightarrow aB\}$. Let us see how this works on the example of the word $aaabbbccc$:

$S \rightarrow Abc$
 $\rightarrow aBbc$
 $\rightarrow aaCbc$
 $\rightarrow aabCc$
 $\rightarrow aabDc$
 $\rightarrow aabEbcc$
 $\rightarrow aaEbbcc$
 $\rightarrow aaBbbcc$
 $\rightarrow aaaCbbcc$
 $\rightarrow aaabCbcc$
 $\rightarrow aaabbCcc$
 $\rightarrow aaabbDcc$
 $\rightarrow aaabbbccc$

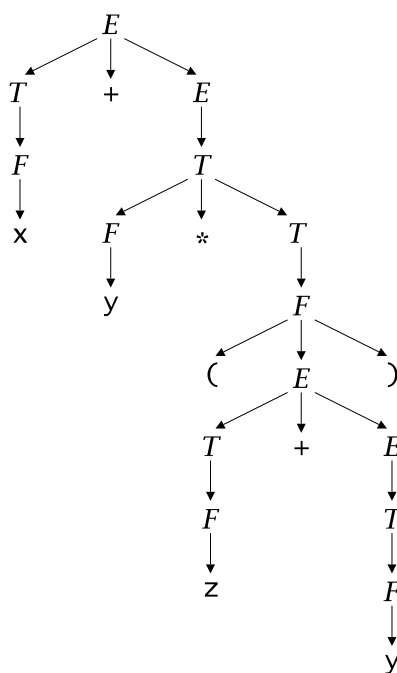


Fig. 19.5. The syntax tree for $x + y * (z + y)$.

We have yet to prove that there is no grammar of type 2 generating this language. We do this by invoking the type 2 pumping lemma 163. Suppose that L is context free. The lemma assures, that there is a number n such that the properties of the lemma will be fulfilled for words of length $\geq n$. Let us choose the word $a^n b^n c^n \in L$ which is certainly longer than n . The lemma tells us that this word must have a structure $uvyzw$ such that $uv^k yz^k w \in L$ for $k = 0, 1, \dots$. But however we choose two subwords in $a^n b^n c^n$, the resulting “pumped-up” word will *not* be in L . Either the equal number of a s, b s and c s will not be maintained, or the order of the letters will not be respected, as can be easily checked. Thus L cannot be context free.

Exercise 89 Prove, by finding a counterexample, that the intersection of two context free languages need not be context free.

The last type 0 is that of completely general phrase structure grammars:

Definition 139 A phrase structure grammar $(\mathcal{T}, \mathcal{N}, S, R)$, with alphabet $\mathcal{A} = \mathcal{T} \cup \mathcal{N}$ is called

- (i) general if there are no further conditions,
- (ii) separated if each of its rules $x \rightarrow y$ has one of the following shapes:
 - a) $x \in \text{Word}(\mathcal{N}) - \{\varepsilon\}$ with $y \in \text{Word}(\mathcal{N})$,
 - b) $x \in \mathcal{N}$ with $y \in \mathcal{T}$, or
 - c) $x \in \mathcal{N}$ with $y = \varepsilon$,
- (iii) normal if each of its rules $x \rightarrow y$ has one of the following shapes:
 - a) $x \in \mathcal{N}$ with $y \in \mathcal{T}$,
 - b) $x \in \mathcal{N}$ with $y = \varepsilon$,
 - c) $x \in \mathcal{N}$ with $y \in \mathcal{N}^2$, or
 - d) $x, y \in \mathcal{N}^2$.

It should be stressed that the definition (i) “general phrase structure grammar” is redundant, but has been used in the computer community as a synonym of “phrase structure grammar”. So the following proposition effectively is a statement about phrase structure grammars without any further attribute.

Proposition 165 For a language L the following four properties are equivalent:

- (i) *There is a (general) phrase structure grammar generating L .*
- (ii) *There is a separated phrase structure grammar generating L .*
- (iii) *There is a normal phrase structure grammar generating L .*

Proof We refer to [43] for a proof. □

Definition 140 *A language L which shares the equivalent properties of proposition 165 is called recursively enumerable or of type 0.*

19.2.2 Backus-Naur Normal Forms

The syntax of most programming languages, e.g., Algol, Pascal, C or Java, can be described by context free grammars. Originally, BNF was used by Peter Naur for the description of Algol 60 in an adaptation of a notation developed by John Backus.

The idea was to set up a standardized formal procedure to create terminal and nonterminal symbols and to describe the rules. Recall that the rules in context free grammars have the particular form $X \rightarrow w$, where $w \in \text{Word}(\mathcal{A})$ and X is a nonterminal symbol. To begin with, the arrow “ \rightarrow ” in a rule is replaced by the sign “ $::=$ ” derived from the mathematical symbol “ $:=$ ” meaning that x is defined by y in the expression $x := y$. The alternative $x \rightarrow y_1|y_2|\dots|y_n$ is written analogously, i.e., by $x ::= y_1|y_2|\dots|y_n$.

The more important contribution of BNF is that the terminal and nonterminal symbols are provided by a standard construction from a given character set CH , in the ASCII encoding, say, i.e., “ $CH = \text{ASCII}$ ”. The procedure is very simple: Terminals are just single characters from CH . Nonterminals are all words of shape $\langle w \rangle$, $w \in \text{Word}(CH)$. The start symbol is mostly clear from the given rule system. For example, the Algol 60 specification of a floating point constant is called “ $\langle \text{unsigned number} \rangle$ ”, and this is the start symbol, which defines the sublanguage of floating-point constants as follows:

```

<unsigned integer> ::= <digit> | <unsigned integer> <digit>
<integer> ::= <unsigned integer> | + <unsigned integer> |
             - <unsigned integer>
<decimal fraction> ::= . <unsigned integer>
<exponent part> ::= _10_ <integer>
<decimal number> ::= <unsigned integer> | <decimal fraction> |

```

```

        <unsigned integer> <decimal fraction>
<unsigned number> ::= <decimal number> | <exponent part> |
        <decimal number> <exponent part>
<digit> ::= 0|1|2|3|4|5|6|7|8|9

```

Here, the start symbol is $S = \langle \text{unsigned number} \rangle$. In comparison, the Extended BNF notation (EBNF) of the same grammar is this:

```

unsigned integer = digit | unsigned integer, digit;
integer = unsigned integer | "+", unsigned integer |
        "-", unsigned integer;
decimal fraction = ".", unsigned integer;
exponent part = "_10_", integer;
decimal number = unsigned integer | decimal fraction |
        unsigned integer, decimal fraction;
unsigned number = decimal number | exponent part |
        decimal number, exponent part;
digit = "0"|"1"|"2"|"3"|"4"|"5"|"6"|"7"|"8"|"9";

```

The EBNF notation also includes some extensions to BNF, which improve readability and conciseness, e.g., the Kleene cross for a sequence of one or more elements of the class so marked, for example

```

unsigned integer = digit+;

```

The changes are evident, the idea behind this standardization is clear and is usually learned by doing some examples.³

It is also customary to represent a BNF grammar by use of syntax diagrams, i.e., groups of flow charts, where the alternatives in a rule $X ::= \gamma_1 | \gamma_2 | \dots | \gamma_n$ are flow ramifications starting from the block of X and terminating at the leftmost symbol in the target replacement γ_i . Here, the leaves are the terminal symbols. Figure 19.6 is a syntax diagram; such diagrams were used for the first time by Jensen and Wirth.

Exercise 90 Define (the fragments of) an alphabet and write the BNF rules corresponding to the flow charts shown in figure 19.6.

³ See <http://www.cl.cam.ac.uk/~mgk25/iso-14977-paper.pdf> for a more complete description of the BNF standard.

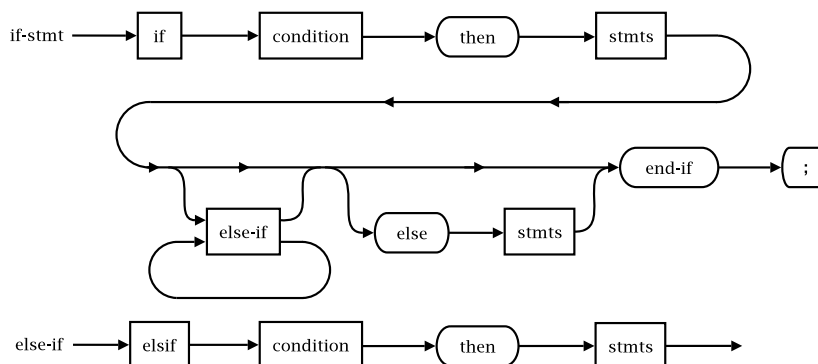


Fig. 19.6. A syntax diagram as used by Kathleen Jensen and Niklaus Wirth in [29].

19.3 Automata and Acceptors

In this section, we shall establish a systematic relation between phrase structure languages and abstract machines, as they are axiomatically described by sequential machines, automata and acceptors. It will turn out that the languages of different Chomsky types are precisely those which may be defined by specific types of machines, such as Turing machines, for instance.

We have defined sequential machines in a preliminary context earlier. Now it is time to give the full-fledged setup of those concepts.

Definition 141 Given a finite alphabet set \mathcal{A} , a finite set S of “states”, and an “initial” state $i \in S$ an automaton over \mathcal{A} with initial state i is a pair (M, i) where M is a set map

$$M : 2^S \times \mathcal{A} \rightarrow 2^S$$

such that M commutes with unions, i.e., for all families $(U_i)_{i \in I}$ of sets of states $U_i \in 2^S$, and for all $a \in \mathcal{A}$, we have $M(\bigcup_i U_i, a) = \bigcup_i M(U_i, a)$; if M is clear from the context, one writes $U \cdot a$ instead of $M(U, a)$. In particular, we always have $\emptyset \cdot a = \emptyset$. The map M is completely determined by its values on singletons $\{s\} \in 2^S$, i.e., on single states s . As with production grammars, we therefore also write $M(\{s\}, a) = s \cdot a$. The corresponding map $S \times \mathcal{A} \rightarrow 2^S$ (now without any further properties), is also denoted by

M and serves as an alternate definition for an automaton, much as this was done for production grammars.

An automaton is called *deterministic* if its images $s \cdot a$ on singletons are always singletons $s \cdot a = \{x\}$ (attention: in particular, the images $s \cdot a$ are never empty!). We then also write $s \cdot a = x$, and correspondingly $M : S \times \mathcal{A} \rightarrow S$. A *nondeterministic automaton* is one which is not deterministic.

The elementary graph of an automaton is the digraph $\Gamma_M : \text{Arr}(M) \rightarrow S^2$ the arrow set of which is $\text{Arr}(M) = \{(s, a, x) \mid a \in \mathcal{A}, s \in S, x \in s \cdot a\}$ with $\Gamma_M((s, a, x)) = (s, x)$. The initial state i is rephrased as a morphism of digraphs $i : 1 \rightarrow \Gamma_M$ pointing to the vertex i . A path

$$p = s_1 \xrightarrow{(s_1, a_1, s_2)} s_2 \xrightarrow{(s_2, a_2, s_3)} s_3 \dots \xrightarrow{(s_{m-1}, a_{m-1}, s_m)} s_m$$

in Γ_M starting at the initial state i (i.e., with $s_1 = i$) is called a *state sequence* of the automaton associated with the word $W_p = a_1 a_2 \dots a_m$. The *lazy path* is associated with the empty word.

Evidently, every automaton (M, i) determines its alphabet \mathcal{A} and state set S , so we do not need to mention them explicitly.

Any automaton defines the *associated power automaton* (M', i') , which, evidently, is always deterministic, with these data: We replace S by $S' = 2^S$ and i by $i' = \{i\}$. Then the same map $M' = M$ defines an automaton, but we take the alternate definition $M' : S' \times \mathcal{A} \rightarrow S'$. this time! Using this definition we don't need to deal with 2^{2^S} , as would have been required by the first definition. This is nothing more than a trick of switching from the first definition to the second one. The point is that, however, not every deterministic automaton is of this type, since its state set need not be a powerset, and the commutation with unions need not work.

Although the elementary graph of an automaton is customary in computer science, it has some serious structural drawbacks which enforce a second digraph associated with an automaton: the *power graph* Γ^M of an automaton (i, M) is the elementary graph of the associated power automaton, $\Gamma^M = \Gamma_{M'}$. Since the power automaton is a priori deterministic, the power graph may be described more economically (it is in fact the Moore graph of the underlying sequential machine, check this out) as follows: vertexes are the sets of states, its arrows are the pairs $(s, a) \in 2^S \times \mathcal{A}$, which are mapped to the state set pairs $(s, s \cdot a)$, and the initial state pointer is $i' : 1 \rightarrow \Gamma^M$.

As to the graph representation $i : 1 \rightarrow \Gamma_M$ of the automaton (M, i) , the representation of the associated deterministic automaton $i' : 1 \rightarrow \Gamma_{M'}$ contains the entire original information, whereas the original graph $i : 1 \rightarrow \Gamma_M$ need not in case where each set $s \cdot a$ is empty.

Definition 142 An acceptor is a triple (M, i, F) , where $(M : S \times \mathcal{A} \rightarrow 2^S, i)$ is an automaton and where $F \subset S$ is a subset of “terminal” or “accepting” states. By definition, the language $(i : M : F)$, or $(i : F)$ if M is clear, accepted by the acceptor (M, i, F) is the set of words W_p associated with state sequences p , which start at i and terminate in an element of F . In particular, the empty word W_i associated with the lazy path at i is accepted iff $i \in F$. If the automaton is given by the first definition, i.e., $(M : 2^S \times \mathcal{A} \rightarrow 2^S, i)$, then a word W_p is accepted iff its path p starts in $\{i\}$ and ends in a set s_n of states such that $s_n \cap F \neq \emptyset$. Two acceptors are called equivalent if they accept the same language.

Observe that an automaton (M, i) can be identified with the acceptor (M, i, \emptyset) . This is why we mostly talk about acceptors from now on, thereby including automata as special cases.

Proposition 166 Every acceptor $(M : 2^S \times \mathcal{A} \rightarrow 2^S, i, F)$ is equivalent to the deterministic power acceptor (M', i', F') with $M' = M, i' = \{i\}, F' = \{s \in 2^S \mid s \cap F \neq \emptyset\}$.

Exercise 91 The proof is left as an exercise.

This proposition ensures that one can build an equivalent deterministic acceptor from a nondeterministic acceptor. This means that there is no fundamental difference between deterministic and nondeterministic automata.

Like automata, acceptors (M, i, F) are also represented in the form of digraphs. The *elementary graph of an acceptor* is the elementary graph Γ_M of the underlying automaton (M, i) , together with the initial state pointer $i : 1 \rightarrow \Gamma_M$, and with the set $f : 1 \rightarrow \Gamma_M$ of digraph morphisms pointing to the elements $f \in F$. Also, the *power graph of an acceptor* (M, i, F) is the power graph Γ^M of the underlying automaton, together with the initial pointer $i' : 1 \rightarrow \Gamma^M$ and the final set pointer $F : 1 \rightarrow \Gamma^M$ pointing to the element $F \in 2^S$.

Example 89 Figure 19.7 shows the elementary graph for an acceptor with states $S = \{A, B, C, D\}$ and alphabet $\{a, b, c, d\}$. Its initial state $i = A$ is drawn as a square, the terminal states, given by $F = \{C, D\}$, as double circles. This acceptor is nondeterministic: from state A for example there are *two* transitions for the letter a .

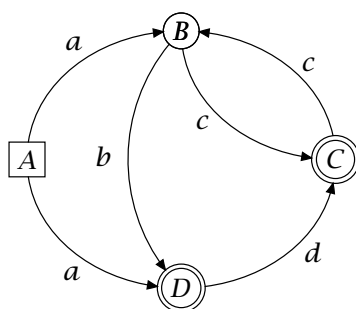


Fig. 19.7. The elementary graph for a nondeterministic acceptor.

	a	b	c	d
{A}	{B, D}	\emptyset	\emptyset	\emptyset
{B}	\emptyset	{D}	{C}	\emptyset
{C}	\emptyset	\emptyset	{B}	\emptyset
{D}	\emptyset	\emptyset	\emptyset	{C}
{A, B}	{B, D}	{D}	{C}	\emptyset
{A, C}	{B, D}	\emptyset	{B, C}	\emptyset
{A, D}	{B, D}	\emptyset	\emptyset	{C}
{B, C}	\emptyset	{D}	{B, C}	\emptyset
{B, D}	\emptyset	{D}	{C}	{C}
{C, D}	\emptyset	\emptyset	{B}	{C}
{A, B, C}	{B, D}	{D}	{B, C}	\emptyset
{B, C, D}	\emptyset	{D}	{B, C}	{C}
{A, C, D}	{B, D}	\emptyset	{B}	{C}
{A, B, D}	{B, D}	{D}	{C}	{C}
{A, B, C, D}	{B, D}	{D}	{B, C}	{C}
\emptyset	\emptyset	\emptyset	\emptyset	\emptyset

Fig. 19.8. The combined states of the power graph.

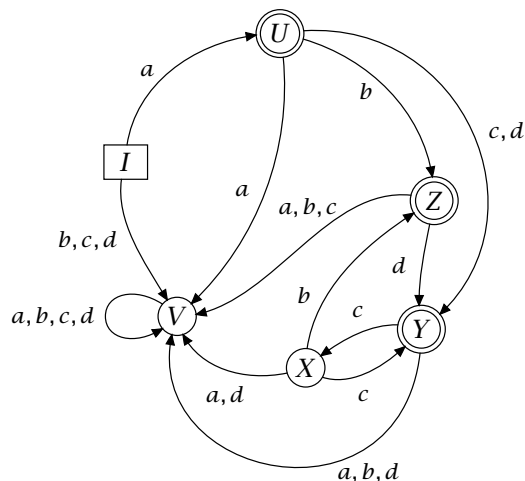


Fig. 19.9. The power graph for the nondeterministic acceptor of figure 19.7. Note that non-accessible states have been removed.

The associated power graph is shown in Figure 19.9. To compute its transition function, it is best to draw up a table of the combined states, as in figure 19.8.

By inspection we see that the only states reachable from the initial state $I = \{A\}$ are $X = \{B\}$, $Y = \{C\}$, $Z = \{D\}$, $U = \{B, D\}$ and $V = \emptyset$, where we have renamed the combined states for easier reference. The new terminal states are $F' = \{U, Y, Z\}$.

Clearly, one is not interested in acceptors of a given language which have superfluous ingredients, for example too many letters a which are never used in that language because their output $s \cdot a$ is always empty, or else states which do not contribute to the language. So we need to compare acceptors and to construct new ones from given ones. This leads to the concept of a morphism of automata and acceptors. To this end we use the following general construction on set maps known from set theory: If $f : X \rightarrow Y$ is a set map, then we have an associated set map $2^f : 2^X \rightarrow 2^Y : U \mapsto f(U)$, moreover, if $g : Y \rightarrow Z$ is a second map, then $2^{g \circ f} = 2^g \circ 2^f$.

Definition 143 If $(i, M : S \times \mathcal{A} \rightarrow 2^S)$ and $(j, N : T \times \mathcal{B} \rightarrow 2^T)$ are two automata, a morphism of automata $(\sigma, \alpha) : (i, M) \rightarrow (j, N)$ is a pair of set maps $\sigma : S \rightarrow T$ and $\alpha : \mathcal{A} \rightarrow \mathcal{B}$ such that

- (i) initial states are conserved: $\sigma(i) = j$, and
(ii) for any $(s, a) \in S \times \mathcal{A}$, we have $2^\sigma(s \cdot a) = \sigma(s) \cdot \alpha(a)$, where the products must be taken in the respective automata, i.e., one has the following commutative diagram

$$\begin{array}{ccc} S \times \mathcal{A} & \xrightarrow{M} & 2^S \\ \sigma \times \alpha \downarrow & & \downarrow 2^\sigma \\ T \times \mathcal{B} & \xrightarrow{N} & 2^T \end{array}$$

If (i, M, F) and (j, N, G) are two acceptors, a morphism of the underlying automata $(\sigma, \alpha) : (i, M) \rightarrow (j, N)$ is a morphism of acceptors iff the terminal sets are also respected, i.e., if $\sigma(F) \subset G$.

Let $(\sigma, \alpha) : (i, M : S \times \mathcal{A} \rightarrow 2^S) \rightarrow (j, N : T \times \mathcal{B} \rightarrow 2^T)$ and $(\tau, \beta) : (j, N : T \times \mathcal{B} \rightarrow 2^T) \rightarrow (k, O : U \times \mathcal{C} \rightarrow 2^U)$ be two morphisms of automata, then their composition $(\tau, \beta) \circ (\sigma, \alpha)$ is defined by $(\tau, \beta) \circ (\sigma, \alpha) = (\tau \circ \sigma, \beta \circ \alpha)$. The same definition is given for the composition of morphisms of acceptors.

For an automaton $(i, M : S \times \mathcal{A} \rightarrow 2^S)$, denote by $Id_{(i, M)}$ the morphism $(Id_S, Id_{\mathcal{A}})$. The notation for the identity of acceptors is correspondingly $Id_{(i, M, F)}$. The morphism (σ, α) is called an isomorphism of automata/acceptors iff there is a morphism $(\tau, \beta) : (j, N) \rightarrow (i, M)$ such that $(\tau, \beta) \circ (\sigma, \alpha) = (Id_S, Id_{\mathcal{A}})$ and $(\sigma, \alpha) \circ (\tau, \beta) = (Id_T, Id_{\mathcal{B}})$.

Sorte 167 Morphisms of automata/acceptors have these standard properties:

- (i) (Associativity) Whenever the composition $(\mu, \gamma) \circ ((\tau, \beta) \circ (\sigma, \alpha))$ is defined, it is equal to $((\mu, \gamma) \circ (\tau, \beta)) \circ (\sigma, \alpha)$ and is denoted by $(\mu, \gamma) \circ (\tau, \beta) \circ (\sigma, \alpha)$.
(ii) (Identity) For any morphism $(\sigma, \alpha) : (i, M) \rightarrow (j, N)$ of automata, or acceptors, the identities $(Id_S, Id_{\mathcal{A}})$ and $(Id_T, Id_{\mathcal{B}})$, are right, respectively left, neutral, i.e.,

$$(Id_T, Id_{\mathcal{B}}) \circ (\sigma, \alpha) = (\sigma, \alpha) = (\sigma, \alpha) \circ (Id_S, Id_{\mathcal{A}}).$$

- (iii) (Isomorphisms) A morphism (σ, α) is iso iff σ and α are both bijections of sets.

Proof This is an easy exercise left to the reader. □

Exercise 92 Let $(\sigma, \alpha) : (i, M : S \times \mathcal{A} \rightarrow 2^S) \rightarrow (j, N : T \times \mathcal{B} \rightarrow 2^T)$ be a morphism of automata, show that the map $(s, a, x) \mapsto (\sigma(s), \alpha(a), \sigma(x))$ on arrows and σ on vertexes defines a morphism $\Gamma_{(\sigma, \alpha)} : \Gamma_M \rightarrow \Gamma_N$ which also maps the initial pointers into each other. Also, if we have sets of final states and therefore acceptors, the corresponding pointers to final states are preserved. Show that on power graphs Γ^M and Γ^N we have a corresponding morphism $\Gamma^{(\sigma, \alpha)} : \Gamma^M \rightarrow \Gamma^N$ defined by $2^\sigma \times \alpha$ on arrows and 2^σ on vertexes. Show that for two morphisms (τ, β) and (σ, α) which can be composed to $(\tau, \beta) \circ (\sigma, \alpha)$, we have $\Gamma_{(\tau, \beta) \circ (\sigma, \alpha)} = \Gamma_{(\tau, \beta)} \circ \Gamma_{(\sigma, \alpha)}$ and $\Gamma^{(\tau, \beta) \circ (\sigma, \alpha)} = \Gamma^{(\tau, \beta)} \circ \Gamma^{(\sigma, \alpha)}$. This kind of passage from one type of objects (automata) to another type (digraphs), which also preserves morphisms and their composition, is called *functorial* and will be discussed extensively in the second volume of this book dedicated to so-called *categories*. Categories are a fundamental subject in modern mathematics and are becoming more and more important in computer science.

Proposition 168 *If $(\sigma : S \rightarrow T, \alpha : \mathcal{A} \rightarrow \mathcal{B})$ is a morphism of acceptors $(\sigma, \alpha) : (i, M, F) \rightarrow (j, N, G)$, then the induced homomorphism of the language set $\text{Word}(\alpha) : \text{Word}(\mathcal{A}) \rightarrow \text{Word}(\mathcal{B})$ maps $(i : M : F)$ into $(j : N : G)$. If in particular (σ, α) is an isomorphism, then $\text{Word}(\alpha)$ induces a bijection $(i : M : F) \xrightarrow{\sim} (j : N : G)$. More specifically, if also $\alpha = \text{Id}_{\mathcal{A}}$, then $(i : M : F) = (j : N : G)$.*

Exercise 93 Use the elementary graph of an acceptor and the morphism $\Gamma_{(\sigma, \alpha)}$ to give a proof of proposition 168.

Definition 144 *If $(\sigma, \alpha) : (i, M, F) \rightarrow (j, N, G)$ is a morphism of acceptors over the alphabets \mathcal{A} and \mathcal{B} respectively, such that $\alpha : \mathcal{A} \hookrightarrow \mathcal{B}$ and $\sigma : S \hookrightarrow T$ are subset inclusions, then we say that (i, M, F) is a subacceptor of (j, N, G) .*

Corollary 169 *If (i, M, F) is a subacceptor of (j, N, G) , then $(i : M : F)$ is a sublanguage of $(j : N : G)$.*

Proof This follows immediately from proposition 168. □

Definition 145 *An acceptor (i, M, F) is simple iff every state $s \in S$ is a vertex of a state sequence p from i to F .*

Proposition 170 *Every acceptor has an equivalent simple subacceptor.*

Proof Since the states not appearing in any state sequence have no meaning for the generated words, one obtains the same language when omitting those states. \square

Besides the simplification procedure for an acceptor, we may also need to look for subacceptors which are present in multiple “copies”, i.e., which play the same role with respect to the language they accept. We now want to eliminate such multiplicities, since it is not reasonable to have machines with equivalent functional units in multiple instantiations.

Definition 146 *If M is a sequential machine and $F \subset S$ a set of final states, then two states $s, t \in S$ are called equivalent if $(s : M : F) = (t : M : F)$. If for an acceptor (i, M, F) any two different states $s \neq t$ on state sequences from i to F are not equivalent, the acceptor is called reduced.*

We now discuss the construction of a reduced acceptor from a given deterministic acceptor $(i, M : S \times \mathcal{A} \rightarrow S, F)$. To begin with, we need a generic sequential machine associated with the alphabet \mathcal{A} .

Definition 147 *For the alphabet \mathcal{A} the sequential machine $LangMachine_{\mathcal{A}}$ of \mathcal{A} is defined by the map*

$$LangMachine_{\mathcal{A}} : Lang(\mathcal{A}) \times \mathcal{A} \rightarrow Lang(\mathcal{A})$$

$$(L, a) \mapsto L/a = \{x \in Word(\mathcal{A}) \mid a \cdot x \in L\}.$$

If (i, M, F) is an acceptor, the associated generic acceptor is defined by $(i_{\mathcal{A}}, LangMachine_{\mathcal{A}}, F_{\mathcal{A}})$, where

- (i) $i_{\mathcal{A}} = (i : M : F)$
- (ii) and $F_{\mathcal{A}} = \{(f : M : F) \mid f \in F\}$.

Exercise 94 *If one defines more generally $L/w = \{x \in Word(\mathcal{A}) \mid w \cdot x \in L\}$ for $w \in Word(\mathcal{A})$, then $v, w \in Word(\mathcal{A})$ implies $(L/v)/w = L/(vw)$.*

Proposition 171 *For a deterministic acceptor $(i, M : S \times \mathcal{A} \rightarrow S, F)$, consider the morphism*

$$(\sigma, Id_{\mathcal{A}}) : (i, M, F) \rightarrow (i_{\mathcal{A}}, LangMachine_{\mathcal{A}}, F_{\mathcal{A}})$$

of acceptors given by $\sigma(s) = (s : M : F)$. Then the image $(i, M, F)_{\mathcal{A}}$ of $(\sigma, Id_{\mathcal{A}})$ is an equivalent reduced deterministic acceptor, more precisely, for each state $s \in S$ we have

$$(s : M : F) = ((s : M : F) : LangMachine_{\mathcal{A}} : F_{\mathcal{A}}).$$

- (i) If (i, M, F) is simple, then so is $(i, M, F)_{\mathcal{A}}$.
- (ii) If (i, M, F) is reduced, then $(\sigma, Id_{\mathcal{A}}) : (i, M, F) \rightarrow (i, M, F)_{\mathcal{A}}$ is an isomorphism.
- (iii) If (i, M, F) and (j, N, G) are reduced, simple, and equivalent, then $(i, M, F)_{\mathcal{A}} = (j, N, G)_{\mathcal{A}}$.

Proof To begin with, we have to check whether $(\sigma, Id_{\mathcal{A}})$ is a morphism. This readily follows from the fact that for a pair $(s, a) \in S \times \mathcal{A}$, we have $(s \cdot a, M, F) = (s, M, F)/a$. To show that

$$(s : M : F) = ((s : M : F) : LangMachine_{\mathcal{A}} : F_{\mathcal{A}}),$$

let $w \in (s : M : F)$. Then $s \cdot w \in F$. Therefore, $(s : M : F) \cdot w = (s \cdot w : M : F) \in F_{\mathcal{A}}$, whence $w \in ((s : M : F) : LangMachine_{\mathcal{A}} : F_{\mathcal{A}})$, i.e., $(s : M : F) \subset ((s : M : F) : LangMachine_{\mathcal{A}} : F_{\mathcal{A}})$. Conversely, if $w \in ((s : M : F) : LangMachine_{\mathcal{A}} : F_{\mathcal{A}})$, then we have $(s : M : F)/w = (f : M : F)$, $f \in F$. In other words, $s \cdot w \cdot v \in F$ iff $f \cdot v \in F$, for any word v . In particular, $v = \varepsilon$, the empty word gives us $f \cdot \varepsilon = f \in F$ whence $s \cdot w \cdot \varepsilon = s \cdot w \in F$, so $w \in (s : M : F)$, whence $(s : M : F) \supset ((s : M : F) : LangMachine_{\mathcal{A}} : F_{\mathcal{A}})$, and equality holds.

The new acceptor is deterministic by construction, and it is reduced since the language we obtain in the image $((s : M : F) : LangMachine_{\mathcal{A}} : F_{\mathcal{A}})$ is exactly the starting state $(s : M : F)$, so no two different starting states can produce the same language. If $(s : M : F)$ is simple, every state t is visited by a state sequence starting at s in a word w . But then the state $(s \cdot w, M, F)$ is reached by w from the starting point (s, M, F) , whence claim (i). As to claim (ii), observe that the fiber of a state (s, M, F) is the set of states t which generate the same language (t, M, F) , i.e., the equivalent states of the (possibly) not reduced acceptor. Therefore, if all fibers are singletons, the map on states is a bijection, and we are done. To prove (iii), observe that the initial states $(i : M : F)$ and $(j, N : G)$ are equal by hypothesis. But the composition rule is the same in $LangMachine_{\mathcal{A}}$, so the terminal states of these acceptors are the same, i.e., the images of the common initial state under the common language. \square

Definition 148 An acceptor over the alphabet \mathcal{A} which has a minimal number of states for a given language $L \in Lang(\mathcal{A})$ is called minimal.

Corollary 172 (Theorem of Myhill-Nerode) Any two minimal acceptors (i, M, F) and (j, N, G) of a given language $L \in Lang(\mathcal{A})$ are isomorphic. In fact, we have $(i, M, F)_{\mathcal{A}} = (j, N, G)_{\mathcal{A}}$.

Proof In fact, a minimal acceptor is reduced and simple, whence the claim by proposition 171. \square

We now have a central theorem which relates acceptors and languages:

Proposition 173 Let $L \in \text{Lang}(\mathcal{A})$ be a language over the alphabet \mathcal{A} . Then the following statements are equivalent.

- (i) The language L is of Chomsky type 3, i.e., regular.
- (ii) There is a minimal (and therefore deterministic, reduced, simple) acceptor (i, M, F) such that $L = (i : M : F)$.
- (iii) There is an acceptor (i, M, F) such that $L = (i : M : F)$.

Proof The equivalence of statements (ii) and (iii) is evident from the above theory. For the equivalence of (i) and (iii), see [43]. \square

19.3.1 Stack Acceptors

Stack acceptors are a special type of acceptors. The characteristic property is that their state space is not finite and is composed in a specific way of three kinds of entities: a (finite) set S of elementary states, a (finite) elementary alphabet \mathcal{A} and a stack alphabet K . The practical relevance of a stack is illustrated by the typical example: Consider a pile of plates in a cafeteria. Each time when a plate is taken away (the so-called *pop* action) from the top of the stack, a spring moves the stack of the remaining plates upwards in order to make available a new plate to the service personnel. When, on the contrary, a clean plate is put onto the existing stack (the so-called *push* action), the augmented stack moves down one unit to receive the new top plate.⁴ The theoretical relevance of such stack acceptors consists in the fact that the languages which are accepted by this type of acceptors are precisely the context free, or type 2, languages.

In the context of stack automata, one starts from three finite sets S, \mathcal{A}, K , the elements of which are called the *elementary states, input letters, and stack elements*, respectively. We then consider the Cartesian product $\text{Word}(S, \mathcal{A}, K) = \text{Word}(S) \times \text{Word}(\mathcal{A}) \times \text{Word}(K)$ of word monoids which is a monoid by factorwise multiplication, i.e., $(u, v, w) \cdot (x, y, z) = (ux, vy, wz)$. If $X \subset \text{Word}(S, \mathcal{A}, K)$ and $x \in \text{Word}(S, \mathcal{A}, K)$, we write X/x for the set of elements y such that $x \cdot y \in X$. This is a construction we already encountered in the theory of generic acceptors. We further need the set $\mathcal{A}_\varepsilon = \mathcal{A} \cup \{\varepsilon\} \subset \text{Word}(\mathcal{A})$, ε being the neutral (empty) word. This is all we need to describe stack acceptors. Observe that as acceptors, stack automata are deterministic. But attention: in theoretical computer

⁴ In computer science the behavior is called LIFO which stands for “Last In, First Out.”

science a slightly different terminology is customary, as will be explained below.

Definition 149 *Given three sets S, \mathcal{A}, K of elementary states, input letters, and stack elements, respectively, a stack acceptor over S, \mathcal{A}, K (also: push down acceptor) consists of*

- (i) *the state space $2^{\mathfrak{S}}$ for the set of configurations $\mathfrak{S} = S \times \text{Word}(\mathcal{A}) \times \text{Word}(K) \subset \text{Word}(S, \mathcal{A}, K)$,*
- (ii) *the stack alphabet $\text{Alpha} = \text{Alpha}(S, \mathcal{A}, K) = S \times \mathcal{A}_\varepsilon \times K$,*
- (iii) *a state transition function $\mu : \text{Alpha} \rightarrow 2^{S \times \text{Word}(K)}$, which defines the following operation on states $M : 2^{\mathfrak{S}} \times \text{Alpha} \rightarrow 2^{\mathfrak{S}}$:*

Let $x \in \text{Alpha}$ and $X \subset \mathfrak{S}$. Then we define

$$\begin{aligned} M(X, x) &= X \cdot x \\ &= \mu(x) \cdot (X/x) \\ &= \{(z, \varepsilon, k) \cdot y \mid y \in X/x, (z, k) \in \mu(x)\} \end{aligned}$$

- (iv) *the initial element (in $2^{\mathfrak{S}}$) is defined by two elementary initial elements $i \in S$ and $k \in K$ and is given by $I_{i,k} = \{i\} \times \text{Word}(\mathcal{A}) \times \{k\}$,*
- (v) *the final set is defined by an elementary final set $E \subset S$ via*

$$\mathcal{E}_E = \{Y \times \{\varepsilon\} \times \{\varepsilon\} \subset \mathfrak{S} \mid Y \cap E \neq \emptyset\} \subset 2^{\mathfrak{S}}.$$

Such a stack acceptor is symbolized by $\text{Stack}(i, \mu, E)$, and again, the elementary sets S, \mathcal{A}, K are determined by μ . Contrary to the strict terminology a stack acceptor is traditionally called *deterministic* iff (1) all $\mu(u, v, w)$ are empty or singletons, and (2), $\mu(u, \varepsilon, w) = \emptyset$ implies that all $\mu(u, v, w), v \in \mathcal{A}$ are singletons; vice versa, if $\mu(u, \varepsilon, w)$ is a singleton, then $\mu(u, v, w) = \emptyset$ for all $v \in \mathcal{A}$. This means in particular that $X \cdot x$ is a singleton or empty if X is a singleton. It is called *nondeterministic* if it is not deterministic. In order to distinguish these confusing wordings, we mark this traditional terminology by the prefix “stack”, i.e., saying “*stack deterministic/nondeterministic*” if a confusion is likely.

Definition 150 *The stack language which is accepted by $\text{Stack}(i, \mu, E)$ will be denoted by $\text{Stack}(i : \mu : E)$ and consists by definition of all words $w \in \text{Word}(\mathcal{A})$ such that there is a word $W_p = (z_1, a_1, k_1)(z_2, a_2, k_2) \dots (z_n, a_n, k_n)$ of a state sequence in $(I_{i,k} : \mu : \mathcal{E}_E)$ with $w = a_1 \cdot a_2 \cdot \dots \cdot a_n$.*

Exercise 95 The initial element $I_{i,k}$ has the property that, if it contains (s, w, t) , then it also contains all elements $(s, v, t), v \in \text{Word}(\mathcal{A})$. Show that this property is inherited under the state operation M . More precisely, if $X \subset \mathfrak{S}$ is such that, if it contains a configuration (s, w, t) , then it also contains all configurations $(s, v, t), v \in \text{Word}(\mathcal{A})$, then so is $X \cdot x$ for any $x \in \text{Alpha}$. In other words, for any state sequence in $(I_{i,k} : \mu : \mathcal{E}_E)$, all its states share the property that any input words are admitted in the middle coordinate. We may therefore adopt the saying that when calculating a state sequence, we may “read a word, or letter, v from the input alphabet”. This means that we are given any triple $(s, w, t) \in X$ and then choose any v , take (s, v, t) , which is also in X , and look for configurations in $X \cdot x$ deduced from (s, v, t) . Therefore, in a more sloppy denotation, one also forgets about the middle coordinate and only denotes the elementary state and stack coordinates. This is what will be done in example 90.

And here is the long awaited proposition about languages and stack automata:

Proposition 174 *A language L over an alphabet \mathcal{A} is context free, i.e., of type 2, iff it is the language $\text{Stack}(i : \mu : E)$ of a stack acceptor over the input alphabet \mathcal{A} .*

Proof For the lengthy proof of this proposition, we refer to [43]. □

Example 90 Reprising our earlier example 87 of the context free language generating arithmetical expressions, we endeavour to construct a stack acceptor $e = \text{Stack}(i_e, \mu_e, E_e)$ for this language. The elementary states, input letters and stack elements are defined as $S_e = \{i, f\}$, $\mathcal{A}_e = \{+, *, (,), x, y, z\}$ and $K_e = \mathcal{A}_e \cup \{E, T, F, k\} = \{+, *, (,), x, y, z, E, T, F, k\}$, respectively. The final set $E_e \subset S$ is $\{f\}$ and the elementary initial elements are $i_e = i$ and $k_e = k$.

Table 19.10 describes the state transition function μ_e .

We now write down a derivation of the word $x + y$. Note that the sets of states become rather large, therefore we abbreviate the sets involved and leave out the states that will play no further role. We begin with the set of states $\{(i, k)\}$. Reading the empty word, i.e., reading nothing at all, we reach the set $\{(f, E)\}$. At each step we apply every matching rule of the transition map μ_e to all states in the current state set. In the following, \rightarrow_a indicates a transition by reading letter a from the input word.

$S_e \times \mathcal{A}_{e_\varepsilon} \times K_e$	\rightarrow_{μ_e}	$2^{S \times \text{Word}(K)}$
(i, ε, k)		$\{(f, E)\}$
(f, ε, E)		$\{(f, T + E), (f, T)\}$
(f, ε, T)		$\{(f, F * E), (f, F)\}$
(f, ε, F)		$\{(f, (E)), (f, x), (f, y), (f, z)\}$
(f, x, x)		$\{(f, \varepsilon)\}$
(f, y, y)		$\{(f, \varepsilon)\}$
(f, z, z)		$\{(f, \varepsilon)\}$
$(f, +, +)$		$\{(f, \varepsilon)\}$
$(f, *, *)$		$\{(f, \varepsilon)\}$
$(f, (, ($		$\{(f, \varepsilon)\}$
$(f,),)$		$\{(f, \varepsilon)\}$

Fig. 19.10. The transition map μ_e for the stack acceptor of example 90.

$$\begin{aligned}
\{(i, k)\} &\rightarrow_\varepsilon \{(f, E)\} \\
&\rightarrow_\varepsilon \{(f, T + E), (f, T)\} \\
&\rightarrow_\varepsilon \{(f, F * E + E), (f, F * E), (f, F + E), (f, F)\} \\
&\rightarrow_\varepsilon \{\dots, (f, x + E), \dots\} \\
&\rightarrow_x \{\dots, (f, +E), \dots\} \\
&\rightarrow_+ \{\dots, (f, E), \dots\} \\
&\rightarrow_\varepsilon \{\dots, (f, T + E), (f, T), \dots\} \\
&\rightarrow_\varepsilon \{\dots, (f, F * E + E), (f, F * E), (f, F + E), (f, F), \dots\} \\
&\rightarrow_\varepsilon \{\dots, (f, y), \dots\} \\
&\rightarrow_y \{\dots, (f, \varepsilon), \dots\}
\end{aligned}$$

In the final line, the current set of states includes a state (f, ε) which satisfies the conditions of a final state, E_e being $\{f\}$ and the stack word being empty.

Note how the form of a rule $(s, a, k) \rightarrow (s, k_1 k)$ justifies the image of “push”, k_1 being pushed on top of k . In the same way a rule $(s, a, k) \rightarrow (s, \varepsilon)$ “pop”s off k .

We leave it to the reader to compare the map μ_e with the set of grammar rules in example 87 and to find out how they relate.

19.3.2 Turing Machines

Intuitively, Turing machines—named after the mathematician Alan Turing (1912–1954), one of the founders of computer science—are finite automata which are provided with an infinite tape memory and sequential access using a read/write head. We shall give a formal definition below, but we should make a point concerning the general formalism of automata and the concrete technical setup where particular types of automata are realized. We had in fact already learned above that stack acceptors are acceptors which are defined by auxiliary operations on complex spaces. The same is true for Turing machines.

