INTRODUCTION TO MODEL THEORY

ALGEBRA, LOGIC AND APPLICATIONS
A series edited by
R. Göbel
Universität Gesamthochschule, Essen, Germany
A. Macintyre
University of Edinburgh, UK

Volume 1
Linear Algebra and Geometry
A.I. Kostrikin and Yu.I. Manin

Volume 2
Model Theoretic Algebra: With Particular Emphasis on Fields, Rings, Modules
Christian U. Jensen and Helmut Lenzing

Volume 3
Foundations of Module and Ring Theory: A Handbook for Study and Research
Robert Wisbauer

Volume 4
Linear Representations of Partially Ordered Sets and Vector Space Categories
Daniel Simson

Volume 5
Semantics of Programming Languages and Model Theory
M. Droste and Y. Gurevich

Volume 6
Exercises in Algebra: A Collection of Exercises in Algebra, Linear Algebra and Geometry
Edited by A.I. Kostrikin

Volume 7
Bilinear Algebra: An Introduction to the Algebraic Theory of Quadratic Forms
Kazimierz Szymiczek

Volume 8
Multilinear Algebra
Russell Merris

Volume 9
Advances in Algebra and Model Theory
Edited by Manfred Droste and Rüdiger Göbel

Please see the back of this book for other titles in the Algebra, Logic and Applications series.

INTRODUCTION TO MODEL THEORY

Philipp Rothmaler
Institut für Logik
Christian-Albrechts-Universität
Kiel, Germany

Published in 2000 by
Taylor & Francis Group
270 Madison Avenue
New York, NY 10016

Published in Great Britain by
Taylor & Francis Group
2 Park Square
Milton Park, Abingdon
Oxon OX14 4RN

Originally published in German in 1995 as Einfuhrung In Die Modelltheorie by Spektrum Akademischer Verlag, Heidelberg.
Printed in the United States of America on acid-free paper
10 9 8 7 6 5 4 3

International Standard Book Number-10: 90-5699-313-5 (Softcover)
International Standard Book Number-13: 978-5699-313-9 (Softcover)

Library of Congress Cataloging-in-Publication Data

informa
Taylor & Francis Group
is the Academic Division of Informa plc.

Visit the Taylor & Francis Web site at
http://www.taylorandfrancis.com

Contents

Preface

The purpose of this book is to give an insight into the model theory of first-order logic and its potential for algebraic applications. Acquaintance with logic—though useful—is not required. Only an undergraduate preparation in algebra (groups, rings, fields, and vector spaces) is assumed on the part of the reader.

The book grew out of a first course in model theory taught at the Christian-Albrechts-Universität in Kiel, Germany in the fall semester of 1992–93. The manuscript for the original German version (published by Spektrum Akademischer Verlag in 1995) was produced in collaboration with one of the students and Word Perfectionists, Frank Reitmaier. Translating it into English and LaTeX I enjoyed the assistance of my student Karsten Guhl. It is my great pleasure to thank them both for their enormous efforts. A number of students, colleagues, and other friends have contributed to both versions of this book with valuable comments and corrections. I am especially indebted to my students Matthias Clasen and Thomas Rohwer. Further thanks are due to Joel Agee, Andreas Baudisch, Paul Moritz Cohn, Ulrich Felgner, Wilfrid Hodges, Rahim Moosa, Arnold Oberschelp, Anand Pillay, Klaus Potthoff, Hans Röpke, Thomas Wilke and (last, but not least) Martin Ziegler. Preparing the final version of the translation I enjoyed the very pleasant hospitality of the Universitá degli Studi di Trento, Italy, and I am most grateful to Stefano Baratella for this. Finally I would like to thank the editors, Rüdiger Göbel and Angus Macintyre, for inviting me to publish a translation into this series.

The English edition differs from the German original in three ways: naturally, corrections and revisions have been made and the bibliography has been updated, second, more exercises, as well as hints and solutions to a selection of them, have been added, and finally, the dimension theory for strongly minimal theories scattered over text and exercises in the original has been made the topic of a separate (penultimate) chapter, which contains, as a particular case, Steinitz' dimension theory for algebraically closed fields.

Attention!
Kozma Prutkov (1803-1863)
Fruits of Reflection (Aphorism 42)

Introduction

Model theory—like mathematical logic in general—is a relatively young field. It deals with the relationship between sets of formal sentences and their models, hence with the relationship between the syntax and semantics of a formal language. We restrict ourselves to the model theory of *first-order* logic, since this has particularly nice features. In their full generality these depend on the axiom of choice, which we apply without much ado, mostly in the equivalent form of Zorn's lemma.

The development of model theory went along with its applications to other mathematical disciplines, mainly to *algebra*, which we concentrate on. Hints to other fields of application and to the model theory of other logics can be found in the references listed at the end of the text. We try to make immediate use of introduced concepts and methods. Therefore the text does not fall into an abstract (theoretical) and a concrete (applied) part, but rather proceeds by letting these two alternate. Thus we present some non-trivial applications of the finiteness theorem (in Part II) before we turn to the central model-theoretic concepts and methods (in Part III).

The table of contents, fairly detailed as it is, may serve as a guide for the beginner when entering new and possibly unfamiliar territory. Its order reflects to a large extent the history of the subject. There are only two exceptions, Cantor's fundamental order- and set-theoretic instruments (§§7.3-6); and the proof of the finiteness (compactness) theorem in §4.3, which is not derived from Gödel's completeness theorem for first-order logic, as is usually the case, but rather uses the later developed ultraproducts. This allows us to completely avoid the necessary calculus of formal derivations and to argue only semantically.[1] Apart from this we get the following chronological picture.

[1] I thank Thomas Wilke for suggesting we banish formal derivations from such a course in model theory.

Part I deals with the basics that were developed in the 20s and 30s. These are the concept of structure (Ch. 1), the corresponding first-order languages (Ch. 2), as well as their connection via Tarski's concept of truth (Ch. 3).

Part II contains the fundamental finiteness theorem (Ch. 4) and first model-theoretic results of the 30s and 40s, which were obtained mainly using this theorem. It is interesting to note, however, that Malcev's deeper group-theoretic results (Ch. 6) (which for linguistic and political reasons were taken note of only later) already anticipated A. Robinson's diagrams and the method of interpretation that was developed in the 50s by Tarski, Mostowski and (R. M) Robinson for decidability theory and that today plays a central role in stability theory. Further, it is explained why the finiteness theorem is also called compactness theorem (§5.7).

Part III is dedicated to the machinery developed in the 50s and to the corresponding results about the relationship between models. One could say that this part deals with the category of models of a theory whose morphisms are the elementary maps—however we will make no further reference to category theory. We then present two important algebraic applications of this material. One is Robinson's proof of Hilbert's Nullstellensatz as a consequence of the model completeness of the theory of algebraically closed fields. The other is a theorem of Chevalley about projections of constructible sets as a consequence of quantifier elimination of the same theory, proved by Tarski and Robinson (§9.5, cf. also §9.6). General model-theoretic applications are e.g. various preservation theorems (§§6.1 and 10.2–3).