For Turing machines, one is essentially given a state set S , a tape alphabet B which includes a special sign $\#$ for a “blank” entry on a specific place on the tape. The tape is thought to be infinite to the left and to the right of the read/write head. On the tape, we may have any elements of B , except that only finitely many differ from $\#$. The set of tape states is therefore described by the subset $B^{(\mathbb{Z})}$ of $B^{\mathbb{Z}}$ of those sequences $t = (t_i), t_i \in B$ with $t_i = \#$ for all but finitely many indexes $i \in \mathbb{Z}$ (mathematicians often say in this case: “for *almost all* indexes”). The read/write head position is by definition the one with index $i = 0$. With this vocabulary, a Turing machine, as we shall define shortly, yields a map

$$\tau : S \times B^{(\mathbb{Z})} \rightarrow 2^{S \times B^{(\mathbb{Z})}},$$

which describes the transition from one pair “state s of automaton plus tape state t ” to a set of possible new pairs s' and t' of the same type. The machine continues updating states and tape until it reaches a “halt” state. This looks somewhat different from the definition of an automaton, which requires a map $S \times \mathcal{A} \rightarrow S$. However, this formalism is also present in the previous description. One must in fact use the natural adjunction of set maps discussed in proposition 59: The sets $\text{Set}(a \times b, c)$ and $\text{Set}(a, c^b)$ are in a natural bijection. Therefore

$$\mathcal{A}d : \text{Set}(S \times B^{(\mathbb{Z})}, 2^{S \times B^{(\mathbb{Z})}}) \xrightarrow{\sim} \text{Set}(S \times B^{(\mathbb{Z})} \times B^{(\mathbb{Z})}, 2^S)$$

which means that the Turing automaton map τ corresponds to a map

$$\mathcal{A}d(\tau) : S \times (B^{(\mathbb{Z})} \times B^{(\mathbb{Z})}) \rightarrow 2^S,$$

and this is precisely the type of maps we need for automata, i.e., the alphabet is $\mathcal{A} = B^{(\mathbb{Z})} \times B^{(\mathbb{Z})}$. We call the bijection $\mathcal{A}d$ the *Turing adjunction*.⁵ The alphabet \mathcal{A} happens to be infinite, but the formalism is the required one. The meaning of $\mathcal{A}d(\tau)$ is this: We are given a present state $s \in S$ and a pair (t, t') of tape states, t is the present tape state, whereas t' is one of the possible successor tape state. The set $\mathcal{A}d(\tau)(s, t, t')$ is precisely the set of those successor states $s' \in S$ such that $(s', t') \in \tau(s, t)$, i.e., which correspond to the successor tape state t' .

But we shall stick to the original map τ in order to maintain the intuitive setup. Here is the formal definition of Turing machines:

Definition 151 *A Turing machine is given by*

- (i) *a finite state set S , an initial state $i \in S$, and a special halt state s_H to be specified,*
- (ii) *a finite tape alphabet B , containing a special blank sign $\#$, together with an input alphabet $\mathcal{A} \subset B$,*
- (iii) *three symbols H, L, R not in B , one writes $B_{HLR} = \{H, L, R\} \cup B$,*
- (iv) *a state transition map $tr : S \times B \rightarrow 2^{S \times B_{HLR}}$ with $tr(s, b) \neq \emptyset$ for all $(s, b) \in S \times B$ and such that only pairs $(s_H, H) \in tr(s, b)$ may appear with second coordinate H .*

A Turing machine is deterministic iff every set $tr(s, b)$ is a singleton, otherwise it is called nondeterministic. The Turing machine is also denoted by $Turing(i, tr, s_H)$, according to our general notation of acceptors.

This definition generates the following map τ . To begin with, the element $q \in B_{HLR}$ of the extended tape alphabet operates on tape states $t \in B^{(\mathbb{Z})}$ as follows (we write $q \cdot t$ for the result of the operation of q on t):

1. if $q \in B$, then $(q \cdot t)_i = t_i$ for $i \neq 0$, and $(q \cdot t)_0 = q$, i.e., the zero position on the tape is replaced by q ;
2. if $q = H$, nothing happens: $q \cdot t = t$;
3. if $q = R$, the tape moves one unit to the right, i.e., $(q \cdot t)_i = t_{i-1}$;
4. if $q = L$, the tape moves one unit to the left, i.e., $(q \cdot t)_i = t_{i+1}$.

⁵ Observe the omnipresence of universal constructions of mathematics in computer science.

Then we have this map:

$$\begin{aligned} \tau : S \times B^{(\mathbb{Z})} &\rightarrow 2^{S \times B^{(\mathbb{Z})}} \\ \tau(s, t) &= \{(s', b' \cdot t) \mid (s', b') \in tr(s, t_0)\} \end{aligned}$$

and the idea is that the change of states should go on until the halt state s_H is obtained. More precisely,

Definition 152 *With the above notations, a state sequence of a Turing machine $Turing(i, tr, s_H)$ is a sequence $(s^{(0)}, t^{(0)}), (s^{(1)}, t^{(1)}), \dots, (s^{(n)}, t^{(n)})$ such that $(s^{(i+1)}, t^{(i+1)}) \in \tau(s^i, t^i)$ for all indexes $i = 0, 1, \dots, n - 1$.*

A word $w = w_1 w_2 \dots w_k \in Word(\mathcal{A})$ is accepted by $Turing(i, tr, s_H)$ iff there is a state sequence $(s^{(0)}, t^{(0)}), (s^{(1)}, t^{(1)}), \dots, (s^{(n)}, t^{(n)})$ which terminates at the halt state, i.e., $s^{(n)} = s_H$, and starts at state $s^{(0)} = i$ and a tape state $t^{(0)}$ with $t_i^{(0)} = \#$, for $i \leq 0$ and $i > k$, and $t_i^{(0)} = w_i$ for $i = 1, 2, \dots, k$. The language of words accepted by the Turing machine $Turing(i, tr, s_H)$ is denoted by $Turing(i : tr : s_H)$. Languages of type $Turing(i, tr, s_H)$ for given Turing machines are also called semi-decidable.

And here is the rewarding proposition relating Turing machines and type 0 languages:

Proposition 175 *A language $L \in Lang(\mathcal{A})$ is of type 0 (i.e., recursively enumerable) iff it is semi-decidable, i.e., iff there is a Turing machine $Turing(i, tr, s_H)$ with input alphabet \mathcal{A} such that $L = Turing(i : tr : s_H)$.*

Proof For the complicated proof of this proposition, we refer to [43]. □

Example 91 A simple example of an actual “program” for a Turing machine shall illustrate the principles of the foregoing discussion, and show that Turing machines are capable of doing “useful” work, in this case the incrementation of a natural number in binary representation, i.e., encoded using the symbols 0 and 1 on the initial tape state, where the least-significant bit of the number is on the right of the tape. Thus the number 151 will appear on the tape as $\dots \bar{\#} \bar{\#} \bar{\#} \bar{\#} 10010111 \bar{\#} \bar{\#} \bar{\#} \bar{\#} \dots$. At the beginning, the machine’s head will be at the position indicated by $\bar{\#}$. The final tape state when the machine terminates in the halt state will be $\dots \bar{\#} \bar{\#} \bar{\#} \bar{\#} 10011000 \bar{\#} \bar{\#} \bar{\#} \bar{\#} \dots$, i.e., 152. Note that we use the Turing machine a little differently than previously described. The object here is not to accept an initial word, but to transform the initial word into the result word.

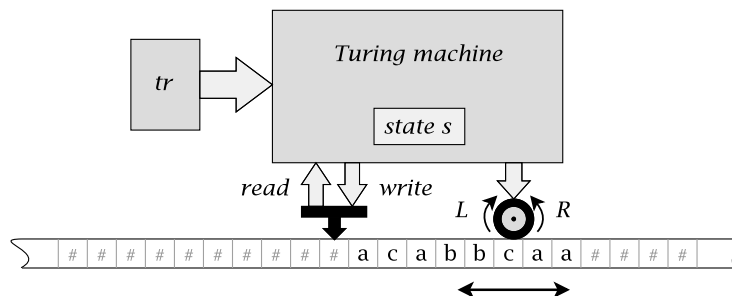


Fig. 19.11. A schematic illustration of a Turing machine, with the transition map (“program”) tr , current state s , input alphabet $\mathcal{A} = \{a, b, c\}$. The gray arrows show the flow of information into and out of the machine.

Now we proceed to describe the required data for the Turing machine T_I : the state set $S_I = \{i, A, B, C, D, E, F, s_H\}$ and the tape alphabet $B_I = \{\#, 1, 0\}$. The state transition map tr_I is shown in table 19.12. We assume that, initially, the tape is not empty, and omit any kind of error handling. The rudimentary character of Turing machines wouldn’t allow much of anything in this direction anyhow. Observe that T_I is essentially deterministic, since the values of the transition map are all singletons, except that we left out the definition of tr_I for some pairs (s, b) that wouldn’t occur anyway.

The reader is invited to simulate this machine on a few input words.

Concluding this short section on Turing machines, we should note that the length of a state sequence which accepts a word w as a function of the word length $l(w)$ may be exorbitant, and one is therefore interested in those languages where the lengths of accepting state sequences for their words are not too long. Here is the precise definition.

Definition 153 *A language $L \in \text{Lang}(\mathcal{A})$ is called of (polynomial) complexity class P if there is a polynomial $P(X)$ and a deterministic Turing machine $\text{Turing}(i, tr, s_H)$ with input alphabet \mathcal{A} and $L = \text{Turing}(i, tr, s_H)$, such that each word $w \in L$ is accepted by a state sequence of length at most $P(l(w))$.*

A language $L \in \text{Lang}(\mathcal{A})$ is called of (nondeterministic polynomial) complexity class NP if there is a polynomial $P(X)$ and a nondeterministic

$S_I \times B_I$	$\xrightarrow{tr_I}$	$2^{S_I \times B_{HLR}}$
$(i, \#)$		$\{(A, L)\}$
$(A, 0)$		$\{(A, L)\}$
$(A, 1)$		$\{(A, L)\}$
$(A, \#)$		$\{(B, R)\}$
$(B, 1)$		$\{(D, 0)\}$
$(B, 0)$		$\{(E, 1)\}$
$(B, \#)$		$\{(E, 1)\}$
$(D, 0)$		$\{(B, R)\}$
$(D, 1)$		$\{(B, R)\}$
$(E, 0)$		$\{(C, R)\}$
$(E, 1)$		$\{(C, R)\}$
$(C, 0)$		$\{(E, 0)\}$
$(C, 1)$		$\{(E, 1)\}$
$(C, \#)$		$\{(F, L)\}$
$(F, 0)$		$\{(s_H, H)\}$
$(F, 1)$		$\{(s_H, H)\}$

Fig. 19.12. The transition map tr_I for the Turing machine of example 91.

Turing machine $Turing(i, tr, s_H)$ with input alphabet \mathcal{A} and language $L = Turing(i, tr, s_H)$, such that each word $w \in L$ is accepted by a state sequence of length at most $P(l(w))$.

A final remark: It is one of the deepest unsolved problems of computer science to understand the relation between class P and class NP. In particular, it is not currently known whether “P = NP”. Recently, it has been shown that there is a deterministic algorithm that decides for every natural number n in $(\log(n))^k$ steps, k being a fixed natural exponent, whether n is prime or not. The logarithm is proportional to the length of n in its binary representation, so it represents the length of the word needed to write down n . This is what is meant, when the result is stated as “PRIME is in P”. See [11] for a lucid exposition.

A comprehensive treatment of automata theory and languages is [28]. For more on the subject of computability, P and NP, see [30].

Categories of Matrixes

The present chapter opens a field of mathematics which is basic to all applications, be it in equation solving, geometry, optimization, calculus, or numerics. This field is called linear algebra. It is the part of algebra, which we do control best—in contrast to non-linear algebra, also called algebraic geometry, which is far more difficult. Linear algebra deals with the structural theory of vectors, and the geometry they describe. We shall, however, see later in the chapters on calculus that even non-linear, or—worse than that—non-algebraic phenomena of continuous and infinitesimal phenomena can in a first approximation be dealt with by linear algebra. So here we enter one of the most powerful domains of mathematics, and also one for which algorithms and corresponding computer programs are most developed.

The structure of linear algebra rests on three pillars, which complement each other, but are of equal importance for a real grasp of the subject: Matrix theory, axiomatic theory of vector spaces, and linear geometry.

First, *matrix theory* is the calculatory backbone, it is nothing less than the mathematical theory of tables. Without knowing the essentials about matrixes, any understanding of arrays, lists, or vectors becomes difficult, and no real, concrete calculation is possible. Strangely enough it turns out that the system of matrixes is at the same time the most concrete and the most abstract structure of this field. This is hinted at by attribute “category” in our title, a specification which will become more and more clear with the evolution of the student’s understanding of fundamental structures in mathematics. Recall that we have already added this attribute in the context of graphs. It is remarkable that things seemingly so dis-

tant as graphs and tables turn out to be core instances of a common substance: the structure of a category. Presently, this is a philosophical allusion rather than hard mathematics. But it is a hint at the beauty of mathematics that any reader should learn to feel in the course of this curriculum.

Second, *axiomatic vector space theory* is the account to the fact that the truths which are “hard coded” in matrix theory can be encountered in much less concrete situations. These situations seem to be unrelated to matrixes, but, when analyzed in view of their structural substance, reveal a fantastic kind of conceptual generalization of matrix calculus. This effect provides very operational perspectives to seemingly abstract situations.

Last, *linear geometry* is a very traditional branch of geometry, which can be traced back to Descartes and his analytical geometry. If one rephrases the structural substance of points, lines, surfaces, together with their operations, metrical properties, and transformational behavior, then it turns out that one gets what is essentially the theory of vector spaces. And this gives the dry calculations of matrix theory and the abstract manipulations of vector space theory a sensorial perspective which is not only beautiful by itself, but in turn helps to understand abstract phenomena of matrix theory in a revealing environment of geometric intuition. We should however point out that the geometric intuition about abstract truths is not the only one: the auditory intuition as cultivated by the experience music is another sensorial image of abstract mathematical truths which for a number of problems is even better suited to yield evidence of “abstract artifacts” (or even what some scientific businessmen call “abstract nonsense”).

20.1 What Matrixes Are

In this and the following sections of this chapter, the zero and unit of a ring R will be denoted by 0 and 1 , respectively, if the context is clear. For any natural number n , we denote by $[1, n]$ the set of natural numbers between 1 and n , including the extremal values 1 and n . For $n = 0$, the set $[1, 0]$ is the empty set.

Definition 154 *Given a ring R and two natural numbers m and n , a matrix of size $m \times n$, or $m \times n$ -matrix, with coefficients in R is a triple*

$(m, n, M : [1, m] \times [1, n] \rightarrow R)$. For non-zero m and n , the value $M(i, j)$, which is also written as M_{ij} , is called the coefficient of M at index pair ij ; the number i is called the row index, while j is called the column index. If the number of rows m and the number of columns n are clear, M is also denoted by (M_{ij}) . For a row number i the i -th row matrix in M is the matrix $M_{i\bullet}$ of size $1 \times n$ with $(M_{i\bullet})_{1j} = M_{ij}$. For a column number j the j -th column matrix in M is the matrix $M_{\bullet j}$ of size $m \times 1$ with $(M_{\bullet j})_{i1} = M_{ij}$.

The set of $m \times n$ -matrixes over R is denoted by $\mathbb{M}_{m,n}(R)$, while the set of all matrixes, i.e., the disjoint union of all sets $\mathbb{M}_{m,n}(R)$, is denoted by $\mathbb{M}(R)$. Clearly, every ring homomorphism $f : R \rightarrow S$ between rings gives rise to a map $\mathbb{M}(f) : \mathbb{M}(R) \rightarrow \mathbb{M}(S)$, which sends a matrix $M = (M_{ij}) \in \mathbb{M}(R)$ to the matrix $f(M) = f \circ M \in \mathbb{M}(S)$ with $(f(M))_{ij} = (f(M_{ij}))$, and which therefore also sends $\mathbb{M}_{m,n}(R)$ into $\mathbb{M}_{m,n}(S)$.

Example 92 There is a number of special matrixes:

- The unique matrix for either m or n equal to 0 is denoted by $0 \square n$, $m \square 0$, or $0 \square 0$, respectively.
- If $m = n$, the matrix is called a *square* matrix.
- The following convention is very useful in matrix calculus: If i and j are two natural numbers, the *Kronecker delta symbol* is the number

$$\delta_{ij} = \begin{cases} 1 \in R & \text{if } i = j, \\ 0 \in R & \text{if } i \neq j. \end{cases}$$

Then the square $n \times n$ -matrix defined by $E_n = (\delta_{ij})$ is called the *unit matrix* of rank n —including the unique matrix $0 \square 0$ as a “degenerate” special case.

- Also important are the so-called *elementary* matrixes. Given an index pair ij , the number of rows m and the number of columns n , the elementary $m \times n$ -matrix for this datum is the matrix $E(i, j)$ such that $(E(i, j))_{uv} = 0$ except for $uv = ij$, where we have $(E(i, j))_{ij} = 1$ (see also figure 20.1).

Usually, a matrix M is represented as a rectangular table, where the entry on row i and column j is the matrix coefficient M_{ij} . Here is the tabular representation of one example of a 2×3 -matrix M over the ring $R = \mathbb{Z}$ of integer numbers:

$$M = \begin{pmatrix} -2 & 5 & 0 \\ 0 & 26 & 3 \end{pmatrix}$$

If we have the canonical ring homomorphism $f = \text{mod}_7 : \mathbb{Z} \rightarrow \mathbb{Z}_7$, then the image $f(M)$ is equal to this matrix over \mathbb{Z}_7 :

$$f(M) = \begin{pmatrix} \text{mod}_7(-2) & \text{mod}_7(5) & \text{mod}_7(0) \\ \text{mod}_7(0) & \text{mod}_7(26) & \text{mod}_7(3) \end{pmatrix}$$

One also writes $M \equiv N \pmod{d}$ if $M, N \in \mathbb{M}_{m,n}(\mathbb{Z})$ and their images under the canonical homomorphism $\text{mod}_d : \mathbb{Z} \rightarrow \mathbb{Z}_d$ coincide; so for example,

$$\begin{pmatrix} -2 & 5 & 0 \\ 0 & 26 & 3 \end{pmatrix} \equiv \begin{pmatrix} 5 & 5 & 0 \\ 0 & 5 & 3 \end{pmatrix} \pmod{7}.$$

Or else, consider the 4×3 -matrix

$$M = \begin{pmatrix} 3.5 + i \cdot 4 & -i & 4 + i \cdot \sqrt{5} \\ -0.5 & 0 & 4 - i \cdot 2 \\ i \cdot 20 & 1 & 3.78 - i \\ 0 & -i & -5 - i \cdot \sqrt{3} \end{pmatrix}$$

with complex coefficients. We have the field automorphism of conjugation $f(z) = \bar{z}$, which gives us the *conjugate matrix*

$$\begin{aligned} f(M) = \overline{M} &= \begin{pmatrix} \overline{3.5 + i \cdot 4} & \overline{-i} & \overline{4 + i \cdot \sqrt{5}} \\ \overline{-0.5} & \overline{0} & \overline{4 - i \cdot 2} \\ \overline{i \cdot 20} & \overline{1} & \overline{3.78 - i} \\ \overline{0} & \overline{-i} & \overline{-5 - i \cdot \sqrt{3}} \end{pmatrix} \\ &= \begin{pmatrix} 3.5 - i \cdot 4 & i & 4 - i \cdot \sqrt{5} \\ -0.5 & 0 & 4 + i \cdot 2 \\ -i \cdot 20 & 1 & 3.78 + i \\ 0 & i & -5 + i \cdot \sqrt{3} \end{pmatrix}. \end{aligned}$$

A third example comes closer to tables in common usage: word processing environments. Suppose that we are given an alphabet C , which, to be concrete, denotes the letters in the Courier font, whereas S denotes the alphabet of letters in the Old German Schwabacher font. In a common word processing software, a text may be converted from Courier to Schwabacher, more formally, we have a map $\gamma : C \rightarrow S$. By the universal property of monoids (proposition 111) and monoid algebras (proposition 120), the map γ induces a ring homomorphism

$$f = \text{Id}_{\mathbb{Z}\langle \text{Word}(\gamma) \rangle} : \mathbb{Z}\langle \text{Word}(C) \rangle \rightarrow \mathbb{Z}\langle \text{Word}(S) \rangle,$$

which we may now apply to a table with Courier-typed text, i.e., elements from the ring $\mathbb{Z}\langle \text{Word}(C) \rangle$, in order to obtain a table with text in the Schwabacher font, i.e., elements from the ring $\mathbb{Z}\langle \text{Word}(S) \rangle$. For example, if

$$M = \begin{pmatrix} \text{Author:} & \text{Shakespeare} \\ \text{Work:} & \text{Hamlet} \end{pmatrix}$$

then

$$f(M) = \begin{pmatrix} \text{Author:} & \text{Shakespeare} \\ \text{Work:} & \text{Hamlet} \end{pmatrix}$$

Definition 155 For any ring R , the transposition is the map

$${}^{\tau} : \mathbb{M}(R) \rightarrow \mathbb{M}(R)$$

defined by $(M^{\tau})_{ij} = M_{ji}$ for all index pairs ij , thereby transforming a $m \times n$ -matrix M into a $n \times m$ -matrix M^{τ} . A matrix M is called symmetric if $M^{\tau} = M$.

Exercise 96 For any natural number n , the identity matrix E_n is symmetric. If $E(i, j)$ is elementary, then $E(i, j)^{\tau} = E(j, i)$. Show that for any matrix M , we have

$$(M^{\tau})^{\tau} = M.$$

In particular, matrix transposition is a bijection on the set $\mathbb{M}(R)$.

$$E_3 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad E(2, 3) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix} \quad E(3, 2) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$

Fig. 20.1. The unit matrix and the elementary matrixes $E(2, 3)$ and $E(3, 2) = E(2, 3)^{\tau}$ in $\mathbb{M}_{3,3}$.

20.2 Standard Operations on Matrixes

We now proceed to the algebraic standard operations on matrixes. We now tacitly assume that the underlying ring R for $\mathbb{M}(R)$ is commutative, and we shall explicitly state any deviation from this general assumption.

Definition 156 Given two matrixes $M, N \in \mathbb{M}_{m,n}(R)$, their sum $M + N \in \mathbb{M}_{m,n}(R)$ is defined as follows: If one of the numbers m or n is 0, then there is only one matrix in $\mathbb{M}_{m,n}(R)$, and we set $M + N = m \square n$. Else, we set $(M + N)_{ij} = M_{ij} + N_{ij}$.

Sorite 176 With the addition defined in definition 156, the set $\mathbb{M}_{m,n}(R)$ becomes an abelian group. For m or n equal to 0, this is the trivial (zero) group. In the general case, the neutral element of $\mathbb{M}_{m,n}(R)$ is the zero matrix $0 = (0)$, whereas the additive inverse $-M$ is defined by $(-M)_{ij} = -M_{ij}$.

Exercise 97 Give a proof of sorite 176.

Definition 157 Given a matrix $M \in \mathbb{M}_{m,n}(R)$, and a scalar $\lambda \in R$, the scalar multiplication $\lambda \cdot M$ is defined as follows: If one of the numbers m or n is 0, then there is only one matrix in $\mathbb{M}_{m,n}(R)$, and we set $\lambda \cdot M = m \square n$. Otherwise we set $(\lambda \cdot M)_{ij} = \lambda \cdot M_{ij}$. The matrix $\lambda \cdot M$ is also called “ M scaled by λ .”

In other words, as is the case with ring homomorphisms and addition, scalar multiplication proceeds coefficient-wise, i.e., by operating on each coefficient of the involved matrixes.

Sorite 177 With the definitions 156 and 157 of addition and scalar multiplication, we have these properties for any $\lambda, \mu \in R$ and $M, N \in \mathbb{M}_{m,n}(R)$:

- (i) Scalar multiplication is homogeneous, i.e., we have $\lambda \cdot (\mu \cdot M) = (\lambda \cdot \mu) \cdot M$, therefore we may write $\lambda \cdot \mu \cdot M$ for this expression.
- (ii) Scalar multiplication is distributive, i.e., we have

$$(\lambda + \mu) \cdot M = \lambda \cdot M + \mu \cdot M$$

and

$$\lambda \cdot (M + N) = \lambda \cdot M + \lambda \cdot N.$$

- (iii) Scalar multiplication and transposition commute: $(\lambda \cdot M)^\top = \lambda \cdot M^\top$.

Exercise 98 Give a proof of sorite 177.

Proposition 178 Given positive row and column numbers m and n , every matrix $M \in \mathbb{M}_{m,n}(R)$ can be uniquely represented as a sum

$$M = \sum_{\substack{i=1,\dots,m \\ j=1,\dots,n}} M_{ij} \cdot E(i, j)$$

of scaled $m \times n$ elementary matrixes $E(i, j)$. I.e., if we have any representation

$$M = \sum_{\substack{i=1,\dots,m \\ j=1,\dots,n}} \mu_{ij} \cdot E(i, j)$$

with $\mu_{ij} \in R$, then $\mu_{ij} = M_{ij}$.

Proof The sum $\sum_{i=1,\dots,m,j=1,\dots,n} M_{ij} \cdot E(i, j)$, when evaluated at an index pair uv , yields $\sum_{i=1,\dots,m,j=1,\dots,n} M_{ij} \cdot (E(i, j)(uv)) = M_{uv}$, since all elementary matrixes vanish, except for $ij = uv$. If any representation $M = \sum_{i=1,\dots,m,j=1,\dots,n} \mu_{ij} \cdot E(i, j)$ of M is given, then $M_{uv} = \mu_{uv}$, and we are done. \square

The next operation is the most important in matrix theory: the product of two matrixes. It is most important to learn this operation by heart.

Definition 158 Let $M \in \mathbb{M}_{m,n}(R)$ and $N \in \mathbb{M}_{n,l}(R)$ be two matrixes, then a matrix $M \cdot N \in \mathbb{M}_{m,l}(R)$, the product of M and N , is defined as follows: If one of the outer numbers, m or l is 0, then $M \cdot N = m \square l \in \mathbb{M}_{m,l}(R)$ is the unique matrix in $\mathbb{M}_{m,l}(R)$. Otherwise, if the middle number $n = 0$, then we set $M \cdot N = 0 \in \mathbb{M}_{m,l}(R)$, the $m \times l$ -matrix with zeros at every position. In the general case (no number m, n, l vanishes), one sets

$$(M \cdot N)_{ij} = \sum_{k=1,\dots,n} M_{ik}N_{kj}$$

for every index pair ij .

$$i \rightarrow \left(\begin{array}{ccc} M_{i1} & \cdots & M_{in} \end{array} \right) \cdot \left(\begin{array}{c} N_{1j} \\ \vdots \\ N_{nj} \end{array} \right) = \left(\begin{array}{c} (MN)_{ij} \end{array} \right) \leftarrow i$$

Fig. 20.2. Multiplication of matrixes $M \in \mathbb{M}_{m,n}$ and $N \in \mathbb{M}_{n,l}$, with the result $MN \in \mathbb{M}_{m,l}$.

So pay attention: The product of matrixes $M \in \mathbb{M}_{m,n}(R)$ and $N \in \mathbb{M}_{n',l}(R)$ is never defined if $n \neq n'$. To make this restriction really evident,

we set up a more telling representation of matrixes. If $M \in \mathbb{M}_{m,n}(R)$, we shall also write

$$M : E_n \rightarrow E_m \text{ or else } E_n \xrightarrow{M} E_m$$

in order to indicate the possibilities to compose matrixes as if they were set maps! In fact, the product of $E_l \xrightarrow{N} E_{n'}$ and $E_n \xrightarrow{M} E_m$ is only possible if $n = n'$, i.e., the “codomain” $E_{n'}$ of N equals the “domain” E_n of M , and then gives us the matrix product $E_l \xrightarrow{M \cdot N} E_m$ which we may visualize by a commutative diagram:

$$\begin{array}{ccc} E_l & \xrightarrow{N} & E_n \\ & \searrow^{M \circ N} & \downarrow M \\ & & E_m \end{array}$$

What looks somehow mysterious here is a very modern point of view of maps: Matrixes look like maps without domains or codomains. We just have composition of maps, and the domains and codomains must be reinvented by a trick. Let us take this approach as it is and postpone its deeper signification to a more advanced discussion of category theory in the second volume of this book. *The only point to retain here is the very useful notational advantage which immediately makes evident which matrixes may be multiplied with each other.* What should however be expected from this arrow formalism is, that matrix products are associative whenever defined. This is of course true:

Sorite 179 Let $A : E_n \rightarrow E_m, B : E_m \rightarrow E_l$ and $C : E_l \rightarrow E_k$ be three matrixes over R . Then

- (i) (Associativity) $(C \cdot B) \cdot A = C \cdot (B \cdot A)$, which we therefore write as $C \cdot B \cdot A$.
- (ii) (Distributivity) If $C' : E_l \rightarrow E_k$ and $B' : E_m \rightarrow E_l$ are two matrixes over R , then $(C + C') \cdot B = C \cdot B + C' \cdot B$ and $C \cdot (B + B') = C \cdot B + C \cdot B'$.
- (iii) (Homogeneity) If $\lambda \in R$ is a scalar, then $\lambda \cdot (C \cdot B) = (\lambda \cdot C) \cdot B = C \cdot (\lambda \cdot B)$, which we therefore write as $\lambda \cdot C \cdot B$.
- (iv) (Neutrality of identity matrixes) We have $A \cdot E_n = E_m \cdot A = A$.
- (v) $(C \cdot B)^\top = B^\top \cdot C^\top$.

Proof Let $A = (A_{tu}), B = (B_{st})$ and $C = (C_{rs})$, with $1 \leq r \leq k, 1 \leq s \leq m$ and $1 \leq t \leq n$. Then $((C \cdot B) \cdot A)_{ru} = \sum_t (C \cdot B)_{rt} \cdot A_{tu} = \sum_t (\sum_s (C_{rs} \cdot B_{st})) \cdot A_{tu} =$

$\sum_t \sum_s (C_{rs} \cdot B_{st}) \cdot A_{tu} = \sum_s \sum_t C_{rs} \cdot (B_{st} \cdot A_{tu}) = \sum_s C_{rs} \cdot (\sum_t B_{st} \cdot A_{tu}) = \sum_s (C_{rs} \cdot (B \cdot A)_{su}) = (C \cdot (B \cdot A))_{ru}$, whence (i).

Then $((C + C') \cdot B)_{rt} = \sum_s (C + C')_{rs} \cdot B_{st} = \sum_s (C_{rs} + C'_{rs}) \cdot B_{st} = \sum_s (C_{rs} \cdot B_{st} + C'_{rs} \cdot B_{st}) = \sum_s C_{rs} \cdot B_{st} + \sum_s C'_{rs} \cdot B_{st} = (C \cdot B)_{rt} + (C' \cdot B)_{rt}$, whence (ii).

Further, $(\lambda \cdot (C \cdot B))_{rt} = \lambda \cdot (C \cdot B)_{rt} = \lambda \cdot \sum_s C_{rs} \cdot B_{st} = \sum_s (\lambda \cdot C_{rs}) \cdot B_{st} = ((\lambda \cdot C) \cdot B)_{rt}$, and similarly $\sum_s (\lambda \cdot C_{rs}) \cdot B_{st} = \sum_s C_{rs} \cdot (\lambda \cdot B_{st}) = (C \cdot (\lambda \cdot B))_{rt}$, whence (iii).

Claim (iv) is left to the reader.

Finally, $((C \cdot B)^T)_{tr} = (C \cdot B)_{rt} = \sum_s C_{rs} \cdot B_{st} = \sum_s B_{ts}^T \cdot C_{sr}^T = (B^T \cdot C^T)_{tr}$. \square

The definition of the matrix product now needs first justifications more practical points of view. Here are some examples.

Example 93 From high school and practical experience it is well known that linear equations are very important. Here is one such equation, which we set up in its concrete shape

$$\begin{aligned} 3.7 &= 23x_1 - x_2 + 45x_4 \\ -8 &= 0.9x_1 + 9.6x_2 + x_3 - x_4 \\ 0 &= 20x_2 - x_3 + x_4 \\ 1 &= 3x_2 + x_3 - 2x_4 \end{aligned}$$

in order to let the student recognize the common situation. This system of equations is in fact an equation among matrixes: On the left hand side of the equation, we have a 4×1 -matrix which is the product

$$\begin{pmatrix} 3.7 \\ -8 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 23 & -1 & 0 & 45 \\ 0.9 & 9.6 & 1 & -1 \\ 0 & 20 & -1 & 1 \\ 0 & 3 & 1 & -2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}$$

of a 4×4 -matrix and a 4×1 -matrix, the latter being the matrix of the unknowns x_1, \dots, x_4 . Hence the theory of matrix products should (and will) provides tools for finding solutions of linear equations. The ideal thing would be to construct a kind of inverse to the 4×4 -matrix of the equation's coefficients and then multiply both sides with this inverse. This is indeed what is done, but we are not yet ready for such a construction and need more theory.

Example 94 This example is taken from graph theory and uses the adjacency matrix introduced in definition 68. Given a digraph $\Gamma : A \rightarrow V^2$,

and fixing a bijection $c : [1, n] \rightarrow V$ with $n = \text{card}(V)$, we obtain a $n \times n$ -matrix $Adj_c = (Adj_c(i, j))$, where $Adj_c(i, j)$ is the number of arrows from the vertex $c(i)$ to the vertex $c(j)$.

What is the role of matrix products in this graph-theoretical context? The entry at index ij of the adjacency matrix is the number of arrows from vertex $c(i)$ to vertex $c(j)$, i.e., the number of paths of length one from i to j . We contend that the square Adj_c^2 of the adjacency matrix has as entry at ij the number of paths of length 2. In fact, any such path must reach $c(j)$ from $c(i)$ through an intermediate vertex, which runs through $c(1), \dots, c(n)$. Now, for each such intermediate vertex $c(k)$, the paths which cross it are one arrow $c(i) \rightarrow c(k)$, composed with one arrow $c(k) \rightarrow c(j)$, and this yields the product $Adj_c(i, k) \cdot Adj_c(k, j)$, therefore the total number of paths of length 2 is the coefficient at ij of the square Adj_c^2 . More generally, the numbers of paths of length r are the coefficients in the r -th power Adj_c^r of the adjacency matrix.

We consider again the adjacency matrix of the graph Γ from example 37 in section 10:

$$Adj_c(\Gamma) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

The square of this matrix is:

$$Adj_c^2(\Gamma) = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 2 \\ 4 & 0 & 0 & 0 & 0 & 2 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 2 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

As an illustration, figure 20.3 shows the four paths from vertex 2 to vertex 0 as indicated by entry $(Adj_c^2(\Gamma))_{3,1}$. Remember that vertex number i is associated with matrix index $i + 1$.

Exercise 99 What does it mean for the adjacency matrix of a digraph if the digraph has three connected components? Try to find a reasonable indexing of the vertexes. How do the powers of such a matrix look like?

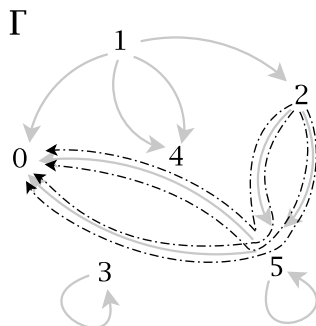


Fig. 20.3. The four paths of length 2 from vertex 2 to vertex 0.

20.3 Square Matrixes and their Determinant

The most important matrixes are the square matrixes M of positive size n , i.e., $M \in \mathbb{M}_{n,n}(R)$. We also tacitly assume $R \neq 0$. We have this general result to start with:

Proposition 180 *Let n be a positive natural number and R a commutative, non-zero ring. Then the set $\mathbb{M}_{n,n}(R)$, together with the sum and product of matrixes is a ring which is not commutative except for $n = 1$. The homomorphism of rings $\Delta : R \rightarrow \mathbb{M}_{n,n}(R) : r \mapsto r \cdot E_n$ identifies R with the diagonal matrixes of size n with $M_{ii} = r$, and $M_{ij} = 0$ for $i \neq j$. We therefore also identify $r \in R$ with its diagonal matrix $r \cdot E_n$ if no confusion is likely. Then, we have $r \cdot M = M \cdot r$ for all $M \in \mathbb{M}_{n,n}(R)$; we say that R commutes with all matrixes of size n . No other matrixes commute with all of $\mathbb{M}_{n,n}(R)$.*

Proof The proposition immediately follows from sorite 179, except for the last statement. Suppose that a $n \times n$ -matrix N commutes with all of $\mathbb{M}_{n,n}(R)$. Then it must commute with all elementary matrixes $E(i, j)$. But $N \cdot E(i, j)$ has zeros except in column j which is $N_{\bullet i}$, whereas the product $E(i, j) \cdot N$ has zeros except in row i which is $N_{j \bullet}$. So, at the intersection of this row and this column, we have the equation $N_{ii} = N_{jj}$, whereas all other coefficients must vanish, whence $N = \lambda = \lambda \cdot E_n$. \square

Definition 159 *The group $\mathbb{M}_{n,n}(R)^*$ of invertible matrixes of size n is called the general linear group of size n and denoted by $GL_n(R)$. An invertible matrix is also called regular.*

In order to handle $GL_n(R)$, one needs a special function, the *determinant* of a square matrix. Here is its construction:

Definition 160 If $M = (M_{ij}) \in \mathbb{M}_{n,n}(R)$, then $\det(M) \in R$ is defined by

$$\det(M) = \sum_{\pi \in S_n} (-1)^{\text{sig}(\pi)} \prod_{j=1 \dots n} M_{\pi(j),j}$$

where S_n is the symmetric group of rank n defined in section 15.2.

Observe that, a priori, this function has $n!$ summands and will therefore require special tools to handle with reasonable effort, especially in computerized implementations. Before delving into the many beautiful properties of this strange formula, let us give some easy examples:

Example 95 For $n = 1$, we have $\det(M) = M_{11}$, and for $n = 2$, we have the formula

$$\det(M) = M_{11}M_{22} - M_{21}M_{12}$$

which is well known from high school. For $n = 3$, we have

$$\begin{aligned} \det(M) = & M_{11}M_{22}M_{33} - M_{11}M_{32}M_{23} - M_{21}M_{12}M_{33} \\ & + M_{21}M_{32}M_{13} + M_{31}M_{12}M_{23} - M_{31}M_{22}M_{13} \end{aligned}$$

Exercise 100 Calculate the determinants of these matrixes, the first over \mathbb{C} , the second over $\mathbb{Z}[X]$:

$$\begin{pmatrix} 2 - i & 3 & \sqrt{3} + i \\ & 3 & 1 - i \end{pmatrix} \quad \begin{pmatrix} X^2 + 2 & 3X - 1 & 0 \\ & -X & X^3 + 5 & 4 \\ & & 0 & X + 12 & -8 \end{pmatrix}$$

Show that for any positive n , $\det(E_n) = 1$.

Proposition 181 For a ring R and a positive natural number n , we have these properties:

- (i) If $M \in \mathbb{M}_{n,n}(R)$, then $\det(M) = \det(M^T)$.
- (ii) (Column Additivity) If for a column $M_{\bullet,j}$ of $M \in \mathbb{M}_{n,n}(R)$, we have $M_{\bullet,j} = N + L$, and if $M|N$ is the matrix obtained from M after replacing $M_{\bullet,j}$ by N , while $M|L$ is the matrix obtained from M after replacing $M_{\bullet,j}$ by L , then $\det(M) = \det(M|N) + \det(M|L)$.

- (iii) (Row Additivity) If for a row $M_{i\bullet}$ of $M \in \mathbb{M}_{n,n}(R)$, we have $M_{i\bullet} = N + L$, and if $M|N$ is the matrix obtained from M after replacing $M_{i\bullet}$ by N , while $M|L$ is the matrix obtained from M after replacing $M_{i\bullet}$ by L , then $\det(M) = \det(M|N) + \det(M|L)$.
- (iv) (Column Homogeneity) If for a column $M_{\bullet j}$ of $M \in \mathbb{M}_{n,n}(R)$, we have $M_{\bullet j} = \lambda \cdot N$, and if $M|N$ is the matrix obtained from M after replacing $M_{\bullet j}$ by N , then $\det(M) = \lambda \cdot \det(M|N)$.
- (v) (Row Homogeneity) If for a row $M_{i\bullet}$ of $M \in \mathbb{M}_{n,n}(R)$, we have $M_{i\bullet} = \lambda \cdot N$, and if $M|N$ is the matrix obtained from M after replacing $M_{i\bullet}$ by N , then $\det(M) = \lambda \cdot \det(M|N)$.
- (vi) (Column Skew Symmetry) If M' is obtained from $M \in \mathbb{M}_{n,n}(R)$ by exchanging two columns $M_{\bullet j}, M_{\bullet k}$, with $j \neq k$, then $\det(M') = -\det(M)$.
- (vii) (Row Skew Symmetry) If M' is obtained from $M \in \mathbb{M}_{n,n}(R)$ by exchanging two rows $M_{i\bullet}, M_{k\bullet}$, with $i \neq k$, then $\det(M') = -\det(M)$.
- (viii) (Column Equality Annihilation) If in $M \in \mathbb{M}_{n,n}(R)$, we have two equal columns $M_{\bullet j} = M_{\bullet k}$ with $j \neq k$, then $\det(M) = 0$.
- (ix) (Row Equality Annihilation) If in $M \in \mathbb{M}_{n,n}(R)$, we have two equal rows $M_{i\bullet} = M_{k\bullet}$ with $i \neq k$, then $\det(M) = 0$.
- (x) (Uniqueness of Determinant) Any function $D : \mathbb{M}_{n,n}(R) \rightarrow R$ with properties (ii), (iv), (viii) is uniquely determined by its value $D(E_n)$, and then we have $D(M) = D(E_n) \cdot \det(M)$.
- (xi) (Product Rule for Determinants) If $M, N \in \mathbb{M}_{n,n}(R)$, then

$$\det(M \cdot N) = \det(M) \cdot \det(N).$$

- (xii) (General Linear Group Homomorphism) The determinant function induces a homomorphism

$$\det : \text{GL}_n(R) \rightarrow R^*$$

onto the multiplicative group R^* of invertible elements of R . Its kernel is denoted by $\text{SL}_n(R)$ and called the special linear group of size n , whence $\text{GL}_n(R)/\text{SL}_n(R) \xrightarrow{\sim} R^*$.

- (xiii) (Invariance under Conjugation) If $M \in \mathbb{M}_{n,n}(R)$ and $C \in \text{GL}_n(R)$, then

$$\det(C \cdot M \cdot C^{-1}) = \det(M),$$

the matrix $C \cdot M \cdot C^{-1}$ being called the C -conjugate of M .¹

¹ Do not confuse matrix conjugation with the conjugation of a complex number.

Proof We have

$$\begin{aligned}
 \det(M) &= \sum_{\pi \in S_n} (-1)^{\text{sig}(\pi)} \prod_{j=1 \dots n} M_{\pi(j)j} \\
 &= \sum_{\pi \in S_n} (-1)^{\text{sig}(\pi)} \prod_{\pi(j)=1 \dots n} M_{\pi(j)\pi^{-1}(\pi(j))} \\
 &= \sum_{\pi \in S_n} (-1)^{\text{sig}(\pi)} \prod_{j=1 \dots n} M_{j\pi^{-1}(j)} \\
 &= \sum_{\pi \in S_n} (-1)^{\text{sig}(\pi)} \prod_{j=1 \dots n} M_{j\pi(j)} \\
 &= \sum_{\pi \in S_n} (-1)^{\text{sig}(\pi)} \prod_{j=1 \dots n} M_{\pi(j)j}^{\tau} \\
 &= \det(M^{\tau}),
 \end{aligned}$$

whence (i).