Part IV starts with work from the late 50s and early 60s, when the concept of type enriched and refined the theory considerably. This part leads more or less directly to, and culminates in Vaught's theorem saying that a countable and complete theory cannot have exactly two countable models. The theorem in itself may seem quite exotic, its proof, however, is intertwined with fundamental model-theoretic methods like saturated and atomic models, omitting types etc. (As often in mathematics, the proof is more consequential than the result.) §11.3 contains, as part of the exercises, a new definition of stability due to I. Herzog and the author.

Part V deals with two rather different applications of the material presented before. They can be studied independently.

The first of these, Ch. 14,[2] concerns the models of the so-called strongly minimal theories—a natural generalization of that of algebraically closed fields, inasmuch as it admits a similar dimension theory. In fact, Steinitz'

[2]This chapter is new. In the German original, some of the material was scattered over text and exercises.

well-known dimension theory for such fields is obtained as a special case of the more general model-theoretic theory here. The most recent result in the book is F. Wagner's confirmation of a conjecture of Podewski about strong minimality of certain fields, which constitutes part of the exercises of this chapter.

The last chapter, Ch. 15, is devoted to the models of a concrete theory, the complete theory of the abelian group of the integers. In passing, some stability-theoretic notions are introduced and their relevance is pointed out by referring to the recent literature.

Exercises are scattered about the text, mostly at the end of the sections. There is an appendix giving hints to some of them, and another one containing selected solutions.

A separate appendix contains the bibliography and some hints to the literature.

While a few historical comments are to be found in the text, for a more complete account the interested reader is referred to the illuminating comments at the ends of chapters in Wilfrid Hodges' *Model Theory.*

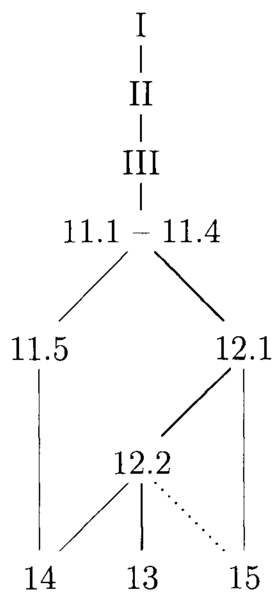
The logical structure of the text is largely linear. Only the last three chapters are more or less independent.[3] Items marked by * can be skipped. One could also skip the field-theoretic applications. However, this would be against the author's intentions, for Steinitz' theory of (algebraically closed) fields can be seen as a paradigm for the model-theoretic classification (or stability) theory of Morley and Shelah, which constitutes one of the central parts of contemporary model theory and which the interested reader may want to go on studying next.

> —After all, nothing's ever said
> that wasn't said before, in olden times.
> Surely, therefore, you will forgive and understand
> if we, though modern, follow where the ancients trod.
> Listen well then, be nice and quiet, pay close attention,
> and it will all be clear...
>
> P. Terentius Afer (195–159 B.C.)
> *Eunuchus* (Prologue)[4]

[3]Note, §11.5 is used only in Ch. 14, and one can pass to Ch. 15 directly after §12.1, cf. the chart on p. xiv.

[4]Translation from the German translation of the Latin original by Joel Agee.

Interdependence chart



Notation

This is a list of notation and terminology that will be assumed to be known.

$X \subseteq Y, Y \supseteq X$	X is a subset of Y, Y contains X
$X \subset Y, Y \supset X$	X is a proper subset of Y, Y contains X properly
$\mathfrak{P}(Y)$	power set of Y, i. e. $\{X : X \subseteq Y\}$
$X \Subset Y$	X is a finite subset of Y
$X \cup Y$	union of X and Y
$X \sqcup Y$	disjoint union of X and Y, i. e., formally, $X \sqcup Y = (X \times \{0\}) \cup (Y \times \{1\})$
$X \cap Y$	intersection of X and Y
$X \setminus Y$	difference of the sets X and Y
$X \times Y$	cartesian product of X and Y, i. e. $\{(x, y) : x \in X \text{ and } y \in Y\}$
$\|X\|$	power (or cardinality) of X
$\emptyset$	empty set
${}^{Y}X$ or X^{Y}	set of all maps from Y to X
$(a_i : i \in I)$	family indexed by I, i. e., formally, a function from ${}^{I}\{a_i : i \in I\}$ with $i \mapsto a_i$ (in case I is well-ordered, this is called a sequence)
$(a_0, \dots, a_{n-1})$	n-tuple, i. e. a sequence of length n
$\bar{a}$	tuple, i. e. a finite sequence
X^n	set of all n-tuples with entries from X
$\mathrm{l}(\bar{a})$	length of the tuple $\bar{a}$, i. e. n if $\bar{a} \in X^n$
$\bar{a}\hat{}\bar{b}$	concatenation of the tuples $\bar{a}$ and $\bar{b}$, i. e. $(a_0, \dots, a_{n-1}, b_0, \dots, b_{m-1})$, if $\bar{a} = (a_0, \dots, a_{n-1})$ and $\bar{b} = (b_0, \dots, b_{m-1})$
$f : x \mapsto y$	f maps x to y
$x \mapsto y$	x is mapped to y
$\operatorname{dom} f$	domain of the map f
$f \upharpoonright X$	restriction of f to $X \subseteq \operatorname{dom} f$
$f[X]$	image of $X \subseteq \operatorname{dom} f$ under the map f, i. e. $\{f(x) : x \in X\}$
$f[\bar{a}]$	$(f(a_0), \dots, f(a_{n-1}))$, where $\bar{a} = (a_0, \dots, a_{n-1})$ and $a_i \in \operatorname{dom} f$ $(i < n)$
fg	composite of the maps f and g, i. e. $(fg)(x) = f(g(x))$ ('first g, then f')
id_X	identical map on a set X

id	identical map if the domain is clear
$\mathbb{N}$, $\mathbb{Z}$, $\mathbb{Q}$, $\mathbb{R}$, $\mathbb{C}$	sets of all natural numbers (including 0), all integers, all rational numbers, all real numbers, and all complex numbers, respectively
$\mathbb{P}$	set of all prime numbers
$\mathcal{R}[x_1, \ldots, x_n]$	ring of all polynomials in the indeterminates $x_1, \ldots, x_n$ and coefficients from $\mathcal{R}$
$\mathcal{H} \lhd \mathcal{G}$	$\mathcal{H}$ is a normal subgroup of the group $\mathcal{G}$
iff	if and only if

Part I

Basics

In this first part we fix our terminology concerning structures and see how languages can be used to talk about them. Anybody acquainted with the beginnings of mathematical logic will only have to leaf through this part in order to confirm terminology and notation.

Chapter 1

Structures

Let us first look at some examples. By specifying certain neutral elements and operations, we may view the set $\mathbb{Z}$ of integers as an additive group, as a multiplicative semigroup, or as a ring. In the three cases given these would be $(0; +)$, $(1; \cdot)$, or $(0, 1; +, \cdot)$, respectively. We could also add the inverse operation $-$ or the ordering relation $<$. It is this choice of *signature*, as we say, that determines which structure on $\mathbb{Z}$ we are dealing with.

1.1 Signatures

A **signature** σ is a quadruple $(\mathbf{C}, \mathbf{F}, \mathbf{R}, \sigma')$ consisting of a set $\mathbf{C}$ of **constant symbols**, a set $\mathbf{F}$ of **function symbols**, a set $\mathbf{R}$ of **relation symbols**, and a **signature function** $\sigma' : \mathbf{F} \cup \mathbf{R} \to \mathbb{N} \setminus \{0\}$, where we assume the sets $\mathbf{C}$, $\mathbf{F}$, and $\mathbf{R}$ to be pairwise disjoint. The elements of $\mathbf{C} \cup \mathbf{F} \cup \mathbf{R}$ are also known as the **non-logical symbols**. For simplicity we often identify a signature with its set of non-logical symbols. Accordingly, by the **cardinality** or **power** of σ, in symbols $|\sigma|$, we simply mean the cardinality of the set $\mathbf{C} \cup \mathbf{F} \cup \mathbf{R}$. Unary relation symbols are also called **predicates**. A signature with $\mathbf{C} = \emptyset$, $\mathbf{F} = \emptyset$, or $\mathbf{R} = \emptyset$ is said to be **without constants**, **without functions**, or **without relations**, respectively. A signature that has neither constants nor functions, is called **(purely) relational**.

The signature function assigns to each symbol from $\mathbf{F} \cup \mathbf{R}$ its arity, i. e. $f \in \mathbf{F}$ is a $\sigma'(f)$-ary function symbol and $R \in \mathbf{R}$ is a $\sigma'(R)$-ary relation symbol. Since a constant symbol $c \in \mathbf{C}$ may be viewed as a (constant) function with unique value c, we may think of c as a 0-place function. Accordingly the signature function can be extended to all non-logical symbols by setting $\sigma'(c) = 0$ for all $c \in \mathbf{C}$.

When explicitly writing down a signature we separate the sets $\mathbf{C}$, $\mathbf{F}$, and $\mathbf{R}$ by a semicolon. E. g. writing $\sigma = (0, 1; +, \cdot; <)$ and $\sigma'(+) = \sigma'(\cdot) = \sigma'(<) = 2$ fixes a signature with the constant symbols 0 and 1, two binary function symbols $+$ and $\cdot$, and a binary relation symbol $<$. If the arities are understood (e. g. by some suggestive choice of symbols as above), we omit the signature function altogether.

1.2 Structures

Let $\sigma = (\mathbf{C}, \mathbf{F}, \mathbf{R}, \sigma')$ be a signature.

Given a set M, we may give σ a 'meaning' in M by choosing elements from M and functions and relations on M which are to be denoted by the non-logical symbols from σ. Every such choice determines a so-called *structure* of that signature on M, which assigns to each constant, function, or relation symbol a well-defined interpretation in M meeting the constraints given by the signature function.

A **σ-structure** $\mathcal{M}$ is a quadruple $(M, \mathbf{C}^{\mathcal{M}}, \mathbf{F}^{\mathcal{M}}, \mathbf{R}^{\mathcal{M}})$ consisting of an arbitrary set M (the **underlying set** or **universe** of $\mathcal{M}$), families $\mathbf{C}^{\mathcal{M}} = (c^{\mathcal{M}} : c \in \mathbf{C})$, $\mathbf{F}^{\mathcal{M}} = (f^{\mathcal{M}} : f \in \mathbf{F})$, and $\mathbf{R}^{\mathcal{M}} = (R^{\mathcal{M}} : R \in \mathbf{R})$, where $c^{\mathcal{M}} \in M$ for all $c \in \mathbf{C}$, $f^{\mathcal{M}}$ is a $\sigma'(f)$-ary function from M to M for all $f \in \mathbf{F}$, and $R^{\mathcal{M}}$ is a $\sigma'(R)$-ary relation on M (hence a subset of $M^{\sigma'(R)}$) for all $R \in \mathbf{R}$. The **cardinality** or **power** $|\mathcal{M}|$ of a σ-structure $\mathcal{M}$ is simply the cardinality $|M|$ of the underlying set M. For P, a non-logical symbol from σ, the object $P^{\mathcal{M}}$ is said to be the **interpretation** of P in $\mathcal{M}$. Given a signature σ without constants, $\emptyset_\sigma$ is used to denote the empty σ-structure.

Note that empty σ-structures exist precisely when $\mathbf{C}$ is empty, i. e. when σ is without constants.

The notation for structures follows the guidelines fixed for signatures in the previous subsection. Further, if $R \in \mathbf{R}$ is an n-ary relation symbol and $(a_0, \ldots, a_{n-1}) \in R^{\mathcal{M}}$, we also write $R^{\mathcal{M}}(a_0, \ldots, a_{n-1})$ or even $\mathcal{M} \models R(a_0, \ldots, a_{n-1})$, this referring to the satisfaction relation to be defined below. In case of an n-place function f, we write $f^{\mathcal{M}}(a_0, \ldots, a_{n-1}) = b$ or $\mathcal{M} \models f(a_0, \ldots, a_{n-1}) = b$, accordingly. As usual, tuples are denoted e. g. by $\bar{a}$; writing $f(\bar{a})$ or $R(\bar{a})$ then tacitly assumes f and R to have the arity corresponding to the length of $\bar{a}$.

Example. Consider $\sigma = (0, 1; +, \cdot; <)$, where $\sigma'(+) = \sigma'(\cdot) = \sigma'(<) = 2$. Every ordered ring $\mathcal{R}$ (see §5.5 below) can be regarded as a σ-structure $(R; 0^{\mathcal{R}}, 1^{\mathcal{R}}; +^{\mathcal{R}}, \cdot^{\mathcal{R}}; <^{\mathcal{R}})$, where the non-logical symbols from σ are interpreted by the corresponding constants, functions, and relations, respectively.

Then, for example, $\mathcal{R} \models 0 < 1$ or, equivalently, $0^{\mathcal{R}} <^{\mathcal{R}} 1^{\mathcal{R}}$.

Exercise 1.2.1. Find a signature appropriate for the description of vector spaces over a given field $\mathcal{K}$.

1.3 Homomorphisms

In order to compare two σ-structures $\mathcal{M}$ and $\mathcal{N}$ we need maps between them that preserve certain features of these structures.

A **homomorphism** from $\mathcal{M}$ to $\mathcal{N}$ is a map $h : M \to N$ satisfying

(i) $h(c^{\mathcal{M}}) = c^{\mathcal{N}}$, for all $c \in \mathbf{C}$,

(ii) $f^{\mathcal{N}}(h(a_0), \dots, h(a_{n-1})) = h(f^{\mathcal{M}}(a_0, \dots, a_{n-1}))$, for all $n \in \mathrm{N}$, all $a_0, \dots, a_{n-1} \in M$, and all $f \in \mathbf{F}$ with $\sigma'(f) = n$,

(iii) $R^{\mathcal{M}}(a_0, \dots, a_{n-1}) \Rightarrow R^{\mathcal{N}}(h(a_0), \dots, h(a_{n-1}))$, for all $n \in \mathrm{N}$, all $a_0, \dots, a_{n-1} \in M$, and all $R \in \mathbf{R}$ with $\sigma'(R) = n$.