If we have $M_{ij} = N_i + L_i$, for all i , then for each product in the determinant function, we have

$$\begin{aligned}
 \prod_{t=1 \dots n} M_{\pi(t)t} &= M_{\pi(j)j} \prod_{t \neq j} M_{\pi(t)t} \\
 &= N_{\pi(j)} \prod_{t \neq j} M_{\pi(t)t} + L_{\pi(j)} \prod_{t \neq j} M_{\pi(t)t} \\
 &= (M|N)_{\pi(j)j} \prod_{t \neq j} M_{\pi(t)t} + (M|L)_{\pi(j)j} \prod_{t \neq j} M_{\pi(t)t} \\
 &= (M|N)_{\pi(j)j} \prod_{t \neq j} (M|N)_{\pi(t)t} + (M|L)_{\pi(j)j} \prod_{t \neq j} (M|N)_{\pi(t)t} \\
 &= \prod_t (M|N)_{\pi(t)t} + \prod_t (M|L)_{\pi(t)t}
 \end{aligned}$$

whence (ii).

To prove (iii), we use (ii) because of (i).

For column homogeneity (iv), we observe that

$$\begin{aligned}
 \prod_t M_{\pi(t)t} &= M_{\pi(j)j} \prod_{t \neq j} M_{\pi(t)t} \\
 &= \lambda(N_{\pi(j)}) \prod_{t \neq j} M_{\pi(t)t} \\
 &= \prod_t (M|N)_{\pi(t)t}.
 \end{aligned}$$

Row homogeneity (v) follows from (iv) and (i).

Suppose that (viii) is true. Then (vi) follows immediately. In fact, take a matrix M and two column indexes k and j . Take the new matrix M' which is derived from M by adding column $M_{\bullet k}$ to column $M_{\bullet j}$ and adding $M_{\bullet j}$ to column $M_{\bullet k}$. Then by (viii), $\det(M') = 0$. But by (iii), $0 = \det(M') = \det(M|k|k) + \det(M|j|j) +$

$\det(M|j|k) + \det(M)$, where $M|k|k$ is the matrix derived from M , where we have the k -th column $M_{\bullet k}$ at column positions k, j , and $M|j|j$ is the matrix derived from M , where we have the j -th column $M_{\bullet j}$ at column positions k and j , while $M|j|k$ is the matrix where the k -th and j -th columns of M have been exchanged. But by (viii) $\det(M|k|k) = \det(M|j|j) = 0$, whence $\det(M|j|k) = -\det(M)$. Also, by (i) and (vi), (vii) follows.

To prove (viii), recall that the number of even permutations is $n!/2$ and that for each even permutation π there is an odd permutation $\pi^* = (\pi(k), \pi(j)) \circ \pi$. This gives us a bijection from even to odd permutations. But then, the product $\prod_t M_{\pi(t)t}$ is equal to the product $\prod_t M_{\pi^*(t)t}$ since both columns at positions k and j are equal. So by the change of signs, i.e., $(-1)^{\pi^*} = -(-1)^\pi$, these products neutralize each other. Further, by (i) and (viii), (ix) follows.

Claim (x) is demonstrated as follows: Each column is the sum of special columns N , where only one coefficient N_i is possibly different from zero. So the function $D(M)$ is determined on matrixes having only one coefficient possibly different from zero. But such a column N is also the scaling $N = N_i \cdot N'$, where N' has coefficient 1 instead of N_i , and zeros else. So by homogeneity in columns, the function D is determined by its values on matrixes which have only columns with a 1 and zero coefficients else. Now, by (viii), the value of our function D must vanish if two columns are equal. If not, the matrix M results from a permutation π of the columns of E_n , and the value must be $D(M) = (-1)^{\text{sig}(\pi)} D(E_n)$. Now, since $D(E_n) \cdot \det(M)$ has all the properties (ii),(iv),(viii) of D , and $D(E_n) \cdot \det(E_n) = D(E_n)$ we must have $D(M) = D(E_n) \det(M)$.

Suppose for (xi) that $M, N \in \mathbb{M}_{n,n}(R)$. Then, fixing M , the function $D(N) = \det(M \cdot N)$ evidently has the properties (ii), (iv), and (viii) by the laws of matrix multiplication. Since its value for $N = E_n$ is $\det(M)$, we are done by (x). Further, since by (xi) the determinant commutes with products of matrixes, it sends the product $E_n = M^{-1} \cdot M$ to $1 = \det(M^{-1}) \cdot \det(M)$, i.e., we have a group homomorphism $\text{GL}_n(R) \rightarrow R^*$, which is surjective, since the matrix $M = (\lambda - 1)E(1, 1) + E_n$ has $\det(M) = \lambda$.

Claim (xii) follows from (xi) and the fact that $\det(E_n) = 1$.

Claim (xiii) is evident from claim (xii). \square

The calculation of the inverse of an invertible square matrix uses the determinant function in a rather complex way. We first establish the necessary auxiliary structures.

Definition 161 For positive n , we denote by ${}_iM^j$ the $(n - 1) \times (n - 1)$ -matrix derived from $M \in \mathbb{M}_{n,n}(R)$ by the cancellation of its i -th row and j -th column. For $n > 1$, the determinant $\det({}_iM^j)$ is called the ij -minor of M . The number $\text{cof}(M)_{ij} = (-1)^{i+j} \det({}_jM^i)$ is called the ij -cofactor of

M . The matrix $Ad(M) = (cof(M)_{ij}) \in \mathbb{M}(R)$ is called the adjoint of M . If $n = 1$, we set $Ad(M) = E_1$.

Lemma 182 For a matrix $M \in \mathbb{M}_{n,n}(R)$, of size $n > 1$, if $1 \leq i \leq n$ is a row index, then

$$\det(M) = \sum_{j=1, \dots, n} M_{ij} cof(M)_{ji}.$$

If $1 \leq j \leq n$ is a column index, then

$$\det(M) = \sum_{i=1, \dots, n} M_{ij} cof(M)_{ji}.$$

Proof We have

$$\begin{aligned} \det(M) &= \sum_{\pi \in S_n} (-1)^{sig(\pi)} \prod_{t=1, \dots, n} M_{\pi(t)t} \\ &= \sum_{i=1, \dots, n} M_{ij} \sum_{\pi \in S_n, \pi(j) \neq i} (-1)^{sig(\pi)} \prod_{t \neq j} M_{\pi(t)t}. \end{aligned}$$

The factor $\sum_{\pi \in S_n, \pi(j) \neq i} (-1)^{sig(\pi)} \prod_{t \neq j} M_{\pi(t)t}$ is easily seen to be $cof(M)_{ji}$, whence the second formula. The first follows from the invariance of the determinant under transposition. \square

Proposition 183 (Cramer's Rule) For a matrix $M \in \mathbb{M}_{n,n}(R)$, of positive size n , we have the following equation:

$$M \cdot Ad(M) = \det(M) \cdot E_n.$$

Proof Cramer's rule is an immediate consequence of lemma 182. If we take the formula $\sum_{j=1, \dots, n} M_{ij} cof(M)_{ji}$ and change the coefficient i to $k \neq i$, then this is the same formula for the matrix deduced from M , where the row at index i is replaced by the row at index k . But such a matrix has determinant zero by row equality annihilation (ix) in proposition 181. So the Cramer formula results. \square

Proposition 184 A matrix $M \in \mathbb{M}_{n,n}(R)$, of positive size n is invertible iff $\det(M) \in R^*$. In particular, if R is a field, this means that $\det(M) \neq 0$. If M is invertible, then the inverse is given by this formula:

$$M^{-1} = \frac{1}{\det(M)} Ad(M).$$

Proof If M is invertible, we know that $\det(M) \in R^*$. Conversely, if $\det(M) \in R^*$, then Cramer's formula yields that the inverse is given by $M^{-1} = \frac{1}{\det(M)} Ad(M)$. \square

Exercise 101 Decide whether the matrix

$$M = \begin{pmatrix} 25 & -1 \\ 12 & 5 \end{pmatrix}$$

over \mathbb{Z} is invertible. Is its image $M \pmod{12}$ over \mathbb{Z}_{12} invertible? Try to calculate the adjoint and, if M is invertible, the inverse matrix of M .

Exercise 102 Show that, if $M \in \mathbb{M}_{n,n}(R)$ is an upper triangular matrix, i.e., $M_{ij} = 0$ for all $i > j$, then $\det(M) = \prod_{i=1}^n M_{ii}$.

Exercise 103 As a special case of matrix multiplications, we have already mentioned linear equations, such as shown in example 93 above. We are now in a position to solve such an equation

$$\begin{pmatrix} 3.7 \\ -8 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 23 & -1 & 0 & 45 \\ 0.9 & 9.6 & 1 & -1 \\ 0 & 20 & -1 & 1 \\ 0 & 3 & 1 & -2 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix}.$$

In fact, if the coefficient matrix

$$\begin{pmatrix} 23 & -1 & 0 & 45 \\ 0.9 & 9.6 & 1 & -1 \\ 0 & 20 & -1 & 1 \\ 0 & 3 & 1 & -2 \end{pmatrix}$$

is invertible, then the solution is

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 23 & -1 & 0 & 45 \\ 0.9 & 9.6 & 1 & -1 \\ 0 & 20 & -1 & 1 \\ 0 & 3 & 1 & -2 \end{pmatrix}^{-1} \cdot \begin{pmatrix} 3.7 \\ -8 \\ 0 \\ 1 \end{pmatrix}.$$

Proposition 185 If $f : R \rightarrow S$ is a ring homomorphism of commutative rings, then for a matrix $M \in \mathbb{M}_{n,n}(R)$ of positive size $n > 0$, we have

$$\det(f(M)) = f(\det(M)).$$

Proof We have $\det(f(M)) = \det(f((M_{ij})))$, but the determinant is a sum of products of the images $f(M_{ij})$, and since f is a ring homomorphism, we have $\det(f((M_{ij}))) = f(\det((M_{ij})))$. \square

Proposition 186 (Cayley-Hamilton Theorem) *If T is a commutative ring, and if $N \in \mathbb{M}_{n,n}(T)$ for positive n , the characteristic polynomial of N is the polynomial $\chi_N(X) = \det(N - X \cdot E_n) \in T[X]$, where $E_n \in \mathbb{M}_{n,n}(T)$ is the unit matrix of size n over T , and X is an indeterminate. Then we have the Cayley-Hamilton equation*

$$\chi_N(N) = 0 \in \mathbb{M}_{n,n}(T),$$

moreover,

$$\chi_N(0) = \det(N).$$

The coefficients of the characteristic polynomial are invariant under conjugation, i.e., if $C \in \text{GL}_n(T)$, then $\chi_N = \chi_{C \cdot N \cdot C^{-1}}$.

Proof Let $R = T[X]$ be the polynomial algebra over the commutative ring T with the indeterminate X . By the universal property of the polynomial algebra, if $N \in \mathbb{M}_{n,n}(T)$, we have a unique ring homomorphism $f : R \rightarrow \mathbb{M}_{n,n}(T)$ defined by $X \mapsto N$. Its image ring S is commutative, since the polynomial ring $T[X]$ is so. The ring S consists of all polynomials in N with coefficients in T . Therefore, by proposition 185, $\det(f(M)) = f(\det(M))$ for any matrix $M \in \mathbb{M}_{n,n}(T[X])$. In particular, if $M = N - X \cdot E_n$ for $N \in \mathbb{M}_{n,n}(T)$ for the unit matrix E_n in $\mathbb{M}_{n,n}(T)$, then $\det(M) = \chi_N(X) \in T[X]$, and we have $f(\chi_N(X)) = \chi_N(N)$. But also $f(\chi_N(X)) = f(\det(M)) = \det(f(M)) = \det(f(N - X \cdot E_n)) = \det(N - N) = 0$. Therefore,

$$\chi_N(N) = 0.$$

As for the invariance under conjugation, we have

$$\begin{aligned} \chi_{C \cdot N \cdot C^{-1}} &= \det(C \cdot N \cdot C^{-1} - X \cdot E_n) \\ &= \det(C \cdot N \cdot C^{-1} - X \cdot C \cdot C^{-1}) \\ &= \det(C \cdot (N - X \cdot E_n) \cdot C^{-1}) \\ &= \det(N - X \cdot E_n) \\ &= \chi_N. \end{aligned}$$

□

Exercise 104 Calculate the characteristic polynomial χ_N for the matrix

$$N = \begin{pmatrix} -1 & 0 \\ 2 & 3 \end{pmatrix}$$

over the integers. Verify the Cayley-Hamilton equation $\chi_N(N) = 0$.

Modules and Vector Spaces

If the matrixes are the backbones of linear algebra, here is the flesh: the axiomatic theory of modules and vector spaces, which encompasses a large variety of comparable structures. They will eventually be cast into matrixes in many interesting cases. The setup is drawn from the historic approach of René Descartes in his analytic geometry. *In this chapter, we shall again stick to commutative rings except when we explicitly state the contrary.*

Definition 162 *Let R be a ring, then a left R -module is a triple $(R, M, \mu : R \times M \rightarrow M)$, where M is an additively written abelian group of so-called vectors, and μ is the scalar multiplication, usually written as $\mu(r, m) = r \cdot m$ if μ is clear, with these properties:*

- (i) *We have $1 \cdot m = m$, for all $m \in M$.*
- (ii) *For all $r, s \in R$ and $m, n \in M$, we have*

$$(r + s) \cdot m = r \cdot m + s \cdot m$$

$$r \cdot (m + n) = r \cdot m + r \cdot n$$

$$r \cdot (s \cdot m) = (rs) \cdot m$$

Given an R -module M , a subgroup $S \subset M$ is called a submodule of M , if for each $r \in R$ and $s \in S$, then $r \cdot s \in S$. It is also an R -module on its own right. If the ring R is a field, the module is called an R -vector space.

Exercise 105 Show that by statement (ii) of definition 162, one always has $0 \cdot m = r \cdot 0 = 0$ in a module.

Example 96 In the course of the last chapter, we have encountered plenty of modules: Each set $M = \mathbb{M}_{m,n}(R)$, together with the sum of matrixes and the scalar multiplication by ring elements defined in sorite 177 is an R -module. In particular, we have zero modules $\mathbb{M}_{0,n}(R)$, $\mathbb{M}_{m,0}(R)$, $\mathbb{M}_{0,0}(R)$, each consisting of a zero group and the only possible scalar multiplication.

But the fundamental idea of defining matrixes as R -valued functions on certain domains ($[1, m] \times [1, n]$ for matrixes) can easily be generalized: Take any set D and consider the set R^D of functions on D with values in R . Then the addition $f + g$ of $f, g \in R^D$ defined by $(f + g)(d) = f(d) + g(d)$ and the scalar multiplication $(r \cdot f)(d) = r \cdot f(d)$ for $r \in R$ defines a module, which generalizes the idea for matrixes.

The following important subset of R^D is also a module under the same addition and scalar multiplication: the set $R^{(D)}$ of functions $f : D \rightarrow R$ such that $f(d) = 0$ except for at most a finite number of arguments. For example, if we consider the monoid algebra $R\langle M \rangle$ of a monoid M , this is precisely $R^{(M)}$, and the sum of its elements is the one we just defined. Moreover, the identification of R -elements r with the element $r \cdot e_M$ yields the scalar multiplication $r \cdot f$ for elements $f \in R\langle M \rangle$.

This idea generalizes as follows: We may view R as an R -module over itself: The vectors are the elements of R , the sum is the given sum in R , and the scalar multiplication is the ring multiplication. This module is of course “isomorphic” to $\mathbb{M}_{1,1}(R)$ (we shall define what a module isomorphism is in a few lines from here). It is called the *free R -module of dimension one* and denoted by ${}_R R$, or simply by R if the context is clear. With this special module in mind, suppose we are given a module M over R . Then for any set D , we have the module $M^{(D)}$ whose elements $f : D \rightarrow M$ vanish except for a finite set of arguments d , and where sum and scalar multiplication are again defined point-wise, i.e., $(f + g)(d) = f(d) + g(d)$ and $(r \cdot f)(d) = r \cdot f(d)$. This module is called the *direct sum of D copies of M* and usually denoted by $M^{\oplus D}$; in the special case where $D = n$ is a natural number, $M^{\oplus n}$ is also written as M^n . We now recognize that part of the structure of complex numbers \mathbb{C} can be regarded as the module \mathbb{R}^2 . In fact, addition of complex numbers is the vector addition on \mathbb{R}^2 , whereas the multiplication of a real number r with a complex number z plays the role of the scalar multiplication $r \cdot z$. The special case $M = {}_R R$ has been introduced above in the module $R^{(D)}$. We also recognize that the matrix module $\mathbb{M}_{m,n}(R)$ identifies with $R^{\oplus [1,m] \times [1,n]}$.

If we are given a finite family $(M_i)_{i=1,\dots,n}$ of R -modules, the Cartesian product $M_1 \times \dots \times M_n$ is given a module structure as follows: Vector addition and scalar multiplication are defined component-wise, i.e.,

$$(m_1, \dots, m_n) + (m'_1, \dots, m'_n) = (m_1 + m'_1, \dots, m_n + m'_n),$$

and

$$r \cdot (m_1, \dots, m_n) = (r \cdot m_1, \dots, r \cdot m_n).$$

This module is denoted by $\bigoplus_{i=1,\dots,n} M_i$ and is called the direct sum of the modules M_1, \dots, M_n .

The following example is a very comfortable restatement of abelian groups in terms of modules: Each abelian group M is automatically a \mathbb{Z} -module by the following scalar multiplication: One defines $z \cdot m = m + m + \dots + m$, z times, for $z > 0$, $-((-z) \cdot m)$ for $z < 0$, and $0 \cdot m = 0$. Verify that this construction satisfies the module axioms.

Already after these first examples one recognizes that many modules are manifestations of essentially the same structure. And this is why we once more return to the principle of morphisms, after we already used it for sets, digraphs, rings, and automata:

Definition 163 *If $(R, M, \mu : R \times M \rightarrow M)$ and $(R, N, \nu : R \times N \rightarrow N)$ are two R -modules, an R -linear homomorphism $f : M \rightarrow N$ is a group homomorphism such that for all $(r, m) \in R \times M$, $f(\mu(r, m)) = \nu(r, f(m))$. If no ambiguity about the scalar multiplications in M and in N is likely, one uses the dot notation, and then linearity reads as $f(r \cdot m) = r \cdot f(m)$.*

The set of R -linear homomorphisms $f : M \rightarrow N$ is denoted by $\text{Lin}_R(M, N)$. By point-wise addition and scalar multiplication, $\text{Lin}_R(M, N)$ is also provided with the structure of an R -module, which we henceforth tacitly assume.

If L is a third R -module, the composition $g \circ f$ of two R -linear homomorphisms $f \in \text{Lin}_R(M, N)$ and $g \in \text{Lin}_R(N, L)$ is defined by their set-theoretic composition; it is again R -linear.

For $M = N$, one writes $\text{End}_R(M) = \text{Lin}_R(M, M)$ and calls its elements R -module endomorphisms. In particular, the identity $\text{Id}_M : M \rightarrow M$ is an R -module endomorphism on M .

An R -linear homomorphism $f : M \rightarrow N$ is called an isomorphism if it has an inverse $g : N \rightarrow M$ such that $g \circ f = \text{Id}_M$ and $f \circ g = \text{Id}_N$. Evidently,

this is the case iff the underlying group homomorphism is a group isomorphism; the inverse is uniquely determined and denoted by $g = f^{-1}$. If, moreover, $M = N$, an isomorphism is called automorphism.

Exercise 106 Show that the matrix R -module $\mathbb{M}_{m,n}(R)$ is isomorphic to R^{mn} . In particular, the column matrix module $\mathbb{M}_{m,1}(R)$ and the row matrix module $\mathbb{M}_{1,m}(R)$ are both isomorphic to R^m , and all modules $\mathbb{M}_{0,n}(R)$, $\mathbb{M}_{m,0}(R)$, $\mathbb{M}_{0,0}(R)$ and R^0 are isomorphic, i.e., they are trivial R -modules.

Exercise 107 Show that the ring $\text{Lin}_R(R, R)$ is isomorphic to R , in particular, observe that therefore these objects are also isomorphic as R -modules.

Sorte 187 If $f \in \text{Lin}_R(N, L)$ and $g \in \text{Lin}_R(M, N)$, then

- (i) if $f = f_1 + f_2$, then $f \circ g = (f_1 + f_2) \circ g = f_1 \circ g + f_2 \circ g$, whereas for $g = g_1 + g_2$, $f \circ g = f \circ (g_1 + g_2) = f \circ g_1 + f \circ g_2$.
- (ii) If $r \in R$, then $r \cdot (f \circ g) = (r \cdot f) \circ g = f \circ (r \cdot g)$.
- (iii) With the addition and composition of linear endomorphisms on M , the set $\text{End}_R(M)$ is a (generally not commutative) ring. If $M \neq 0$, R identifies with the subring $R \cdot \text{Id}_M$ by the ring isomorphism $R \xrightarrow{\sim} R \cdot \text{Id}_M : r \mapsto r \cdot \text{Id}_M$, and it commutes with every endomorphism. The group of invertible elements in $\text{End}_R(M)$ is denoted by $\text{GL}(M)$, it is called the general linear group of M .

Proof Let $g : M \rightarrow N$ and $f : N \rightarrow L$ be R -linear homomorphisms. If $f = f_1 + f_2$, then for $x \in M$, $((f_1 + f_2) \circ g)(x) = (f_1 + f_2)(g(x)) = f_1(g(x)) + f_2(g(x)) = (f_1 \circ g)(x) + (f_2 \circ g)(x) = ((f_1 \circ g) + (f_2 \circ g))(x)$. If $g = g_1 + g_2$, then $(f \circ (g_1 + g_2))(x) = f((g_1 + g_2)(x)) = f(g_1(x) + g_2(x)) = f(g_1(x)) + f(g_2(x)) = (f \circ g_1)(x) + (f \circ g_2)(x) = ((f \circ g_1) + (f \circ g_2))(x)$, whence (i).

If $r \in R$, then $(r \cdot (f \circ g))(x) = r \cdot ((f \circ g)(x)) = r \cdot (f(g(x)))$. But also $((r \cdot f) \circ g)(x) = ((r \cdot f)(g(x))) = r \cdot f(g(x))$. Finally $(f \circ (r \cdot g))(x) = f((r \cdot g)(x)) = f(r \cdot g(x)) = r \cdot f(g(x))$, whence (ii).

Claim (iii) follows now immediately from (i) and (ii). \square

Example 97 Consider the R -modules $\mathbb{M}_{n,1}(R)$ and $\mathbb{M}_{m,1}(R)$ of n -element and m -element columns. Then an $m \times n$ -matrix $M : E_n \rightarrow E_m$ defines a map $f_M : \mathbb{M}_{n,1}(R) \rightarrow \mathbb{M}_{m,1}(R)$ by the matrix multiplication $f_M(X) = M \cdot X$. This gives us an a posteriori justification of the functional notation of a

matrix. The general laws of matrix multiplication stated in sorite 179 imply that this map is R -linear. Moreover, applying f_M to the elementary column $E(i, 1) \in \mathbb{M}_{n,1}(R)$ gives us the column $M_{i\bullet}$ of M . Therefore the map $M \mapsto f_M$ is injective. We shall soon see that the map is often also surjective, i.e., the matrixes are essentially the same thing as linear homomorphisms! This is not always the case, but for a large class of rings, the fields, and therefore for all vector spaces, this is true.

Analogous to groups, one can also build quotient and image modules as follows:

Proposition 188 *Given an R -linear homomorphism $f : M \rightarrow N$, the image group $Im(f)$ is a submodule of N . The kernel $Ker(f)$ of the underlying group homomorphism is a submodule of M .*

If $S \subset M$ is a submodule of the R -module M , then the quotient group M/S is also an R -module by the scalar multiplication $r \cdot (m + S) = r \cdot m + S$. This will be called the quotient R -module.

Proof The fact that $Im(f)$ and $Ker(f)$ are submodules is immediate and left to the reader. If $S \subset M$ is a submodule, then the scalar multiplication is well defined, in fact, $m + S = m' + S$ iff $m - m' \in S$. But then $r \cdot m + S = r \cdot m' + S$ since $r \cdot m - r \cdot m' = r(m - m') \in S$. That this scalar multiplication satisfies the module axioms is then immediate. \square

And here is the universal property of quotient modules:

Proposition 189 *Let M be an R -module and $S \subset M$ a R -submodule of M . Then for every R -module N , we have a bijection*

$$Lin_R(M/S, N) \xrightarrow{\sim} \{f \mid f \in Lin_R(M, N) \text{ and } S \subset Ker(f)\}.$$

Proof Let $p : M \rightarrow M/S$ be the canonical projection. If $g : M/S \rightarrow N$ is R -linear, then the composition $g \circ p : M \rightarrow N$ is in the set on the right hand side. Since p is surjective, the map $g \mapsto g \circ p$ is injective (see the characterization of surjections of sets in sorite 16). Conversely, if $f : M \rightarrow N$ is such that $S \subset Ker(f)$, then we may define a map $g : M/S \rightarrow N : m + S \mapsto f(m)$. Is this map well defined? If $m + S = m' + S$, then $m - m' \in S$, hence $f(m) - f(m') = f(m - m') = 0$. It is evidently R -linear, and we are done, since $f = g \circ p$. \square

Here is the (double) universal property of the finite direct sum:

Proposition 190 *If M_1, \dots, M_n is a finite family of R -modules M_i , we have R -linear injections $\iota_j : M_j \rightarrow \bigoplus_i M_i : m \mapsto (0, \dots, 0, m, 0, \dots, 0)$ for each*

$j = 1, \dots, n$, which map an element $m \in M_j$ to the n -tuple having zeros except at the j -th position, where the element m is placed. For any R -module N , this defines a bijection

$$\text{Lin}_R\left(\bigoplus_i M_i, N\right) \xrightarrow{\sim} \bigoplus_i \text{Lin}_R(M_i, N)$$

between sets of homomorphisms defined by $f \mapsto (f \circ \iota_i)_i$.

Dually, we have R -linear projections $\pi_j : \bigoplus_i M_i \rightarrow M_j : (m_i)_i \mapsto m_j$, for each $j = 1, \dots, n$. For any R -module N , this defines a bijection

$$\text{Lin}_R\left(N, \bigoplus_i M_i\right) \xrightarrow{\sim} \bigoplus_i \text{Lin}_R(N, M_i)$$

between sets of homomorphisms defined by $f \mapsto (\pi_i \circ f)_i$.

Proof The isomorphisms $f \mapsto (f \circ \iota_i)_i$ and $f \mapsto (\pi_i \circ f)_i$ indicated in the proposition allow an immediate verification of the claims. We leave the details to the reader. \square

Exercise 108 Suppose we are given two subspaces $U, V \subset M$ and consider the homomorphism $f : U \oplus V \rightarrow M$ guaranteed by proposition 190 and these two inclusions. Then show that f is an isomorphism iff (i) $U \cap V = 0$ and (ii) $M = U + V$, which means that every $m \in M$ is a sum $m = u + v$ of $u \in U$ and $v \in V$. In this case M is also called the *inner direct sum of U and V* , and U and V are said to be *complements* to each other.

Remark 25 If we consider the two injections $i_M : M \rightarrow M \oplus N$ and $i_N : N \rightarrow M \oplus N$, an element $m \in M$ is mapped to $i_M(m) = (m, 0)$, while an element $n \in N$ is mapped to $i_N(n) = (0, n)$. If it is clear from the context that m belongs to the direct summand M and n to N , then one also identifies m with $i_M(m)$ and n to $i_N(n)$. With this identification, one may add m to n and write $m + n$ instead of $i_M(m) + i_N(n)$. From now on, this comfortable and economic notation will often be used without special emphasis.

The following enables the reduction of linear algebra to matrix algebra for a class of modules called *free*:

Definition 164 For a finite number n , an R -module M is called *free of dimension n* if it is isomorphic to R^n .

Attention: for general rings, a module is not necessarily free. A very simple example is a finite abelian group, such as \mathbb{Z}_n , which, as a \mathbb{Z} -module, cannot be free since any free non-zero \mathbb{Z} module is infinite. At present, you cannot know whether the dimension of a free module is uniquely determined. That it is in fact unique is shown by this result:

Proposition 191 *If an R -module M is free of dimension n and free of dimension m , then $n = m$, and this uniquely defined dimension is also called $\dim(M)$.*

Proof This follows from the properties of the determinant. Let $f : R^n \rightarrow R^m$ be an R -linear isomorphism with $n < m$. Any element $x = (x_1, \dots, x_n) \in R^n$ can be written as $x = \sum_{i=1, \dots, n} x_i e_i$, where $e_i = (0, \dots, 0, 1, 0, \dots, 0)$ has 1 at the i -th position. Then since every element $y \in R^m$ is an image under f , it can be written as $y = \sum_i x_i f(e_i)$. Now, consider the unit matrix E_m , for which we have $\det(E_m) = 1$. By the above, we may write each row $E(1, i) \in R^m$ as $E(1, j) = \sum_{i=1, \dots, n} x(j)_i f(e_i)$, for $x(j)_i \in R$, i.e., as a combination of less than m row vectors $f(e_i), i = 1, \dots, n$. Therefore, by the properties of the determinant, especially equal row annihilation, the determinant must vanish, a contradiction, therefore $n \geq m$. A symmetric argument shows $m \geq n$, whence the claim. \square

Example 98 The free \mathbb{R} -module \mathbb{R}^2 is called the real plane, and \mathbb{R}^3 is called the real three-dimensional space.

It will be shown in the course of the next chapter that every vector space has a dimension.

Proposition 192 *If M is a free R -module of dimension n , and if N is a free R -module of dimension m , then the R -module $\text{Lin}_R(M, N)$ is isomorphic to $\mathbb{M}_{m,n}(R)$, i.e., free of dimension mn .*

Proof We may wlog suppose that $M = R^n$ and $N = R^m$. Then by proposition 190, we have an R -linear isomorphism $\text{Lin}_R(M, N) \xrightarrow{\sim} \bigoplus_{i=1, \dots, m, j=1, \dots, n} \text{Lin}_R(R, R)$, with $\text{Lin}_R(R, R) \xrightarrow{\sim} R$, and therefore $\dim(\text{Lin}_R(M, N)) = n \cdot m$. One now maps the homomorphism f defined by the sequence $(m_{ij})_{i=1, \dots, m, j=1, \dots, n} \in \bigoplus_{i=1, \dots, m, j=1, \dots, n} R$ to the matrix M_f with $(M_f)_{ij} = m_{ij}$. This map is evidently a linear bijection. \square

We are now in the position to define the determinant of any linear endomorphism of a module M of dimension n by the following observation: If we have a free R -module M of dimension n and a linear endomorphism $f : M \rightarrow M$, then, if $u : M \rightarrow R^n$ is an isomorphism, we may consider the linear homomorphism $u \circ f \circ u^{-1} : R^n \rightarrow R^n$. This corresponds to a matrix $M_{f,u} \in \mathbb{M}_{n,n}(R)$. If we take another isomorphism $u' : M \rightarrow R^n$, then

we have the corresponding matrix $M_{f,u'} \in \mathbb{M}_{n,n}(R)$, and it easily follows that

$$M_{f,u'} = (u'u^{-1}) \cdot M_{f,u} \cdot (u'u^{-1})^{-1}.$$

Therefore, the determinant of f , if defined by

$$\det(f) = \det(M_{f,u})$$

is well defined by our previous result on conjugation of matrixes, see (xiii) of theorem 181.

Free modules are therefore fairly transparent, since their theory seems to reduce to matrix theory. However, we still have some unknown situations even with this easy type of modules: For example, if $f : R^n \rightarrow R^m$ is an R -linear map, what is the structure of $\text{Ker}(f)$ or $\text{Im}(f)$? Are these modules free? In general, they are not, but again, for vector spaces, this is true. This will be shown in the next chapter.

Linear Dependence, Bases, and Dimension

In practice, modules often do not occur as free constructions, but as subspaces, more precisely: kernels or even quotient spaces related to linear homomorphisms. For example, if we are given a matrix $M \in \mathbb{M}_{m,n}(R)$, the corresponding linear homomorphism $f_M : R^n \rightarrow R^m$ has a kernel $\text{Ker}(f_M)$ which plays a crucial role in the theory of linear equations. A linear equation is a matrix equation of this type:

$$\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

where the matrixes (y_i) and $M = (a_{ij})$ are given and one looks for the unknown column matrix (x_i) . One reinterprets this equation by the linear homomorphism $f_M : R^n \rightarrow R^m$ associated with M . We are given an element $y = (y_1, y_2, \dots, y_m) \in R^m$ and look for the inverse image $f_M^{-1}(y)$ of y under f_M . The solutions of the above equations are by definition the elements $x \in f_M^{-1}(y)$. So this solution space is structured as follows: If \tilde{x} is a particular solution, the solution space is

$$f_M^{-1}(y) = \tilde{x} + \text{Ker}(f_M).$$

This means that we have to deal with the following two problems: (1) Deciding if there is at least one particular solution and possibly find it. (2) Describing the kernel of f_M .

22.1 Bases in Vector Spaces

To tackle these problems, we shall restrict our theory to vector spaces, i.e., module over fields R during the rest of our discussion of linear algebra.

Definition 165 A finite sequence $(x_i) = (x_1, x_2, \dots, x_k)$, $k \geq 1$, of elements $x_i \in M$ of an R -vector space M is called linearly independent if one of the following equivalent properties holds:

- (i) A linear combination $\sum_{i=1, \dots, k} \lambda_i x_i$ of the vectors x_i equals 0 iff we have $\lambda_i = 0$ for all scalars λ_i .
- (ii) The R -linear homomorphism $f : R^k \rightarrow M$ defined by $f(\lambda_1, \dots, \lambda_k) = \sum_{i=1, \dots, k} \lambda_i x_i$ is injective, i.e., $\text{Ker}(f) = 0$.

If (x_i) is not linearly independent, the sequence is called linearly dependent.

Exercise 109 Give a proof of the equivalence of the properties in definition 165.

Exercise 110 Show that a linearly independent sequence (x_i) cannot contain the zero vector, nor can it contain the same vector twice. If (x_i) is linearly independent, then so is every permutation of this sequence. This means that linear independence is essentially a property of the underlying set $\{x_i, i = 1, \dots, k\}$ of vectors. However, there are many reasons to keep the sequential order here and in the following definitions of generators and bases, as well.

Exercise 111 Show that in the real vector space $\mathbb{R}^{(\mathbb{N})}$, every sequence $(e_i)_{i=0,1,2,\dots,k}$, $k \geq 1$ with $e_i = (0, \dots, 0, 1, 0, \dots)$ having a 1 exactly in position $i \in \mathbb{N}$ and 0 else, is linearly independent.

Exercise 112 Show that in a free vector space R^n of dimension n , the sequence $(x_i)_{i=1,2,\dots,n}$ of the vectors $x = (1, 1, \dots, 1, 0, 0 \dots 0)$ whose components are 1 up to and including the i -th coordinate, and 0 thereafter, is linearly independent.

Exercise 113 Consider the \mathbb{Q} -vector space \mathbb{R} , defined by the usual addition of "vectors", i.e., real numbers, and the usual scalar multiplication, but restricted to rational scalars. Show that the two vectors 1 and $\sqrt{2}$

are linearly independent over \mathbb{Q} . Use the results of exercise 69 about the irrationality of $\sqrt{2}$.

Definition 166 A finite sequence $(x_i) = (x_1, x_2, \dots, x_k)$, $k \geq 1$, of elements $x_i \in M$ of an R -vector space M is said to generate M if one of the following equivalent properties holds:

- (i) The vector space M equals the subspace of all linear combinations $\sum_{i=1, \dots, k} \lambda_i x_i$ of the vectors x_i (also called the space generated by (x_i)).
- (ii) The R -linear homomorphism $f : R^k \rightarrow M$ defined by $f(\lambda_1, \dots, \lambda_k) = \sum_{i=1, \dots, k} \lambda_i x_i$ is surjective, i.e., $\text{Im}(f) = M$.

A vector space M is called finitely generated if there is a finite sequence $(x_i) = (x_1, x_2, \dots, x_k)$, $k \geq 1$ of elements $x_i \in M$ which generates M .

Exercise 114 Give a proof of the equivalence of the properties in definition 166.

Exercise 115 Consider the \mathbb{R} -vector space $M = \mathbb{R}[X, Y]/(X^{12}, Y^{12})$. Show that it is generated by the images of $X^i Y^j$, $0 \leq i, j \leq 11$, in M .

Definition 167 A finite sequence $(x_i) = (x_1, x_2, \dots, x_k)$, $k \geq 1$ of elements $x_i \in M$ of an R -vector space M is called a basis of M iff it is linearly independent and generates M . Equivalently, (x_i) is a basis, iff the R -linear homomorphism $f : R^k \rightarrow M : (\lambda_1, \dots, \lambda_k) \mapsto \sum_{i=1, \dots, k} \lambda_i x_i$ is an isomorphism. Since by proposition 191, the dimension of M is uniquely determined, every basis of M must have the same number of elements, i.e., $k = \dim(M)$.

Remark 26 We have excluded the zero vector spaces here, because in those no finite sequence (x_1, \dots, x_k) , $k \geq 1$, can be linearly independent. To complete the general terminology, one also says that the empty sequence is linearly independent, and that it forms a basis for a zero space, but this is merely a convention.

Exercise 116 Show that in a free vector space R^n of dimension n , the sequence $(x_i)_{i=1, 2, \dots, n}$ of the vectors $x = (1, 1, \dots, 1, 0, 0 \dots 0)$ whose components are 1 up to and including the i -th coordinate, and 0 thereafter, is a basis of R^n . Show that the elementary matrixes $E(i, j)$ of $\mathbb{M}_{m, n}(R)$, $m, n > 0$, form a basis of this vector space (R being any field).

Here is the guarantee that bases always exist:

Proposition 193 *A vector space which is finitely generated has a basis, more precisely, for every finite sequence (x_i) of generators, there is a subsequence which is a basis of M .*

Proof Let $(x_1, \dots, x_k), k \geq 1$, be a generating sequence, then consider the first $x_i \neq 0$ (if there is none, we have the case of a zero space, and the “empty basis” does the job). The one-element sequence (x_{i_1}) is linearly independent since $\lambda \cdot x_{i_1} = 0$, with $\lambda \neq 0$, implies $\lambda^{-1} \cdot \lambda \cdot x_{i_1} = x_{i_1} = 0$, a contradiction. Suppose we have found a subsequence $(x_{i_1}, x_{i_2}, \dots, x_{i_r}), i_1 < i_2 < \dots < i_r$, of linearly independent vectors of maximal length. Then this generates the space for the following reason: If i is an index $i_1 < i < i_r$, then x_i is linearly dependent on the vectors $x_{i_j}, i_j \leq i$, by construction. If $i > i_r$, then there is a non-trivial linear combination $0 = \mu \cdot x_i + \sum_{j=1, \dots, r} x_{i_j}$ by the maximality of our sequence. But then $\mu \neq 0$, otherwise, we would have linear dependence of the maximal sequence. Therefore x_i is contained in the space generated by the maximal sequence $(x_{i_1}, x_{i_2}, \dots, x_{i_r})$, and we are done. \square

Here is the famous Steinitz exchange theorem, which guarantees that vector subspaces are always embedded in a distinct way:

Proposition 194 (Steinitz Exchange Theorem) *If (y_1, y_2, \dots, y_l) is a sequence of linearly independent vectors in a finitely generated vector space M , and if (x_1, x_2, \dots, x_k) is a basis of M (guaranteed by proposition 193), then $l \leq k$, and there is a (possibly empty) subsequence $(x_{i_1}, x_{i_2}, \dots, x_{i_{k-l}})$ of (x_1, x_2, \dots, x_k) such that $(y_1, y_2, \dots, y_l, x_{i_1}, x_{i_2}, \dots, x_{i_{k-l}})$ is a basis of M .*

Proof There is a representation $y_1 = \sum_i \lambda_i x_i$. Since $y_1 \neq 0$, there is a t , such that $\lambda_t \neq 0$. Then x_t is in the space generated by the sequence $(y_1, x_1, \dots, \hat{x}_t, \dots, x_r)$ (refer to the footnote of page 182 for the $\hat{}$ notation). But this is again a basis, since it generates the whole space and it is linearly independent. In fact, if $0 = \mu \cdot y_1 + \sum_{i=1, \dots, \hat{t}, \dots, r} \lambda_i x_i$ is a non-trivial linear combination, then necessarily, $\mu \neq 0$, but then y_1 also has a representation as a linear combination without x_t , so $0 = y_1 - y_1$ would have a non-trivial representation by the basis, a contradiction! Therefore we have a new basis $(y_1, x_1, \dots, \hat{x}_t, \dots, x_r)$. Suppose now that we have found a new basis $(y_1, y_2, \dots, y_r, x_{i_1}, x_{i_2}, \dots, x_{i_{k-l}}), r \leq l$. If $r = l$ we are done, otherwise we may proceed as initially with y_1 , however, we must show that we can still eliminate one of the remaining x_{i_j} . But if $y_{r+1} = \sum_{e=1, \dots, r} \mu_e y_e + \sum_{f=1, \dots, k-r} \lambda_f x_{i_f}$, then there must exist a $\lambda_{f_0} \neq 0$, otherwise, the y would be linearly dependent. So we may eliminate $x_{i_{f_0}}$ and we may proceed until all y are integrated in the basis. \square

Remark 27 The proof of the Steinitz theorem is quite algorithmic. Let us sketch the procedure: Suppose that we can find a linear combination

$y_j = \sum_{i=1, \dots, k} \lambda_{ji} x_i$ for each y_j . Then, starting with y_1 , take the first non-vanishing coefficient $\lambda_{1i(1)}$ in the sequence (λ_{1i}) , which exists since y_1 is not zero. Clearly, replacing $x_{i(1)}$ by y_1 in the basis (x_i) gives us a new basis. Now, suppose that we have already replaced some x_i by y_1, \dots, y_r and still have a basis. Now, y_{r+1} (if such a vector is left) is also a linear combination of these new basis elements. However, it is impossible that all coefficients of the remaining x_i vanish since then the (y_j) would not be linearly independent. So we may go on as in the beginning and replace one such x_i whose coefficient is not zero. This procedure eventually yields the new basis which contains all y_j .

Corollary 195 *If N is a subspace of a vector space M of dimension $\dim(M)$, then N is also finitely generated and $\dim(N) \leq \dim(M)$, equality holding iff $N = M$. Moreover, there is a subspace $C \subset M$ complementary to N , i.e., the homomorphism $N \oplus C \rightarrow M$ defined by the inclusions $C, N \subset M$ via the universal property of direct sums (proposition 190) is an isomorphism; in other words, M is the inner direct sum of N and C (see exercise 108).*

Proof We first show that N has a basis. If $N = 0$, we are done. Otherwise we take a maximal sequence (y_1, \dots, y_r) of linearly independent vectors in N . Then $r \leq k$ by Steinitz. This must generate N , otherwise, let $z \in N$ be a vector which is not a linear combination of y_1, \dots, y_r . Then evidently (y_1, \dots, y_r, z) is linearly independent, a contradiction. So let (y_1, \dots, y_l) be a basis of N , and (x_1, \dots, x_k) a basis of M . Then, by Steinitz, $l \leq k$. If $N = M$, then by uniqueness of dimension, $l = k$. If $l = k$, we may replace all of (x_1, \dots, x_k) by the basis elements y_1, \dots, y_r , and therefore $N = M$.

To find a complement of N , take the space spanned by the $k - l$ elements x of the old basis in the basis $(y_1, y_2, \dots, y_l, x_{i_1}, x_{i_2}, \dots, x_{i_{k-l}})$. This is clearly a complement. \square

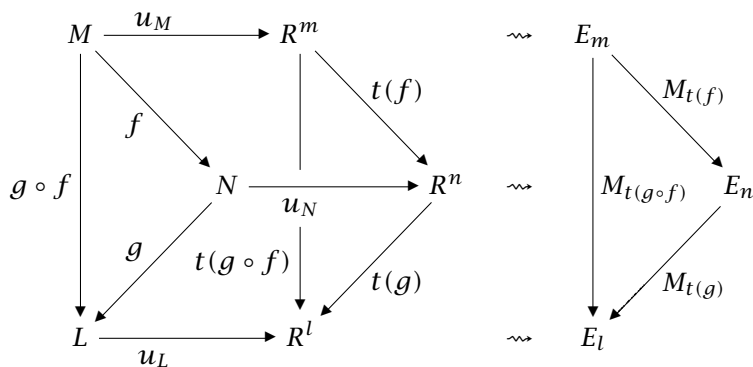
Remark 28 So we have this image in the case of finite-dimensional vector spaces over a fixed field R : If we fix an isomorphism $u_M : M \xrightarrow{\sim} R^{\dim(M)}$ for each R -vector space M , we obtain an isomorphism

$$t_{u_M, u_N} : \text{Lin}_R(M, N) \xrightarrow{\sim} \mathbb{M}_{\dim(N), \dim(M)}(R)$$

defined by the conjugation $f : M \rightarrow N \mapsto u_N \circ f \circ (u_M)^{-1}$ and the canonical interpretation of linear maps $R^n \rightarrow R^m$ as matrixes. In this setup, if we are given a second linear map $g : N \rightarrow L$, $\dim(L) = l$, then we have

$$t_{u_M, u_L}(g \circ f) = t_{u_N, u_L}(g) \cdot t_{u_M, u_N}(f) \text{ and } t_{u_M, u_M}(Id_M) = E_{\dim(M)},$$

i.e., the matrix product commutes with the composition of linear maps. This may be visualized by commutative diagrams:



We therefore have restated vector space structures in terms of matrixes, as predicted. But what happens if we change the basis of a vector space M ? Let us discuss this for endomorphisms of M which we transform into square matrixes in $\mathbb{M}_{n,n}(R)$ of size $n = \dim(M)$. Suppose we are given two bases which induce the isomorphisms $u, v : M \xrightarrow{\sim} R^n$. Then $u \circ v^{-1} : R^n \xrightarrow{\sim} R^n$ defines a matrix X such that we have

$$t_{v,v}(f) = X^{-1} \cdot t_{u,u}(f) \cdot X,$$

i.e., conjugation with the base change matrix X gives us the second matrix of f . In particular, if $M = R^n$, and if $v = Id_{R^n}$, then this formula gives us the matrix of f when calculated in the matrix representation from the new basis, whose elements are the column vectors of X . In other words:

Corollary 196 *If we have a new basis (x_i) of the vector space R^n given in terms of a matrix C of columns $C_{\cdot i}$ which correspond to x_i , then the representation of a linear map matrix $f : R^n \rightarrow R^n$ in terms of the basis (x_i) is $C^{-1} \cdot f \cdot C$.*

We can now state the relation between the dimensions of the kernel and the image of a linear map.

Corollary 197 *Let $f : M \rightarrow N$ be a linear homomorphism defined on a finite-dimensional R -vector space M (the vector space N need not be finite-dimensional). Then we have*

$$\dim(M) = \dim(\text{Im}(f)) + \dim(\text{Ker}(f)).$$

More precisely, there is a subspace $U \subset M$, which is a complement of $\text{Ker}(f)$, i.e., $M \cong U \oplus \text{Ker}(f)$, and such that $f|_U : U \rightarrow \text{Im}(f)$ is an isomorphism.

Proof Let U be a complement of $\text{Ker}(f)$ in M . Then $\dim(U) + \dim(\text{Ker}(f)) = \dim(M)$, by corollary 195. But the restriction $f|_U : U \rightarrow N$ is evidently a surjection onto $\text{Im}(f)$ since $\text{Ker}(f)$ is mapped to zero. Moreover, $\text{Ker}(f) \cap U = 0$ means that $\text{Ker}(f|_U) = 0$. Therefore $f|_U : U \xrightarrow{\sim} \text{Im}(f)$ is an isomorphism, and we are done. \square

Example 99 A simple example of this fact is illustrated in figure 22.1. Here we have a projection π of the 3-dimensional space \mathbb{R}^3 onto the 2-dimensional plane \mathbb{R}^2 . A point x in \mathbb{R}^3 is mapped to a point $\pi(x)$ in the plane of dimension 2, which is $\text{Im}(\pi)$. The points on the line through the origin O parallel to the projection axis are all mapped to O , i.e., these are all the points y such that $\pi(y) = O$. Thus this line of dimension 1 is $\text{Ker}(\pi)$. As predicted by corollary 197, $\dim(\text{Im}(\pi)) + \dim(\text{Ker}(\pi)) = \dim(\mathbb{R}^3)$, i.e., $2 + 1 = 3$.

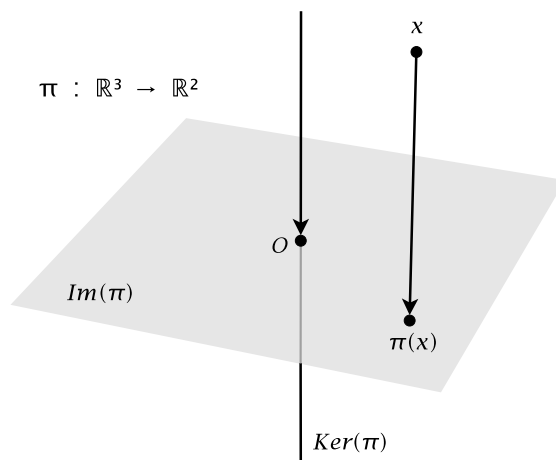


Fig. 22.1. The image and kernel of a projection.

Definition 168 The dimension $\dim(\text{Im}(f))$ of a linear homomorphism $f : M \rightarrow N$ of R -vector spaces is called the rank of f .

Here is the numerical criterion which allows us to calculate the rank of f in terms of its matrix:

Definition 169 The rank $rk(M)$ of a matrix $M \in \mathbb{M}_{m,n}(R)$, for positive m and n over a field R is the maximal r such that there is a square $r \times r$ -submatrix D of M with $\det(D) \neq 0$, $r = 0$ meaning that $M = 0$. By definition, such a submatrix is obtained by eliminating $m - r$ rows and $n - r$ columns from M .

Proposition 198 The rank of a linear homomorphism $f : M \rightarrow N$ of non-trivial R -vector spaces coincides with the rank of an associated matrix $t(f) \in \mathbb{M}_{\dim(N), \dim(M)}(R)$ with respect to selected bases.

Proof We know that the matrix $t(f)$ of a linear homomorphism $f : M \rightarrow N$ with respect to selected bases of M and N is described as follows: The associated linear map $g : R^n \rightarrow R^m$ with $n = \dim(M)$ and $m = \dim(N)$, has the images $g(e_j)^\tau = t(f)_{\bullet j}$ for the canonical basis $e_i = E(i, 1)_{\bullet 1}$, $i = 1, \dots, n$, of R^n , and of course $rk(g) = rk(f)$. If we take the submatrix of $t(f)$ defined by selecting $q > rk(f)$ columns, then these q columns are linearly dependent. This remains true if we cancel all but q rows of this matrix. After canceling, we have a $q \times q$ submatrix with linearly dependent columns, which clearly has zero determinant. So the rank of $t(f)$ is at most $rk(f)$. Take $k = rk(f)$ columns, which generate a basis of $Im(g)$. Then, using Steinitz, complete we can complete this basis by elementary columns $e_{\pi(j)}$, $j = k + 1, \dots, m$. This yields a $m \times m$ matrix whose determinant does not vanish. But the determinant only changes its sign if we permute columns or rows. We may obviously exchange rows and columns such that the new matrix has as last $m - k$ columns the elementary columns e_{k+1}, \dots, e_m , filling up the diagonal with 1s after the columns associated with the basis of $Im(g)$. But then the determinant is the necessarily non-zero determinant of the upper left $k \times k$ -block of the matrix, so the rank of $t(f)$ is at least k and we are done. \square

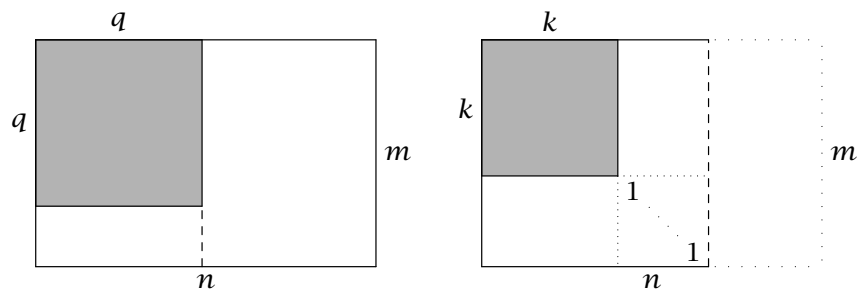


Fig. 22.2. Illustrating the proof of proposition 198. Left: A $q \times q$ submatrix with linearly dependent column vectors. Right: A $k \times k$ submatrix extended to a regular $m \times m$ matrix.

22.2 Equations

We may now decide upon the existence of a solution of the linear equation

$$\begin{pmatrix} \mathcal{Y}_1 \\ \mathcal{Y}_2 \\ \vdots \\ \mathcal{Y}_m \end{pmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{pmatrix}$$

with given matrixes (\mathcal{Y}_i) , $M = (a_{ij})$ and unknown (x_i) : Let (\mathcal{Y}, M) be the $m \times (n + 1)$ -matrix obtained by prepending the column \mathcal{Y} to the left of M . Here is the somewhat redundant but useful list of cases for solutions of the system:

Proposition 199 *With the preceding notations, the linear equation $\mathcal{Y} = M \cdot x$*

- (i) *has a solution iff $rk(M) = rk((\mathcal{Y}, M))$;*
- (ii) *has at most one solution if $rk(M) = n$;*
- (iii) *has exactly one solution iff $rk((\mathcal{Y}, M)) = rk(M) = n$;*
- (iv) *has the unique solution $x = M^{-1} \cdot \mathcal{Y}$ in the case $m = n = rk(M)$ (so-called "regular equation").*
- (v) *Given one solution x_0 , and a basis $(z_t)_{t=1, \dots, s}$ of the kernel of the linear map f_M associated with M , the solution space consists of the coset $x_0 + Ker(f_M)$, i.e., all the vectors $x_0 + \sum_{t=1, \dots, s} \lambda_t z_t$, $\lambda_t \in R$. The elements of $Ker(f_M)$ are also called the solutions of the homogeneous equation $0 = M \cdot x$ associated with $\mathcal{Y} = M \cdot x$.*

Proof If the equation $\mathcal{Y} = M \cdot x$ has a solution, it is of the form $\mathcal{Y} = \sum_{j=1, \dots, d} \lambda_j M_{\bullet j}$ for a basis $(M_{\bullet j_1}^T, \dots, M_{\bullet j_d}^T)$ of the image of the homomorphism $f_M : R^n \rightarrow R^m$ associated with M . But then the matrix (\mathcal{Y}, M) has no regular square submatrix containing the column \mathcal{Y} , by the common column equality annihilation argument, whence $rk(M) = rk((\mathcal{Y}, M))$. Conversely, consider the linear map $h : R^n \oplus R \rightarrow R^m$ which on the first summand is f_M , and on the second summand just maps the basis 1 to \mathcal{Y} . Then since $rk(M) = rk((\mathcal{Y}, M))$, $dim(Im(h)) = dim(f_M)$, so the images are equal, and \mathcal{Y} is in the image of f_M , this proves (i).

As to (ii), if $rk(M) = n$, then by corollary 197, $Ker(f_M) = 0$ and f_M is injective.

If in (iii) we suppose $rk((\mathcal{Y}, M)) = rk(M) = n$, then by (i) and (ii), there is exactly one solution. Conversely, if there is exactly one solution, then there is a solution, and (i) shows $rk((\mathcal{Y}, M)) = rk(M)$, while if $rk(M) < n$ would yield a non-trivial

kernel, and for each solution y , we get another solution $y+w$ for $w \in \text{Ker}(f_M) - \{0\}$. Therefore also $\text{rk}(M) = n$.

Statements (iv) and (v) are clear, since the difference of any two solutions is in the kernel, and any solution y , when changed to $y+w$, $w \in \text{Ker}(f_M)$, yields another solution. \square

In chapter 23, we shall present more algorithmic methods for finding solutions.

22.3 Affine Homomorphisms

It is sometimes customary to distinguish between vectors and points when dealing with elements of an R -vector space M . Why? Because vectors may play different roles within a space. So far we know that vectors are elements of a space which may be added and scaled. But all vectors play the same role. The representation of a vector $x \in M$ by means of its coordinate sequence $f(x) = (\lambda_1, \dots, \lambda_n)$ for a given basis (x_i) of M and the associated isomorphism $f: M \xrightarrow{\sim} R^n$ positions the vector in a coordinate system, whose axes are the 1-dimensional base spaces $R \cdot x_i$. This is the common image of traditional analytical geometry.

In “affine geometry” however, one adopts a slightly different point of view in that the addition $u + y$ of two vectors is restated as an operation of u upon y . This operation is denoted by $T^u: M \rightarrow M: y \mapsto T^u(y) = u + y$, and is called the *translation by u* . The exponential notation has its justification in the obvious formula $T^u \circ T^v = T^{u+v}$. In this understanding, y plays the role of a point which is shifted by the vector u to a new point $T^u(y) = u + y$. Clearly, we therefore have an injection $T: M \rightarrow \text{Sym}(M)$ of the additive group of M into the symmetric group of M ; denote by T^M the image *group of translations on M* . This identification of a vector u with its translation T^u creates two kinds of vectors in affine geometry: the given ones, y , and the associated operators T^y . In this way, addition of vectors is externalized as an operator on the “point set M ”. Like linear algebra, affine algebra deals with modules and in particular vector spaces, but the morphisms between such spaces are a little more general, since they also include translations. Here is the formal definition.

Definition 170 *If M and N are R -vector spaces, then a map $f: M \rightarrow N$ is called an R -affine homomorphism, if there is a map $g \in \text{Lin}_R(M, N)$ and a vector $u \in N$ such that $f = T^u \circ g$. The homomorphism g is called*

the linear part, whereas T^u is called the translation part of f . The set of affine homomorphisms $f : M \rightarrow N$ is denoted by $\text{Aff}_R(M, N)$. The group of invertible elements in the ring $\text{Aff}_R(M, M)$ of affine endomorphisms of M is denoted by $\text{GA}(M)$ and called the general affine group of M .

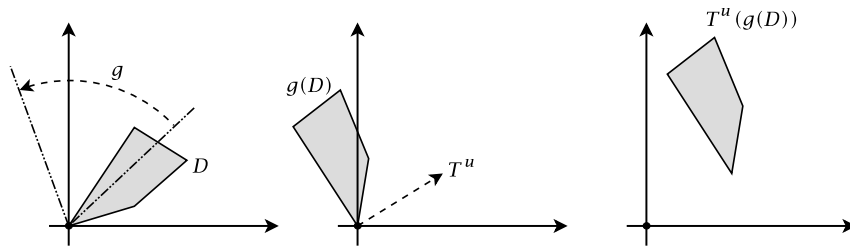


Fig. 22.3. An affine homomorphism $f = T^u \circ g$ on \mathbb{R}^2 is shown as a rotation g around the origin, followed by a translation T^u .

For an affine homomorphism $f = T^u \circ g$, the translation part $u = f(0)$ and the linear part $g = T^{-u} \circ f$ are uniquely determined by f . They are denoted by $u = \tau(f)$ and $g = \lambda(f)$, i.e.,

$$f = T^{\tau(f)} \circ \lambda(f).$$

Exercise 117 Show that together with the point-wise addition and scalar multiplication, $\text{Aff}_R(M, N)$ is also an R -vector space.

Lemma 200 If $f : M \rightarrow N$ and $g : N \rightarrow L$ are R -affine homomorphisms, then their composition $g \circ f : M \rightarrow L$ is R -affine and we have this formula:

$$g \circ f = (T^{\tau(g)} \circ \lambda(g)) \circ (T^{\tau(f)} \circ \lambda(f)) = T^{\tau(g) + \lambda(g)(\tau(f))} \circ (\lambda(g) \circ \lambda(f)).$$

The inverse of an affine isomorphism $f = T^{\tau(f)} \circ \lambda(f)$ (i.e., its linear part is an isomorphism) is given by the formula

$$f^{-1} = T^{-\lambda(f)^{-1}(\tau(f))} \circ \lambda(f)^{-1}.$$

Proof The lemma follows without any trick from the explicit formulas, which it presents. We therefore leave it to the reader. □

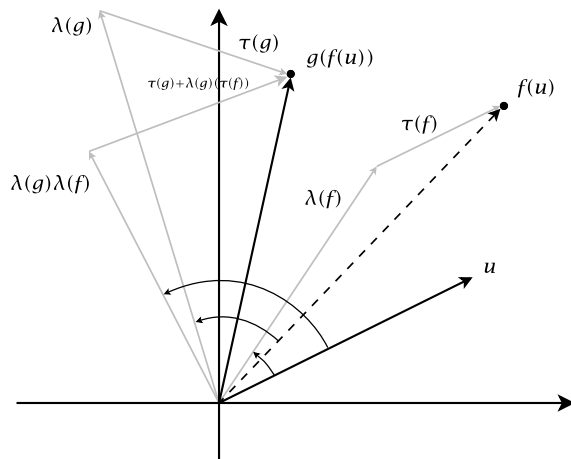


Fig. 22.4. Composition $g \circ f$ of affine homomorphisms g and f acting on a vector u .

Exercise 118 Show that the group T^M of translations is a normal subgroup of $\text{GA}(M)$. Show that $\text{GA}(M)/T^M \simeq \text{GL}(M)$.

It has turned out advantageous to represent linear maps by matrixes, so let us see how this can be obtained for affine maps. Of course, the usual representation does not work, because, in general, affine maps do leave origin fixed. But there is a beautiful trick to interpret an affine map as if it would. The trick consists in inventing a new origin and embedding the given vector space in a larger space, where we have a linear map which on the embedded space behaves like the given affine map. The new system is related to what is known as the method of “homogeneous coordinates”. We have the affine injection

$$\eta_M : M \rightarrow M \oplus R : m \mapsto (m, 1)$$

of *homogenization*, i.e., $\eta_M = T^{(0,1)} \circ i_1$, where i_1 is the usual linear embedding of M as first summand of $M \oplus R$. Then, for a given affine homomorphism $f : M \rightarrow N$, we consider the linear homomorphism

$$\hat{f} : M \oplus R \rightarrow N \oplus R : (m, r) \mapsto (\lambda(f)(m), 0) + r(\tau(f), 1)$$

associated with an affine homomorphism $f : M \rightarrow N$. Clearly, \hat{f} sends $\eta_M(M)$ to $\eta_N(N)$. More precisely, we have

$$\hat{f} \circ \eta_M = \eta_N \circ f,$$

which is best represented as a commutative diagram

$$\begin{array}{ccc} M & \xrightarrow{\eta_M} & M \oplus R \\ \downarrow f & & \downarrow \hat{f} \\ N & \xrightarrow{\eta_N} & N \oplus R \end{array}$$

We may get back f by the formula

$$\pi_1 \circ \hat{f} \circ \eta_M = f$$

with the first projection $\pi_1 : N \oplus R \rightarrow N$, since $\pi_1 \circ \eta_N = Id_N$. Because $\eta_M(m) = (m, 1)$, then we have $\hat{f}((m, 1)) = (f(m), 1)$. If M is identified with R^n and N is identified with R^m by the choice of two bases, then the linear part $\lambda(f)$ identifies with a $m \times n$ -matrix (a_{ij}) , the translation vector $\tau(f)$ with a column vector (t_{i1}) and \hat{f} can be represented by the matrix

$$\begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} & t_{11} \\ \vdots & \vdots & & \vdots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & t_{m1} \\ 0 & 0 & \dots & 0 & 1 \end{pmatrix}$$

which evidently sends a column (y_{i1}) from the image $\eta_M(M)$ with last coordinate 1 to a column with the same last coordinate. The *homogeneous coordinates* of a column vector are the given coordinates plus the new last coordinate 1.

Exercise 119 Show that $\widehat{Id}_M = Id_{M \oplus R}$. Given two affine homomorphisms $f : M \rightarrow N$ and $g : N \rightarrow L$, show that $\widehat{g \circ f} = \widehat{g} \circ \widehat{f}$.

Example 100 Anticipating the concept of angles and associated matrixes, the counter-clockwise rotation R_0 by 60 degrees around the origin $(0, 0)$ in \mathbb{R}^2 is given by the matrix

$$M_{R_0} = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix}$$

This subject will be treated properly in section 24, but for the moment, we may just recall the high school education in mathematics. We now

consider the counter-clockwise rotation R_p of 60 degrees around any point $p = (x, y)$. Figure 22.5 illustrates the case of $p = (2, 1)$ and the value $R_p(a)$ of the point $a = (-\frac{1}{2}, -1)$. The rotation R_p is an affine trans-

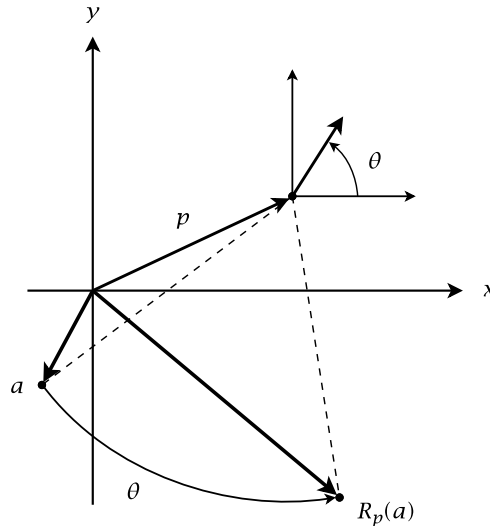


Fig. 22.5. The rotation $R_p(a)$ of the point $a = (-\frac{1}{2}, -1)$ by $\theta = 60$ degrees around the point $p = (2, 1)$.

formation on \mathbb{R}^2 by the following argument: consider the composition $T^{-p} \circ R_p \circ T^p$. This map fixes the origin and is in fact the counter-clockwise rotation R_0 of 60 degrees around the origin $(0, 0)$. Then the equation $R_0 = T^{-p} \circ R_p \circ T^p$ yields $R_p = T^p \circ R_0 \circ T^{-p} = T^{\Delta_p} \circ R_0$ with $\Delta_p = p - R_0(p)$. Let us calculate the numeric values and the 3×3 -matrix of \hat{R}_p in terms of homogeneous coordinates for the concrete vector $p = (2, 1)$. We have

$$\Delta_p = \begin{pmatrix} 2 \\ 1 \end{pmatrix} - \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 + \frac{\sqrt{3}}{2} \\ \frac{1}{2} - \sqrt{3} \end{pmatrix}$$

and therefore the matrix $M_{\hat{R}_p}$ of \hat{R}_p is

$$M_{\hat{R}_p} = \begin{pmatrix} M_{R_0} & \Delta_p \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 + \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} - \sqrt{3} \\ 0 & 0 & 1 \end{pmatrix}.$$

The transformation corresponding to $M_{\hat{R}_p}$ is shown in figure 22.6. If applied to a vector $a = (-\frac{1}{2}, -1)$, rewritten in homogeneous coordinates $\hat{a} = (-\frac{1}{2}, -1, 1)$, we get the product

$$\hat{R}_p(\hat{a})^\tau = \begin{pmatrix} \frac{1}{2} & -\frac{\sqrt{3}}{2} & 1 + \frac{\sqrt{3}}{2} \\ \frac{\sqrt{3}}{2} & \frac{1}{2} & \frac{1}{2} - \sqrt{3} \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} -\frac{1}{2} \\ -1 \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{3}{4} + \sqrt{3} \\ -\frac{5\sqrt{3}}{4} \\ 1 \end{pmatrix}.$$

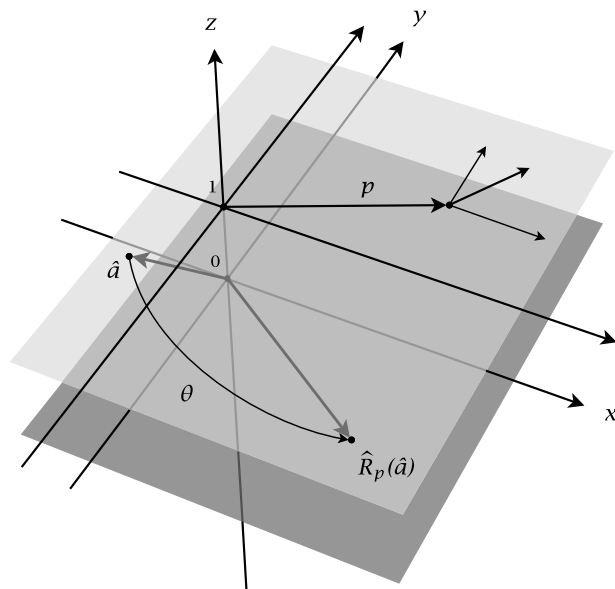


Fig. 22.6. The same transformation as in figure 22.5, but this time in homogeneous coordinates, i.e., $\hat{R}_p(\hat{a})$, where $\hat{a} = (-\frac{1}{2}, -1, 1)$.

Algorithms in Linear Algebra

In practice, one needs algorithmic methods to calculate matrixes which arise from problems in linear algebra or vector spaces. In particular, it is important to obtain solutions of linear equations, to calculate determinants and inverse matrixes. This is a vast field of numerical mathematics and of ongoing research, since fast and reliable algorithms for special matrixes are of crucial importance for the entire computer-aided science, be it physics, economics, chemistry or biology. There are also very important applications of matrix calculations in special technological fields such as signal processing (e.g., Fast Fourier Transform, see later in the second volume of this book), which are particularly sensitive to fast methods because they pertain to real-time applications. We shall only discuss some very elementary algorithms here in order to give a first idea of the subject. For object-oriented implementations of these and other algorithms, we refer to [6].

23.1 Gauss Elimination

The Gauss algorithm is based on two simple observations about the solutions of linear equations:

1. If we are given a linear equation $y = f(x)$ for an R -linear map $f : M \rightarrow N$, the solutions x are unchanged if we compose f with a $g \in \text{GL}(N)$ and thus obtain the new equation $g(y) = g(f(x)) = (g \circ f)(x)$.

2. If we have a well-known $h \in \text{GL}(M)$, then we may rewrite the equation as $y = f(x) = (f \circ h^{-1}) \circ h(x)$, and hope that the new unknown $h(x)$ is easier to manage than the original one.

The first observation can be applied to matrixes as follows: We are given a system of linear equations

$$y_i = \sum_{j=0, \dots, n} a_{ij} \cdot x_j$$

with m equations ($i = 1, \dots, m$) and n unknowns ($j = 1, \dots, n$). This can be written as matrix equation

$$y = A \cdot x$$

where $A = (a_{ij})$, $x = (x_j)$ and $y = (y_i)$. Then the solution set is left unchanged if we multiply both sides by an invertible $m \times m$ -matrix B , obtaining the new matrix equation

$$y' = B \cdot y = (B \cdot A) \cdot x = A' \cdot x.$$

The second observation suggests that we take an invertible $n \times n$ -matrix C and rewrite

$$y = A \cdot x = (A \cdot C^{-1}) \cdot (C \cdot x) = A' \cdot x'$$

with the new unknowns

$$x' = C \cdot x.$$

The method of Gauss elimination consists of a clever choice of a series of transformations B and C which successively change the equation until it has a particularly simple shape. Mostly, Gauss elimination is applied in the case of a regular system of n equations with n unknowns, i.e., the coefficient matrix A is supposed to be invertible, such that we have exactly one solution.

The idea of this procedure is to obtain an upper triangular coefficient matrix A , i.e., $a_{ij} = 0$ for $i > j$. This means that the diagonal elements $a_{ii} \neq 0$, because their product is the non-zero determinant. Then we may solve the equation system by *backward substitution*, which means that we first calculate x_n , then x_{n-1} , etc. until we obtain the solution of x_1 . In fact, a triangular matrix yields these n equations:

$$y_i = \sum_{j=i,i+1,\dots,n} a_{ij}x_j$$

which can be solved recursively, beginning with

$$x_n = \frac{1}{a_{nn}}y_n$$

and yielding

$$x_i = \frac{y_i - \sum_{j=i+1,\dots,n} a_{ij}x_j}{a_{ii}}$$

from the calculation of $x_n, x_{n-1}, \dots, x_{i+1}$.

So we are left with the construction of an upper triangular coefficient matrix. Again, this is a recursive construction, proceeding along the size n . To begin with, one would like to have $a_{11} \neq 0$. If this is not the case, we look for a first index $k > 1$ such that $a_{1k} \neq 0$. This exists since otherwise, the determinant of A would vanish. Now we rename the unknowns: x_1 becomes x_k , and x_k becomes x_1 . This is achieved by replacing A with $A' = A \cdot P(1, k)$ and the column x with $x' = P(1, k) \cdot x$. The matrix $P(1, k)$ corresponds to the transposition $(1k)$ in the symmetric group S_n and is defined by $P(1, k) = E_n - E(1, 1) - E(k, k) + E(k, 1) + E(1, k)$. Multiplying A from the right by $P(1, k)$ exchanges column 1 and k of A , whereas multiplying x by $P(1, k)$ from the left, exchanges x_1 and x_k , and this does not change anything in the product $y = A' \cdot x'$, since $P(1, k)^2 = E_n$. Therefore, up to renaming x_1 to x'_k , and x_k to x'_1 , and leaving all other unknowns $x_j = x'_j$, we have achieved that $a_{11} \neq 0$. Next, we find an invertible B such that $A' = B \cdot A$ has zeros $a'_{ij} = 0$ $i > j$. This is the matrix:

$$B = E_n - \sum_{i=2,\dots,n} \frac{a_{i1}}{a_{11}} \cdot E(i, 1).$$

It is invertible (the determinant is 1) and the product $A' = B \cdot A$ has zeros $a'_{i1} = 0$, for $i > 1$. Proceeding recursively, the situation is restricted to the equations $i = 2, \dots, n$ involving only the unknowns x_2, \dots, x_n . This settles the problem.

Example 101 Consider the system of equations

$$\begin{aligned} 2x_1 + 3x_2 + 4x_3 &= 8 \\ 4x_1 + 2x_2 - x_3 &= -3 \\ 2x_1 - 3x_2 + 3x_3 &= 17 \end{aligned}$$

This can be rewritten as

$$A \cdot x = y$$

where

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 4 & 2 & -1 \\ 2 & -3 & 3 \end{pmatrix} \text{ and } y = \begin{pmatrix} 8 \\ -3 \\ 17 \end{pmatrix}$$

To perform Gauss elimination, we first have to transform a into an upper triangular matrix. In a first step, we will multiply the equation with a matrix B that nullifies the second and third entries in the first column. After that, a matrix C will be used that nullifies the third entry in the second column, i.e.,

$$C \cdot B \cdot A \cdot x = C \cdot B \cdot y$$

where $C \cdot B \cdot A$ will be an upper triangular matrix. As defined above, the matrix B is given by

$$B = E_3 - \sum_{i=2,3} \frac{a_{i1}}{a_{11}} \cdot E(i,1)$$

i.e.,

$$B = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} - \left[\frac{4}{2} \cdot \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} + \frac{2}{2} \cdot \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \right] = \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$$

A simple calculation shows

$$B \cdot A = \begin{pmatrix} 2 & 3 & 4 \\ 0 & -4 & -9 \\ 0 & -6 & -1 \end{pmatrix} \text{ and } B \cdot y = \begin{pmatrix} 8 \\ -19 \\ 9 \end{pmatrix}$$

Now we have to nullify the third element in the second row of $B \cdot A$. For this, we look at the submatrix A' of A :

$$A' = \begin{pmatrix} -4 & -9 \\ -6 & -1 \end{pmatrix}$$

Using the same procedure as above, we get

$$C' = E_2 - \frac{a'_{21}}{a'_{11}} \cdot E(2,1) = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} - \frac{3}{2} \cdot \begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix} = \begin{pmatrix} 0 & 0 \\ -\frac{3}{2} & 0 \end{pmatrix}$$

This yields

$$C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -\frac{3}{2} & 1 \end{pmatrix}$$

hence

$$C \cdot B \cdot A = \begin{pmatrix} 2 & 3 & 4 \\ 0 & -4 & -9 \\ 0 & 0 & \frac{25}{2} \end{pmatrix} \text{ and } C \cdot B \cdot \mathbf{y} = \begin{pmatrix} 8 \\ -19 \\ \frac{75}{2} \end{pmatrix}$$

Now we have modified our original equation to

$$A \cdot \mathbf{x} = \mathbf{y}$$

where

$$A = \begin{pmatrix} 2 & 3 & 4 \\ 0 & -4 & -9 \\ 0 & 0 & \frac{25}{2} \end{pmatrix} \text{ and } \mathbf{y} = \begin{pmatrix} 8 \\ -19 \\ \frac{75}{2} \end{pmatrix}$$

Going for the third unknown

$$x_3 = \frac{1}{a_{33}} y_3 = \frac{2}{25} \frac{75}{2} = 3$$

$$x_2 = \frac{y_2 - a_{23}x_3}{a_{22}} = \frac{-19 + 9 \cdot 3}{-4} = \frac{8}{-4} = -2$$

and finally

$$x_1 = \frac{y_1 - a_{12}x_2 - a_{13}x_3}{a_{11}} = \frac{8 - 3 \cdot (-2) - 4 \cdot 3}{2} = \frac{2}{2} = 1$$

It is left as an exercise for the reader to check that these values satisfy the original equation.

23.2 The LUP Decomposition

This algorithm computes a decomposition of a regular matrix $A = (a_{ij})$ which is also useful for calculating its determinant. The decomposition yields a product:

$$A = L \cdot U \cdot P$$

where the factors are as follows: $L = (l_{ij})$ is a lower triangular $n \times n$ -matrix (i.e., $l_{ij} = 0$ for $i < j$), U is an upper triangular $n \times n$ -matrix, and P is a permutation matrix (see figure 23.1). This means that $P = \sum_{i=1, \dots, n} E(i, \pi(i))$ for a permutation $\pi \in S_n$.

$$\begin{pmatrix} & & & \\ & & & \\ & & & \\ & & & \end{pmatrix} = \begin{matrix} \diagdown & & & \\ & \square & & \\ & & \square & \\ & & & \square \end{matrix} \cdot \begin{matrix} \square & & & \\ \diagdown & & & \\ & \square & & \\ & & \square & \\ & & & \square \end{matrix} \cdot \begin{pmatrix} & & & 1 \\ & & & \\ & & & \\ 1 & & & \\ & & & \\ & & & \\ & & & \\ & & & 1 \end{pmatrix}$$

$$A = L \cdot U \cdot P$$

Fig. 23.1. The LUP decomposition of a matrix A .

Such a decomposition yields $\det(A)$ as a product of the diagonal coefficients of L , times the product of the diagonal elements of U , times the determinant of P , which is the signature $\text{sig}(\pi)$ of the given permutation.

The solution of an equation $(y_i) = A \cdot (x_i)$ proceeds in three steps: first one solves the auxiliary equation $(y_i) = L \cdot (z_i)$. The lower diagonal L allows this recursive calculation of z_i by *forward substitution*:

$$z_1 = \frac{1}{l_{11}} y_1$$

and producing

$$z_i = \frac{y_i - \sum_{j=1, \dots, i-1} l_{ij} z_j}{l_{ii}}$$

from the calculation of z_1, z_2, \dots, z_{i-1} . Then, we observe that the permutation $P \cdot (x_i)$ is nothing but a renaming of the indexes of the unknowns. Apart from this renaming, the remaining problem is an equation $(z_i) = U \cdot (x_i)$, which is solved by the above backward substitution.

The algorithm for the *LUP* decomposition runs as follows: First, we rewrite A as $A = A \cdot E_n = A \cdot P^2$ by use of a permutation matrix P , which permutes two columns 1 and k of A as described above in 23.1, such that $(A \cdot P)_{11} \neq 0$. Therefore wlog we can assume that $a_{11} \neq 0$. One then writes A as a block-configuration of four submatrixes:

$$A = \begin{pmatrix} a_{11} & w \\ v & A' \end{pmatrix}$$

where $v \in \mathbb{M}_{n-1,1}(R)$, $w \in \mathbb{M}_{1,n-1}(R)$, and $A' \in \mathbb{M}_{n-1,n-1}(R)$. The two matrixes v and w define their product matrix $v \cdot w \in \mathbb{M}_{n-1,n-1}(R)$. Supposing that $a_{11} \neq 0$, we now have this equation:

$$A = \begin{pmatrix} a_{11} & w \\ v & A' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ \frac{1}{a_{11}} \cdot v & E_{n-1} \end{pmatrix} \cdot \begin{pmatrix} a_{11} & w \\ 0 & A' - \frac{1}{a_{11}} \cdot v \cdot w \end{pmatrix} \quad (23.1)$$

where the regular submatrix $A' - \frac{1}{a_{11}} \cdot v \cdot w$ is called the *Schur complement of A with respect to the pivot element a_{11}* . By induction, we assume that this complement has a *LUP* decomposition

$$A' - \frac{1}{a_{11}} \cdot v \cdot w = L' \cdot U' \cdot P'$$

which we now use to obtain the desired *LUP* decomposition

$$A = \begin{pmatrix} 1 & 0 \\ \frac{1}{a_{11}} \cdot v & L' \end{pmatrix} \cdot \begin{pmatrix} a_{11} & w \cdot (P')^{-1} \\ 0 & U' \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 \\ 0 & P' \end{pmatrix}$$

of A .

Example 102 The goal is to calculate the *LUP* decomposition of the matrix $A \in \mathbb{M}_{3,3}(\mathbb{Q})$:

$$A = \begin{pmatrix} 2 & -3 & 1 \\ 1 & -2 & -3 \\ 1 & 4 & 1 \end{pmatrix}.$$

Equation 23.1 yields the following values:

$$a_{11} = 2 \quad A' = \begin{pmatrix} -2 & -3 \\ 4 & 1 \end{pmatrix} \quad v = \begin{pmatrix} 1 \\ 1 \end{pmatrix} \quad w = (-3, 1)$$

The Schur complement B is then computed as

$$\begin{aligned} B &= A' - \frac{1}{a_{11}} \cdot v \cdot w \\ &= \begin{pmatrix} -2 & -3 \\ 4 & 1 \end{pmatrix} - \frac{1}{2} \cdot \begin{pmatrix} 1 \\ 1 \end{pmatrix} \cdot (-3, 1) \\ &= \begin{pmatrix} -\frac{1}{2} & -\frac{7}{2} \\ \frac{11}{2} & \frac{1}{2} \end{pmatrix} \end{aligned}$$

thus

$$A = L \cdot U = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & -3 & 1 \\ 0 & -\frac{1}{2} & -\frac{7}{2} \\ 0 & \frac{11}{2} & \frac{1}{2} \end{pmatrix}.$$

The next step is to recursively construct the *LUP* decomposition of the Schur complement B . First the required parts are extracted:

$$b_{11} = -\frac{1}{2} \quad B' = \frac{1}{2} \quad v' = \frac{11}{2} \quad w' = -\frac{7}{2}$$

The Schur complement C at this (last) stage is simply a 1×1 -matrix:

$$\begin{aligned}
 C &= B' - \frac{1}{b_{11}} \cdot v' \cdot w' \\
 &= \frac{1}{2} + 2 \cdot \frac{11}{2} \cdot -\frac{7}{2} \\
 &= -38
 \end{aligned}$$

therefore

$$B = L' \cdot U' = \begin{pmatrix} 1 & 0 \\ -11 & 1 \end{pmatrix} \cdot \begin{pmatrix} -\frac{1}{2} & -\frac{7}{2} \\ 0 & -38 \end{pmatrix}.$$

Finally, the *LUP* decomposition is built up using the components just determined. Luckily, during the entire procedure there has never been a need for an exchange of columns; all permutation matrixes are therefore unit matrixes and can be omitted:

$$A = \begin{pmatrix} 1 & 0 & 0 \\ \frac{1}{2} & 1 & 0 \\ \frac{1}{2} & -11 & 1 \end{pmatrix} \cdot \begin{pmatrix} 2 & -3 & 1 \\ 0 & -\frac{1}{2} & -\frac{7}{2} \\ 0 & 0 & -38 \end{pmatrix}$$

It is now easy to calculate the determinant of *A*:

$$\det(A) = \det(L) \cdot \det(U) = 1 \cdot 1 \cdot 1 \cdot 2 \cdot -\frac{1}{2} \cdot -38 = 38$$

Of course, in this case we could have applied the formula for determinants of 3×3 -matrixes, but in larger sized matrixes, the *LUP* decomposition provides a much more efficient procedure than using the definition of determinants directly.

Linear Geometry

The previous mathematical development has covered a considerable number of familiar objects and relations, such as sets, numbers, graphs, grammars, or rectangular tables, which are abstractly recast in the matrix calculus. The axiomatic treatment of modules and, more specifically, vector spaces, has also allowed us to rebuild what is commonly known as coordinate systems. However, one very important aspect of everyday's occupation with geometric objects has not been even alluded to: distance between objects, angles between straight lines, lengths of straight lines connecting two points in space. Even more radically, the concept of a neighborhood has not been thematized, although it is a central concept in the comparison of positions of objects in a sensorial space, such as the visual, tactile, gestural, or auditive space-time. The following chapter is devoted to the very first steps towards the concept of a mathematical model of geometric reality (the Greek etymology of "geometry" being "to measure the earth"). *In this spirit, we shall exclusively deal with real vector spaces in the last two chapters of this part, i.e., the coefficient set is \mathbb{R} . We shall also assume that the vector spaces are always of finite dimension—unless the contrary is stated.*

24.1 Euclidean Vector Spaces

We begin with some preliminary definitions. For a real vector space V , the vector space of linear homomorphisms $\text{Lin}_{\mathbb{R}}(V, \mathbb{R})$ is called the *dual space of V* and denoted by V^* . If V is finite-dimensional of dimension

$\dim(V) = n$, then we know that $V^* \xrightarrow{\sim} \mathbb{M}_{1,n}(\mathbb{R}) \xrightarrow{\sim} \mathbb{R}^n \xrightarrow{\sim} V$. A linear map $l \in V^*$ is called an \mathbb{R} -linear form on V .

In order to generate the basic metric structures, one first needs bilinear forms:

Definition 171 Given a real vector space V , a map $b : V \times V \rightarrow \mathbb{R}$ is called \mathbb{R} -bilinear iff for each $v \in V$, both maps $b(v, ?) : V \rightarrow \mathbb{R} : x \mapsto b(v, x)$ and $b(?, v) : V \rightarrow \mathbb{R} : x \mapsto b(x, v)$ are \mathbb{R} -linear forms. A bilinear form is called symmetric iff $b(x, y) = b(y, x)$ for all $(x, y) \in V \times V$. It is called positive definite iff $b(x, x) > 0$ for all $x \neq 0$.