We write $h : \mathcal{M} \to \mathcal{N}$ for short.

A homomorphism $h : \mathcal{M} \to \mathcal{N}$ is said to be **strong** if for all $n \in \mathrm{N}$, all $R \in \mathbf{R}$ with $\sigma'(R) = n$, and all $b_0, \dots, b_{n-1} \in h[M]$ with $R^{\mathcal{N}}(b_0, \dots, b_{n-1})$, there are $a_0, \dots, a_{n-1} \in M$ such that $R^{\mathcal{M}}(a_0, \dots, a_{n-1})$ and $h(a_i) = b_i$ for all $i < n$.

The structure $\mathcal{N}$ is a (**strong**) **homomorphic image** of the structure $\mathcal{M}$ if there is a (strong) homomorphism from $\mathcal{M}$ *onto* $\mathcal{N}$.

The difference between the notations $g : M \to N$ and $h : \mathcal{M} \to \mathcal{N}$ thus is that g is merely a map between the universes (as sets), while h is a homomorphism of *structures*.

Using the notation $h[(a_0, \dots, a_{n-1})] = (h(a_0), \dots, h(a_{n-1}))$ from the list before Chapter 1, we can rewrite (ii) and (iii) more concisely as follows.

(ii') $f^{\mathcal{N}}(h[\bar{a}]) = h(f^{\mathcal{M}}(\bar{a}))$ for all $f \in \mathbf{F}$ and $\bar{a} \in M^{\sigma'(f)}$,

(iii') $R^{\mathcal{M}}(\bar{a}) \Rightarrow R^{\mathcal{N}}(h[\bar{a}])$ for all $R \in \mathbf{R}$ and $\bar{a} \in M^{\sigma'(R)}$.

A **monomorphism** from $\mathcal{M}$ to $\mathcal{N}$ is, by definition, an injective and strong homomorphism, i. e. an injective homomorphism that satisfies also the inverse implication of (iii). We use $h : \mathcal{M} \hookrightarrow \mathcal{N}$ to denote that h is a monomorphism from $\mathcal{M}$ to $\mathcal{N}$.

An **isomorphism** from $\mathcal{M}$ to $\mathcal{N}$ (or between $\mathcal{M}$ and $\mathcal{N}$) is, by definition, a surjective monomorphism from $\mathcal{M}$ onto $\mathcal{N}$. We write $h : \mathcal{M} \cong \mathcal{N}$ if h is such an isomorphism. If h fixes a set $X \subseteq M \cap N$ pointwise (i. e. h extends the map id_X), we write $h : \mathcal{M} \cong_X \mathcal{N}$ and speak of an **isomorphism over** X or just an X-**isomorphism**. The structures $\mathcal{M}$ and $\mathcal{N}$ are said to be **isomorphic** (over X), and $\mathcal{N}$ is called an **isomorphic image** of $\mathcal{M}$ (over

X), if there is an isomorphism between $\mathcal{M}$ and $\mathcal{N}$ (over X). The notation $\mathcal{M} \cong \mathcal{N}$ (or $\mathcal{M} \cong_X \mathcal{N}$) is used to indicate this. An **isomorphism type** of σ-structures is an equivalence class of σ-structures modulo the equivalence relation $\cong$.

Remark. That $\cong$ is indeed an equivalence relation is easy to see.

Isomorphic structures have of course the same cardinality.

Warning. A bijective homomorphism need not be an isomorphism!

Example. Consider two sets M and N of the same power and a signature σ that consists of a unique predicate symbol R only. Turn these sets into σ-structures $\mathcal{M}$ and $\mathcal{N}$ by setting $R^{\mathcal{M}} = \emptyset$ and $R^{\mathcal{N}} = N$. Any bijection $h : M \to N$ is clearly a homomorphism but not a strong one, hence it cannot be an isomorphism.

Remark. In case $\mathbf{R} = \emptyset$, any bijective homomorphism is already an isomorphism, since then every homomorphism is strong.

Lemma 1.3.1. *Let $\mathcal{M}$ and $\mathcal{N}$ be σ-structures and let $h : M \to N$ be a bijection.*

Then $h : \mathcal{M} \to \mathcal{N}$ and $h^{-1} : \mathcal{N} \to \mathcal{M}$ hold if and only if $h : \mathcal{M} \cong \mathcal{N}$ and $h^{-1} : \mathcal{N} \cong \mathcal{M}$ hold.

Proof. $\Longrightarrow$. The homomorphic condition (iii) for h^{-1} implies that h is an isomorphism.

$\Longleftarrow$. Any isomorphism is a homomorphism. $\square$

Remark. Let $\mathcal{M}$ and $\mathcal{N}$ be σ-structures.

Then $h : \mathcal{M} \to \mathcal{N}$ is an isomorphism if and only if there is $h' : \mathcal{N} \to \mathcal{M}$ such that $hh' = \mathrm{id}_N$ and $h'h = \mathrm{id}_M$.

An **endomorphism** of $\mathcal{M}$ is, by definition, a homomorphism from $\mathcal{M}$ to itself. The endomorphisms of $\mathcal{M}$ form—with respect to composition of maps—a semigroup (see §5.2 below) whose identity element is id_M. An **automorphism** of $\mathcal{M}$ is an isomorphism of $\mathcal{M}$ onto itself. Given $X \subseteq M$, an X-isomorphism from $\mathcal{M}$ onto itself is called **automorphism over** X or just X-**automorphism** of $\mathcal{M}$. The automorphisms of $\mathcal{M}$ form a group with respect to composition of maps, the so-called **automorphism group** of M, denoted by $\mathrm{Aut}\,\mathcal{M}$. Given $X \subseteq M$, $\mathrm{Aut}_X\mathcal{M}$ denotes the subgroup formed by the X-automorphisms. A monomorphism $h : \mathcal{M} \hookrightarrow \mathcal{N}$ is also called **isomorphic embedding** of $\mathcal{M}$ in $\mathcal{N}$ (cf. the end of §1.4 below). In case $h \restriction X = \mathrm{id}_X$ for some $X \subseteq M$, we speak of **embeddings over** X or X-**embeddings**, denoted $h : \mathcal{M} \hookrightarrow_X \mathcal{N}$. $\mathcal{M}$ is said to be (**isomorphically**) **embeddable** (over X) in $\mathcal{N}$, if there is an embedding (over X) of $\mathcal{M}$ in $\mathcal{N}$. We then write $\mathcal{M} \hookrightarrow \mathcal{N}$ (resp. $\mathcal{M} \hookrightarrow_X \mathcal{N}$).

Thus every automorphism is a surjective and injective endomorphism, and again, the converse need not be true (check!).

As in group theory or other algebraic theories known to the reader, every homomorphic image of a structure $\mathcal{M}$ is (isomorphic to) a factor structure of $\mathcal{M}$ modulo a *congruence relation* on $\mathcal{M}$. In order to have a 1-1 correspondence between the isomorphism types of homomorphic images and factor structures, one needs to restrict oneself to strong homomorphisms (which, of course, is irrelevant if there are no relation symbols around). See §2.4 of Malcev's *Algebraic Systems* for this.

Exercise 1.3.1. Given $X \subseteq M$, let $\mathrm{Aut}_{\{X\}}\mathcal{M}$ be the set $\{h \in \mathrm{Aut}\,\mathcal{M} : h[X] = X\}$. Show that $\mathrm{Aut}_X\mathcal{M}$ is a normal subgroup of $\mathrm{Aut}_{\{X\}}\mathcal{M}$. What happens if, instead of $h[X] = X$, we require only $h[X] \subseteq X$?

Exercise 1.3.2. Find a structure with a bijective endomorphism that is not an automorphism.

Exercise 1.3.3. Find an infinite structure $\mathcal{M}$ with a trivial automorphism group, i. e. $\mathrm{Aut}\,\mathcal{M} = \{\mathrm{id}_M\}$.

1.4 Restrictions onto subsets

Often one wants to think of a subset of a given structure as a structure of the same signature, in its own right. This is possible only if the subset meets the following requirement.

Let $\mathcal{M} = (M, \mathbf{C}^{\mathcal{M}}, \mathbf{F}^{\mathcal{M}}, \mathbf{R}^{\mathcal{M}})$ be a σ-structure and N a subset of M. We say N is **closed** in $\mathcal{M}$ **under functions**[1] (from σ) if $\mathbf{C}^{\mathcal{M}} \subseteq N$ and $\mathbf{F}^{\mathcal{M}}[N] \subseteq N$ (i. e., $c^{\mathcal{M}} \in \mathcal{N}$, for all $c \in \mathbf{C}$, and $f^{\mathcal{M}}(\bar{a}) \in N$, for all $f \in \mathbf{F}$ and all $\sigma'(f)$-tuples $\bar{a}$ in N). If this is the case, we can turn N into a σ-structure $\mathcal{N}$ by setting $c^{\mathcal{N}} = c^{\mathcal{M}}$, $f^{\mathcal{N}}(\bar{a}) = f^{\mathcal{M}}(\bar{a})$, and $R^{\mathcal{N}}(\bar{b})$ precisely if $R^{\mathcal{M}}(\bar{b})$, for all $c \in \mathbf{C}$, all $f \in \mathbf{F}$, and all $\sigma'(f)$-tuples $\bar{a}$ from N, as well as all $R \in \mathbf{R}$ and all $\sigma'(R)$-tuples $\bar{b}$ from N.

Remark. If the signature of $\mathcal{M}$ is purely relational, i. e. $\mathbf{C} = \mathbf{F} = \emptyset$, *every* subset $N \subseteq M$ can be made such a uniquely determined structure $\mathcal{N}$. If the signature of $\mathcal{M}$ is without constants, i. e. $\mathbf{C} = \emptyset$, the empty set can be made such a structure (isomorphic to $\emptyset_\sigma$ from §1.2).

[1] This terminology is justified by the aforementioned fact that constants can be regarded as nullary functions.

For this relationship between $\mathcal{M}$ and $\mathcal{N}$ we introduce the following terminology.

$\mathcal{N}$ is called a **restriction** (or **relativisation**) of $\mathcal{M}$ onto N, denoted $\mathcal{M} \restriction N$. The structure $\mathcal{N}$ is said to be a **substructure** of $\mathcal{M}$, if $N \subseteq M$ and $\mathcal{N}$ is the restriction of $\mathcal{M}$ onto N. We write $\mathcal{N} \subseteq \mathcal{M}$ for short.

$\mathcal{M}$ is said to be a **superstructure** or an **extension** of $\mathcal{N}$, if $\mathcal{N}$ is a substructure of $\mathcal{M}$. We write $\mathcal{M} \supseteq \mathcal{N}$ then.

Remark. The image $h[M]$ of any homomorphism $h : \mathcal{M} \to \mathcal{N}$ of σ-structures is closed under functions in $\mathcal{N}$ and thus the universe of a canonical substructure of $\mathcal{N}$. Hence such an image is itself a σ-structure and as such a homomorphic image of $\mathcal{M}$. This structure on $h[M]$ is denoted by $h(\mathcal{M})$. The homomorphism h is a monomorphism if and only if it is an isomorphism between $\mathcal{M}$ and $h(\mathcal{M})$.

Exercise 1.4.1. Describe the difference between substructures of $\mathbb{Z}$ according to whether $\mathbb{Z}$ is considered in the signature $(0; +)$ or in the signature $(0; +, -)$.

1.5 Reductions onto subsignatures

We obtain a different kind of canonical structure, if, instead of shrinking the *universe*, we make the *signature* smaller and just 'forget' the interpretation of the symbols left out from the signature. As in the preceding section, we consider also the reverse process—which is no longer canonical where we have to assign interpretations to symbols added to the signature.

Let σ_0 and σ_1 be signatures such that $\sigma_0 \subseteq \sigma_1$ (i. e. with $\mathbf{C}_0 \subseteq \mathbf{C}_1$, $\mathbf{F}_0 \subseteq \mathbf{F}_1$, $\mathbf{R}_0 \subseteq \mathbf{R}_1$, and $\sigma_0' = \sigma_1' \restriction \operatorname{dom} \sigma_0'$). Then every σ_1-structure $\mathcal{M}$ can be canonically regarded as a σ_0-structure. More precisely, the **reduct** of $\mathcal{M}$ onto σ_0 (or the σ_0-**reduct** of $\mathcal{M}$) is the structure $\mathcal{M} \restriction \sigma_0 =_{\mathrm{def}} (M, \mathbf{C}_0^{\mathcal{M}}, \mathbf{F}_0^{\mathcal{M}}, \mathbf{R}_0^{\mathcal{M}})$ (where $\mathbf{C}_0^{\mathcal{M}} = \{c^{\mathcal{M}} : c \in \mathbf{C}_0\}$ and similarly for $\mathbf{F}_0^{\mathcal{M}}$ and $\mathbf{R}_0^{\mathcal{M}}$). Given a σ_0-structure $\mathcal{N}$ and a σ_1-structure $\mathcal{M}$, the structure $\mathcal{M}$ is said to be an **expansion** of the structure $\mathcal{N}$ to σ_1, if $\mathcal{N}$ is the reduct of $\mathcal{M}$ onto σ_0.

The relationship between these concepts is illustrated by the following.

restriction $\longleftrightarrow$ extension	—	changing universes
reduct$\longleftrightarrow$ expansion	—	changing signatures

Exercise 1.5.1. Given a signature σ, find a signature $\sigma_1 \supseteq \sigma$ such that all σ-structures $\mathcal{M}$ and $\mathcal{N}$ with $\mathcal{N} \subseteq \mathcal{M}$ have expansions $\mathcal{M}'$ and $\mathcal{N}'$ to σ_1 such that $\mathcal{N}' \subseteq \mathcal{M}'$ and $\operatorname{Aut} \mathcal{M}' = \operatorname{Aut}_{\{N\}} \mathcal{M}$ (cf. notation from Exercise 1.3.1).