Given a symmetric, positive definite bilinear form b , the pair (V, b) is called a Euclidean vector space.

An isometry $f : (V, b) \rightarrow (W, c)$ between Euclidean spaces is a linear map $f \in \text{Lin}_{\mathbb{R}}(V, W)$ such that for all $(x, y) \in V \times V$, we have $c(f(x), f(y)) = b(x, y)$. The set of isometries $f : (V, b) \rightarrow (W, c)$ is denoted by $O_{b,c}(V, W)$ or $O(V, W)$ if the bilinear forms are clear. If $(V, b) = (W, c)$, one writes $O(V)$ instead.

For a vector x in a Euclidean vector space (V, b) , the norm of x is the non-negative real number $\|x\| = \sqrt{b(x, x)}$.

Lemma 201 For a Euclidean space (V, b) , the norm has this property for any vectors $x, y \in V$:

$$\|x + y\|^2 = \|x\|^2 + 2 \cdot b(x, y) + \|y\|^2,$$

On the other hand, the form b is determined by the associated norm with the formula

$$b(x, y) = \frac{1}{2}(\|x + y\|^2 - \|x\|^2 - \|y\|^2).$$

Proof We have

$$\begin{aligned} \|x + y\|^2 &= b(x + y, x + y) \\ &= b(x, x) + b(x, y) + b(y, x) + b(y, y) \\ &= \|x\|^2 + 2 \cdot b(x, y) + \|y\|^2. \end{aligned}$$

□

Lemma 202 For a finite-dimensional Euclidean space (V, b) , the map $*b : V \rightarrow V^* : v \mapsto b(v, ?)$ is a linear isomorphism and is equal to the map $b^* : V \rightarrow V^* : v \mapsto b(?, v)$.

Proof The map $v \mapsto b(v, ?)$, where $b(v, ?) : V \rightarrow \mathbb{R} : w \mapsto b(v, w)$, maps into V^* , where $\dim(V^*) = \dim(V)$ according to the remark at the beginning of this section. So it is sufficient to show that $*b$ is a linear injection. If $b(v, ?) = 0$, then also $b(v, v) = 0$, but then, $v = 0$, since b is positive definite. Further $b(v_1 + v_2, w) = b(v_1, w) + b(v_2, w)$, and $b(\lambda \cdot v, w) = \lambda \cdot b(v, w)$, so $*b$ is linear. By symmetry of b , we also have $b^* = *b$. \square

Exercise 120 Let (V, b) be a Euclidean space and $f : V \rightarrow V$ is a linear endomorphism. Prove that for any $x \in V$, the map $y \mapsto b(x, f(y))$ is a linear form. By lemma 202, there is a vector ${}^\tau f(x) \in V$ such that $b(x, f(y)) = b({}^\tau f(x), y)$ for all y . Show that ${}^\tau f$ is a linear map. It is called the *adjoint of f* .

Proposition 203 An isometry $f \in O(V, W)$ is always injective, and it is an isomorphism, whose inverse is also an isometry, if V and W have the same finite dimension n . The composition $g \circ f$ of two isometries $f : V \rightarrow W$ and $g : W \rightarrow X$ is an isometry, and $O(V)$ is a subgroup of $GL(V)$, called the orthogonal group of V .

Proof For an isometry $f : (V, b) \rightarrow (W, c)$ and $v \in V$, $\|f(v)\| = \|v\|$ in the respective norms, i.e., f conserves norms. But then, if $v \neq 0$, $f(v) \neq 0$, so f is injective. If both spaces have the same finite dimension, f must also be surjective, and the inverses of such isometries also conserve norms, and norms define the bilinear forms. So they are also isometries. Further, the composition of isometries is an isometry, since conservation of norms is a transitive relation. Hence $O(V)$ is a subgroup of $GL(V)$. \square

Exercise 121 For $V = \mathbb{R}^n, n > 0$, we have the standard bilinear form $(?, ?) : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}$, or *scalar product* with

$$((x_1, \dots, x_n), (y_1, \dots, y_n)) = (x_1, \dots, x_n) \cdot (y_1, \dots, y_n)^\top = \sum_{i=1, \dots, n} x_i y_i,$$

the product of a row and a column matrix, where we omit the parentheses surrounding the resulting number. Show that the standard bilinear form is symmetric and positive definite.

The bilinear form of any non-zero Euclidean space can be calculated by means of matrix products as follows: Let $(e_i)_{i=1, \dots, n}$ be a basis of V , and define the *associated matrix of the bilinear form* by $B = (B_{ij}) \in \mathbb{M}_{n, n}(\mathbb{R})$ with $B_{ij} = b(e_i, e_j)$. Then the bilinearity of b implies the following formula for the representations $x = \sum_i \xi_i e_i$ and $y = \sum_i \eta_i e_i$ of the vectors x and y by their n -tuples in \mathbb{R}^n :

$$b(x, y) = (\xi_i) \cdot (B_{ij}) \cdot (\eta_j)^T$$

One recognizes that the scalar product defined above is the special case of a bilinear form where $(B_{ij}) = (\delta_{ij}) = E_n$. The question, whether one may find a basis for every Euclidean space such that its associated matrix becomes this simple, can be answered by “yes!”, but we need some auxiliary theory which will also justify the hitherto mysterious wording “orthogonal group”.

Definition 172 In a Euclidean space (V, b) , a vector x is said to be orthogonal to a vector y iff $b(x, y) = 0$, in signs: $x \perp y$. Since b is symmetric, orthogonality is a symmetric relation. A sequence (x_1, \dots, x_k) of vectors in V is called orthogonal iff $x_i \perp x_j$ for all $i \neq j$. It is called orthonormal iff it is orthogonal and we have $\|x_i\| = 1$ for all i .

Two subspaces $U, W \subset V$ are called orthogonal to each other, in signs $U \perp W$ iff $u \perp w$ for all $u \in U$ and $w \in W$. The subspace of all vectors which are orthogonal to a subspace U is the largest subspace orthogonal to U and is denoted by U^\perp .

Proposition 204 (Gram-Schmidt Orthonormalization) If $(x_1 \dots x_k)$ with $k > 0$ is a sequence of linearly independent vectors in a Euclidean space (V, b) , then there is an orthonormal sequence $(e_1 \dots e_k)$ of linearly independent vectors such that for every index $i = 1, \dots, k$, $(x_1 \dots x_i)$ and $(e_1 \dots e_i)$ generate the same subspace. In particular, if $(x_1 \dots x_n)$ is a basis of V , then there is a orthonormal basis (e_1, \dots, e_n) such that $(x_1 \dots x_i)$ and $(e_1 \dots e_i)$ generate the same subspaces for all $i = 1, \dots, n$.

Proof The construction is by induction on the length k of the sequence. For $k = 1$, $e_1 = \frac{1}{\|x_1\|} x_1$ does the job. Suppose that all x_1, x_2, \dots, x_i are represented by an orthonormal sequence e_1, e_2, \dots, e_i in the required way. Setting $e_{i+1} = x_{i+1} + \sum_{j=1, \dots, i} \lambda_j e_j$, if we find a solution, then clearly the space generated by x_1, \dots, x_{i+1} coincides with the space generated by e_1, \dots, e_{i+1} . But the condition that $e_{i+1} \perp e_j$, for all $j = 1, \dots, i$, means that $b(e_j, x_{i+1}) + \lambda_j \cdot \|e_j\|^2 = 0$, which yields $\lambda_j = -b(e_j, x_{i+1})$, since $\|e_j\| = 1$. Now, the resulting e_{i+1} is orthogonal to all previous e_j and cannot vanish, because of the dimension $i + 1$ of the subspace generated by e_1, \dots, e_{i+1} . So, to obtain the correct norm 1 of e_{i+1} , replace it by $\frac{1}{\|e_{i+1}\|} e_{i+1}$, and everything is perfect. \square

Observe that the proof of proposition 204 is constructive, i.e., algorithmic, and should be kept in mind together with the result.

Example 103 Let $x_1 = (2, 2, 0)$, $x_2 = (1, 0, 2)$ and $x_3 = (0, 2, 1)$ be a basis of \mathbb{R}^3 . We compute an orthonormal basis $\{e_1, e_2, e_3\}$ using the Gram-Schmidt procedure. For the linear form b we use the ordinary scalar product.

The computation of e_1 is simple. It consists in normalizing x_1 :

$$\begin{aligned} e_1 &= \frac{1}{\|x_1\|} x_1 \\ &= \frac{1}{\sqrt{2^2 + 2^2 + 0^2}} (2, 2, 0) \\ &= \frac{1}{2\sqrt{2}} (2, 2, 0) \\ &= \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right) \end{aligned}$$

For the second vector e_2 we first compute an intermediate value e'_2 using the formula from the proof of proposition 204:

$$\begin{aligned} e'_2 &= x_2 - (e_1, x_2) \cdot e_1 \\ &= (1, 0, 2) - \left(\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right), (1, 0, 2) \right) \cdot \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right) \\ &= (1, 0, 2) - \frac{\sqrt{2}}{2} \cdot \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right) \\ &= \left(1 - \frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2}}{2}, -\frac{\sqrt{2}}{2} \cdot \frac{\sqrt{2}}{2}, 2 \right) \\ &= \left(\frac{1}{2}, -\frac{1}{2}, 2 \right) \end{aligned}$$

and then normalize to get e_2 :

$$\begin{aligned} e_2 &= \frac{1}{\|e'_2\|} e'_2 \\ &= \frac{1}{\sqrt{1/2^2 + 1/2^2 + 2^2}} \left(\frac{1}{2}, -\frac{1}{2}, 2 \right) = \frac{\sqrt{2}}{3} \left(\frac{1}{2}, -\frac{1}{2}, 2 \right) \\ &= \left(\frac{\sqrt{2}}{6}, -\frac{\sqrt{2}}{6}, \frac{2\sqrt{2}}{3} \right) \end{aligned}$$

The formula for the last vector becomes more complex:

$$\begin{aligned}
e'_3 &= x_3 - (e_1, x_3) \cdot e_1 - (e_2, x_3) \cdot e_2 \\
&= x_3 - \left(\left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right), (0, 2, 1) \right) \cdot e_1 - \left(\left(\frac{\sqrt{2}}{6}, -\frac{\sqrt{2}}{6}, \frac{2\sqrt{2}}{3} \right), (0, 2, 1) \right) \cdot e_2 \\
&= x_3 - \sqrt{2} \cdot \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right) - \frac{\sqrt{2}}{3} \cdot \left(\frac{\sqrt{2}}{6}, -\frac{\sqrt{2}}{6}, \frac{2\sqrt{2}}{3} \right) \\
&= (0, 2, 1) - (1, 1, 0) - \left(\frac{1}{9}, -\frac{1}{9}, \frac{4}{9} \right) \\
&= \left(-\frac{10}{9}, \frac{10}{9}, \frac{5}{9} \right)
\end{aligned}$$

Normalizing e'_3 finally yields e_3 :

$$\begin{aligned}
e_3 &= \frac{1}{\|e'_3\|} e'_3 \\
&= \frac{1}{\sqrt{\left(-\frac{10}{9}\right)^2 + \left(\frac{10}{9}\right)^2 + \left(\frac{5}{9}\right)^2}} e'_3 \\
&= \frac{3}{5} \cdot \left(-\frac{10}{9}, \frac{10}{9}, \frac{5}{9} \right) \\
&= \left(-\frac{2}{3}, \frac{2}{3}, \frac{1}{3} \right)
\end{aligned}$$

Summarizing, the orthonormal basis is:

$$e_1 = \left(\frac{\sqrt{2}}{2}, \frac{\sqrt{2}}{2}, 0 \right), e_2 = \left(\frac{\sqrt{2}}{6}, -\frac{\sqrt{2}}{6}, \frac{2\sqrt{2}}{3} \right), e_3 = \left(-\frac{2}{3}, \frac{2}{3}, \frac{1}{3} \right)$$

Figure 24.1 shows both bases. It is left to the reader to check that the e_i are indeed pairwise orthogonal.

Exercise 122 Show that, for an endomorphism $f : V \rightarrow V$ on a Euclidean space, if A is the matrix of f with respect to an orthonormal basis, the adjoint endomorphism ${}^t f$ has the transpose A^T as its matrix with respect to this basis.

Corollary 205 For an n -dimensional Euclidean space (V, b) , if (e_i) is an orthonormal basis (which exists according to Gram-Schmidt), then the group $O(V)$ identifies with the subgroup $O_n(\mathbb{R}) \subset GL_n(\mathbb{R})$ consisting of all matrixes A with $A^T \cdot A = E_n$. In particular, $\det(f) = \pm 1$ for $f \in O(V)$. The orthogonal group is the disjoint union of the normal subgroup $SO(V) \subset O(V)$ of the isometries f with $\det(f) = 1$, called rotations, and the coset $O^-(V)$ of the isometries f with $\det(f) = -1$. $SO(V)$ is called the special orthogonal group of V .

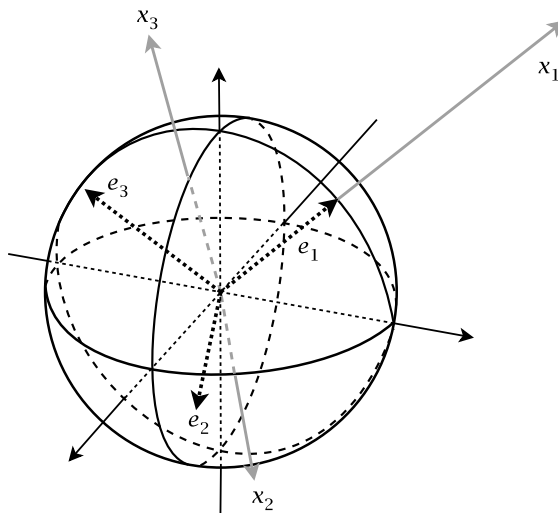


Fig. 24.1. The base x_i and its orthonormalization e_i from example 103.

Proof Given an orthonormal basis (e_1, \dots, e_n) of (V, b) , if $f \in O(V)$ is represented by the matrix A relative to this basis, then $\delta_{ij} = b(e_i, e_j) = b(f(e_i), f(e_j)) = \sum_{t=1, \dots, n} A_{ti} A_{tj} = \sum_{t=1, \dots, n} A_{it}^T A_{tj} = (A^T \cdot A)_{ij}$. This means that $A^T \cdot A = E_n$. Conversely, if the latter equation holds, then reading these equalities in the other direction, we have $\delta_{ij} = b(e_i, e_j) = b(f(e_i), f(e_j))$, and therefore f conserves the bilinear form's values for the orthonormal basis (e_1, \dots, e_n) . But then, by bilinearity, it conserves bilinear form values $b(x, y)$ for any x and y . The rest is straightforward. \square

Corollary 206 In a Euclidean space (V, b) , if U is a subspace, then U^\perp is a complement of U in V , i.e., we have an isomorphism $U \oplus U^\perp \xrightarrow{\sim} V$. In particular,

$$\dim(U) + \dim(U^\perp) = \dim(V).$$

Exercise 123 Give a proof of the corollary 206 by using proposition 204 and the Steinitz theorem 194.

We are now ready to describe hyperplanes in a Euclidean space.

Definition 173 In a Euclidean space (V, b) of dimension $n > 0$, a hyperplane is the translate $H = T^v(W)$ of a vector subspace $W \subset V$ of codimension 1, i.e., $\dim(W) = n - 1$.

Proposition 207 In a Euclidean space (V, b) , a hyperplane H can be defined by an equation

$$H = \{h \mid h \in V, k \perp (h - v)\}$$

where v and k are vectors with $k \neq 0$.

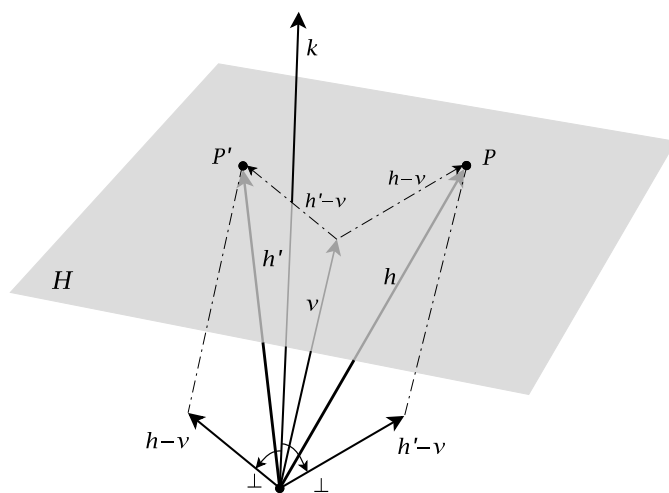


Fig. 24.2. The construction of a 2-dimensional hyperplane H in a 3-dimensional Euclidean space, according to proposition 207.

Exercise 124 Give a proof of proposition 207. Use this idea: We know that $H = T^v(W)$ for a subspace W of codimension 1. Since $h \in H$ means that $(h - v) \in W$, and since $W = (W^\perp)^\perp$ with $\dim(W^\perp) = 1$, we have $h - v \perp k$ for any generator k of W^\perp .

Exercise 125 Rewrite the equation in proposition 207 in terms of an equation using coordinates for an orthonormal basis.

A special type of isometries are the reflections at a hyperplane H of a non-zero Euclidean space (V, b) . By definition, a hyperplane is a vector subspace $H \subset V$ of dimension $\dim(H) = \dim(V) - 1$. By corollary 206, we have a 1-dimensional complement $H^\perp = \mathbb{R} \cdot x$, where $x \neq 0$ is any vector in H^\perp , and $V = H \oplus H^\perp$. This defines a linear map ρ_H on V by

$$\rho_H(y) = y - 2 \cdot \frac{b(y, x)}{b(x, x)} \cdot x$$

where any other generator $x' \in H^\perp$ is a scaling of x , $x' = \lambda x$, and therefore yields the same map, so the map only depends on H . In fact, $\rho_H|_H = Id_H$ and $\rho_H|_{H^\perp} = -Id_{H^\perp}$. Therefore $\rho_H \in O^-(V)$ and $\rho_H^2 = Id_V$. The isometry ρ_H is therefore called the *reflection at H* .

Exercise 126 Show that if $x \neq y$ are two different vectors of equal norm $\|x\| = \|y\|$, then the reflection ρ_H at $H = (x - y)^\perp$ exchanges x and y , i.e., $\rho_H(x) = y$.

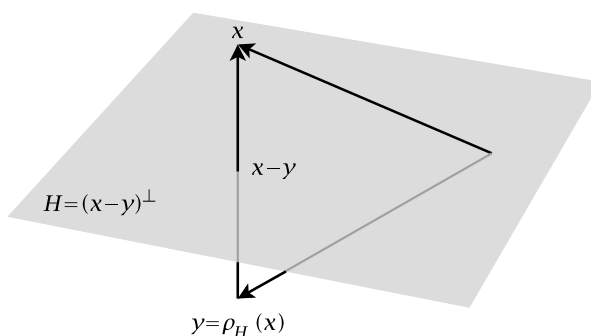


Fig. 24.3. If $\|x\| = \|y\|$, $x \neq y$, and $H = (x - y)^\perp$, then $\rho_H(x) = y$.

This exercise entails the theorem about the fundamental role of reflections:

Proposition 208 Every $f \in O(V)$ for a non-zero Euclidean space V is the product of at most $\dim(V)$ reflections (the identity for zero reflections). Every rotation is the product of an even number of reflections. In particular, for $\dim(V) = 2$, a rotation is a product of two reflections.

Proof Suppose that $f \neq Id_V$. Then there is x such that $f(x) = y \neq x$. Following exercise 126, the reflection ρ_H at $H = (x - y)^\perp$ exchanges x and y and fixes the orthogonal space H pointwise. Therefore $\rho_H \circ f$ fixes the line $\Delta = \mathbb{R}(x - y)$ pointwise, and, since it is an isometry, also H , but not necessarily pointwise. Then the restriction of $g = \rho_H \circ f$ to H is an isometry of H , which has dimension $\dim(V) - 1$. So, by recursion, we have $g = Id_\Delta \times g_H$ with $g_H \in O(H)$. If $g_H = Id_H$, we have $f = \rho_H$, and we are finished. Else, we have $g_H = \rho_{H_1} \times \dots \times \rho_{H_k}$,

$k \leq \dim(V) - 1$, for hyperplanes $H_j \subset H$. But each reflection ρ_{H_i} extends to a reflection at $H_i \oplus \Delta$, leaving Δ pointwise fixed, since $\Delta \perp H$, and therefore $\Delta \perp H_i^\perp$, where H_i^\perp is the line orthogonal to H_i in H . So we are done, since $f = \rho_H \circ g$. Finally, since a rotation in a 2-dimensional space cannot be one single reflection, it must be the product of two of them, since it is the product of at most two of them. \square

In figure 24.4, the geometrical object x is first reflected through the axis R , then through S . This corresponds to a rotation by an angle a .

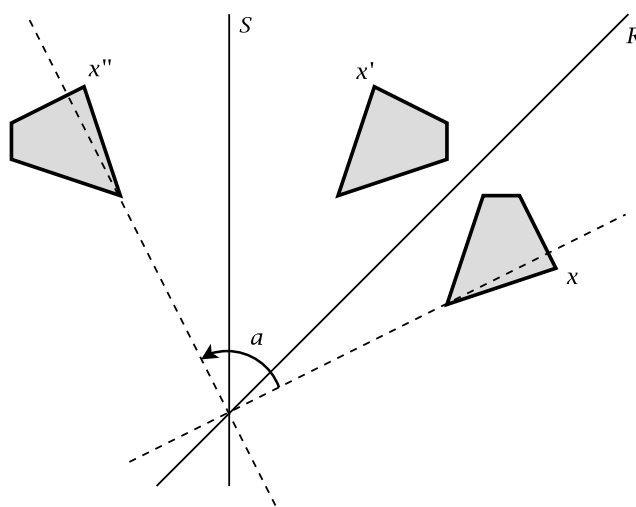


Fig. 24.4. A rotation in \mathbb{R}^2 is a product of two reflections.

Among the orthonormal bases (e_i) and (d_i) of a Euclidean space (V, b) we have an equivalence relation $(e_i) \sim (d_i)$ iff the transition isometry $e_i \mapsto d_i$ has determinant 1. Bases in one of the two equivalence classes are said to have the *same orientation*, i.e., each of these two equivalence classes defines an orientation.

24.2 Trigonometric Functions from Two-Dimensional Rotations

In this section we deal exclusively with the case $\dim(V) = 2$, i.e., the plane geometry, and the structure of the group $SO(V)$ of rotations.

Proposition 209 Given an orthonormal basis (e_i) of V , let $M_f \in \mathbb{M}_{2,2}(\mathbb{R})$ be the associated matrix of an isometry $f \in \text{GL}(V)$. Then

- (i) we have $f \in \text{SO}(V)$ (a rotation) iff $M_f = \begin{pmatrix} a & -b \\ b & a \end{pmatrix}$ and $a^2 + b^2 = 1$,
- (ii) we have $f \in \text{O}^-(V)$ (a reflection) iff $M_f = \begin{pmatrix} a & b \\ b & -a \end{pmatrix}$ and $a^2 + b^2 = 1$.
- (iii) The group $\text{SO}(V)$ is abelian and the product matrix for two rotations f and g is

$$M_g \cdot M_f = \begin{pmatrix} u & -v \\ v & u \end{pmatrix} \cdot \begin{pmatrix} a & -b \\ b & a \end{pmatrix} = \begin{pmatrix} au - bv & -(av + bu) \\ av + bu & au - bv \end{pmatrix}.$$

- (iv) The number a is independent of the chosen orthonormal basis, and so is $|b|$. If another orthonormal basis (e'_i) with the same orientation is chosen, then b does not change.
- (v) For any two vectors $x, y \in S(V) = \{z \mid z \in V, \|z\| = 1\}$, there is exactly one $f \in \text{SO}(V)$ such that $f(x) = y$.

Proof Clearly, concerning points (i) and (ii), according to corollary 205 the matrixes in the described form define isometries f which are rotations or reflections, respectively. Conversely, if $\begin{pmatrix} a \\ b \end{pmatrix}$ is the first column of M_f , then we must have $a^2 + b^2 = 1$, since the norm must be 1. On the other hand, the second column must be orthogonal to the first, and $\begin{pmatrix} -b \\ a \end{pmatrix}$ is orthogonal to the first and has norm 1. So $\begin{pmatrix} -b \\ a \end{pmatrix}$ must be $\pm f(e_2)$. But the determinant must be 1 if f is a rotation, so the second column must be $\begin{pmatrix} -b \\ a \end{pmatrix}$. Since the determinant must be -1 , if f is a reflection, the second column must be the negative of the first, so (i) and (ii) are settled.

For point (iii), knowing that rotation matrixes have the shape described in (i), the formula in (iii) shows that the product of rotations is commutative.

For (iv), we shall see in proposition 215 that the characteristic polynomial $\chi_{M_f}(X)$ of f is independent of its matrix M_f (that result does not presuppose the present one). But the coefficient of X in $\chi_f(X)$ is $-2a$ with the notation described in (i) and (ii). So a is uniquely determined, and therefore also $|b|$. If one changes the base by a rotation, then the new matrix of f is the conjugate of the old matrix by a rotation matrix. A straight calculation shows that b is also invariant.

Statement (v) follows from the fact that the orthogonal spaces x^\perp and y^\perp are 1-dimensional. So there are two possibilities to map a unit vector in x^\perp to a

unit vector in γ^\perp . Exactly one of them has positive determinant, and this is our candidate from $SO(V)$. \square

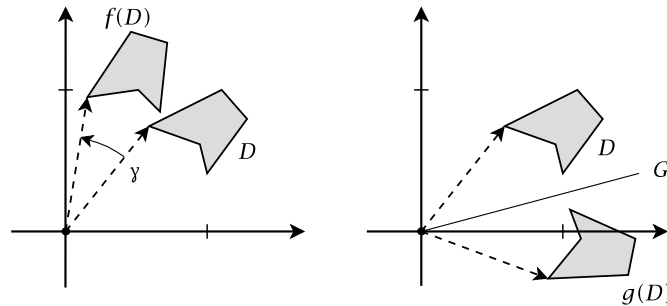


Fig. 24.5. Using $a = \frac{\sqrt{3}}{2}$ and $b = \frac{1}{2}$ in the matrixes of proposition 209, on the left is a rotation $f \in SO(\mathbb{R}^2)$ by the angle γ , and on the right a reflection $g \in O^-(\mathbb{R}^2)$ about the axis G .

Exercise 127 In the case (ii) of a reflection in proposition 209, calculate the reflection formula ρ_H .

We now have to justify the word “rotation” and want to define the cosine and sine functions, together with the associated angles. To this end, recall that U is the unit circle $S(\mathbb{C})$, i.e., the multiplicative group of complex numbers z with norm $|z| = 1$.

Proposition 210 *Suppose that we are given an orthonormal basis of V , $\dim(V) = 2$. Consider the maps $\cos : SO(V) \rightarrow \mathbb{R} : f \mapsto a = (M_f)_{11}$ and $\sin : SO(V) \rightarrow \mathbb{R} : f \mapsto b = (M_f)_{21}$, then the map*

$$\text{cis} : SO(V) \rightarrow U : f \mapsto \cos(f) + i \cdot \sin(f)$$

is an isomorphism of groups. The isomorphism remains unchanged if we select another orthonormal basis of the same orientation.

Proof The only point is the multiplication of complex numbers, which corresponds to the product of rotations, but this is evident from the product formula in proposition 209, (iii). \square

Now we know that rotations identify with complex numbers on the unit circle group U in \mathbb{C} , but we would like to have the concept of an angle which gives rise to a complex number in U . In fact, we have this result:

Proposition 211 *There is a surjective group homomorphism $A : \mathbb{R} \rightarrow U$ whose kernel is $\text{Ker}(A) = 2\pi\mathbb{Z}$, where π is the positive number 3.1415926... which will in the chapter on limits and topology of the second volume of this book. So with $f \in \text{SO}(V)$ a coset of numbers $\theta \in \mathbb{R}$ is associated such that $\cos(f) + i \sin(f) = A(\theta)$. We also write $\cos(\theta)$ and $\sin(\theta)$ instead of $\cos(f)$ and $\sin(f)$ respectively for the rotation associated with θ . Any such θ is called angle of f . The matrix of f is:*

$$M_f = \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}.$$

So the rotation angle θ is determined up to multiples of 2π . The product formula in statement (iii) of proposition 209 translates into the classical goniometric formulas for $\cos(\theta \pm \eta)$ and $\sin(\theta \pm \eta)$:

$$\cos(\theta \pm \eta) = \cos(\theta) \cos(\eta) \mp \sin(\theta) \sin(\eta),$$

$$\sin(\theta \pm \eta) = \sin(\theta) \cos(\eta) \pm \cos(\theta) \sin(\eta).$$

Proof The only point to be proved here are the formulas for $\cos(\theta \pm \eta)$ and $\sin(\theta \pm \eta)$. But both relate to the cos and sin functions of sums/differences of angles, and this means that one looks for the product of the rotations (or the product of one with the inverse of the other) associated with these angles, i.e., $\cos(f \circ g^{\pm 1})$ and $\sin(f \circ g^{\pm 1})$. Then we know the formulas from the product formula in proposition 209, (iii). \square

24.3 Gram's Determinant and the Schwarz Inequality

Given an n -dimensional Euclidean space (V, b) and an orthonormal basis (e_i) , we may consider the determinant of the linear map associated with a sequence $x_\bullet = (x_i)$ of length n by $f : e_i \mapsto x_i$. On the other hand, we also have the *Gram determinant* $\text{Gram}(x_i)$ of (x_i) defined by

$$\text{Gram}(x_i) = \det \begin{pmatrix} b(x_1, x_1) & b(x_1, x_2) & \dots & b(x_1, x_n) \\ b(x_2, x_1) & b(x_2, x_2) & \dots & b(x_2, x_n) \\ \vdots & \vdots & & \vdots \\ b(x_n, x_1) & b(x_n, x_2) & \dots & b(x_n, x_n) \end{pmatrix}$$

But the Gram matrix $(b(x_i, x_j))$ clearly equals $M_f^T \cdot M_f$. Therefore we have the Gram equation

$$\text{Gram}(x_i) = \det(f)^2$$

For $n = 2$ we immediately deduce the Schwarz inequality:

Proposition 212 (Schwarz Inequality) *If x and y are two vectors in a Euclidean space (V, b) , then*

$$|b(x, y)| \leq \|x\| \|y\|,$$

equality holding iff x and y are linearly dependent.

Proof

$$\begin{aligned} 0 &\leq \det(f)^2 \\ &= \text{Gram}(x, y) \\ &= \det \begin{pmatrix} b(x, x) & b(x, y) \\ b(y, x) & b(y, y) \end{pmatrix} \\ &= b(x, x) \cdot b(y, y) - b(x, y)^2 \end{aligned}$$

Therefore, $b(x, y)^2 \leq b(x, x) \cdot b(y, y)$, which, due to the definition of the norm, implies the desired result. \square

This result may be reinterpreted in terms of the cosine function. If $x, y \neq 0$, choose a 2-dimensional subspace W of V containing x and y , and carrying the induced bilinear form $b|_{W \times W}$. Then we have $\left| \frac{b(x, y)}{\|x\| \|y\|} \right| \leq 1$, and defining $c(x, y) = \frac{b(x, y)}{\|x\| \|y\|}$, we have $b(x, y) = c(x, y) \cdot \|x\| \cdot \|y\|$ with $|c(x, y)| \leq 1$. If $f \in \text{SO}(W)$ is the unique rotation with $f(x/\|x\|) = y/\|y\|$, then $c(x, y) = \cos(f) = \cos(\theta(x, y))$, which means that the angle is determined up to integer multiples of 2π by the unit vectors $x/\|x\|$ and $y/\|y\|$ or equivalently the half lines \mathbb{R}_+x and \mathbb{R}_+y through x and y . This gives us the famous cosine formula for the bilinear form, where one has chosen an orthogonal basis on a plane containing x and y :

$$b(x, y) = \cos(\theta(x, y)) \cdot \|x\| \cdot \|y\|.$$

We also obtain the following intuitive fact: the triangle inequality for norms.

Corollary 213 (Triangle Inequality) *If x and y are two vectors in a Euclidean space (V, b) , then*

$$\|x + y\| \leq \|x\| + \|y\|.$$

Proof Since both sides are non-negative numbers, we may prove that the squares of these numbers fulfill the inequality. But by the Schwarz inequality from proposition 212, we have

$$\begin{aligned}
 \|x + y\|^2 &= b(x + y, x + y) \\
 &= b(x, x) + 2b(x, y) + b(y, y) \\
 &= \|x\|^2 + 2b(x, y) + \|y\|^2 \\
 &\leq \|x\|^2 + 2\|x\|\|y\| + \|y\|^2 \\
 &= (\|x\| + \|y\|)^2.
 \end{aligned}$$

□

Defining the *distance* between two vectors x and y in a Euclidean space by

$$d(x, y) = \|x - y\|,$$

we obtain these characteristic properties of a distance function:

Proposition 214 *Given a Euclidean space (V, b) , the distance function $d(x, y) = \|x - y\|$ has these properties for all $x, y, z \in V$:*

- (i) $d(x, y) \geq 0$, and $d(x, y) = 0$ iff $x = y$,
- (ii) (Symmetry) $d(x, y) = d(y, x)$,
- (iii) (Triangle Inequality) $d(x, z) \leq d(x, y) + d(y, z)$.

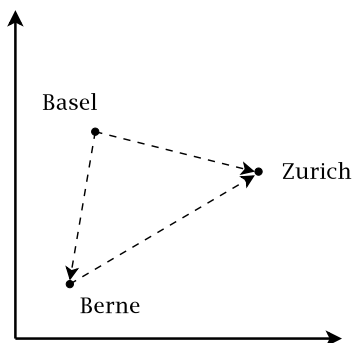


Fig. 24.6. Triangle Inequality: The beeline between Basel and Zurich is shorter than a detour over Berne.

Proof Claim (i) is true because the norm is always non-negative, and strictly positive if the argument is not zero (b is positive definite).

For (ii), we have $d(x, y)^2 = \|x - y\|^2 = b(x - y, x - y) = (-1)^2 b(y - x, y - x) = \|y - x\|^2 = d(y, x)^2$.

For (iii), again by the Schwarz inequality from proposition 212, we have $d(x, z) = \|x - z\| = \|(x - y) + (y - z)\| \leq \|x - y\| + \|y - z\|$. \square

These three properties are those which will define a metric in the chapter on topology in volume II. So proposition 214 guarantees that a Euclidean space has a metric which is induced by the norm derived from the given positive definite bilinear form.

Eigenvalues, the Vector Product, and Quaternions

This chapter deals with the geometry in three dimensions, a very important case since we all live in 3D space. Computer graphics makes extensive use of this space and its transformations. *Again, in this chapter we shall only deal with finite-dimensional \mathbb{R} -vector spaces.*

25.1 Eigenvalues and Rotations

We begin with an analysis of the special orthogonal group $SO(V)$ where $\dim(V) = 3$; recall that for the standard scalar product (\cdot, \cdot) on \mathbb{R}^3 this group is also denoted by $SO_3(\mathbb{R})$.

First, we want to show that every $f \in SO(V)$ is really what one would call a rotation: it has a rotation axis and turns the space around this axis by a specific angle as introduced in the last chapter. In other words we are looking for a *rotation axis*, i.e., a 1-dimensional vector subspace $R = \mathbb{R}x$ which remains fixed under f , i.e., $f(x) = x$. Rewriting this equation as $f(x) - 1 \cdot x = 0$ means that there is a solution of the linear equation $(f - 1 \cdot Id_V)(x) = 0$, i.e., the linear map $f - 1 \cdot Id_V$ has a non-trivial kernel, namely $x \in Ker(f - 1 \cdot Id_V) - \{0\}$. We know from the theory of linear equations that the condition $Ker(f - 1 \cdot Id_V) \neq 0$ is equivalent to the vanishing of the determinant $\det(f - 1 \cdot Id_V)$. This means that we have to look for solutions of the equation $\det(f - X \cdot Id_V) = 0$. Let us make this equation more precise.

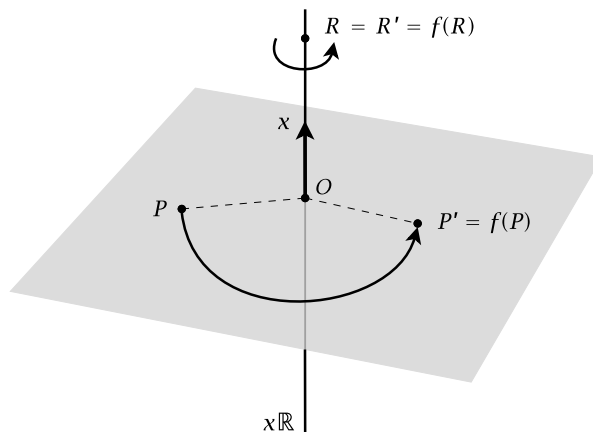


Fig. 25.1. Points on the rotation axis $x\mathbb{R}$ like R are not affected by the rotation.

Lemma 215 *If V is a real vector space of finite dimension n , if $f \in \text{End}(V)$, and if $M \in \mathbb{M}_{n,n}(\mathbb{R})$ is the matrix representation of f with respect to a basis, then the characteristic polynomial $\chi_M = \det(M - X \cdot E_n)$ is independent of the chosen basis. We therefore also write χ_f instead of χ_M . We have*

$$\chi_f = \sum_{i=0, \dots, n} t_i X^i = (-1)^n X^n + (-1)^{n-1} \text{tr}(f) X^{n-1} \pm \dots + \det(f)$$

where the second coefficient $\text{tr}(f) = \sum_{i=1, \dots, n} M_{ii}$ is called the trace of f (or of the matrix which represents f).

Proof If we change the basis of V , the new matrix M' of f is the conjugate $M' = C^{-1} \cdot M \cdot C$ of M by the basis change matrix C , this is corollary 196. But we know from corollary 186 that the characteristic polynomial does not change under conjugation. The general formula for χ_f follows from an argument by induction on the dimension n . \square

So our problem is this: to find special solutions of the characteristic polynomial equation $\chi_f(X) = 0$.

Definition 174 *If V is a real vector space of finite dimension n , and if $f \in \text{End}(V)$, the zeros λ of the characteristic polynomial χ_f*

$$\chi_f(\lambda) = 0$$

are called the eigenvalues of f . The non-zero elements x of the non-trivial kernel $\text{Ker}(f - \lambda \cdot \text{Id}_V)$ for an eigenvalue λ are called eigenvectors of f (corresponding to λ). They are characterized by $f(x) = \lambda x$.

Remark 29 In the case where all eigenvalues of f are real, i.e., $\lambda_i \in \mathbb{R}$ for all i , the corresponding eigenvectors form a basis of \mathbb{R}^n , and in this basis the matrix of f is diagonal, where the diagonal elements are the eigenvalues.

We are more concretely interested in the case $\dim(V) = 3$, where we have the polynomial

$$\chi_f = -X^3 + \text{tr}(f)X^2 + t_1X + \det(f),$$

and we are looking for solutions λ such that $-\lambda^3 + \text{tr}(f)\lambda^2 + t_1\lambda + \det(f) = 0$. It will be shown in the second volume of this book that every polynomial $P \in \mathbb{R}[X]$ of odd degree has at least one root in \mathbb{R} . Therefore, the characteristic polynomial has a real eigenvalue λ_0 . Let us show that in the case of $f \in \text{SO}(V)$, there is a *positive* eigenvalue. The equation factorizes to

$$\begin{aligned} -X^3 + \text{tr}(f)X^2 + t_1X + \det(f) &= -X^3 + \text{tr}(f)X^2 + t_1X + 1 \\ &= -(X - \lambda_0)(X^2 + bX + c) \\ &= -X^3 + (\lambda_0 - b)X^2 + (\lambda_0b - c)X + \lambda_0c \end{aligned}$$

and therefore $\lambda_0 \cdot c = 1$. If $\lambda_0 < 0$, then $c < 0$, and then we have two more real solutions $\lambda_{1,2} = \frac{-b \pm \sqrt{b^2 - 4c}}{2}$ since $b^2 - 4c \geq 0$, and square roots of non-negative elements exist in \mathbb{R} (see corollary 90 in chapter 9). Therefore $1 = \lambda_0\lambda_1\lambda_2$, which means that one of these solutions must be positive. So, after renaming the roots, we take a positive eigenvalue λ_0 and look at a corresponding eigenvector x , i.e., $f(x) = \lambda_0x$. Since f is an isometry, we have $\|f(x)\| = |\lambda_0|\|x\| = \|x\|$, whence $\lambda_0 = 1$, and we have a 1-dimensional subspace $\mathbb{R}x$ which is left pointwise fixed under f . This shows:

Proposition 216 Every rotation $f \in \text{SO}(V)$ for an Euclidean space (V, b) with $\dim(V) = 3$, has a rotation axis $A_f = \mathbb{R}x$, i.e., $f|_{A_f} = \text{Id}_{A_f}$. If $f \neq \text{Id}_V$, the rotation axis is unique.

Proof The only open point here is uniqueness of a rotation axis. But if we had two rotation axes, the plane H generated by these axes would be fixed pointwise,

so H^\perp would have to be reflected or remain unchanged. In both cases, this would not yield a non-trivial transformation in $SO(V)$. \square

Since $A_f \oplus A_f^\perp \xrightarrow{\sim} V$, the isometry f which leaves the rotation axis point-wise invariant also leaves invariant the 2-dimensional orthogonal plane A_f^\perp , i.e., each point of A_f^\perp is mapped to another point of A_f^\perp . Taking an orthonormal basis (a_1, a_2) of A_f^\perp , we may rewrite f in terms of the orthonormal basis (a_0, a_1, a_2) , $a_0 = x/\|x\|$ as a matrix

$$M_f = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{pmatrix}$$

which makes explicit the rotation in the plane A_f^\perp orthogonal to the rotation axis A_f . This representation is unique for all orthonormal bases (a'_0, a'_1, a'_2) of same orientation, which represent the orthogonal decomposition defined by the rotation axis, i.e., $A_f = \mathbb{R}a'_0$ and $A_f^\perp = \mathbb{R}a'_1 + \mathbb{R}a'_2$.

The above formula and the invariance of the trace $tr(f)$ with respect to basis changes implies this uniqueness result:

Corollary 217 *Every rotation in $SO(V)$ for an Euclidean space (V, b) with $dim(V) = 3$, has a rotation angle θ whose $\cos(\theta)$ is uniquely determined, i.e.,*

$$\cos(\theta) = \frac{1}{2}(tr(f) - 1).$$

So we are left with the elements $f \in O^-(V)$ for $dim(V) = 3$. There are two cases for the characteristic polynomial:

1. Either it has a positive solution $\lambda_0 = 1$, then we have again a rotation axis A_f and the orthogonal complement is left invariant not by a rotation, but by a reflection at a line $B_f \in A_f^\perp$. This line and the axis A_f are left point-wise invariant, whereas the orthogonal complement line $C = B_f^\perp$ in A_f^\perp is inverted by -1 . This means that f is a *reflection* ρ_H at the plane H generated by A_f and B_f .
2. Or else, there is no positive eigenvalue. Then we have the eigenvalue -1 and an eigenvector x with $f(x) = -x$. Then the reflection ρ_H at the plane $H = x^\perp$ yields an isometry $g = \rho_H \circ f \in SO(V)$, with rotation axis $A_g = \mathbb{R}x$. Therefore, $f = \rho_H \circ g$ is the *composition of a rotation g and a reflection ρ_H orthogonal to the rotation axis of g* .

So this classification covers all cases of isometries in $O(V)$, $dim(V) = 3$.

25.2 The Vector Product

If we are given a triple (x_1, x_2, x_3) of vectors $x_i \in V$ in a Euclidean space (V, b) of dimension 3, and if (e_i) is an orthonormal basis of V , then the linear map $f : e_i \mapsto x_i$ has a determinant $\det(f)$, which we also write as $\det_{(e_i)}(x_1, x_2, x_3)$. We know that this determinant does only depend on the orientation defined by (e_i) . So if we fix the orientation ω of (e_i) , we have a function $\det_\omega(x_1, x_2, x_3)$. In the basis (e_i) , The columns of M_f , the matrix of f , are the vectors x_1, x_2 and x_3 , therefore we can deduce the following properties from the general properties of the determinant described in theorem 181 of chapter 20:

1. The function $\det_\omega(x_1, x_2, x_3)$ is linear in each argument.
2. The function is skew-symmetric, i.e., $\det_\omega(x_1, x_2, x_3) = 0$ if two of the three x_i are equal. This entails that $\det_\omega(x_{\pi(1)}, x_{\pi(2)}, x_{\pi(3)}) = \text{sig}(\pi) \det_\omega(x_1, x_2, x_3)$ for a permutation $\pi \in S_3$.
3. $\det_\omega(x_1, x_2, x_3) = 1$ for any orthonormal basis $(x_1, x_2, x_3) \in \omega$.

Therefore we define the vector product as follows:

Definition 175 *Given a 3-dimensional Euclidean space (V, b) , with the previous notations, fix a pair (x_1, x_2) of vectors in V . Then under the isomorphism $*b : V \xrightarrow{\sim} V^*$, the linear form*

$$d_{(x_1, x_2)} \in V^* \text{ defined by } d_{(x_1, x_2)}(x) = \det_\omega(x_1, x_2, x)$$

*corresponds to a vector $(*b)^{-1}d_{(x_1, x_2)} \in V$ which we denote by $x_1 \wedge x_2$, and which is characterized by the equation*

$$b(x_1 \wedge x_2, x) = \det_\omega(x_1, x_2, x),$$

and which is called the vector product of x_1 and x_2 (in this order).¹ The orientation ω is not explicitly denoted in the vector product expression, but should be fixed in advance, otherwise the product is only defined up to sign.

Given the representations $x_1 = \sum_i x_{1i}e_i$ and $x_2 = \sum_i x_{2i}e_i$, the vector product has these coordinates in terms of the basis (e_i) :

¹ Some texts also use the notation $x_1 \times x_2$ instead of $x_1 \wedge x_2$.

$$(x_1 \wedge x_2)_1 = x_{12}x_{23} - x_{13}x_{22},$$

$$(x_1 \wedge x_2)_2 = x_{13}x_{21} - x_{11}x_{23},$$

$$(x_1 \wedge x_2)_3 = x_{11}x_{22} - x_{12}x_{21}.$$

From the definition, it follows that $x_1 \wedge x_2$ is linear in each argument and skew-symmetric, i.e., $x_1 \wedge x_2 = -x_2 \wedge x_1$.

Exercise 128 Show that $(x \wedge y) \perp x$ and $(x \wedge y) \perp y$. Calculate $e_2 \wedge e_1$ for two basis vectors of an orthonormal basis (e_1, e_2, e_3) of the given orientation.

Exercise 129 Calculate the vector product $(1, -12, 3) \wedge (0, 3, 6)$ of two vectors in \mathbb{R}^3 with the standard scalar product $b = (,)$ and the orientation of the standard basis $(e_1 = (1, 0, 0), e_2 = (0, 1, 0), e_3 = (0, 0, 1))$.

Proposition 218 Given a 3-dimensional Euclidean space (V, b) with a fixed orientation, the vector product satisfies these identities for any three vectors u, v and w :

$$(i) \quad u \wedge (v \wedge w) = b(u, w)v - b(u, v)w,$$

$$(ii) \quad (\text{Jacobi Identity}) \quad u \wedge (v \wedge w) + v \wedge (w \wedge u) + w \wedge (u \wedge v) = 0.$$

Proof Since the expressions in question are all linear in each argument, it suffices to verify them for $u = e_i, v = e_j, w = e_k, i, j, k = 1, 2, 3$, where e_1, e_2, e_3 is an orthonormal basis. Further, (i) is skew-symmetric in v and w whence some cases can be omitted. We leave to the reader the detailed verification, which follows from the formulas given in this section for the coordinatewise definition of the vector product. \square

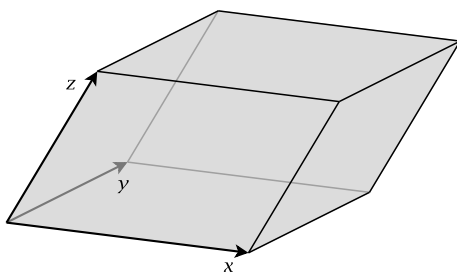


Fig. 25.2. The parallelepiped spanned by the vectors x, y and z .

Remark 30 We should add a remark on surfaces of parallelograms and volumes of parallelepipeds which have not yet been defined and which belong to the chapters on calculus in the second volume of this book. However, from high school the reader may temporarily recall the definitions of surfaces and volumes and read these remarks which will be fully justified later. If we are given $x, y, z \in V$, a Euclidean space of dimension 3, then we can consider the parallelogram spanned by x and y , i.e., the set $Parallel(x, y) = \{t \mid t = \lambda x + \mu y, 0 \leq \lambda, \mu \leq 1\}$ as well as the parallelepiped spanned by x, y and z , i.e., the set $Parallel(x, y, z) = \{t \mid t = \lambda x + \mu y + \nu z, 0 \leq \lambda, \mu, \nu \leq 1\}$. The surface of $Parallel(x, y)$ is $|x \wedge y|$, whereas the volume of $Parallel(x, y, z)$ is $|b(x \wedge y, z)|$, and these numbers are independent of the orientation.

25.3 Quaternions

We have learned that 2-dimensional rotations are essentially products of complex numbers of norm 1. We shall see in this section that in three dimensions, we also have an algebraic structure, the quaternions, which can be used to describe rotations in three dimensions. It seems that there is a deep relation between algebra and geometry, a relation which is classical in algebraic geometry, but which turns out to become more intense for the groups of transformations in linear geometry. It is not by chance that this field of research is named geometric algebra.

The quaternions were invented by the mathematician William Rowan Hamilton (1805-1865), who was in search for a correspondence to the geometric interpretation of complex numbers for three dimensions. Although quaternions were overrated by the so-called “quaternionists” (see [12]) as being a kind of magic objects, they are firmly established in computer graphics and also in aerospace science (see [35]).

Let us first motivate how quaternions relate to the vector product. As is shown by the Jacobi identity in proposition 218, the anticommutative (i.e., $x \wedge y = -y \wedge x$) vector product is not associative. In particular, $0 = (x \wedge x) \wedge y$, while $x \wedge (x \wedge y)$ may be different from 0. Quaternions were invented while looking for the construction of an algebraic multiplication structure which is associative, such that the square of an element x cannot be annihilated (as it happens with the vector product, since always $x \wedge x = 0$). As already seen with the complex numbers, such

a desideratum may be met by adding supplementary dimensions to the given space. Hamilton's solution was to add one dimension to the three dimensions of \mathbb{R}^3 and to delegate the non-vanishing part of the square to that new dimension. His philosophical justification of this procedure was that the three space coordinates must be supplemented by one time coordinate to describe the comprising four-dimensional space-time. It is not by chance (recall the definition of the vector product, which is intimately related to the bilinear form of the Euclidean space) that the new component of the Hamilton product of $u, v \in \mathbb{R}^3$ was just the negative of the scalar product $-(u, v)$. There is in fact nothing arbitrary in Hamilton's construction. It can not only be shown that \mathbb{C} is the only field of dimension two over the reals, but that the set of quaternions \mathbb{H} is the only skew field of dimension four over the reals! Here is the formal definition:

Definition 176 *The quaternions are a skew field \mathbb{H} (for Hamilton), whose additive group is \mathbb{R}^4 , and whose multiplication is defined as follows. One identifies \mathbb{R}^4 with the direct sum $\mathbb{R} \oplus \mathbb{R}^3$ defined by the projections $R : \mathbb{H} \rightarrow \mathbb{R} : (r, x, y, z) \mapsto r$ and $P : \mathbb{H} \rightarrow \mathbb{R}^3 : (r, x, y, z) \mapsto (x, y, z)$, such that every $q \in \mathbb{H}$ is uniquely decomposed as $q = r + p$ with $r = R(q) \in R(\mathbb{H}) \xrightarrow{\sim} \mathbb{R}$ and $p = P(q) \in P(\mathbb{H}) \xrightarrow{\sim} \mathbb{R}^3$. The summand p is called the pure part of q , and if $r = 0$, then q is called a pure quaternion. The summand r is called the real part of q , and if $p = 0$, then q is called a real quaternion. If the context is clear, the additive notation $q = r + p$ is always meant in this understanding.² The pure part is provided with the standard scalar product (\cdot, \cdot) in \mathbb{R}^3 , together with the orientation given by the canonical basis $((1, 0, 0), (0, 1, 0), (0, 0, 1))$. The quaternion product is now defined by*

$$(r + p) \cdot (r' + p') = (r \cdot r' - (p, p')) + (r \cdot p' + r' \cdot p + p \wedge p'),$$

i.e.,

$$R((r + p) \cdot (r' + p')) = r \cdot r' - (p, p'),$$

and

$$P((r + p) \cdot (r' + p')) = r \cdot p' + r' \cdot p + p \wedge p'.$$

On \mathbb{H} , conjugation is defined by $\bar{\cdot} : \mathbb{H} \rightarrow \mathbb{H} : q = r + p \mapsto \bar{q} = r - p$. The norm $\|q\|$ of a quaternion q is defined by

$$\|q\| = \sqrt{q \cdot \bar{q}},$$

² Remember remark 25 concerning addition in direct sums.

which makes sense since $q \cdot \bar{q} = r^2 + (p, p)$ is a non-negative real quaternion, which coincides with the square norm of q when interpreted in the standard Euclidean space $(\mathbb{R}^4, (\cdot, \cdot))$.

The immediate properties of \mathbb{H} , in particular the skew field properties, are summarized in the following sorite:

Sorite 219 Let $q, q' \in \mathbb{H}$ be quaternions. Then

- (i) Conjugation $\bar{\cdot} : \mathbb{H} \rightarrow \mathbb{H}$ is a linear anti-involution, i.e., $\overline{\bar{q}} = q$ and $\overline{q \cdot q'} = \bar{q}' \cdot \bar{q}$.
- (ii) $\bar{q} = q$ iff q is real, i.e., $q = R(q)$.
- (iii) $\bar{q} = -q$ iff q is pure, i.e., $q = P(q)$.
- (iv) q is pure iff q^2 is a real number ≤ 0 .
- (v) q is real iff q^2 is a real number ≥ 0 .
- (vi) We have $\|q \cdot q'\| = \|q\| \cdot \|q'\|$.
- (vii) With the quaternion product, \mathbb{H} becomes a skew field with $1_{\mathbb{H}} = 1+0$. The inverse of $q \neq 0$ is the quaternion $q^{-1} = \frac{1}{\|q\|^2} \bar{q}$.
- (viii) The injection $\mathbb{R} \rightarrow \mathbb{H} : r \mapsto r \cdot 1_{\mathbb{H}}$ identifies the subfield \mathbb{R} of real quaternions, which commutes with all quaternions, i.e., $r \cdot q = q \cdot r$ for all $r \in \mathbb{R}, q \in \mathbb{H}$. Conversely, if $q' \cdot q = q \cdot q'$ for all q , then q' is real.
- (ix) By linearity in each factor, the multiplication $x \cdot y$ on \mathbb{H} is entirely determined by the following multiplication rules for the four basis vectors $1_{\mathbb{H}}, i, j, k$. Observe that i, j, k are pure and, therefore, their vector product is defined.

$$\begin{aligned} 1_{\mathbb{H}} \cdot t &= t \cdot 1_{\mathbb{H}} = t \text{ for all } t = 1_{\mathbb{H}}, i, j, k, \\ i^2 &= j^2 = k^2 = -1_{\mathbb{H}}, \\ ij &= -ji = i \wedge j = k, \\ jk &= -kj = j \wedge k = i, \\ ki &= -ik = k \wedge i = j. \end{aligned}$$

The permutation (i, j, k) induces an automorphism of \mathbb{H} .

- (x) Setting $i = (0, 1, 0, 0), j = (0, 0, 1, 0), k = (0, 0, 0, 1)$, the three injections

$$(i) : \mathbb{C} \rightarrow \mathbb{H} : a + ib \mapsto a + ib,$$

$$(j) : \mathbb{C} \rightarrow \mathbb{H} : a + ib \mapsto a + jb,$$

$$(k) : \mathbb{C} \rightarrow \mathbb{H} : a + ib \mapsto a + kb$$

define identifications of the field \mathbb{C} with the subfields $\mathbb{R} + i\mathbb{R}$, $\mathbb{R} + j\mathbb{R}$, $\mathbb{R} + k\mathbb{R}$, respectively, which are related with each other by the automorphism of \mathbb{H} induced by the permutation (i, j, k) .

$$(xi) \text{ We have } (q, q') = \frac{1}{2}(\bar{q}q' + \bar{q}'q).$$

Proof For (i) and a quaternion $q = r + p$, we have $\bar{q} = \overline{(r + p)} = r - (-p) = r + p = q$. And for $q' = r' + p'$, we have

$$\begin{aligned} \overline{q \cdot q'} &= \overline{(r \cdot r' - (p, p')) + (r \cdot p' + r' \cdot p + p \wedge p')} \\ &= (r \cdot r' - (p, p')) - (r \cdot p' + r' \cdot p + p \wedge p') \\ &= (r' \cdot r - (p', p)) + (r \cdot (-p) + r' \cdot (-p') + p' \wedge p) \\ &= \overline{q'} \cdot \bar{q} \end{aligned}$$

Points (ii) and (iii) are obvious.

For (iv), we have this general formula: $q^2 = (r^2 - (p, p)) + 2rp$. If $q = p$, then $q \cdot q = -\|p\|^2 \leq 0$. Conversely, if $r \neq 0$, then if $p \neq 0$, $P(q^2) = 2rp \neq 0$. If q is real, then $q^2 = r^2 > 0$.

The proof of point (v) proceeds along the same straightforward calculation, we therefore leave it as an exercise for the reader.

Next, we prove (viii). The commutativity of real quaternions with all quaternions is immediate from the definition of the quaternion product. Conversely, if $q \cdot q' = q' \cdot q$ for all q' , then $p \wedge p' = p' \wedge p = -p \wedge p' = 0$ for all p' . But for $p \neq 0$, taking p, p' linearly independent yields a contradiction, so $p = 0$.

As to (vi), $\|q \cdot q'\| = \sqrt{q \cdot q' \cdot \bar{q} \cdot \bar{q}'} = \sqrt{q \cdot q' \cdot \bar{q}' \cdot \bar{q}} = \sqrt{q \cdot \|q'\|^2 \cdot \bar{q}}$, but real quaternions commute with all quaternions by (viii), so $\sqrt{q \cdot \|q'\|^2 \cdot \bar{q}} = \sqrt{\|q'\|^2 \cdot q \cdot \bar{q}} = \sqrt{\|q'\|^2 \cdot \|q\|^2} = \|q'\| \cdot \|q\|$, and we are done.

Next, we prove (ix). The first point is \mathbb{R} -linearity in each argument. But in view of the defining formula of multiplication, this follows from linearity of the scalar product $(?, ?)$ and of the vector product \wedge . The permutation (i, j, k) of the basis ($1_{\mathbb{H}}$ is fixed) transforms the equation $i \cdot j = k$ into $j \cdot k = i$ and this into $k \cdot i = j$, and this latter into the first, furthermore $i^2 = j^2 = k^2 = -1$ is invariant under all permutations of i, j, k , and so are the products with $1_{\mathbb{H}}$.

Point (vii) is straightforward, except that we must verify associativity. This follows from the associativity of multiplication rules for the basis $(1_{\mathbb{H}}, i, j, k)$ in (ix), since multiplication is \mathbb{R} -linear in each factor. The reader should verify all equalities $x \cdot (y \cdot z) = (x \cdot y) \cdot z$ for three arbitrary basis elements, but should also use the permutation (i, j, k) to minimize the cases to be studied.

Point (x) is obvious.

Point (xi) is an immediate consequence of the definition of the product. \square

We have now established all the necessary algebraic ingredients to tackle the promised geometric features. To this end, recall that we have a group isomorphism of $SO_2(\mathbb{R})$ (the special orthogonal group of \mathbb{R}^2) and the multiplicative group $U \subset \mathbb{C}$ of complex numbers of norm 1, i.e., the *unit circle* $U = S(\mathbb{C}) = \{z \mid \|z\| = 1\}$. Therefore we suggest to consider the group $S(\mathbb{H}) = \{q \mid \|q\| = 1\}$ of unit quaternions (in a Euclidean space, this subset is called the *unit sphere*). This is a group because of point (vi) in sorite 219. The relation between quaternion multiplication and 3-dimensional isometries is this:

Proposition 220 *Let $s, s' \in \mathbb{H}^*$ be non-zero quaternions.*

- (i) *If $s \in P(\mathbb{H})$ is pure, then the map $q \mapsto -s \cdot q \cdot s^{-1}$ leaves the pure quaternion space $P(\mathbb{H})$ invariant and its restriction to $P(\mathbb{H})$ is the reflection ρ_{s^\perp} in $P(\mathbb{H})$ at the plane s^\perp orthogonal to s .*
- (ii) *The map $q \mapsto \text{Int}_s(q) = s \cdot q \cdot s^{-1}$ leaves the pure quaternion space $P(\mathbb{H})$ invariant, and its restriction Int_s^P to $P(\mathbb{H})$ is a rotation in $SO_3(\mathbb{R})$.*
- (iii) *We have $\text{Int}_s^P = \text{Int}_{s'}^P$ iff $\mathbb{R}s = \mathbb{R}s'$.*
- (iv) *The restriction Int_s^{PS} of the map Int_s^P to arguments s in the unit sphere $S(\mathbb{H})$ is a surjective group homomorphism*

$$\text{Int}_s^{PS} : S(\mathbb{H}) \rightarrow SO_3(\mathbb{R})$$

with kernel $\text{Ker}(\text{Int}^{PS}) = \{\pm 1\}$.

Proof (i) By criterion (iv) in sorite 219, we suppose q^2 is a real number ≤ 0 . So $(\pm s \cdot q \cdot s^{-1})^2 = s \cdot q^2 \cdot s^{-1} = q^2$ is also real and ≤ 0 , i.e., $\pm s \cdot q \cdot s^{-1}$ is pure. These maps are evidently \mathbb{R} -linear. Since $\|\pm s \cdot q \cdot s^{-1}\| = \|s\| \cdot \|q\| \cdot \|s\|^{-1} = \|q\|$, the maps $q \mapsto \pm s \cdot q \cdot s^{-1}$ conserve norms and therefore their restrictions to $P(\mathbb{H})$ are in $O_3(\mathbb{R})$. Now, if s is pure, then $-s \cdot s \cdot s^{-1} = -s$, whereas for a pure $q \perp s$, by (xi) we have $0 = (q, s) = \frac{1}{2}(\bar{q} \cdot s + \bar{s} \cdot q) = \frac{-1}{2}(q \cdot s + s \cdot q)$, i.e., $-q \cdot s = s \cdot q$, whence $-s \cdot q \cdot s^{-1} = q \cdot s \cdot s^{-1} = q$, and therefore the plane s^\perp orthogonal to s remains fixed, i.e., $q \mapsto -s \cdot q \cdot s^{-1}$ on $P(\mathbb{H})$ is a reflection at s^\perp .

(ii) For any quaternion s the map $\text{Int}_s^P \in O_3(\mathbb{R})$, because

$$\begin{aligned} (sq s^{-1}, sq' s^{-1}) &= \frac{1}{2}(\overline{sq s^{-1}} \cdot sq' s^{-1} + \overline{sq' s^{-1}} \cdot sq s^{-1}) \\ &= \frac{1}{2}(s^{-1} \cdot \bar{s} q \cdot sq' s^{-1} + s^{-1} \cdot \bar{s} q' \cdot sq s^{-1}) \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{2}(\overline{s^{-1}} \cdot \overline{q} \cdot \overline{s} \cdot s q' s^{-1} + \overline{s^{-1}} \cdot \overline{q'} \cdot \overline{s} \cdot s q s^{-1}) \\
&= \frac{1}{2} \left(\frac{s}{\|s\|^2} \overline{q} \cdot (\|s\|^2 s^{-1}) \cdot s q' s^{-1} + \frac{s}{\|s\|^2} \overline{q'} \cdot (\|s\|^2 s^{-1}) \cdot s q s^{-1} \right) \\
&= \frac{1}{2} (s \overline{q} s^{-1} \cdot s q' s^{-1} + s \overline{q'} s^{-1} \cdot s q s^{-1}) \\
&= \frac{1}{2} s (\overline{q} q' + \overline{q'} q) s^{-1} \\
&= \frac{1}{2} (\overline{q} q' + \overline{q'} q) \\
&= (q, q')
\end{aligned}$$

The map Int_s for a real s is the identity, so suppose s is not real. Consider for $\mu \in [0, 1]$, the real unit interval, the quaternion $s_\mu = (1 - \mu) + \mu \cdot s$, so $s_0 = 1$, $s_1 = s$, and $s_\mu \neq 0$ for all $\mu \in [0, 1]$. So $Int_{s_\mu}^P \in O_3(\mathbb{R})$ for all $\mu \in [0, 1]$, i.e., $f(\mu) = \det(Int_{s_\mu}^P)$ is never zero, and it is always ± 1 . We shall show in the chapter on topology in volume II of this book, that f has the property of being continuous (a concept to be introduced in that chapter). Evidently, $f(0) = 1$. On the other hand, if we had $f(1) = -1$, then by continuity of f , there would exist $\mu \in [0, 1]$ with $f(\mu) = 0$, a contradiction, so $Int_s^P \in SO_3(\mathbb{R})$.

(iii) If $s' = \lambda s$, $\lambda \in \mathbb{R}^*$, then $s' \cdot q \cdot s'^{-1} = \lambda s \cdot q \cdot \lambda^{-1} s^{-1} = s \cdot q \cdot s^{-1}$. Conversely, if $Int_{s'}^P = Int_s^P$, then $-Int_{s'}^P = -Int_s^P$, so the rotation axis $\mathbb{R}s = \mathbb{R}s'$ is uniquely determined.

As to (iv), the map Int^{PS} is a group homomorphism, since $(s' \cdot s) \cdot q \cdot (s' \cdot s)^{-1} = s' \cdot s \cdot q \cdot s^{-1} \cdot s'^{-1}$. It is surjective, since every rotation is the product of an even number of reflections at hyperplanes (see proposition 208). So since the reflections are represented by the quaternion multiplications $-s \cdot q \cdot s^{-1}$ by (i), rotations are represented by the product of an even number of such reflections which yields the required map. Since two rotations are equal iff their quaternions s define the same line $\mathbb{R}s$, and since any line has exactly two points $t, -t$ on the sphere $S(\mathbb{H})$, the kernel is $\{\pm 1\}$. \square

So we recognize that rotations in $SO_3(\mathbb{R})$ can be described by conjugations Int_s^{PS} with quaternions, and the composition of rotations corresponds to the conjugation of products. This is exactly what was expected from the 2-dimensional case of complex multiplication. The identification of the quotient group $S(\mathbb{H})/\{\pm 1\}$ and $SO_3(\mathbb{R})$ gives us an interesting geometric interpretation of $SO_3(\mathbb{R})$: Observe that each line $\mathbb{R} \cdot x \subset \mathbb{R}^3$ intersects the unit sphere $S(\mathbb{R}^3)$ exactly in two points $s, -s$ of norm 1. These points are identified via the two-element group $\{\pm 1\}$. So the quotient group identifies with the set $\mathbb{P}_3(\mathbb{R})$ of lines through the origin (1-dimensional subspaces) in \mathbb{R}^4 . This space is called the three-dimensional *projective space of \mathbb{R}^3* . So we have this corollary:

Corollary 221 *The group $SO_3(\mathbb{R})$ is isomorphic to the projective space $\mathbb{P}_3(\mathbb{R})$ with the group structure induced from the quaternion multiplication.*

At this point we know that every rotation can be described by the conjugation with a quaternion s . However we do not yet know how the rotation axis and the rotation angle related to s .

Proposition 222 *Let $s = r + p$ be a non-zero quaternion. If s is real, it induces the identity rotation. So let us suppose $p \neq 0$. Then $\mathbb{R}p$ is the rotation axis of Int_s^{PS} . The rotation angle $\theta \in [0, \pi]$ is given by*

$$\tan\left(\frac{\theta}{2}\right) = \sin\left(\frac{\theta}{2}\right) / \cos\left(\frac{\theta}{2}\right) = \|p\|/r \text{ if } r \neq 0,$$

and by

$$\theta = \pi \text{ if } r = 0,$$

the second case corresponding to a reflection orthogonally through $\mathbb{R}p$.

Proof Wlog, we may suppose that $\|s\| = 1$. If $p \neq 0$, then $s \cdot p \cdot s^{-1} = (r + p) \cdot p \cdot (r - p) = p \cdot (r^2 - p^2) = p(r^2 + (p, p)) = p$. So p generates the rotation axis. We omit the somewhat delicate proof of the tangent formula and refer to [4]. \square

Remark 31 Of course, this is a fairly comfortable situation for the parametric description of rotations via quaternions. However, there is an ambiguity by the two-element kernel of the group homomorphism Int^{PS} . But there is no right inverse group homomorphism $f : SO_3(\mathbb{R}) \rightarrow S(\mathbb{H})$, i.e., such that $Int_s^{PS} \circ f = Id_{SO_3(\mathbb{R})}$. For details, see [4].

For practical purposes it is advantageous to restate the fact that $s \in S(\mathbb{H})$. If $s = r + p$ is the real-pure decomposition, then $s \in S(\mathbb{H})$ means $r^2 + \|p\|^2 = 1$. Thus, for the rotation angle θ calculated in proposition 222, we have $r = \cos\left(\frac{\theta}{2}\right)$ and $p = \sin\left(\frac{\theta}{2}\right)(ix + jy + kz)$, where $(x, y, z) \in S^2 = S(\mathbb{R}^3)$, the unit sphere in \mathbb{R}^3 . In other words, writing $s_\theta = \left(\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right)\right)$ and $s_{dir} = (x, y, z)$ we have a representation of s as $s_S = (s_\theta, s_{dir}) \in S^1 \times S^2$, the Cartesian product of the unit circle and the unit sphere.

Example 104 Let us now consider a rotation of 120 degrees about the axis defined by $(1, 1, 1)$. This rotation should take the x -axis to the y -axis, the y -axis to the z -axis, and the z -axis to the x -axis.

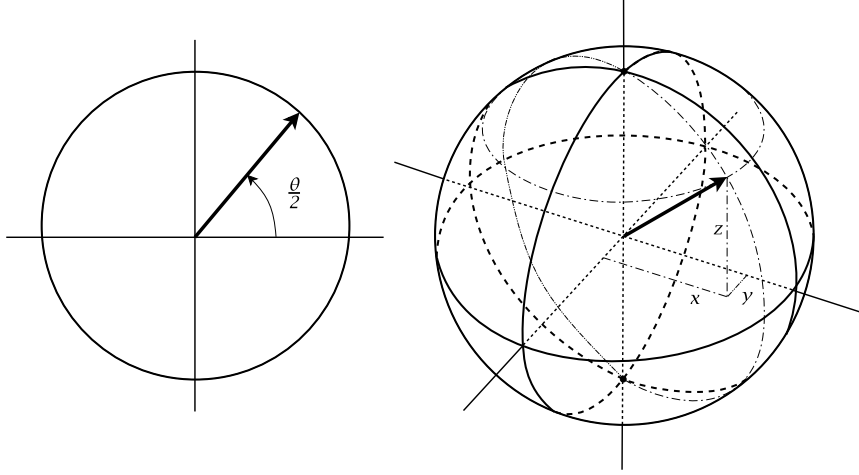


Fig. 25.3. The representation of $s \in S(\mathbb{H})$ as $s_S(s_\theta, s_{dir})$, where s_θ is a vector on the unit circle S^1 and s_{dir} a vector on the unit sphere S^2 .

According to the representation above, the quaternion s describing this rotation can be written as

$$s_S = (s_\theta, s_{dir})$$

where

$$s_\theta = \left(\cos\left(\frac{\theta}{2}\right), \sin\left(\frac{\theta}{2}\right) \right) \text{ and } s_{dir} = \frac{1}{\sqrt{3}}(1, 1, 1).$$

Now to rotate a point $p = (p_x, p_y, p_z)$ we first have to write it as a pure quaternion $\hat{p} = ip_x + jp_y + kp_z$ and then calculate

$$\hat{p}' = s \cdot \hat{p} \cdot \bar{s},$$

knowing from sorite 219, (vii), that for $s \in S(\mathbb{H})$, $s^{-1} = \bar{s}$. Some calculations using the multiplication rules lined out in sorite 219, (x), (we advise the student to do them as an exercise) lead to the following result:

$$\begin{aligned} \hat{p}' = & i(p_x(u^2 - v^2) + 2p_y(v^2 - uv) + 2p_z(v^2 + uv)) + \\ & j(2p_x(v^2 + uv) + p_y(v^2 - u^2) + 2p_z(v^2 - uv)) + \\ & k(2p_x(v^2 - uv) + 2p_y(v^2 + uv) + p_z(u^2 - v^2)), \end{aligned}$$

where

$$u = \cos\left(\frac{\theta}{2}\right) \text{ and } v = \frac{1}{\sqrt{3}} \sin\left(\frac{\theta}{2}\right).$$

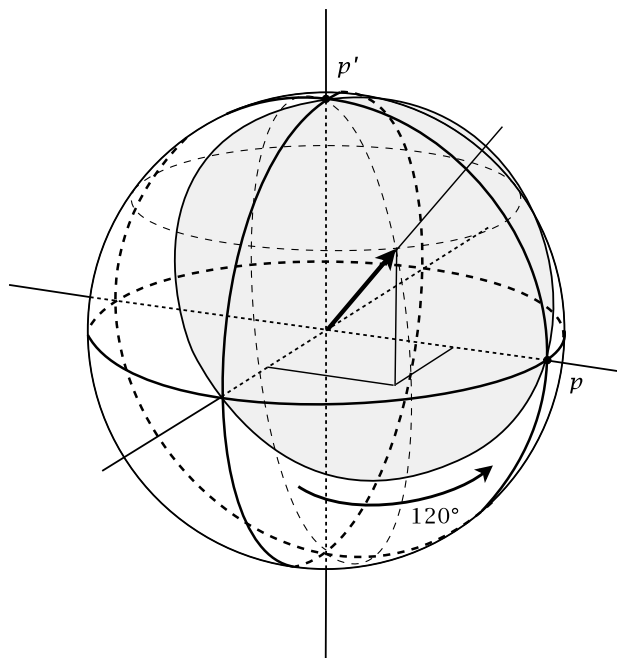


Fig. 25.4. The rotation by $\theta = \frac{2\pi}{3}$ around the axis defined by the vector $(\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}})$ maps a point p on the x -axis to a point p' on the y -axis.

If we look at the coordinates of \hat{p}' using the value of $\frac{2\pi}{3}$ for the 120° -rotation which we want to perform, we find that

$$\begin{aligned}
 u^2 - v^2 &= \cos^2\left(\frac{\pi}{3}\right) - \frac{1}{3} \sin^2\left(\frac{\pi}{3}\right) \\
 &= \left(\frac{1}{2}\right)^2 - \frac{1}{3} \left(\frac{\sqrt{3}}{2}\right)^2 \\
 &= \left(\frac{1}{2}\right)^2 - \left(\frac{1}{2}\right)^2 \\
 &= 0 \\
 v^2 - uv &= \frac{1}{3} \sin^2\left(\frac{\pi}{3}\right) - \cos\left(\frac{\pi}{3}\right) \cdot \frac{1}{\sqrt{3}} \cdot \sin\left(\frac{\pi}{3}\right) \\
 &= \frac{1}{3} \left(\frac{\sqrt{3}}{2}\right)^2 - \frac{1}{2} \cdot \frac{1}{\sqrt{3}} \cdot \frac{\sqrt{3}}{2} \\
 &= 0 \\
 v^2 + uv &= \frac{1}{3} \sin^2\left(\frac{\pi}{3}\right) + \cos\left(\frac{\pi}{3}\right) \cdot \frac{1}{\sqrt{3}} \cdot \sin\left(\frac{\pi}{3}\right)
 \end{aligned}$$

$$\begin{aligned} &= \frac{1}{3} \left(\frac{\sqrt{3}}{2} \right)^2 + \frac{1}{2} \cdot \frac{1}{\sqrt{3}} \cdot \frac{\sqrt{3}}{2} \\ &= \frac{1}{2} \end{aligned}$$

Here we use the trigonometric facts that $\sin(\frac{\pi}{3}) = \frac{\sqrt{3}}{2}$, and $\cos(\frac{\pi}{3}) = \frac{1}{2}$. Putting this together again, we get

$$\hat{p}' = ip_z + jp_x + kp_y$$

and for $p = (1, 0, 0)$ we do indeed get $p' = (0, 1, 0)$.

Second Advanced Topic

In this second advanced chapter, we shall briefly describe the theory of finite fields, called Galois fields, and then give two important applications of this theory: The first is Reed's and Solomon's error correction code, which is of great importance in media encoding for CDs, DVDs, ADSL, etc. The second is the encryption algorithm developed by Rivest, Shamir and Adleman, which is of practical use in digital signature generation, for example.

26.1 Galois Fields

Galois fields are by definition commutative fields with finite cardinality. (In field theory, it can be shown that every finite skew field is in fact automatically commutative, so no other finite fields exist.) Their name has been chosen in honor of the French mathematician Evariste Galois (1811-1832), who invented the algebraic theory of polynomial equations and thereby was, among others, able to solve some of the most difficult questions in mathematics: The conjecture of the impossibility of a generally valid construction by compass and straightedge of the trisection of an angle or the Delian problem concerning the duplication of the cube by the same type of construction. Unfortunately, he was killed in a duel at the age of 20. His research had not been understood during his short lifetime. The famous mathematician Siméon-Denis Poisson commented on a Galois paper: "His argument is neither sufficiently clear nor sufficiently developed to allow us to judge its rigor." Only in 1843, the mathemati-

cian Joseph Liouville recognized Galois' work as a solution "as correct as it is deep of this lovely problem: Given an irreducible equation of prime degree, decide whether or not it is soluble by radicals." This is in fact the abstract statement implying the solution of the old problems mentioned above.

We shall outline the theory of Galois fields, because they play a crucial role in coding theory, the theory dealing with the methods for storing and transmitting digital information through noisy media. In fact, bit strings can be represented by polynomials with coefficients in \mathbb{Z}_2 , which are elements of a particular Galois field. Thus we can take over the results from the theory of Galois fields to the realm of bit strings. In the next section, this will be exemplified by the famous Reed-Solomon error correction code.

We start with a few definitions:

Definition 177 Let K be a field, where 0_K is the additive neutral element, and 1_K is the multiplicative unit. Consider the ring homomorphism $p : \mathbb{Z} \rightarrow K$, with

$$p(n) = n \cdot 1_K = \underbrace{1_K + \cdots + 1_K}_{n \text{ times}}.$$

The characteristic of K , denoted by $\text{char}(K)$ is the smallest positive n , such that $p(n) = 0_K$. If there is no such n , then, by definition, $\text{char}(K) = 0$.

Definition 178 Let K be a field. The prime field of K , denoted by $P(K)$, is the smallest subfield of K .

Since $1_K \in P(K)$, we have $\text{char}(P(K)) = \text{char}(K)$. The following lemma provides a classification of prime fields.

Lemma 223 Given a field K , consider the unique homomorphism of rings $p : \mathbb{Z} \rightarrow K : n \mapsto n \cdot 1_K$ from definition 177.

Then $\text{Ker}(p) = (\text{char}(K))$ is the principal ideal generated by $\text{char}(K)$. There are two cases to consider:

1. $\text{Ker}(p)$ is trivial, i.e., $\text{char}(K) = 0$. Then $P(K)$ is isomorphic to \mathbb{Q} , because it contains all integers and, being a field, all inverses of integers.
2. The characteristic $\text{char}(K) > 0$. Since $\mathbb{Z}/(\text{char}(K))$ is a field, $\text{char}(K)$ is a prime by lemma 133. By proposition 124, $\text{Im}(p) \cong \mathbb{Z}/(\text{char}(K))$.

Since $P(K) = \text{Im}(p)$ ($\text{Im}(p) \subset P(K)$, but because $P(K)$ is minimal, $P(K) = \text{Im}(p)$), we have $P(K) \cong \mathbb{Z}/(\text{char}(K))$.

Thus a Galois field (which is just another name for finite field) K has positive prime characteristic p . Clearly, a Galois field is a vector space over its prime field, and of finite dimension n . So, since the prime field $P(K)$ of K is isomorphic to \mathbb{Z}_p , we have the vector space isomorphism $K \cong \mathbb{Z}_p^n$, and therefore $\text{card}(K) = \text{card}(\mathbb{Z}_p^n) = p^n$.

Moreover, since the group K^* has order $p^n - 1$, and every element a of the group has an order which divides $p^n - 1$, we have $a^{p^n - 1} = 1$ (see sorite 117 (iii)). Therefore all elements of K^* are roots of the polynomial $X^{p^n - 1} - 1$.

Together with 0_K , the elements of a field K with characteristic p and p^n elements constitute the set of roots of the polynomial $X(X^{p^n - 1} - 1) = X^{p^n} - X \in \mathbb{Z}_p[X]$. Since the degree of this polynomial is p^n , it has at most p^n different roots, and therefore it decomposes into a product

$$X^{p^n} - X = \prod_{x \in K} (X - x)$$

of p^n different linear factors. It remains to establish whether the characteristic and the polynomial uniquely determine K up to field isomorphisms.

To this end, we sketch a short but comprehensive run through the theory dealing with the existence and uniqueness of fields defined by roots of certain polynomials. In all these discussions, a *field extension* is a pair of fields K, L such that $K \subset L$. If $K \subset L$ is such a field extension, then L is called an *extension of K* . If $T \subset L$, we denote by $K(T)$ the smallest extension of K in L containing T and call it the *extension of K generated by T* . If $T = \{x\}$, then we also write $K(x)$ instead of $K(\{x\})$.

To begin with, we have the following definition:

Definition 179 *If $K \subset L$ is an extension and $f \in K[X] - \{0\}$, then $x \in L$ is called an algebraic element (over K) if $f(x) = 0$.*

An extension $K \subset L$ is called an algebraic extension if every element of L is algebraic over K .

We have this lemma about the uniqueness of extensions.

Lemma 224 *If $K \subset L$ is an extension, and if $x \in L$ is algebraic, then there is a unique irreducible polynomial $r = X^m + a_{m-1}X^{m-1} + \dots + a_0 \in K[X]$ such that the extension $K(x) \subset L$ of K is isomorphic to $K[X]/(r)$, and which, as a K -vector space, has dimension $m = \deg(r)$, a basis being defined by the sequence $(x^{m-1}, \dots, x^2, x, 1)$. The polynomial r is called the defining polynomial of x .*

If the dimension of an extension $K \subset L$ as a K -vector space is finite, then L is algebraic over K . In particular the extension $K(x)$ by an algebraic element x is algebraic. If extensions $K \subset L$ and $L \subset M$ are algebraic, then so is $K \subset M$.

Conversely, given an irreducible polynomial $r = X^m + a_{m-1}X^{m-1} + \dots + a_0 \in K[X]$, there is an algebraic extension $K \subset L = K(x)$ such that the defining polynomial of x is r , i.e., $K(x)$ is isomorphic to $K[X]/(r)$.

Example 105 A Galois field K of characteristic p and dimension n is an algebraic extension of its prime field $P(X)$ which is isomorphic to \mathbb{Z}_p . We have seen that every element $x \in K$ is a root of a polynomial in $\mathbb{Z}_p[X]$ (to be precise, the polynomial $X^{p^n} - X$), i.e., every element is algebraic.

Exercise 130 Show that the $\mathbb{R} \subset \mathbb{C}$ is an algebraic extension.

Proposition 225 *Consider two field extensions $K \subset L = K(x_1, x_2, \dots, x_r)$ and $K' \subset L' = K'(x'_1, x'_2, \dots, x'_r)$. Let k be an isomorphism $k : K \xrightarrow{\sim} K'$ with this property: there is a polynomial $f \in K[X]$, such that $f = (X - x_1)(X - x_2) \dots (X - x_r) \in L[X]$, which is mapped to the polynomial $f' \in K'[X]$ by the extension of k to the polynomial ring $K[X]$, such that $f' = (X - x'_1)(X - x'_2) \dots (X - x'_r) \in L'[X]$. Then there is an extension $l : L \xrightarrow{\sim} L'$ of k , i.e., $l|_K = k$.*

The proof of proposition 225 is by induction on the maximal degree of the irreducible factors in the decomposition of f in $K[X]$, and then on the number of such maximal degree factors. The inductive step is in fact provided by the above lemma 224.

The next proposition ensures that every polynomial over a field K gives rise to a field extension of K .

Proposition 226 *If K is a field, and if $f \in K[X]$ is a polynomial, there exists a splitting field extension $K \subset L$ of f , i.e., $L = K(x_1, \dots, x_r)$ where x_i are roots of f such that in $L[X]$, $f = a(X - x_1)(X - x_2) \dots (X - x_r)$.*

The proof of this proposition is again an immediate consequence of lemma 224. In fact, for each irreducible factor g of f in $K[X]$, we may embed K in the field $K[X]/(g)$, which, by construction, contains a root of g . And so forth until f has been split into linear factors.

We are now ready to state the main result of this section:

Corollary 227 *For a given prime characteristic p and a given exponent $n > 0$, there is a Galois field of the cardinality p^n . It is a splitting field of the polynomial $X^{p^n} - X$, and any two such fields are isomorphic. They are denoted by $\text{GF}(p^n)$.*

Summarizing, for every prime p and every positive natural exponent n , there is one, and, up to field isomorphisms, only one Galois field $\text{GF}(p^n)$ of p^n elements. Let us now show that in fact, $\text{GF}(p^n) = \mathbb{Z}_p(\zeta)$, i.e., an algebraic extension by a single element ζ .

Proposition 228 *The multiplicative group $\text{GF}(p^n)^*$ of a Galois field is cyclic, i.e., isomorphic to the additive group \mathbb{Z}_{p^n-1} . A generator ζ of this group is called a $(p^n - 1)$ -th primitive root of unity.*

The proof is easy, see [46, page 42] for details.