1.6 Products

Here we see how one can, in a canonical way, patch various structures together, provided all of them have the same signature.

Let I be a nonempty set and $(\mathcal{M}_i : i \in I)$ a family of σ-structures. We define the **direct** (or **cartesian**) **product** $\mathcal{M} = \prod_{i\in I} \mathcal{M}_i$ of this family to be the following σ-structure $\mathcal{M}$. The universe M of $\mathcal{M}$ is the set of all maps $a : I \to \bigcup_{i\in I} M_i$ that have the property that $a(i) \in M_i$ for all $i \in I$. We often write a as $(a(i) : i \in I)$.

Given $c \in \mathbf{C}$, we let $c^{\mathcal{M}}$ be the element $a \in M$ for which $a(i) = c^{\mathcal{M}_i}$ for all $i \in I$.

Given an n-place function symbol $f \in \mathbf{F}$ and a tuple $\bar{a} = (a_0, \ldots, a_{n-1})$ from M, we let $f^{\mathcal{M}}(\bar{a})$ be the element $b \in M$ such that, for all $i \in I$, we have $b(i) = f^{\mathcal{M}_i}(a_0(i), \ldots, a_{n-1}(i))$.

Given an n-place relation symbol $R \in \mathbf{R}$ and a tuple $\bar{a} = (a_0, \ldots, a_{n-1})$ from M, set $R^{\mathcal{M}}(\bar{a})$ in case $R^{\mathcal{M}_i}(a_0(i), \ldots, a_{n-1}(i))$ holds for all $i \in I$.

For $\prod_{i<n} \mathcal{M}_i$ we also write $\mathcal{M}_0 \times \ldots \times \mathcal{M}_{n-1}$. Given $i \in I$, the structure $\mathcal{M}_i$ is called the ith (**direct**) **factor** of the direct product $\mathcal{M}$. If $\mathcal{M}_i = \mathcal{N}$ for all $i \in I$, the product $\prod_{i\in I} \mathcal{M}_i$ is also called Ith **direct power** of $\mathcal{N}$, in symbols $\mathcal{N}^I$.

Remark. The axiom of choice ensures that $\prod_{i\in I} \mathcal{M}_i \neq \emptyset$ if none of the M_i is empty. (In case I is finite, the axiom of choice is not necessary for this.)

In connection with the above direct product we have the following canonical homomorphisms.

Given $j \in I$, the map $p_j : \prod_{i\in I} M_i \to M_j$ defined by $p_j(a) = a(j)$ is, by definition, the **projection onto the jth factor**.

Remark. Any such projection p_j is a homomorphism of $\mathcal{M} = \prod_{i\in I} \mathcal{M}_i$ to $\mathcal{M}_j$, which is surjective if $\mathcal{M} \neq \emptyset$.

Exercise 1.6.1. Show that $\mathcal{M} = \prod_{i\in I} \mathcal{M}_i$ is uncountable as soon as no M_i is empty and infinitely many of the M_i have at least two elements.

Exercise 1.6.2. Find an embedding $e : \mathcal{M} \to \mathcal{M}^I$ such that $p_i e = \mathrm{id}_M$ for all $i \in I$.

Chapter 2

Languages

In the first three sections of this chapter we build a formal language for each signature $\sigma = (\mathbf{C}, \mathbf{F}, \mathbf{R}, \sigma')$. More precisely, we build a *first-order* (also called an *elementary*) language $L = \mathrm{L}(\sigma)$, whose building blocks are the symbols from a certain *alphabet*, which depends on the signature σ, and whose syntactic categories are *terms* and *formulas*. (What we here simply call a **language**, logicians also call an *object language*—as opposed to the *metalanguage*, in which the main text of this book is written and in which we usually argue.)

Fix an arbitrary signature $\sigma = (\mathbf{C}, \mathbf{F}, \mathbf{R}, \sigma')$.

2.1 Alphabets

The language $\mathrm{L}(\sigma)$ we are going to define will consist of certain strings of symbols. The set of these symbols is called the **alphabet** of $\mathrm{L}(\sigma)$. It consists of the following.

Logical symbols: the **connectives** $\neg$ [read: *not*] for **negation**, $\wedge$ [read: *and*] for **conjunction**, the **existential quantifier** $\exists$ [read: *there exists* or *there is*], and the **equality symbol** $=$;

countably many **variables** (see below);

the **non-logical symbols** from the signature σ (i. e. the constant, function, and relation symbols from σ);

the **parentheses** (and).

(Thus alphabets can differ only in their non-logical symbols.) Some symbols may seem to be missing in the above, however we will see in due course that this choice suffices.

Although the alphabet contains a fixed (countable) set of variables, there will be no need to know their formal names. We denote them by $x_0, x_1, x_2, \dots$ or x, y, z, or the like. These symbols thus serve as (meta) variables for variables from the alphabet and are formally not part of the alphabet.

2.2 Terms

The **terms** of signature σ (also called σ-**terms**) are defined recursively as follows.

(i) All variables are terms.
(ii) All constant symbols are terms.
(iii) If $t_0, \dots, t_{n-1}$ are terms and $f \in \mathbf{F}$ with $\sigma'(f) = n$, then $f(t_0, \dots, t_{n-1})$ is a term too.
(iv) t is a term, if it can be built in finitely many steps using (i)–(iii).

An example of a term in the language with two binary function symbols f_1 and f_2 is $f_1(f_1(f_2(z, f_2(x, f_2(x, y))), f_2(x, x)), f_2(x, x))$. If f_1 is $+$ and f_2 is $\cdot$, then this term can be regarded as the polynomial $(z \cdot (x \cdot (x \cdot y)) + (x \cdot x)) + (x \cdot x)$. Be aware that a term is a syntactic object with no meaning attached initially. So we cannot drop the parentheses 'assuming associativity or commutativity of the operations', for terms 'know' nothing about operations, they are just built from function symbols that will later, on the semantic level of Chapter 3, be interpreted as operations in structures. In such a structure any term will then be interpreted as an element of that structure.

Given a set X of variables, the **term algebra** of L **over** X (or with **basis** X, is the L-structure $\mathrm{Term}_L(X)$ defined as follows. The domain of $\mathrm{Term}_L(X)$ is the set of all L-terms whose variables are in X. We interpret the constant and the function symbols of L by themselves, i. e., $c^{\mathrm{Term}_L(X)} = c$ and $f^{\mathrm{Term}_L(X)}(t_1, \dots, t_n) = f(t_1, \dots, t_n)$ for each $c \in \mathbf{C}$ and $f \in \mathbf{F}$. The relational symbols of L are interpreted trivially, that is we let them have empty domains, i. e., $R^{\mathrm{Term}_L(X)} = \emptyset$ for all $R \in \mathbf{R}$.

Exercise 2.2.1. Show that any map h_0 from X to an L-structure $\mathcal{M}$ can be uniquely extended to a homomorphism h from $\mathrm{Term}_L(X)$ to $\mathcal{M}$.

Exercise 2.2.2. Let h_0 and h be as above and let $t(\bar{x})$ be an L-term whose variables $\bar{x}$ are in X. Prove that $t^{\mathcal{M}}(h_0[\bar{x}]) = h(t(\bar{x}))$.

Exercise 2.2.3. (About unique legibility) Prove that no proper initial segment of a term (regarded as a string of symbols of the alphabet) can be a term. Derive that for every term there is a unique way of building it up from its constituents according to the above recursion.

2.3 Formulas

Next we build formulas from terms. While terms will later (in Chapter 3) be interpreted as *elements* of structures, formulas will be interpreted as *statements* about these elements. Thus formulas will turn out to be the objects of our languages, whose interpretations have a truth value. But, as for terms, formulas themselves are syntactical objects that 'know' nothing about elements or structures.

We define **formulas** of signature σ (or σ-**formulas**) recursively as follows.

(i) If t_1 and t_2 are σ-terms, then $t_1 = t_2$ is a formula.
(ii) If $t_0, \ldots, t_{n-1}$ are σ-terms and $R \in \mathbf{R}$ with $\sigma'(R) = n$, then $R(t_0, \ldots, t_{n-1})$ is a formula too.
(iii) If φ and ψ are formulas and x is a variable, then $\neg\varphi$, $(\varphi \wedge \psi)$, and $\exists x\, \varphi$ are formulas too.
(iv) φ is a formula if it can be obtained from (i)–(iii) in finitely many steps.

The formulas from (i) and (ii) are said to be **atomic**. We denote the class of atomic formulas by **at**. The formulas from (i) are also known as **term equations** and those from (ii) as **relational atomic formulas**. Atomic formulas and their negations are called **literals**. The only proper **subformula** of the formulas $\neg\varphi$ and $\exists x\, \varphi$ is the formula φ, the only proper **subformulas** of $(\varphi \wedge \psi)$ are its two **conjuncts**, φ and ψ.

Sometimes we are interested only in atomic formulas in which no variables have been replaced by terms other than variables. These are called **unnested** and defined formally as follows. Variables x_i and constant symbols $c \in \mathbf{C}$ are **unnested terms**, as well as terms of the form $f(x_0, \ldots, x_n)$, where the x_i are variables (and $f \in \mathbf{F}$). No other terms are unnested. An atomic formula is said to be **unnested** if it is either an unnested term equation (i. e. an equation of unnested terms) or else a relational formula of the form $R(x_0, \ldots, x_{n-1})$, where $R \in \mathbf{R}$ and the x_i are variables.

An example of an atomic formula in $\mathrm{L}(+, \cdot)$ is $(z \cdot (x \cdot (x \cdot y))) + y = (z \cdot (x \cdot (x \cdot y)) + (x \cdot x)) + (x \cdot x)$; it is not unnested.

The terms and formulas of the signature σ together form the **expressions** of $\mathrm{L}(\sigma)$ (or $\mathrm{L}(\sigma)$-**expressions**). Formally we define the **language** $\mathrm{L}(\sigma)$ to be the set of all σ-formulas. All concepts from Chapter 1 corresponding to signatures (like **reducts**, signatures **without constants**, **purely relational** signatures etc.) can thus be applied to languages as well.

Note that all the expressions of the language are *finite* strings of symbols, hence one can check in finitely many steps and effectively if a string of symbols is a term or a formula of the given language.

The correspondence between signatures and languages being one-to-one (in fact, uniquely determined by the set of non-logical symbols), we may write $\sigma = \sigma(L)$ instead of $L = \mathrm{L}(\sigma)$.

Given a language L without constants, $\emptyset_L$ denotes the empty L-structure (cf. §1.2).

Exercise 2.3.1. Verify that there are only finitely many unnested atomic sentences in L, provided the signature of L is finite.

Exercise 2.3.2. (About unique legibility)

Prove that no proper initial segment of a formula (regarded as a string of symbols of the alphabet) can be a formula. Derive that for every formula there is a unique way of building it up from its constituents according to the above recursion.

2.4 Abbreviations

We now introduce some common notation as abbreviations, like the **nullary connectives** $\top$ [read: *true* or *verum*] and $\bot$ [read: *false* or *falsum*], the binary connectives $\vee$ [read: *or*] for the **disjunction**, $\rightarrow$ [read: *if, then*] for the **implication** or **subjunction**, and $\leftrightarrow$ [read: *if and only if*] for the **equivalence** or **equijunction**, the many-place connectives $\bigwedge$ and $\bigvee$ for **multiple conjunction** and **disjunction**, and the **universal quantifier** $\forall$ [read: *for all*].

The formal definitions are as follows. Given terms t_1 and t_2, formulas φ and ψ, and a natural number $n > 0$ we write
$t_1 \neq t_2$ for $\neg t_1 = t_2$,
$\bot$ for $\exists x\, x \neq x$,
$\top$ for $\neg\bot$,
$\varphi \vee \psi$ for $\neg(\neg\varphi \wedge \neg\psi)$,
$\varphi \rightarrow \psi$ for $\neg\varphi \vee \psi$,
$\varphi \leftrightarrow \psi$ for $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$, $\forall x\, \varphi$ for $\neg\exists x\, \neg\varphi$,
$\exists x_0 \ldots x_{n-1}\, \varphi$ (also $\exists \bar{x}\, \varphi$ in case $\bar{x} = (x_0, \ldots, x_{n-1})$) for $\exists x_0 \ldots \exists x_{n-1}\, \varphi$,
$\bigwedge_{i<n} \varphi_i$ for $(\ldots(\varphi_0 \wedge \varphi_1) \wedge \ldots) \wedge \varphi_{n-1}$ (here $\varphi_0, \ldots, \varphi_{n-1}$ are arbitrary formulas),
$\bigvee_{i<n} \varphi_i$ for $(\ldots(\varphi_0 \vee \varphi_1) \vee \ldots) \vee \varphi_{n-1}$,
$\exists^{\geq n} x\, \varphi$ for $\exists x_0 \ldots x_{n-1}(\bigwedge_{i<j<n} x_i \neq x_j \wedge \bigwedge_{i<n} \varphi(x_i))$,
$\exists^{\leq n-1} x\, \varphi$ for $\neg\exists^{\geq n} x\, \varphi$,
$\exists^{=n} x\, \varphi$ for $\exists^{\geq n} x\, \varphi \wedge \exists^{\leq n} x\, \varphi$.
Sometimes we also write $\exists^{>n}$ for $\exists^{\geq n+1}$ (similarly for $\exists^{<n}$) and $\exists!$ for $\exists^{=1}$.

The **disjuncts** of a disjunction $\varphi \vee \psi$ are the **subformulas** φ and ψ. The **subformulas** of an implication $\varphi \to \psi$ are the **premise** φ and the **conclusion** ψ.