Therefore a Galois field $\text{GF}(p^n)$ is an algebraic extension $\text{GF}(p^n) = \mathbb{Z}_p(\zeta)$ by a primitive root of unity ζ , whose defining polynomial $Z \in \mathbb{Z}_p[X]$ is of degree n , the powers $1, \zeta, \zeta^2, \dots, \zeta^{n-1}$ forming a basis over the prime field. But observe that there may be different defining polynomials for the same Galois field.

26.1.1 Implementation

We shall discuss the Reed-Solomon error correction code for the special prime field \mathbb{Z}_2 which may be identified with the bit set $\text{Bit} = \{0, 1\}$ for computerized implementation. In this case, we need to master the arithmetics in $\text{GF}(2^n)$ on the basis of bit-wise encoding.

Identifying $\text{GF}(2^n)$ with $\mathbb{Z}_2[X]/(Z)$, where Z is the defining polynomial of a primitive $(2^n - 1)$ -th root of unity ζ , we know that elements $u \in \text{GF}(2^n)$ are uniquely represented in the basis $(x^{n-1}, \dots, x^2, x^1, 1)$, where x is the image of $X \in \mathbb{Z}_2[X]$ in $\mathbb{Z}_2[X]/(Z)$ by the canonical homomorphism and corresponds to ζ . Through the identification of $\text{GF}(2^n)$ with $\mathbb{Z}_2[X]/(Z)$,

and the vector space identification $\text{GF}(2^n) \xrightarrow{\sim} \mathbb{Z}_2^n$, we can identify $u(x) \in \mathbb{Z}_2[X]/(Z)$ with the vector $u = (u_{n-1}, \dots, u_2, u_1, u_0) \in \mathbb{Z}_2^n$, i.e., with a bit sequence of length n , which encodes the class of the polynomial residue $u(x) = u_{n-1}x^{n-1} + \dots + u_2x^2 + u_1x + u_0$ in $\mathbb{Z}_2[X]/(Z)$.

Arithmetic in this representation is as follows:

- Addition is the coordinate-wise addition of bits and has nothing to do with the defining polynomial, except for its degree n :

$$(u_{n-1}, \dots, u_1, u_0) + (v_{n-1}, \dots, v_1, v_0) = (u_{n-1} + v_{n-1}, \dots, u_1 + v_1, u_0 + v_0)$$

This addition may also be viewed as the logical exclusive alternative *xor* on the Boolean algebra $\text{Bit} = 2$, via $a + b = \neg(a \Leftrightarrow b) = \text{xor}(a, b)$.

- Multiplication $u(x) \cdot v(x)$ is defined as follows:

The multiplication of $u(x)$ by a constant $v \in \mathbb{Z}_2$ is the coordinate-wise multiplication of bits, i.e., corresponding to logical conjunction $a \cdot b = a \& b$.

For the multiplication $u(x) \cdot x$, first the residue $r(x)$ in $x^n = 1 \cdot Z + r(x)$ has to be calculated. This has to be done only once for the given n and Z . With $r = (r_{n-1}, \dots, r_2, r_1, r_0)$ the following two steps are performed:

1. a shift of $u = (u_{n-1}, \dots, u_2, u_1, u_0)$ to the left, yielding $l(u) = (u_{n-2}, \dots, u_1, u_0, 0)$,
2. the addition $l(u) + r$ in case $u_{n-1} \neq 0$.

The full multiplication $u(x) \cdot v(x)$ is written as a succession of these elementary operations (easily implemented in hardware using a shift register, and a simple logical unit capable of $\&$ and *xor* operations). It is the realization of the stepwise multiplication of $u(x)$ by the terms of $v(x)$ (also known as Horner scheme):

$$\begin{aligned} &u(x) \cdot v_{n-1} \\ &(u(x) \cdot v_{n-1}) \cdot x \\ &((u(x) \cdot v_{n-1}) \cdot x) + v_{n-2} \\ &(((u(x) \cdot v_{n-1}) \cdot x) + v_{n-2}) \cdot x \\ &((((u(x) \cdot v_{n-1}) \cdot x) + v_{n-2}) \cdot x) + v_{n-3} \\ &\vdots \end{aligned}$$

$$(\dots (u(x) \cdot v_{n-1}) \cdot x) \dots) + v_0$$

Example 106 We consider the Galois field $\text{GF}(2^4)$. It has 16 elements, which are represented as four-bit words, e.g., $u = (1, 1, 0, 1)$ corresponds to $u(x) = x^3 + x^2 + 1$. The primitive root of unity can be defined by the polynomials $Z = X^4 + X + 1$ or $Z = X^4 + X^3 + 1$. Let us take $Z = X^4 + X + 1$. Then we have the remainder formula $X^4 = 1 \cdot Z + (X + 1)$ (observe that $1 = -1$ in \mathbb{Z}_2). So $x^4 = x + 1$ in $\mathbb{Z}_2[X]/(Z)$ which is represented as $(0, 0, 1, 1)$. Therefore the multiplication $u(x) \cdot x$ is implemented as a left shift of u followed by the addition of $(0, 0, 1, 1)$.

Exercise 131 Calculate the 16 powers x^i of x for $i = 1, 2, 3, \dots, 16$ in the example 106 by means of the multiplication algorithm described above.

The product in $\text{GF}(p^n)$ is also easily encoded by observing the fact that the multiplicative group $\text{GF}(p^n)^*$ is isomorphic to the additive group \mathbb{Z}_{p^n-1} . In the representation $\mathbb{Z}_p[X]/(Z)$ of $\text{GF}(p^n)$, x is a generator, i.e., every non-zero element of $\mathbb{Z}_p[X]/(Z)$ occurs as a power x^i . Thus, we have the group isomorphism $\varphi : \text{GF}(p^n)^* \rightarrow \mathbb{Z}_{p^n-1}$, defined by $\varphi(x^i) = i$, and $\varphi(x^i x^j) = i + j$. Therefore, multiplication in $\mathbb{Z}_p[X]/(Z)$ can be replaced by addition in \mathbb{Z}_{p^n-1} .

As an example, the $3^2 - 1 = 8$ elements of $\text{GF}(3^2)^*$ when represented as elements of $\mathbb{Z}_3[X]/(X^3 + X + 1)$ can be calculated as the powers of x , i.e., by successive multiplication by x , starting with 1 (all operations are in $\mathbb{Z}_3[X]/(X^3 + X + 1)$):

$$\begin{array}{ll} x^0 = 1, & x^4 = (2x + 2) \cdot x = 2x^2 + 2x, \\ x^1 = x, & x^5 = (2x^2 + 2x) \cdot x = 2x^2 + x + 1, \\ x^2 = x \cdot x = x^2, & x^6 = (2x^2 + x + 1) \cdot x = x^2 + 2x + 1, \\ x^3 = x^2 \cdot x = 2x + 2, & x^7 = (x^2 + 2x + 1) \cdot x = 2x^2 + 2. \end{array}$$

26.2 The Reed-Solomon (RS) Error Correction Code

The Reed-Solomon error correction code was invented in 1960 by Lincoln Laboratory (MIT) members Irving S. Reed and Gustave Solomon and published in [39]. When it was written, digital technology was not advanced enough to implement the concept. The key to the implementation of Reed-Solomon codes was the invention of an efficient decoding algorithm

by Elwyn Berlekamp, a professor of electrical engineering at the University of California, Berkeley (see his paper [5]). The Reed-Solomon code is used in storage devices (including tape, Compact Disk, DVD, barcodes, etc.), wireless or mobile communications (such as cellular telephones or microwave links), satellite communications, digital television, high-speed modems such as ADSL. The encoding of digital pictures sent back by the Voyager space mission in 1977 was the first significant application.

The following development is akin to the exposition in [42]. The Reed-Solomon code adds redundant information to a message, such that errors in the entire transmitted message can be found and even corrected. We call the code $RS(s, k, t)$, where s, k, t are natural numbers which define specific choices as follows:

We start from a sequence $S = (b_i)_i$ of bits, which are subdivided into words of s bits each. So the sequence S is interpreted as a sequence $(c_j)_j$, where $c_j = (b_{s \cdot j}, b_{s \cdot j+1}, \dots, b_{s \cdot j+s-1}) \in GF(2^s)$. This sequence is split into blocks $c = (c_0, c_1, \dots, c_{k-1}) \in GF(2^s)^k$ of k elements $c_j \in GF(2^s)$ each (see figure 26.1). The encoding is an injective $GF(2^s)$ -linear map

$$\epsilon : GF(2^s)^k \rightarrow GF(2^s)^{k+2t}$$

with k, t such that $k + 2t \leq 2^s - 1$.

To define ϵ , one takes a primitive $(2^s - 1)$ -th root ζ of unity in the cyclic group $GF(2^s)^*$, and considers the polynomial $p(X) = (X - \zeta)(X - \zeta^2) \dots (X - \zeta^{2^s-1}) \in GF(2^s)[X]$. One then encodes the block vector $c = (c_0, c_1, \dots, c_{k-1}) \in GF(2^s)^k$ as a polynomial $c(X) = \sum_{i=0, \dots, k-1} c_i X^i \in GF(2^s)[X]$. Then, $p(X) \cdot c(X) = \sum_{i=0, \dots, k+2t-1} d_i X^i$, and we set

$$\epsilon(c) = d = (d_0, d_1, \dots, d_{k+2t-1})$$

which is obviously linear in c and injective, since c can be recovered from d : denote the polynomial $\sum_{i=0, \dots, k+2t-1} d_i X^i$ as $d(X)$, i.e., $p(X) \cdot c(X) = d(X)$, whence $c(X) = d(X)/p(X)$.

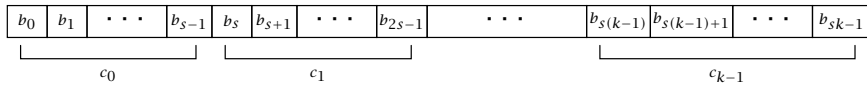


Fig. 26.1. One block c consists of k elements c_i with s bits each.

Proposition 229 (2t-error detection) *Given the above notations, if the measured value f differs from the encoded value d in at most $2t$ positions, and we denote the noise $e = f - d \in \text{GF}(2^s)^{k+2t}$, then $e = 0$ (i.e., there has been no error from noise) iff $(f(\zeta^i))_{i=1,\dots,2t} = 0$, where $f(\xi)$ is the evaluation of $f(X)$ at ξ .*

In fact, evaluating $e(X)$ at ζ^i for all i , we have $e(\zeta^i) = f(\zeta^i) + d(\zeta^i)$, and, since $d(\zeta^i) = 0$ by construction, $e(\zeta^i) = f(\zeta^i)$. This can be rewritten as matrix equation:

$$(f(\zeta^i))^\tau = \begin{pmatrix} 1 & \zeta & \zeta^2 & \dots & \zeta^{k+2t-1} \\ 1 & \zeta^2 & \zeta^4 & \dots & \zeta^{2(k+2t-1)} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & \zeta^{2t} & \zeta^{4t} & \dots & \zeta^{2t(k+2t-1)} \end{pmatrix} \cdot (e_i)^\tau.$$

Now, if all $e_i = 0$ except of at most $2t$ indexes $i_1 < i_2 < \dots < i_{2t}$, then the above equation reduces to

$$(f(\zeta^i))^\tau = \begin{pmatrix} \zeta^{i_1} & \zeta^{i_2} & \dots & \zeta^{i_{2t}} \\ \zeta^{2i_1} & \zeta^{2i_2} & \dots & \zeta^{2i_{2t}} \\ \vdots & \vdots & & \vdots \\ \zeta^{2ti_1} & \zeta^{2ti_2} & \dots & \zeta^{2ti_{2t}} \end{pmatrix} \cdot (e_{i_j})^\tau = Z \cdot (e_{i_j})^\tau$$

where (e_{i_j}) is a row vector of length $2t$. But the $(2t \times 2t)$ -matrix Z of the ζ -powers is of rank $2t$, therefore Z is invertible, and we have $(e_{i_j})^\tau = Z^{-1}f(\zeta^i)^\tau = 0$, whence the claim, that $(f(\zeta^i))_{i=1,\dots,2t} = 0$ implies $e = 0$.

Let us understand why Z is regular. We have

$$\det(Z) = \zeta^{i_1+i_2+\dots+i_{2t}} \cdot \det \begin{pmatrix} 1 & 1 & \dots & 1 \\ \zeta^{i_1} & \zeta^{i_2} & \dots & \zeta^{i_{2t}} \\ \zeta^{2i_1} & \zeta^{2i_2} & \dots & \zeta^{2i_{2t}} \\ \vdots & \vdots & & \vdots \\ \zeta^{(2t-1)i_1} & \zeta^{(2t-1)i_2} & \dots & \zeta^{(2t-1)i_{2t}} \end{pmatrix}.$$

using the properties of the determinant presented in proposition 181. The matrix to the right of the power of ζ is a Vandermonde matrix, whose determinant is known to be $\prod_{1 \leq u < v \leq 2t} (\zeta^{i_u} - \zeta^{i_v})$. See [23] for a proof. Since $i_u < k + 2t$ by construction of the matrix and $k + 2t \leq 2^s - 1$ by hypothesis, no two powers ζ^{i_u}, ζ^{i_v} are equal for $u \neq v$, and the Vandermonde matrix is regular.

Proposition 230 (*t*-error correction) *If with the above notations, the encoded value d is altered in at most t positions, then e can be calculated from f , and therefore the original value d and, hence, c can be reconstructed.*

We already know that, under these assumptions, the above Vandermonde $2t \times 2t$ -matrixes for subsequences i_\bullet of indexes $i_1 < i_2 < \dots < i_{2t}$ are regular. Take all these subsequences i_\bullet and calculate the vectors $e_{i_\bullet} = Z_{i_\bullet}^{-1} f(\zeta^{i_\bullet})^\tau$. Then, if one such vector has at most t non-vanishing entries, we are done, i.e., the error vector vanishes outside the indexes i_\bullet and is e_{i_\bullet} for the indexes i_\bullet . In fact, any two solutions $e_{i'_\bullet}$ and $e_{i''_\bullet}$, if they are different, yield two different solutions to a $2t$ index sequence i_\bullet containing the union of indexes, where the two solutions do not vanish, a contradiction to the regularity of the corresponding Vandermonde matrix.

Example 107 Let us consider an example from the digital storage of music on compact disks. It can be shown that one hour of stereo music with 16-bit resolution and 44100 Hz sampling rate needs roughly $635\text{MB} = 635 \times 8 \times 10^6 = 5080000000$ bits. Suppose that we want to be able to correct a burst of up to 200 bit errors. This means that if a block has s bits, such a burst hits up to $\lceil 200/s \rceil + 1$ blocks, where $\lceil x \rceil$ is the least integer $\geq x$. We may reconstruct such errors if $t > \lceil 200/s \rceil$. Moreover, the condition $k + 2t \leq 2^s - 1$ implies $k + 2\lceil 200/s \rceil + 2 \leq 2^s - 1$ and can be met by $k = 2^s - 3 - 2\lceil 200/s \rceil$, whence also $t = \lceil 200/s \rceil + 1$. So one block of $k \cdot s$ bits makes a total of $\lceil 5080000000 / (k \cdot s) \rceil$ blocks, and each being expanded to $k + 2t$ blocks, we get a number of

$$\begin{aligned} N(s) &= \lceil 5080000000 / (k \cdot s) \rceil \cdot (k + 2t) \cdot s \\ &= \lceil 5080000000 / (2^s - 3 - 2\lceil 200/s \rceil) \cdot s \rceil \cdot (2^s - 1) \cdot s \end{aligned}$$

bits. This yields the following numbers of MB required to meet this task for $s = 6 \dots 12$:

s	6	7	8	9	10	11	12
$N(s)$	-5715.00	1203.66	797.66	700.83	662.19	647.66	640.63

Observe that below $s \leq 6$ no reasonable bit number is possible, and that for $s = 12$, we get quite close to the original size.

26.3 The Rivest-Shamir-Adleman (RSA) Encryption Algorithm

The RSA algorithm was published in 1978 by Ron Rivest, Adi Shamir, and Leonhard Adleman [40]. It can be used for public key encryption and digital signatures. Its security is based on the difficulty of factoring large integers, and again uses the theory of Galois fields.

The **first step** is the *generation of the public and private key* and runs as follows:

1. Generate two different large primes, p and q , of approximately equal size, such that their product $n = pq$ is of a bit length, e.g., 1024 bits, which is required for the representation of the message as a big number, see below in the next paragraph.
2. One computes $n = pq$ and $\phi = (p - 1)(q - 1)$.
3. One chooses a natural number e , $1 \leq e \leq \phi$, such that $\gcd(e, \phi) = 1$. Then by definition, the couple (e, n) is the *public key*.
4. From the previous choice and the verification that $\gcd(e, \phi) = 1$, one computes the inverse d , $1 \leq d \leq \phi$, i.e. $de = 1 \pmod{\phi}$. By definition, the couple (d, n) is the *private key*.
5. The values p , q , ϕ , and d are kept secret.

The **second step** describes the *encryption* of the message sent from A to B . The private knowledge (p_B, q_B, ϕ_B) is attributed to B who will be able to decipher the encrypted message from A .

1. B 's public key (n_B, e_B) is transmitted to A .
2. The message is represented by a (possibly very large) natural number $1 \leq m \leq n_B$.
3. The encrypted message (the "cyphertext") is defined by

$$c = m^{e_B} \pmod{n_B}.$$

4. The cyphertext c is sent to B .

Since B knows that the message number m is unique in \mathbb{Z}_{n_B} , then once we have recalculated $m \pmod{n_B}$, we are done.

The **third step** deals with the *decryption of the original message number* m by B .

1. Receiver B must use his or her private key (d_B, n_B) and calculate the number $c^{d_B} \bmod n_B$, where he must use the fact that $m = c^{d_B} \bmod n_B$.
2. He then reconstructs A 's full message text from the numeric representation of m .

Why is it true that $m = c^{d_B} \bmod n_B$? By construction, we have $c^{d_B} = m^{e_B d_B} = m^{1+s \cdot \phi}$ in \mathbb{Z}_{n_B} . Consider now the projection

$$\pi : \mathbb{Z} \rightarrow \mathbb{Z}_{p_B} \times \mathbb{Z}_{q_B} : z \mapsto (z \bmod p_B, z \bmod q_B)$$

of rings. The kernel is the principal ideal $(p_B q_B)$ by the prime factorization theory. So we have an injection of rings $\mathbb{Z}_{n_B} \rightarrow \mathbb{Z}_{p_B} \times \mathbb{Z}_{q_B}$, and because both rings have equal cardinality, this is an isomorphism. To show that $m = m^{e_B d_B}$ in \mathbb{Z}_{n_B} is therefore equivalent to show this equation holds in each factor ring \mathbb{Z}_{p_B} and \mathbb{Z}_{q_B} . Now, if $m = 0$ in \mathbb{Z}_{p_B} or in \mathbb{Z}_{q_B} , the claim is immediate, so let us assume that $m \neq 0 \bmod p_B$. Then we have $m = m^{e_B d_B} = m^1 m^{s \phi}$, and it suffices to show that $m^\phi = m^{(p_B-1)(q_B-1)} = 1$ in \mathbb{Z}_{p_B} . This follows readily from the small Fermat theorem 134 $m^{p_B-1} = 1$ in \mathbb{Z}_{p_B} , for $m \neq 0 \bmod p_B$. The same argument holds for q_B , and we are done.

For an in-depth treatment of cryptography, theory and implementation, see [44] and [41].

Appendix

APPENDIX A

Further Reading

Set theory. Keith Devlin's *The Joy of Sets* [20] is a modern treatment of set theory, including non-well-founded sets. As the title indicates, the style is rather relaxed, but the treatment is nevertheless elaborate.

Graph theory. Harris', Hirst's and Mossinghoff's *Combinatorics and Graph Theory* [26] includes recent results and problems that emphasize the cross-fertilizing power of mathematics. Frank Harary's *Graph theory* [25] is a very classical book including many interesting problems and written by one of the leading graph theorists.

Abstract algebra. Among the wealth of books on algebra, Bhattacharya's, Jain's and Nagpaul's *Basic Abstract Algebra* [7] is very readable. A text with a more practical focus is *Discrete Mathematics* [8] by Norman Biggs.

Number theory. The branch of mathematics that deals with natural numbers and their properties, such as primes, is called *number theory*. Despite its elementary basis, number theory quickly becomes very involved, with many unsolved problems. Andrews' *Number Theory* [2] provides a clear introduction to the subject.

Formal logic. Alonzo Church and Willard Van Orman Quine were pioneers of mathematical logic. Church's *Introduction to Mathematical Logic* [15] and Quine's *Mathematical Logic* [37] deal with with classical propositional and predicate logic, and their relation to set theory, with a slight philosophical flavor. A modern text is Dirk van Dalen's *Logic and Structure* [17], which also provides an exposition of natural deduction and intuitionistic logic.

Languages, grammars and automata. Hopcroft's, Motwani's and Ullman's *Introduction to Automata Theory, Languages, and Computation* [28] provides a comprehensive exposition of formal languages and automata. Although the theory of computation has its origin in the work of Alan Turing and Alonzo Church, the modern treatment is much indebted to Martin Davis' *Computability and Unsolvability* [19] from 1958.

Linear algebra. Werner Greub's *Linear Algebra* is a classical text, which is written in a lucid and precise style, also extending to multilinear algebra in a second volume. The two volumes *Basic Linear Algebra* [9] and *Further Linear Algebra* [10] by Blyth and Robertson are more recent texts for first year students, working from concrete examples towards abstract theorems, via tutorial-type exercises. Marcel Berger's *Geometry* [4] is one of the best introductions to linear geometry, including a large number of examples, figures, and applications from different fields, including the arts and classical synthetic geometry.

Computer mathematics. Mathematics in the context of computers can be roughly divided into three domains: algorithms, numerics and computer algebra.

Donald Knuth's series of *The Art of Computer Programming* [31, 32, 33] has the great merit of introducing to computer science a more rigorous mathematical treatment of algorithms such as sorting and searching or machine arithmetic. His books have become the yardstick for all subsequent literature in this branch of computer science. They are, however, not for the faint of heart. A more recent book is Cormen's, Leiserson's, Rivest's and Stein's *Introduction to Algorithms* [16].

Numerics is probably the oldest application of computers, accordingly the literature is extensive. A recent publication is Didier Besset's *Object-Oriented Implementation of Numerical Methods* [6] which exploits object-oriented techniques.

In contrast to numerics, computer algebra focusses on the symbolic solution of many of the problems presented in this book. A recent work is von zur Gathen's and Gerhard's *Modern Computer Algebra* [47]. Computer algebra is the foundation of such symbolic computation systems as Maple or Mathematica.

APPENDIX B

Bibliography

- [1] Aczel, Peter. *Non-Well-Founded Sets*. CSLI LN 14, Stanford, Cal. 1988.
- [2] Andrews, George E. *Number Theory*. Dover, New York 1994.
- [3] Barwise, Jon & Moss, Lawrence. *Vicious Circles*. CSLI Publications, Stanford, Cal. 1996.
- [4] Berger, Marcel. *Geometry I, II*. Springer, Heidelberg et al. 1987.
- [5] Berlekamp, Elwyn. "Bit-Serial Reed-Solomon Encoders." *IEEE Transactions on Information Theory*, IT 28, 1982, pp. 869-874.
- [6] Besset, Didier H. *Object-Oriented Implementation of Numerical Methods*. Morgan Kaufmann, San Francisco et al. 2001.
- [7] Bhattacharya, P. B., Jain, S. K. & Nagpaul S. R. *Basic Abstract Algebra*. Cambridge University Press, Cambridge 1994.
- [8] Biggs, Norman L. *Discrete Mathematics*. Oxford University Press, Oxford 2002.
- [9] Blyth, Thomas S. & Robertson, Edmund F. *Basic Linear Algebra*. Springer, Heidelberg et al. 2002.
- [10] Blyth, Thomas S. & Robertson, Edmund F. *Further Linear Algebra*. Springer, Heidelberg et al. 2001.
- [11] Bornemann, Folkmar. "PRIME Is in P: A Breakthrough for 'Everyman'." *Notices of the AMS*, vol. 50, No. 5, May 2003.
- [12] Bourbaki, Nicolas. *Éléments d'histoire des mathématiques*. Hermann, Paris 1969.

- [13] Cap, Clemens H. *Theoretische Grundlagen der Informatik*. Springer, Heidelberg et al. 1993.
- [14] Chomsky, Noam. "Three models for the description of language." *I.R.E. Transactions on information theory*, volume 2, pp. 113-124, IT, 1956.
- [15] Church, Alonzo. *Introduction to Mathematical Logic*. Princeton University Press, Princeton 1996.
- [16] Cormen, Thomas H., Leiserson, Charles E., Rivest, Ronald L. & Stein, Clifford. *Introduction to Algorithms*. MIT Press, Cambridge 2001.
- [17] van Dalen, Dirk. *Logic and Structure*. Springer, Heidelberg et al. 1994.
- [18] Date, C. J. *An Introduction to Database Systems*. Addison-Wesley, Reading 2003.
- [19] Davis, Martin. *Computability and Unsolvability*. Dover, New York 1985.
- [20] Devlin, Keith J. *The Joy of Sets: Fundamentals of Contemporary Set Theory*. Springer, Heidelberg et al. 1999.
- [21] Goldblatt, Robert. *Topoi—The Categorical Analysis of Logic*. North-Holland, Amsterdam, 1984.
- [22] Greub, Werner. *Linear Algebra*. Springer, Heidelberg et al. 1975.
- [23] Gröbner, Wolfgang. *Matrizenrechnung*. Bibliographisches Institut, Mannheim 1966.
- [24] Groff, James R. & Weinberg, Paul N. *SQL: The Complete Reference*. McGraw-Hill Osborne, New York 2002.
- [25] Harary, Frank. *Graph Theory*. Addison-Wesley, Reading 1972.
- [26] Harris, John M., Hirst, Jeffrey L. & Mossinghoff, Michael J. *Combinatorics and Graph Theory*. Springer, Heidelberg et al. 2000.
- [27] Hilbert, David & Ackermann, Wilhelm. *Grundzüge der theoretischen Logik*. Springer, Heidelberg et al. 1967.
- [28] Hopcroft, John E., Motwani, Rajeev & Ullman, Jeffrey D. *Introduction to Automata Theory, Languages and Computation*. Addison Wesley, Reading 2000.
- [29] Jensen, Kathleen & Wirth, Niklaus. *PASCAL—User Manual and Report ISO Pascal Standard*. Springer, Heidelberg et al. 1974.

- [30] Garey, Michael R. & Johnson, David S. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W H Freeman & Co., New York 1979.
- [31] Knuth, Donald Ervin. *The Art of Computer Programming. Volume I: Fundamental Algorithms*. Addison-Wesley, Reading 1997.
- [32] Knuth, Donald Ervin. *The Art of Computer Programming. Volume II: Seminumerical Algorithms*. Addison-Wesley, Reading 1998.
- [33] Knuth, Donald Ervin. *The Art of Computer Programming. Volume III: Sorting and Searching*. Addison-Wesley, Reading 1998.
- [34] Kruse, Rudolf et al. *Foundations of Fuzzy Systems*. John Wiley & Sons, New York 1996.
- [35] Kuipers, Jack B. *Quaternions and Rotation Sequences*. Princeton University Press, Princeton-Oxford 1998.
- [36] Mac Lane, Saunders & Moerdijk, Ieke. *Sheaves in Geometry and Logic*. Springer, Heidelberg et al. 1992.
- [37] Quine, Willard Van Orman. *Mathematical Logic*. Harvard University Press, Cambridge 1981.
- [38] Rasiowa, Helena & Sikorski, Roman. *The Mathematics of Metamathematics*. Polish Scientific Publishers, Warsaw 1963.
- [39] Reed, Irving S. & Solomon, Gustave. "Polynomial Codes over Certain Finite Fields." *Journal of the Society for Industrial and Applied Mathematics*, Vol. 8, 1960, pp. 300-304.
- [40] Rivest, Ronald L., Shamir, Adi & Adleman, Leonard A. "A method for obtaining digital signatures and public-key cryptosystems." *Communications of the ACM*, Vol. 21, Nr. 2, 1978, pp. 120-126.
- [41] Schneier, Bruce. *Applied Cryptography: Protocols, Algorithms, and Source Code in C*. John Wiley & Sons, New York 1996.
- [42] Steger, Angelika. *Diskrete Strukturen I*. Springer, Heidelberg et al. 2001.
- [43] Stetter, Franz. *Grundbegriffe der Theoretischen Informatik*. Springer, Heidelberg et al. 1988.
- [44] Stinson, Douglas R. *Cryptography: Theory and Practice*. CRC Press, Boca Raton 1995.

-
- [45] Surma, Stanislaw. *Studies in the History of Mathematical Logic*. Polish Academy of Sciences, 1973.
- [46] van der Waerden, Bartel Leendert. *Algebra I*. Springer, Heidelberg et al. 1966.
- [47] von zur Gathen, Joachim & Gerhard, Jürgen. *Modern Computer Algebra*. Cambridge University Press, Cambridge 1999.
- [48] Zadeh, Lotfi A. "Fuzzy Sets." *Information and Control* 8:338-363, 1965.

Index

Symbols

$!$ 35, 193
 $n!$ 166
 (G) 181
 (M_{ij}) 263
 (i) 335
 $(i : M : E)$ 227
 $(i : M : F)$ 245
 (j) 335
 (k) 335
 (x, y) 29
 $*$ 159
 $?^*$ 107, 225
 V^* 311
 b^* 312
 $*b$ 312
 $+$ 73, 84, 94, 160
 a^+ 18
 \rightarrow 193
 $-\frac{a}{b}$ 88
 $-x$ 82
 G/H 168
 a/b 87
 0 74, 171
 0_R 171
 1 74, 171
 $1_{\mathbb{H}}$ 335
 1_R 171
 $G : H$ 169
 $f : A$ 211
 $x : A$ 211
 $::=$ 241
 $<$ 44, 53, 82, 88, 95
 \leftrightarrow 206, 221

\leq 41
 $=$ 16
 \cong 212
 A_n 167
 B_{HLR} 257
 $E(i, j)$ 263
 E_n 263
 $K(T)$ 345
 K_n 123
 $K_{n,m}$ 123
 M^* 164
 $P(\mathbb{H})$ 334
 Q^n 139
 $R(\mathbb{H})$ 334
 R^* 171
 $S(\mathbb{C})$ 160
 $S(\mathbb{H})$ 337
 $S(EX)$ 193
 S^2 143
 S_n 164
 S_{AX} 203
 $T(\phi)$ 218
 T^u 296
 $R[A]$ 174
 $R[X]$ 174
 $[1, n]$ 262
 $[n]$ 118
 $[s]$ 42
 $\#$ 256
 $\#(a)$ 53
 $\&$ 193
 $M_{\bullet j}$ 263
 $M_{i\bullet}$ 263
 \mathbb{C} 102

- \mathbb{C}^* 164
 \mathbb{H} 334
 $\mathbb{M}(R)$ 263
 $\mathbb{M}(f)$ 263
 $\mathbb{M}_{m,n}(R)$ 263
 \mathbb{N} 53
 $\mathbb{P}_3(\mathbb{R})$ 338
 \mathbb{Q} 87
 \mathbb{Q}^* 164
 \mathbb{R} 93
 \mathbb{R}^* 164
 \mathbb{R}_+ 95
 \mathbb{R}_- 95
 \Rightarrow 197
 Σ 211
 \mathbb{Z} 82
 \mathbb{Z}^* 164
 \mathbb{Z}_n 170
 $H \setminus G$ 168
 \cap 20
 \cup 17
 $\bigcup^n x$ 111
 \mathbf{U} 160, 322, 337
 U^\perp 314
 $U \perp W$ 314
 $x \perp y$ 314
 \perp 196
 \cap 19
 $M \cdot N$ 267
 \cdot 73, 85, 94
 χ_f 328
 $\cos(\theta)$ 323
 $\cos(f)$ 322
 \cup 18
 δ_{ij} 263
 $\det(M)$ 272
 $\det_\omega(x_1, x_2, x_3)$ 331
 \emptyset 12
 $r \equiv s$ 170
 η_M 298
 $\frac{a}{b}$ 87
 $\text{GA}(M)$ 297
 $\text{GF}(2^n)$ 347
 $\text{GF}(p^n)$ 347
 $\text{GL}(M)$ 282
 $\text{GL}_n(\mathbb{R})$ 271
 $\text{O}(V)$ 313
 $\text{O}(V, W)$ 312
 $\text{O}_n(\mathbb{R})$ 316
 $\text{SO}(V)$ 316
 $\text{SO}_3(\mathbb{R})$ 327
 AX 203
 $\text{Adj}(G)$ 116, 122
 $\text{Aff}_R(M, N)$ 297
 $\text{Aut}(G)$ 163
 $\text{Aut}(M)$ 162
 $\text{BipDi}(V_1, V_2)$ 110
 $\text{BipDi}(\Gamma, \Delta)$ 133
 $\text{Bip}(V_1, V_2)$ 114
 $\text{Bip}(\Gamma, \Delta)$ 133
 $\text{ComWord}(A)$ 174
 $\text{CompDi}(V)$ 109
 $\text{Comp}(V)$ 114
 $\text{DiDi}(V)$ 109
 $\text{Di}(V)$ 114
 EX 193, 213
 $\text{End}(G)$ 163
 $\text{End}(M)$ 161
 $\text{End}(X)$ 160
 $\text{End}_R(M)$ 281
 $\text{Free}(\phi)$ 215
 FunType 211
 Fun 211
 $\text{Fuzzy}(0, 1)$ 199
 $\text{Gram}(x_i)$ 323
 $\text{Group}(G, H)$ 163
 Id 35
 Id_Γ 121
 $\text{Im}(f)$ 35, 283
 $\text{Im}(x)$ 103
 Int_s^P 337
 Int^{PS} 337
 Int_s^{PS} 337
 Int_s 337
 $\text{Ker}(f)$ 283
 $\text{LSK}(V)$ 133
 LangMachine_A 250
 $\text{Lin}_R(M, N)$ 281
 $\text{Loop}(L)$ 137
 $\text{Loop}(n)$ 137
 $\text{Monoid}(M, N)$ 161
 $\text{Moore}(M)$ 226

- $Parallel(x, y)$ 333
 $Path(\Gamma)$ 139
 $Path_v(\Gamma)$ 139
 $RS(s, k, t)$ 350
 $RelType$ 211
 Rel 211
 $Re(x)$ 103
 $Ring(R, S)$ 172
 $Sequ(S)$ 211
 $Set(a, b)$ 55
 $Stack(i, \mu, E)$ 253
 $Stack(i : \mu : E)$ 253
 $Stream(A)$ 224
 $Sym(X)$ 160, 164
 $Term(P)$ 213
 $Turing(i, tr, s_H)$ 257
 $Turing(i : tr : s_H)$ 258
 $Word(A)$ 139
 $card(a)$ 53
 $codom(f)$ 35
 $deg(f)$ 174
 $deg(v)$ 126
 $deg^+(v)$ 125
 $deg^-(v)$ 125
 $dim(M)$ 285
 $dom(f)$ 35
 $gcd(a, b)$ 183
 $head_\Gamma$ 108
 $lcm(a, b)$ 183
 $left(w)$ 195
 $lev(x)$ 111
 $log_a(x)$ 99
 $ord(G)$ 166
 $pr(a)$ 32, 58
 $right(w)$ 195
 $rk(M)$ 294
 $round(x)$ 153
 $sig(x)$ 168
 $tail_\Gamma$ 108
 tr 257
 $tr(f)$ 328
 $value(x)$ 201
 $var(x)$ 201
 xor 348
 \in 15
 ∞ 91
 $\lambda(f)$ 297
 $\langle S \rangle$ 162
 $\langle X \rangle$ 166
 $R\langle M \rangle$ 173
 $R\langle X_1, \dots, X_n \rangle$ 174
 \mathcal{C} 176
 \mathcal{O} 176
 $\mathcal{V}(A, L)$ 201
 \aleph 217
 \neg 197
 \neq 16
 $x \dagger y$ 183
 $\not\subset$ 16
 \notin 15
 ω 331
 $M^{\oplus D}$ 280
 $\bigoplus_{i=1, \dots, n} M_i$ 281
 $M^{\oplus n}$ 280
 \bar{a}_m 100
 \bar{q} 334
 \bar{x} 105
 $\bar{?}$ 107
 $\bar{\Gamma}$ 113
 π 323
 \prod 87
 $t \xrightarrow{a} h$ 108
 $v \rangle$ 120
 ρ_H 318
 \rightarrow 232
 $E_n \xrightarrow{M} E_m$ 268
 $f : a \rightarrow b$ 35
 \succ 211
 $\tilde{\sim}$ 38
 \sim 42
 $\sin(\theta)$ 323
 $\sin(f)$ 322
 \sqcup 59
 $\sqrt[n]{a}$ 98
 \sqrt{a} 98
 $0 \square 0$ 263
 $0 \square n$ 263
 $m \square 0$ 263
 \subset 16
 \subsetneq 16
 \sum 85

- τ 256
 M^τ 265
 τf 313
 $\tau(f)$ 297
 $\Gamma \times \Delta$ 129
 \times 30
 \times_c 67
 \top 196
 $2 \models s$ 202
 $BA \models s$ 202
 $HA \models s$ 202
 $L \models s$ 202
 $\mathfrak{N} \models \phi[\gamma]$ 219
 $v \models s$ 202
 $\models s$ 202
 $\frac{}{\vdash} 203$
 $\frac{}{\vdash_{AX}} 204$
 $\frac{}{\vdash_{IL}} 204$
 \vee 197
 \wedge 197
 $x_1 \wedge x_2$ 331
 $\{\}$ 15
 $]0, 1[$ 144
 $d(x, y)$ 325
 i 103, 335
 j 335
 k 335
 2V 107
 $(\frac{a}{b})^{-1}$ 88
 2^a 18
 2^f 247
 $B^{(\mathbb{Z})}$ 256
 $R^{(D)}$ 280
 Γ^M 244
 a^b 73
 $a^{\frac{1}{n}}$ 98
 b^a 55
 $f^\infty(I)$ 230
 ${}_R R$ 280
 $|$ 193
 $a|b$ 86
 $x|y$ 183
 $|?$ 107
 $|\Gamma|$ 113
 $|a|$ 53, 82, 96
 $|n|$ 123
 $|x|$ 105
 $||$ 207
 $\|q\|$ 334
 $\|x\|$ 312
 $@$ 9
 $RS(s, k, t)$ 350
 \hat{f} 298
 \hat{x}_i 182
- A**
 \mathcal{A}_ε 252
 A_n 167
 abelian group 163, 281
 ABNF 236
 absolute value
 of integer 82
 of rational number 89
 of real number 96
 absorbing 224
 accepted language 245
 accepting state 245
 acceptor 245
 elementary graph of - 245
 generic - 250
 minimal - 251
 push down - 253
 reduced - 250
 simple - 249
 stack - 253
 acceptors
 equivalent - 245
 isomorphism of - 248
 morphism of - 248
 Aczel, Peter 112
 addition 73
 algorithm for - 152
 additive inverse
 of a rational number 88
 of an integer 82
 additive monoid 160
 $Adj_c(\Gamma)$ 116, 122
 adjacency matrix 116, 122, 269
 adjoint
 linear homomorphisms 313
 matrix 276
 adjunction 197

- Turing - 257
- Adleman, Leonhard 353
- $Aff_R(M, N)$ 297
- affine homomorphism 296
- algebra
 - Boolean - 196, 198
 - Heyting - 197
 - linear - 261
 - logical - 192, 199
 - monoid - 172
- algebraic
 - element 345
 - extension 345
 - geometry 261
- algorithm 76
 - Euclidean - 183
 - for addition 152
- alphabet
 - input - 257
 - predicative - 213
 - propositional - 193
 - tape - 256
- alternating group 167
- alternative
 - left - 138
 - right - 138
- alternative set 45
- AND 5, 197
- angle 323
- ANSI 66, 228
- antisymmetry 26
- application of rules 232
- Archimedean ordering 89
- architecture of concepts 8
- arity 211
- arrow 108
- ASCII 228
- associated
 - digraph 113
 - graph 113
 - matrix of bilinear form 313
 - power automaton 244
- associativity 20
- atomic proposition 211
- attribute of sets 18
- $Aut(G)$ 163
- $Aut(M)$ 162
- automata
 - isomorphism of - 248
 - morphism of - 247
- automaton 139
 - deterministic - 244
 - elementary graph of - 244
- automorphism
 - group 164
 - of groups 163
 - of modules 282
 - of monoids 162
 - of rings 172
- AX 203
- axiom 13, 203
 - of choice 19, 44, 80
 - of empty set 17
 - of equality 17
 - of infinity 18
 - of pairs 18
 - of powersets 18
 - of subsets for attributes 18
 - of union 17
- axiomatic 13
 - set theory 17
 - vector space theory 262
- axiomatics 203
- axis, rotation 327
- B**
- B_{HLR} 257
- b -adic representation 100
- BA 198
- Backus, John 241
- Backus-Naur form 236, 241
- backward substitution 304
- Barwise, Jon 112
- basis
 - of logarithm 99
 - of vector space 289
- Berlekamp, Elwyn 350
- Bernstein-Schröder theorem 40
- bias 150
- bijective 35
- bilinear form 312
 - associated matrix of - 313

- positive definite - 312
- standard - 313
- symmetric - 312
- binary 229
 - normal form 77
 - relation 41
 - representation 100
- binding strength 202
- $Bip(\Gamma, \Delta)$ 133
- $Bip(V_1, V_2)$ 114
- bipartite
 - complete - 110
 - digraph 110
 - graph 114
- $BipDi(\Gamma, \Delta)$ 133
- $BipDi(V_1, V_2)$ 110
- blank 256
- BMP 229
- BNF 236
- Boole, George 196
- Boolean
 - algebra 25, 196, 198
 - operations 197
 - valid 202
- bound variable 215
- bounded set 97
- bracket 193, 213
- C**
- C 176
- \mathbb{C} 102
- \mathbb{C}^* 164
- $card(a)$ 53
- cardinality 38
 - of a set 53
- Cartesian product 30
 - of digraphs 129
 - of functions 38
 - universal property of - 58
- category 249, 261
- Cauchy sequence 91
- Cauchy, Augustin 91
- Cayley-Hamilton equation 278
- chain 123
 - directed - 118
 - length of - 118, 123
- characteristic 344
- characteristic polynomial 328
 - of a matrix 278
- child 138
- Chomsky
 - hierarchy 233
 - normal form 236
 - types 233
- Chomsky, Noam 224
- circular 12
- CL 203
- class, equivalence 42
- classical logic 203
- classically valid 202
- closed under concatenation 225
- closure
 - existential - 217
 - universal - 217
- co-root of digraph 119
- coding theory 344
- $codom(f)$ 35
- codomain 35
- coefficient of a matrix 263
- cofactor 275
- column
 - index 263
 - matrix 263
- comma 213
- common denominator 88
- commutative
 - group 163
 - ring 171
- commutativity 20
- commute 271
- $Comp(V)$ 114
- compact disk 352
- $CompDi(V)$ 109
- complement 284, 317
 - Schur - 309
 - set - 21
- complete
 - bipartite digraph 110
 - digraph 109
 - graph 114
- complete relation 41
- completeness theorem 205

- complex number 102
 - complex numbers
 - product of - 102
 - sum of - 102
 - component
 - connected - 124, 145
 - of digraph morphism 115
 - of graph 122
 - of sentence 196
 - composition
 - of affine homomorphisms 297
 - of automata morphisms 248
 - of digraph morphisms 114
 - of functions 36
 - of graph morphisms 121
 - of graphs 33
 - of isometries 313
 - of linear homomorphisms 281
 - of monoid monomorphisms 161
 - of paths 120
 - of ring homomorphisms 172
 - computer 223
 - ComWord(A)* 174
 - concatenation, closed under 225
 - concept 8
 - architecture 8
 - component 8
 - existence of - 11
 - instance 8
 - name 8
 - configuration 253
 - conjugate matrix 264, 273
 - conjugation
 - in \mathbb{C} 105
 - of quaternions 334
 - conjunction 5
 - formula 215
 - symbol 193
 - conjunctive normal form 207
 - connected
 - component 124, 145
 - digraph 123
 - graph 123
 - constant 210, 211
 - sequence 91
 - construction by induction 56
 - context
 - free grammar 236
 - sensitive grammar 238
 - continuous curve 143
 - contraction, elementary 148
 - contradiction, principle of 4
 - convergent sequence 91, 97
 - coordinate 29
 - coordinates, homogeneous 298
 - coproduct 59
 - of digraphs 130
 - of graphs 130
 - universal property of - 59
 - corollary 13
 - $\cos(f)$ 322
 - $\cos(\theta)$ 323
 - coset 93
 - left - 168
 - right - 168
 - cosine
 - formula 324
 - function 322
 - covering
 - of a set 133
 - skeleton of - 133
 - curve, continuous 143
 - cycle 165
 - Euler - 125
 - Hamilton - 125
 - in a digraph 118
 - in a graph 123
 - cyclic
 - group 170
 - cyclic group 166
 - cyphertext 353
- D**
- $d(x, y)$ 325
 - database, relational 66
 - De Morgan's Laws 206
 - decadic normal form 78
 - decimal 229
 - representation 100
 - decomposition, LUP 307
 - decryption 354
 - defining polynomial 346

- definition 13
 - $deg(f)$ 174
 - $deg(v)$ 126
 - $deg^+(v)$ 125
 - $deg^-(v)$ 125
 - degree 125
 - head - 125
 - tail - 125
 - denominator 87
 - common - 88
 - density of rational numbers in reals 96
 - denumerable set 79
 - Descartes, René 279
 - $\det_{\omega}(x_1, x_2, x_3)$ 331
 - $\det(M)$ 272
 - determinant 272
 - Gram - 323
 - deterministic
 - automaton 244
 - production grammar 230
 - stack acceptor 253
 - Turing machine 257
 - $Di(V)$ 114
 - diagonal
 - graph 32
 - procedure 80
 - $DiDi(V)$ 109
 - difference
 - set - 21
 - symmetric set - 27
 - digraph 108
 - associated - 113
 - bipartite - 110
 - co-root of - 119
 - complete - 109
 - connected - 123
 - coproduct 130
 - discrete - 109
 - Finsler - 112
 - induced - 115
 - isomorphism 115
 - join - 133
 - loop - 137
 - morphism 114
 - component of - 115
 - root of - 119
 - sink of - 119
 - source of - 119
 - digraphs, Cartesian product of 129
 - $dim(M)$ 285
 - dimension of a module 285
 - direct sum 281
 - inner - 284
 - directed
 - chain 118
 - graph 108
 - tree 119
 - directed subgraph 115
 - discrete
 - digraph 109
 - graph 114
 - disjoint 19
 - sum 59
 - disjunction 5
 - formula 215
 - symbol 193
 - disjunctive normal form 207
 - distance 325
 - distributivity 26, 27, 171, 197
 - division of integers 86
 - division theorem, Euclid's 76, 177
 - divisor 183
 - zero - 183
 - $dom(f)$ 35
 - domain 35
 - integral - 183
 - drawing 144
 - drawn graph 144
 - dual
 - graph 109
 - normal form 77
 - space 311
- E**
- E_n 263
 - $E(i, j)$ 263
 - EBNF 236
 - edge 113
 - set 107
 - eigenvalue 329
 - eigenvector 329

- element 15
 - algebraic - 345
 - neutral - 160
 - order of - 167
 - elementary
 - contraction 148
 - graph of acceptor 245
 - graph of automaton 244
 - matrix 263
 - state 252
 - elements, stack 252
 - elimination, Gauss 303
 - empty
 - list 63
 - relation 41
 - word 161
 - encryption 353
 - $End_R(M)$ 281
 - $End(G)$ 163
 - $End(M)$ 161
 - $End(X)$ 160
 - endomorphism
 - of groups 163
 - of modules 281
 - of monoids 161
 - of rings 172
 - epimorphism 35
 - equal sets 16
 - equation 27, 82, 84, 90, 95, 102
 - Cayley-Hamilton - 278
 - Gram - 323
 - homogeneous - 295
 - Lagrange - 169
 - linear - 287, 303
 - polynomial - 102
 - equipollent 38
 - equivalence
 - class 42
 - relation 42
 - generated - 66
 - equivalent
 - acceptors 245
 - formulas 221
 - sentences 206
 - states 250
 - error
 - correction 352
 - detection 351
 - Euclid 182
 - Euclid's division theorem 76, 177
 - Euclidean
 - algorithm 183
 - vector space 312
 - Euler
 - cycle 125
 - formula 145
 - Euler, Leonhard 126
 - evaluation of polynomials 175
 - EX 193, 213
 - excluded third, principle of 4
 - existence of a concept 11
 - existence quantifier 210, 213, 215
 - existential closure 217
 - exponentials, universal property of
 - 60
 - exponentiation 73
 - expression 193, 213
 - extension
 - algebraic - 345
 - of a field 345
- F**
- factorial 166
 - factorization, unique 185
 - false 3, 197
 - falsity formula 214
 - family
 - of elements 62
 - of sets 62
 - Fermat's little theorem 185
 - fiber 60
 - product 66
 - universal property of - 66
 - field
 - extension 345
 - generated - 345
 - Galois - 343
 - prime - 344
 - skew - 177
 - splitting - 346

- final
 - set 58
 - state 227
 - finite
 - group 166
 - set 53
 - finite sequence 63
 - finitely generated 166, 289
 - Finsler digraph 112
 - Finsler, Paul 112
 - first order predicate logic 210
 - floating point representation 101, 150
 - normalized - 150
 - form
 - bilinear - 312
 - linear - 312
 - formula 214
 - conjunction - 215
 - cosine - 324
 - disjunction - 215
 - falsity - 214
 - goniometric - 323
 - implication - 215
 - interpolation - 187
 - negation - 215
 - relational - 214
 - truth - 214
 - formulas
 - equivalent - 221
 - validity for - 219
 - forward substitution 308
 - founded set 46
 - free
 - module 284
 - range 218
 - variable 215
 - $Free(\phi)$ 215
 - Fun 211
 - function 35
 - 0-ary - 210
 - identity - 35
 - polynomial - 175
 - symbol 211
 - functional graph 34
 - functions
 - Cartesian product of - 38
 - composition of - 36
 - functorial 249
 - fundamental theorem of algebra 102, 186
 - $FunType$ 211
 - fuzzy 199
 - fuzzy logic 197
 - $Fuzzy(0, 1)$ 199
- G**
- Gödel, Kurt 192
 - $GA(M)$ 297
 - Galois field 343
 - Galois, Evariste 343
 - Gauss elimination 303
 - Gauss, Carl Friedrich 102
 - $gcd(a, b)$ 183
 - general
 - affine group of a vector space 297
 - grammar 240
 - linear group 271
 - of a module 282
 - generate 289
 - generated
 - equivalence relation 66
 - field extension 345
 - finitely - 166
 - group 166
 - ideal 181
 - language 230, 232
 - vector space 289
 - generic acceptor 250
 - geometry
 - algebraic - 261
 - linear - 262
 - $GF(2^n)$ 347
 - $GF(p^n)$ 347
 - $GL_n(R)$ 271
 - $GL(M)$ 282
 - goniometric
 - formula 323
 - Gram
 - determinant 323
 - equation 323

- Gram*(x_i) 323
 Gram-Schmidt orthonormalization 314
 grammar
 context free - 236
 context sensitive - 238
 general - 240
 left linear - 233
 left regular - 233
 normal - 240
 phrase structure - 232
 production - 230
 reduced - 236
 right linear - 233
 right regular - 233
 separated - 240
 graph (graph theory) 113
 associated - 113
 bipartite - 114
 complete - 114
 components of - 122
 connected - 123
 coproduct 130
 directed - 108
 discrete - 114
 drawn - 144
 dual - 109
 isomorphism 122
 join - 133
 Moore - 140, 226
 morphism 121
 planar - 144
 skeletal - 143
 graph (relation theory) 32
 diagonal 32
 functional 34
 inverse 32
 graphs, composition of 33
 greatest common divisor (gcd) 183
 greatest lower bound (g.l.b.) 26, 197
 Greibach normal form 236
 group 163
 abelian - 163, 281
 alternating - 167
 automorphism 163, 164
 commutative - 163
 cyclic 170
 cyclic - 166
 endomorphism 163
 finite - 166
 general affine - 297
 general linear - 271
 general linear - of a module 282
 generated - 166
 homomorphism 163
 image 168
 kernel 168
 inverse 163
 isomorphism 163
 of invertible elements 164
 of translations 296
 order 166
 orthogonal - 313
 special orthogonal - 316
 symmetric - 164
 trivial - 166
Group(G, H) 163
- H**
 \mathbb{H} 334
 HA 197
 halt state 257
 Hamilton cycle 125
 Hamilton, William Rowan 333
head_r 108
 head degree 125
 head, read/write 256
 hexadecimal 229
 normal form 78
 representation 100
 Heyting
 algebra 197
 valid 202
 Heyting, Arend 197
 hierarchy, Chomsky 233
 homogeneous 266
 coordinates 298
 equation 295
 homogenization 298
 homomorphism
 affine - 296
 linear - 281

- of groups 163
- of monoids 161
- of rings 172
- hyperplane 317
- I**
- Id 35
- Id_{Γ} 121
- ideal 93, 176
 - generated - 181
 - maximal - 179
 - prime - 181
 - principal - 176
- idempotency 27, 226
- identity
 - function 35
 - of languages 232
- IEEE 101, 150
 - standard #754 150
- IL 204
- $Im(f)$ 35, 283
- $Im(x)$ 103
- image 35
 - of group homomorphism 168
 - of linear homomorphism 283
- imaginary
 - complex number 104
 - part 103
 - unit 103
- implication 5, 197
 - formula 215
 - symbol 193
- IMPLIES 5, 197
- indeterminate 173
- index
 - column - 263
 - of subgroup 169
 - row - 263
 - set 62
- indirect proof 13
- induced
 - digraph 115
 - relation 44
- induction, construction by 56
- inequality
 - Schwarz - 324
 - triangle - 87, 89, 96, 105, 324, 325
- Inf 151
- infinite word 224
- infix 224
- initial
 - language 230
 - set 58
 - state 141, 227, 243
- injective 35
- input
 - alphabet 257
 - letters 252
 - place 111
- instance
 - name 8
 - value 8
- Int_s^P 337
- Int_s^{PS} 337
- Int_s^{PS} 337
- Int_s 337
- integer 82
 - absolute value of - 82
 - additive inverse of - 82
 - negative - 82
 - positive - 82
 - prime - 86
- integers
 - division of - 86
 - product of - 85
 - sum of - 84
- integral domain 183
- interior of unit interval 144
- interpolation 188
 - formula 187
- intersection 19, 20
- intuitionistic logic 204
- inverse graph 32
- invertible 163
- involution 27
- irreducible polynomial 181
- ISO 229
- isometry 312
- isomorphism 35
 - of acceptors 248
 - of automata 248
 - of digraphs 115

- of graphs 122
 - of groups 163
 - of modules 281
 - of monoids 161
 - of rings 172
- J**
- Java 112
 - Jensen, Kathleen 242
 - join 197
 - of digraphs 133
 - of graphs 133
 - join (SQL) 69
- K**
- K_n 123
 - $K_{n,m}$ 123
 - $K(T)$ 345
 - $\text{Ker}(f)$ 283
 - kernel
 - of group homomorphism 168
 - of linear homomorphism 283
 - key
 - private - 353
 - public - 353
 - Kleene
 - cross 242
 - operator 225
 - Knuth, Donald 125
 - Kronecker delta 263
 - Kuratowski, Kazimierz 147
- L**
- L-system 230
 - L -valid 202
 - labeled transition system 109
 - Lagrange equation 169
 - LangMachine_A 250
 - language
 - accepted
 - by sequential machine 227
 - accepted - 245
 - generated - 230, 232
 - initial - 230
 - of type 0 241
 - of type 1 238
 - of type 2 236
 - of type 3 234
 - predicative - 215
 - propositional - 193
 - recursively enumerable - 241
 - semi-decidable - 258
 - sequential machine of - 250
 - stream - 225
 - terminal - 230
 - word - 225
 - lazy
 - path 118
 - walk 123
 - $\text{lcm}(a, b)$ 183
 - leading coefficient 174
 - leaf 119
 - least common multiple (lcm) 183
 - least upper bound (l.u.b.) 26, 197
 - $\text{left}(w)$ 195
 - Leibniz, Gottfried Wilhelm 196
 - lemma 13
 - length
 - of chain 118, 123
 - of path 118
 - of sequence 63
 - of walk 123
 - letters, input 252
 - $\text{lev}(x)$ 111
 - level 111
 - lexicographic ordering 64
 - lim 91
 - $\text{Lin}_R(M, N)$ 281
 - Lindenmayer, Aristid 230
 - linear
 - algebra 261
 - equation 287, 303
 - form 312
 - geometry 262
 - homomorphism 281
 - adjoint - 313
 - image of - 283
 - kernel of - 283
 - rank of - 293
 - ordering 43
 - part 297

- linearly
 - dependent 288
 - independent 288
- Liouville, Joseph 344
- list 63
 - empty - 63
- $\log_a(x)$ 99
- logarithm 99
 - basis of - 99
- logic
 - classical - 203
 - fuzzy - 197
 - intuitionistic - 204
 - propositional - 4
- logical
 - algebra 192,199
 - connective symbol 193
- loop 111
 - digraph 137
 - in a digraph 118
 - in a graph 123
- $Loop(L)$ 137
- $Loop(n)$ 137
- lower triangular matrix 307
- $LSK(V)$ 133
- LTS 109
- LUP decomposition 307

- M**
- \mathfrak{M} 217
- $\mathbb{M}_{m,n}(R)$ 263
- $\mathbb{M}(f)$ 263
- $\mathbb{M}(R)$ 263
- M^* 164
- machine 223
 - sequential - 226
 - Turing - 257
- mantissa 150
- map, transition 257
- matrix 262
 - adjacency - 116,122,269
 - adjoint - 276
 - characteristic polynomial of - 278
 - coefficient 263
 - column - 263
 - conjugate - 264,273
 - elementary - 263
 - lower triangular - 307
 - product 267
 - rank of - 294
 - regular - 271
 - row - 263
 - scaled - 266
 - square - 271
 - sum 266
 - symmetric - 265
 - tabular representation of - 263
 - theory 261
 - unit - 263
 - upper triangular - 277
 - Vandermonde - 351
- maximal ideal 179
- meet 197
- minimal 43
 - acceptor 251
- minor 275
- mod n 170
- model 219
- module 279
 - dimension of - 285
 - free - 284
- modus ponens 13,203
- monoid 159
 - additive - 160
 - algebra 172
 - universal property of - 174
 - automorphism 162
 - endomorphism 161
 - homomorphism 161
 - isomorphism 161
 - multiplicative - 160
 - stream - 224
 - word - 139,160
- $Monoid(M,N)$ 161
- monomial 173
- monomorphism 35
- monotony
 - additive - 74
 - multiplicative - 74
- Moore graph 140,226
- $Moore(M)$ 226

- morphism
 - of acceptors 248
 - of automata 247
 - of digraphs 114
 - of graphs 121
- Moss, Lawrence 112
- multiplication 73
 - scalar - 279
- multiplicative
 - inverse of a rational number 88
 - monoid 160
- music storage 352
- Myhill-Nerode, theorem of - 251
- N**
- \mathbb{N} 53
- n -ary predicate 210
- n -ary relation 65
- n -ary tree 138
- n -bit word 139
- n -cube 139
- n -th root 98
- NaN 151
- NaNQ 151
- NaNS 151
- natural number 50
- Naur, Peter 241
- negation 4, 197
 - formula 215
 - symbol 193
- negative
 - integer 82
 - real number 95
- neutral element 27, 160
 - additive - 74
 - exponential - 74
 - multiplicative - 74
- Newton interpolation formula 187
- nondeterministic
 - automaton 244
 - production grammar 230
 - stack acceptor 253
 - Turing machine 257
- nondeterministic polynomial complexity 259
- nonterminal symbol 232
- norm 312
 - of a complex number 105
 - quaternion - 334
- normal
 - grammar 240
 - subgroup 170
- normal form
 - adic - 77
 - ary - 77
 - binary - 77
 - Chomsky - 236
 - conjunctive - 207
 - decadic - 78
 - disjunctive - 207
 - dual - 77
 - Greibach - 236
 - hexadecimal - 78
 - prenex - 221
 - Skolem - 221
- normalized floating point representation 150
- NOT 4, 197
- not-a-number 150, 151
- NP 259
- 0-ary function 210
- 0-ary function 210
- number
 - complex - 102
 - natural - 50
 - rational - 87
 - real - 93
- numerator 87
- O**
- \emptyset 176
- $O_n(\mathbb{R})$ 316
- $O(V)$ 313
- $O(V, W)$ 312
- octal 229
- one-to-one 35
- onto 35
- operator, Kleene 225
- OR 5, 197
- $ord(G)$ 166
- order
 - of element 167

- of group 166
- ordered pair 29
- ordering
 - Archimedean - 89
 - lexicographic - 64
 - linear - 43
 - partial - 43
- ordinal set 47
- orientation 320, 331
- orthogonal
 - group 313
 - vector spaces 314
 - vectors 314
- orthonormal 314
- orthonormalization, Gram-Schmidt 314
- output place 111
- overflow
 - exponent - 153
 - fractional - 153
- P**
- P 259
- $\mathbb{P}_3(\mathbb{R})$ 338
- $P(\mathbb{H})$ 334
- pair, ordered 29
- Parallel*(x, y) 333
- parallelepiped, volume of 333
- parallelogram, surface of 333
- parent 138
- partial ordering 43
- partition 42
- path 118
 - lazy - 118
 - length of - 118
- Path*(Γ) 139
- Path* _{v} (Γ) 139
- paths, composition of 120
- pattern 232
- Paul, Finsler 222
- Peano axioms 51
- permutation 164
- Petri net 111
- phrase structure grammar 232
- $\pi = 3.1415926$ 323
- place 111
 - input - 111
 - output - 111
- planar graph 144
- Poisson, Siméon-Denis 343
- polyhedron 144
- polynomial
 - characteristic - 328
 - commutative - 174
 - complexity (P) 259
 - constant - 174
 - cubic - 174
 - defining - 346
 - equation 102
 - evaluation 175
 - function 175
 - irreducible - 181
 - linear - 174
 - non-commutative - 173
 - quadratic - 174
 - root of - 186
- pop 252
- positive
 - integer 82
 - real number 95
- positive definite bilinear form 312
- Post, Emil 205
- power
 - automaton, associated 244
 - graph 244, 245
 - rational - 98
- powerset 18
- $pr(a)$ 32, 58
- predicate 209
 - n -ary - 210
 - logic, first order 210
- predicative
 - alphabet 213
 - language 215
 - sentence 216
- prefix 224
- prenex normal form 221
- prime 181
 - ideal 181
 - integer 86
- prime field 344
- primitive root of unity 347

- principal
 - ideal 176
 - ideal ring 176
 - principle
 - of contradiction 4
 - of excluded third 4
 - private key 353
 - product
 - Cartesian - 30
 - fiber - 66
 - of complex numbers 102
 - of integers 85
 - of matrixes 267
 - of natural numbers 74
 - of rational numbers 88
 - of real numbers 94
 - relation 65
 - scalar - 313
 - production grammar 230
 - deterministic - 230
 - nondeterministic - 230
 - projection 32, 58
 - stereographic - 144
 - projective space 338
 - proof 13
 - by induction 52
 - indirect - 13
 - sequence 203
 - proposition 3, 13
 - atomic - 211
 - propositional
 - alphabet 193
 - language 193
 - logic 4
 - variable 193
 - public key 353
 - pullback 66
 - pumping lemma 235, 237
 - pure
 - part 334
 - quaternion 334
 - set 12
 - push 252
 - push down acceptor 253
- Q**
- \mathbb{Q} 87
 - \mathbb{Q}^* 164
 - \mathbb{Q}^n 139
 - quantifier
 - existence - 210, 213, 215
 - universal - 210, 213, 215
 - quaternion 334
 - conjugation 334
 - norm 334
 - pure - 334
 - real - 334
 - quotient module 283
- R**
- \mathbb{R} 93
 - R^* 171
 - \mathbb{R}^* 164
 - \mathbb{R}_+ 95
 - \mathbb{R}_- 95
 - $R(\mathbb{H})$ 334
 - R -algebra 173
 - range, free 218
 - rank
 - of a linear homomorphism 293
 - of a matrix 294
 - rational number 87
 - absolute value of - 89
 - rational numbers
 - product of - 88
 - sum of - 88
 - rational power 98
 - $Re(x)$ 103
 - reachable 118, 123
 - read/write head 256
 - real
 - part 103, 334
 - quaternion 334
 - real number 93
 - absolute value of - 96
 - negative - 95
 - positive - 95
 - real numbers
 - product of - 94
 - sum of - 94
 - recursion theorem 56

- recursively enumerable language
 - 241
 - reduced
 - acceptor 250
 - grammar 236
 - Reed, Irving S. 349
 - Reed-Solomon 349
 - reflection 319
 - reflexivity 26
 - regular matrix 271
 - Rel* 211
 - relation 210
 - n -ary - 65
 - 0-ary - 210
 - binary - 41
 - complete - 41
 - empty - 41
 - equivalence - 42
 - induced - 44
 - product - 65
 - symbol 211
 - relational
 - database 66
 - relational formula 214
 - RelType* 211
 - representation
 - binary - 100
 - decimal - 100
 - floating point - 101, 150
 - hexadecimal - 100
 - restriction 38
 - rewriting system 231
 - right(w)* 195
 - ring 171
 - commutative - 171
 - endomorphism 172
 - homomorphism 172
 - isomorphism 172
 - principal ideal - 176
 - Ring(R, S)* 172
 - Rivest, Ronald 353
 - rk(M)* 294
 - root
 - of a polynomial 186
 - of digraph 119
 - rotation 316
 - axis 327
 - round(x)* 153
 - rounding 153
 - row
 - index 263
 - matrix 263
 - rules, application of 232
 - Russell, Bertrand 11
- S**
- $S_{AX}(EX)$ 203
 - S_n 164
 - $S(EX)$ 193
 - $S(\mathbb{H})$ 337
 - S^2 143
 - scalar
 - multiplication 266, 279
 - product 313
 - scaled matrix 266
 - Schur complement 309
 - Schwarz inequality 324
 - scope of variable 216
 - SELECT (SQL) 71
 - selection 8
 - semantics 192, 217
 - semi-decidable language 258
 - sentence 193
 - component of - 196
 - predicative - 216
 - sentences
 - equivalent - 206
 - separated grammar 240
 - Sequ(S)* 211
 - sequence 63
 - Cauchy - 91
 - constant - 91
 - convergent - 91, 97
 - finite - 63
 - length of a - 63
 - proof - 203
 - state - 244
 - terminal - 203
 - zero - 91
 - sequential machine 140, 226
 - of a language 250

- set
 - alternative - 45
 - attribute 18
 - bounded - 97
 - cardinality 53
 - complement 21
 - denumerable - 79
 - difference 21
 - symmetric - 27
 - final - 58
 - finite - 53
 - founded - 46
 - initial - 58
 - ordinal - 47
 - pure - 12
 - singleton - 39
 - state - 256
 - theory, axiomatic 17
 - totally finite - 111
 - transitive - 45
- $Set(a, b)$ 55
- sets
 - equal - 16
 - family of - 62
- Shamir, Adi 353
- Sheffer stroke operator 207
- $sig(x)$ 168
- Σ -structure 217
- signature 211
- signification 192, 218
- simple acceptor 249
- $\sin(f)$ 322
- $\sin(\theta)$ 323
- sine function 322
- single precision representation 150
- singleton set 39
- sink of digraph 119
- skeletal graph 143
- skeleton of a covering 133
- skew field 177
 - commutative - 177
- skew-symmetric 331
- Skolem normal form 221
- $SO_3(\mathbb{R})$ 327
- $SO(V)$ 316
- Solomon, Gustave 349
- solution of linear equation 287
- sorite 13
- sort 211
 - of term 213
- soundness theorem 205
- source of digraph 119
- space
 - dual - 311
 - projective - 338
 - state - 109, 140, 226
 - vector - 279
- spanning 127
 - tree 127
- special orthogonal group 316
- sphere 143
 - unit - 337
- splitting field 346
- square matrix 271
- stack acceptor 253
 - deterministic - 253
 - nondeterministic - 253
- stack elements 252
- $Stack(i, \mu, E)$ 253
- $Stack(i : \mu : E)$ 253
- standard bilinear form 313
- state
 - accepting - 245
 - elementary - 252
 - final - 227
 - halt - 257
 - initial - 141, 227, 243
 - sequence 244
 - of Turing machine 258
 - set 256
 - space 109, 140, 226
 - terminal - 245
 - transition function 253
- states 243
 - equivalent - 250
- Steinitz exchange theorem 290
- stereographic projection 144
- stream 224
 - language 225
 - monoid 224
- $Stream(A)$ 224
- strength of binding 202

- structural transport 44
 - subacceptor 249
 - subdigraph 115
 - subgraph
 - generated by a vertex 120
 - subgroup 163
 - normal - 170
 - submodule 279
 - submonoid 162
 - generated - 162
 - subobject classifier 61
 - subring 172
 - subset 16
 - substitution
 - backward - 304
 - forward - 308
 - successor 18
 - suffix 224
 - sum
 - of complex numbers 102
 - of integers 84
 - of matrixes 266
 - of natural numbers 74
 - of rational numbers 88
 - of real numbers 94
 - supremum 97
 - surface of parallelogram 333
 - surjective 35
 - $Sym(X)$ 160, 164
 - symbol
 - conjunction - 193
 - disjunction - 193
 - function - 211
 - implication - 193
 - negation - 193
 - nonterminal - 232
 - relation - 211
 - terminal - 232
 - symmetric
 - bilinear form 312
 - group 164
 - of rank n 164
 - matrix 265
 - set difference 27
 - syntactics 211
 - syntax 191
 - diagram 242
- T**
- tabular representation of a matrix 263
 - $tail_{\Gamma}$ 108
 - tail degree 125
 - tape 256
 - alphabet 256
 - tautology 202
 - term 213
 - sort of - 213
 - $Term(P)$ 213
 - terminal
 - language 230
 - sequence 203
 - state 245
 - symbol 232
 - theorem 13, 203
 - completeness - 205
 - of Myhill-Nerode 251
 - of recursion 56
 - soundness - 205
 - Steinitz exchange - 290
 - topos theory 218
 - totally finite set 111
 - tr 257
 - $tr(f)$ 328
 - trace 328
 - transition 111
 - function, state 253
 - map 257
 - transitive
 - set 45
 - transitivity 26
 - translation 296
 - part 297
 - translations
 - group of - 296
 - transposition 165, 265
 - tree 123
 - n -ary - 138
 - directed - 119
 - triangle inequality 87, 89, 96, 105, 324, 325
 - trivial group 166

- true 3, 197
 - truth
 - formula 214
 - table 5
 - Turing
 - adjunction 257
 - machine 257
 - deterministic - 257
 - nondeterministic - 257
 - state sequence of - 258
 - $Turing(i, tr, s_H)$ 257
 - $Turing(i : tr : s_H)$ 258
 - Turing, Alan 256
 - turtle graphics 230
 - type 211
 - type 0 language 241
 - type 1 language 238
 - type 2 language 236
 - type 3 language 234
- U**
- U 160, 322, 337
 - underflow, exponent 153
 - Unicode 229
 - unique factorization 185
 - unit 263
 - circle 160, 337
 - imaginary - 103
 - interval, interior of 144
 - universal
 - closure 217
 - quantifier 210, 213, 215
 - universal property
 - of Cartesian product 58
 - of coproduct 59
 - of exponentials 60
 - of fiber product 66
 - of monoid algebra 174
 - of word monoid 162
 - upper bound 97
 - upper triangular matrix 277
- V**
- $\mathcal{V}(A, L)$ 201
 - valid 219
 - Boolean - 202
 - classically - 202
 - Heyting - 202
 - validity for formulas 219
 - valuation 201
 - value 197
 - $value(x)$ 201
 - Vandermonde matrix 351
 - $var(x)$ 201
 - variable 211
 - bound - 215
 - free - 215
 - propositional - 193
 - scope of - 216
 - vector 279
 - vector space 279
 - basis of - 289
 - Euclidean - 312
 - generated - 289
 - Venn diagram 26
 - vertex 108, 113
 - subgraph generated by - 120
 - volume of parallelepiped 333
- W**
- walk 123
 - lazy - 123
 - well defined 84
 - well-ordering 43
 - Wirth, Niklaus 242
 - Wittgenstein, Ludwig 3
 - word
 - n -bit - 139
 - empty - 161
 - infinite - 224
 - language 225
 - word monoid 139, 160
 - universal property of - 162
 - $Word(A)$ 139
- X**
- xor 348
- Z**
- \mathbb{Z} 82
 - \mathbb{Z}^* 164
 - \mathbb{Z}_n 170
 - Zermelo, Ernst 44
 - zero
 - divisor 183
 - sequence 91

Universitext

- Aguilar, M.; Gitler, S.; Prieto, C.:* Algebraic Topology from a Homotopical Viewpoint
- Aksoy, A.; Khamsi, M. A.:* Methods in Fixed Point Theory
- Aletras, D.; Padberg M. W.:* Linear Optimization and Extensions
- Andersson, M.:* Topics in Complex Analysis
- Aoki, M.:* State Space Modeling of Time Series
- Arnold, V. I.:* Lectures on Partial Differential Equations
- Arnold, V. I.; Cooke, R.:* Ordinary Differential Equations
- Audin, M.:* Geometry
- Aupetit, B.:* A Primer on Spectral Theory
- Bachem, A.; Kern, W.:* Linear Programming Duality
- Bachmann, G.; Narici, L.; Beckenstein, E.:* Fourier and Wavelet Analysis
- Badescu, L.:* Algebraic Surfaces
- Balakrishnan, R.; Ranganathan, K.:* A Textbook of Graph Theory
- Balser, W.:* Formal Power Series and Linear Systems of Meromorphic Ordinary Differential Equations
- Bapat, R.B.:* Linear Algebra and Linear Models
- Benedetti, R.; Petronio, C.:* Lectures on Hyperbolic Geometry
- Benth, F. E.:* Option Theory with Stochastic Analysis
- Berberian, S. K.:* Fundamentals of Real Analysis
- Berger, M.:* Geometry I, and II
- Bliedtner, J.; Hansen, W.:* Potential Theory
- Blowey, J. F.; Coleman, J. P.; Craig, A. W. (Eds.):* Theory and Numerics of Differential Equations
- Blyth, T. S.:* Lattices and Ordered Algebraic Structures
- Börger, E.; Grädel, E.; Gurevich, Y.:* The Classical Decision Problem
- Böttcher, A.; Silbermann, B.:* Introduction to Large Truncated Toeplitz Matrices
- Boltyanski, V.; Martini, H.; Soltan, P. S.:* Excursions into Combinatorial Geometry
- Boltyanskii, V. G.; Efremovich, V. A.:* Intuitive Combinatorial Topology
- Bonnans, J. F.; Gilbert, J. C.; Lemaréchal, C.; Sagastizábal, C. A.:* Numerical Optimization
- Booss, B.; Bleecker, D. D.:* Topology and Analysis
- Borkar, V. S.:* Probability Theory
- Brunt B. van:* The Calculus of Variations
- Carleson, L.; Gamelin, T. W.:* Complex Dynamics
- Cecil, T. E.:* Lie Sphere Geometry: With Applications of Submanifolds
- Chae, S. B.:* Lebesgue Integration
- Chandrasekharan, K.:* Classical Fourier Transform
- Charlap, L. S.:* Bieberbach Groups and Flat Manifolds
- Chern, S.:* Complex Manifolds without Potential Theory
- Chorin, A. J.; Marsden, J. E.:* Mathematical Introduction to Fluid Mechanics
- Cohn, H.:* A Classical Invitation to Algebraic Numbers and Class Fields
- Curtis, M. L.:* Abstract Linear Algebra
- Curtis, M. L.:* Matrix Groups
- Cyganowski, S.; Kloeden, P.; Ombach, J.:* From Elementary Probability to Stochastic Differential Equations with MAPLE
- Da Prato, G.:* An Introduction to Infinite Dimensional Analysis
- Dalen, D. van:* Logic and Structure
- Das, A.:* The Special Theory of Relativity: A Mathematical Exposition

- Debarre, O.:* Higher-Dimensional Algebraic Geometry
- Deitmar, A.:* A First Course in Harmonic Analysis
- Demazure, M.:* Bifurcations and Catastrophes
- Devlin, K. J.:* Fundamentals of Contemporary Set Theory
- DiBenedetto, E.:* Degenerate Parabolic Equations
- Diener, F.; Diener, M. (Eds.):* Nonstandard Analysis in Practice
- Dimca, A.:* Sheaves in Topology
- Dimca, A.:* Singularities and Topology of Hypersurfaces
- DoCarmo, M. P.:* Differential Forms and Applications
- Duistermaat, J. J.; Kolk, J. A. C.:* Lie Groups
- Dumortier.:* Qualitative Theory of Planar Differential Systems
- Edwards, R. E.:* A Formal Background to Higher Mathematics Ia, and Ib
- Edwards, R. E.:* A Formal Background to Higher Mathematics IIa, and IIb
- Emery, M.:* Stochastic Calculus in Manifolds
- Endler, O.:* Valuation Theory
- Engel, K.-J.; Nagel, R.:* A Short Course on Operator Semigroups
- Erez, B.:* Galois Modules in Arithmetic
- Everest, G.; Ward, T.:* Heights of Polynomials and Entropy in Algebraic Dynamics
- Farenick, D. R.:* Algebras of Linear Transformations
- Foulds, L. R.:* Graph Theory Applications
- Franke, J.; Hrdle, W.; Hafner, C. M.:* Statistics of Financial Markets: An Introduction
- Frauenthal, J. C.:* Mathematical Modeling in Epidemiology
- Friedman, R.:* Algebraic Surfaces and Holomorphic Vector Bundles
- Fuks, D. B.; Rokhlin, V. A.:* Beginner's Course in Topology
- Fuhrmann, P. A.:* A Polynomial Approach to Linear Algebra
- Gallot, S.; Hulin, D.; Lafontaine, J.:* Riemannian Geometry
- Gardiner, C. F.:* A First Course in Group Theory
- Gårding, L.; Tambour, T.:* Algebra for Computer Science
- Godbillon, C.:* Dynamical Systems on Surfaces
- Godement, R.:* Analysis I, and II
- Goldblatt, R.:* Orthogonality and Spacetime Geometry
- Gouvêa, F. Q.:* p -Adic Numbers
- Gross, M. et al.:* Calabi-Yau Manifolds and Related Geometries
- Gustafson, K. E.; Rao, D. K. M.:* Numerical Range. The Field of Values of Linear Operators and Matrices
- Gustafson, S. J.; Sigal, I. M.:* Mathematical Concepts of Quantum Mechanics
- Hahn, A. J.:* Quadratic Algebras, Clifford Algebras, and Arithmetic Witt Groups
- Hájek, P.; Havránek, T.:* Mechanizing Hypothesis Formation
- Heinonen, J.:* Lectures on Analysis on Metric Spaces
- Hlawka, E.; Schoißengeier, J.; Taschner, R.:* Geometric and Analytic Number Theory
- Holmgren, R. A.:* A First Course in Discrete Dynamical Systems
- Howe, R., Tan, E. Ch.:* Non-Abelian Harmonic Analysis
- Howes, N. R.:* Modern Analysis and Topology
- Hsieh, P.-F.; Sibuya, Y. (Eds.):* Basic Theory of Ordinary Differential Equations
- Humi, M., Miller, W.:* Second Course in Ordinary Differential Equations for Scientists and Engineers
- Hurwitz, A.; Kritikos, N.:* Lectures on Number Theory
- Huybrechts, D.:* Complex Geometry: An Introduction
- Isaev, A.:* Introduction to Mathematical Methods in Bioinformatics

- Istas, J.:* Mathematical Modeling for the Life Sciences
- Iversen, B.:* Cohomology of Sheaves
- Jacob, J.; Protter, P.:* Probability Essentials
- Jennings, G. A.:* Modern Geometry with Applications
- Jones, A.; Morris, S. A.; Pearson, K. R.:* Abstract Algebra and Famous Impossibilities
- Jost, J.:* Compact Riemann Surfaces
- Jost, J.:* Dynamical Systems. Examples of Complex Behaviour
- Jost, J.:* Postmodern Analysis
- Jost, J.:* Riemannian Geometry and Geometric Analysis
- Kac, V.; Cheung, P.:* Quantum Calculus
- Kannan, R.; Krueger, C. K.:* Advanced Analysis on the Real Line
- Kelly, P.; Matthews, G.:* The Non-Euclidean Hyperbolic Plane
- Kempf, G.:* Complex Abelian Varieties and Theta Functions
- Kitchens, B. P.:* Symbolic Dynamics
- Kloeden, P.; Ombach, J.; Cyganowski, S.:* From Elementary Probability to Stochastic Differential Equations with MAPLE
- Kloeden, P. E.; Platen, E.; Schurz, H.:* Numerical Solution of SDE Through Computer Experiments
- Kostrikin, A. I.:* Introduction to Algebra
- Krasnoselskii, M. A.; Pokrovskii, A. V.:* Systems with Hysteresis
- Kurzweil, H.; Stellmacher, B.:* The Theory of Finite Groups. An Introduction
- Lang, S.:* Introduction to Differentiable Manifolds
- Luecking, D. H., Rubel, L. A.:* Complex Analysis. A Functional Analysis Approach
- Ma, Zhi-Ming; Roeckner, M.:* Introduction to the Theory of (non-symmetric) Dirichlet Forms
- Mac Lane, S.; Moerdijk, I.:* Sheaves in Geometry and Logic
- Marcus, D. A.:* Number Fields
- Martinez, A.:* An Introduction to Semiclassical and Microlocal Analysis
- Matoušek, J.:* Using the Borsuk-Ulam Theorem
- Matsuki, K.:* Introduction to the Mori Program
- Mazzola, G.; Milmeister G.; Weissman J.:* Comprehensive Mathematics for Computer Scientists 1
- Mazzola, G.; Milmeister G.; Weissman J.:* Comprehensive Mathematics for Computer Scientists 2
- McCarthy, P. J.:* Introduction to Arithmetical Functions
- McCrimmon, K.:* A Taste of Jordan Algebras
- Meyer, R. M.:* Essential Mathematics for Applied Field
- Meyer-Nieberg, P.:* Banach Lattices
- Mikosch, T.:* Non-Life Insurance Mathematics
- Mines, R.; Richman, F.; Ruitenburg, W.:* A Course in Constructive Algebra
- Moise, E. E.:* Introductory Problem Courses in Analysis and Topology
- Montesinos-Amilibia, J. M.:* Classical Tessellations and Three Manifolds
- Morris, P.:* Introduction to Game Theory
- Nikulin, V. V.; Shafarevich, I. R.:* Geometries and Groups
- Oden, J. J.; Reddy, J. N.:* Variational Methods in Theoretical Mechanics
- Øksendal, B.:* Stochastic Differential Equations
- Øksendal, B.; Sulem, A.:* Applied Stochastic Control of Jump Diffusions
- Poizat, B.:* A Course in Model Theory
- Polster, B.:* A Geometrical Picture Book
- Porter, J. R.; Woods, R. G.:* Extensions and Absolutes of Hausdorff Spaces
- Radjavi, H.; Rosenthal, P.:* Simultaneous Triangularization
- Ramsay, A.; Richtmeyer, R. D.:* Introduction to Hyperbolic Geometry
- Rautenberg, W.:* A Concise Introduction to Mathematical Logic

- Rees, E. G.:* Notes on Geometry
- Reisel, R. B.:* Elementary Theory of Metric Spaces
- Rey, W. J. J.:* Introduction to Robust and Quasi-Robust Statistical Methods
- Ribenboim, P.:* Classical Theory of Algebraic Numbers
- Rickart, C. E.:* Natural Function Algebras
- Rotman, J. J.:* Galois Theory
- Rubel, L. A.:* Entire and Meromorphic Functions
- Ruiz-Tolosa, J. R.; Castillo E.:* From Vectors to Tensors
- Runde, V.:* A Taste of Topology
- Rybakowski, K. P.:* The Homotopy Index and Partial Differential Equations
- Sagan, H.:* Space-Filling Curves
- Samelson, H.:* Notes on Lie Algebras
- Sauvigny, F.:* Partial Differential Equations I
- Sauvigny, F.:* Partial Differential Equations II
- Schiff, J. L.:* Normal Families
- Sengupta, J. K.:* Optimal Decisions under Uncertainty
- Séroul, R.:* Programming for Mathematicians
- Seydel, R.:* Tools for Computational Finance
- Shafarevich, I. R.:* Discourses on Algebra
- Shapiro, J. H.:* Composition Operators and Classical Function Theory
- Simonnet, M.:* Measures and Probabilities
- Smith, K. E.; Kahanpää, L.; Kekäläinen, P.;*
Traves, W.: An Invitation to Algebraic Geometry
- Smith, K. T.:* Power Series from a Computational Point of View
- Smoryński, C.:* Logical Number Theory I. An Introduction
- Stichtenoth, H.:* Algebraic Function Fields and Codes
- Stillwell, J.:* Geometry of Surfaces
- Stroock, D. W.:* An Introduction to the Theory of Large Deviations
- Sunder, V. S.:* An Invitation to von Neumann Algebras
- Tamme, G.:* Introduction to Étale Cohomology
- Tondeur, P.:* Foliations on Riemannian Manifolds
- Toth, G.:* Finite Möbius Groups, Minimal Immersions of Spheres, and Moduli
- Verhulst, F.:* Nonlinear Differential Equations and Dynamical Systems
- Wong, M. W.:* Weyl Transforms
- Xambó-Descamps, S.:* Block Error-Correcting Codes
- Zaenen, A. C.:* Continuity, Integration and Fourier Theory
- Zhang, F.:* Matrix Theory
- Zong, C.:* Sphere Packings
- Zong, C.:* Strange Phenomena in Convex and Discrete Geometry
- Zorich, V. A.:* Mathematical Analysis I
- Zorich, V. A.:* Mathematical Analysis II