Parentheses are used in formulas (and terms) to indicate their syntactic structure. However, in order to avoid too many, we adopt the rule that the unary connective $\neg$ binds more strongly than the binary connectives $\wedge$ and $\vee$, and the latter bind more strongly than $\to$ and $\leftrightarrow$. Further, outer parentheses around formulas may also be omitted (as we have already done in the above abbreviations). On the other hand, parentheses may be added if this serves their readability.

Note that the abbreviations introduced are not formally part of the language. This has the advantage of avoiding many cases when doing proofs by induction on the complexity of a formula, in which case we have to deal only with $\neg$, $\wedge$, and $\exists$. However, we do use the other connectives and quantifier for the following syntactical classification of formulas.

Let Σ be a set of formulas. A **boolean combination** of formulas from Σ is, by definition, a formula that can be obtained from formulas from Σ by using $\vee$, $\wedge$ and $\neg$ only. (Obviously, we may also allow $\to$ and $\leftrightarrow$, and we could do without $\vee$.) A **positive boolean combination** of formulas from Σ is a formula that can be obtained from formulas from Σ by using only $\wedge$ and $\vee$. The **boolean closure** of Σ is the set of all boolean combinations of formulas from Σ, denoted by $\widetilde{\Delta}$.

A formula is **positive** if it can be obtained from atomic formulas using only $\wedge$, $\vee$, $\exists$, and $\forall$. The class of all positive formulas (of all possible languages) is denoted by $+$.

A **negative** formula is a negated positive formula. The class of all such is denoted by $-$.

A formula is **quantifier-free** if it contains no quantifiers, where, for technical reasons, we assume $\top$ and $\bot$ to be quantifier-free too.[1]

The class of all quantifier-free formulas (of arbitrary signature) is denoted by **qf**.

Thus **qf** is the class of all boolean combinations of atomic formulas. The class of all positive formulas from **qf** is the class of all positive boolean combinations of atomic formulas.

[1] This is relevant only in case of quantifier-free *sentences* in languages without constants, cf. Remark (3) in §3.3.

2.5 Free and bound variables

A main ingredient of our formal language are the placeholders for elements of a structure—the variables. (Note that the term *first-order* indicates that only variables for elements occur, as opposed to *second-order logic*, which also has variables for sets of elements.) They allow us, as common in mathematics, to formulate in our formal language relations between elements without naming them concretely. Accordingly, we have to distinguish between two different kinds of occurrences of variables in a formula—an occurrence as such a placeholder and an occurrence as an operator variable for a quantifier.

More formally, we make the following definition. In the formula $\exists x\, \varphi$, the subformula φ is said to be the **scope** of the quantifier. The occurrence of x after the quantifier is x's occurrence as **operator variable**. This occurrence as well as each occurrence of x in the scope of the quantifier is a **bound occurrence** of this variable, while any occurrence that is not bound is said to be a **free occurrence** of this variable. A **free variable** of a formula is, by definition, a variable that has a free occurrence in this formula.

Example. All occurrences of x in the formula $\forall x(x = y \vee \exists y(x \neq y))$ are bound, while the first of the occurrences of y is free and the other two are bound. Hence y is the only free variable of this formula.

A particular role play expressions without free variables. A term t is said to be **constant** if it contains no variables at all. A formula φ is said to be a **sentence** if it contains no free variables.

Remark.

(1) (i) Every constant symbol is a constant term.
 (ii) If $t_0, \ldots, t_{n-1}$ are constant terms and $f \in \mathbf{F}$ with $\sigma'(f) = n$, then $f(t_0, \ldots, t_{n-1})$ is also a constant term.
 (iii) Obviously, a term t is constant iff it can be obtained in finitely many steps from (i) and (ii).

(2) Atomic sentences, i. e. atomic formulas that are sentences, are either equations of constant terms or relational sentences of the form $R(t_0, \ldots, t_{n-1})$, where the t_i are constant terms.

(3) Languages without constants thus have no atomic *sentences*. According to our convention about $\top$ and $\bot$ (from §2.4) the only quantifier-free sentences in this case are $\top$ and $\bot$ and their boolean combinations.

For technical reasons we introduce the following division among formulas of a given language L. Given a tuple $\bar{x}$ of variables, $L_{\bar{x}}$ is to denote the set of L-formulas, whose free variable are among the variables from $\bar{x}$. Further, L_n is used to denote the collection of all L-formulas that have *precisely* n free variables (no matter which). For $\bigcup_{k\leq n} L_k$ we also write $L_{\leq n}$.

Thus, L_0 is the set of L-sentences, and $L = \bigcup_{n\in\mathbb{N}} L_n$. Further, $L_{\bar{x}}$ is the set of all L-formulas whose free variables are from $\bar{x}$, while $L_{l(\bar{x})}$ contains only those formulas from $L_{\bar{x}}$ in which *all* entries of $\bar{x}$ occur free (however, $L_{l(\bar{x})}$ also contains all other L-formulas with precisely $l(\bar{x})$ free variables). If, for instance, $\bar{x} = (x_0, \ldots, x_7)$, then $x_0 = x_1$ is in $L_{\bar{x}} \subseteq L_{\leq 8}$, but $x_7 = x_{27}$ is not, while for arbitrary variables x and y the formula $x = y$ is in L_2, but not in L_8. This notational difference is rather technical, for one can always add so-called *dummy variables*, as we will see in §3.2 below.

The **cardinality** or **power** of the language L is, by definition, the cardinal[2] number $|L|$. In §7.6 we will see that $|L| = |L_n| = |L_0| = \max\{\aleph_0, |\sigma|\} = \max\{\aleph_0, |\mathbf{C} \cup \mathbf{F} \cup \mathbf{R}|\}$. Hence a language is countably infinite[3] precisely if the set of symbols from the signature is countable (that is, finite or countably infinite).

Exercise 2.5.1. Find a recursive definition of free variable that is according to the syntactical complexity of the formula under consideration.

2.6 Substitutions

An important syntactic operation on expressions of the language is that of substitution of variables by terms. For example, in the language $L(+,\cdot)$, substituting x by $z \cdot z$ in the term $x + z$ we obtain the term $(z \cdot z) + z$. Similarly in polynomial equations (which are formulas in that language). Care has to be taken only in the case of bound variables, e. g., substituting x by $z \cdot z$ in the formula $\exists z(x + z = 0)$ would bring the variable z of the term $z \cdot z$ under the scope of $\exists z$. This phenomenon is called **collision of variables**. Renaming is a remedy for this: in the above example replace first z by y, say, which yields $\exists y(x + y = 0)$, and only then substitute. The resulting formula would then be $\exists y((z \cdot z) + y = 0)$.

The reader will have no great difficulty to understand how to perform this renaming of bound variables in general (but note, the result is not unique,

[2] We introduce cardinals only in Chapter 6. For the time being it suffices to replace every statement about cardinals by a statement about sets having the same power (which is defined, as usual, by the existence of a bijection).

[3] Every language being infinite anyway (due to infinite supply of variables at the least), we may simply say 'countable' here.