[write η, ζ, θ etc. for linear

ıples and strings. If $\bar{a}$ and $\bar{b}$ are
isting of $\bar{a}$ followed by $\bar{b}$; but
lash between the usual notation
ɔups: in model theory xy is the
ɔups it is x times y. One has to
ɜuity I have used $x \hat{} y$ for the
ɔduct.
d when we need it. The most
ive sections of Chapter 2.

1

Naming of parts

Every person had in the beginning one only proper name, except the savages of Mount Atlas in Barbary, which were reported to be both nameless and dreamless.

William Camden

In this first chapter we meet the main subject-matter of model theory: structures.

Every mathematician handles structures of some kind – be they modules, groups, rings, fields, lattices, partial orderings, Banach algebras or whatever. This chapter will define basic notions like 'element', 'homomorphism', 'substructure', and the definitions are not meant to contain any surprises. The notion of a (Robinson) 'diagram' of a structure may look a little strange at first, but really it is nothing more than a generalisation of the multiplication table of a group.

Nevertheless there is something that the reader may find unsettling. Model theorists are forever talking about symbols, names and labels. A group theorist will happily write the same abelian group multiplicatively or additively, whichever is more convenient for the matter in hand. Not so the model theorist: for him or her the group with '·' is one structure and the group with '+' is a different structure. Change the name and you change the structure.

This must look like pedantry. Model theory is an offshoot of mathematical logic, and I can't deny that some distinguished logicians have been pedantic about symbols. Nevertheless there are several good reasons why model theorists take the view that they do. For the moment let me mention two.

In the first place, we shall often want to compare two structures and study the homomorphisms from one to the other. What is a homomorphism? In the particular case of groups, a homomorphism from group G to group H is a map that carries multiplication in G to multiplication in H. There is an obvious way to generalise this notion to arbitrary structures: a homomorphism from structure A to structure B is a map which carries each operation of A to *the operation with the same name in B*.

Secondly, we shall often set out to build a structure with certain properties. One of the maxims of model theory is this: *name the elements of your* *structure first, then decide how they should behave.* If the names are well chosen, they will serve both as a scaffolding for the construction, and as raw materials.

Aha – says the group theorist – I see you aren't really talking about *written* symbols at all. For the purposes you have described, you only need to have formal labels for some parts of your structures. It should be quite irrelevant what kinds of thing your labels are; you might even want to have uncountably many of them.

Quite right. In fact we shall follow the lead of A. I. Mal'tsev and put no restrictions at all on what can serve as a name. For example any ordinal can be a name, and any mathematical object can serve as a name of itself. The items called 'symbols' in this book need not be written down. They need not even be dreamed.

1.1 Structures

We begin with a definition of 'structure'. It would have been possible to set up the subject with a slicker definition – say by leaving out clauses (1.2) and (1.4) below. But a little extra generality at this stage will save us endless complications later on.

A **structure** A is an object with the following four ingredients.

(1.1) A set called the **domain** of A, written $\text{dom}(A)$ or $\text{dom}\,A$ (some people call it the **universe** or **carrier** of A). The elements of $\text{dom}(A)$ are called the **elements** of the structure A. The **cardinality** of A, in symbols $|A|$, is defined to be the cardinality $|\text{dom}\,A|$ of $\text{dom}(A)$.

(1.2) A set of elements of A called **constant elements**, each of which is named by one or more **constants**. If c is a constant, we write c^A for the constant element named by c.

(1.3) For each positive integer n, a set of n-ary relations on $\text{dom}(A)$ (i.e. subsets of $(\text{dom}\,A)^n$), each of which is named by one or more n-ary **relation symbols**. If R is a relation symbol, we write R^A for the relation named by R.

(1.4) For each positive integer n, a set of n-ary operations on $\text{dom}(A)$ (i.e maps from $(\text{dom}\,A)^n$ to $\text{dom}(A)$), each of which is named by one or more n-ary **function symbols**. If F is a function symbol, we write F^A for the function named by F.

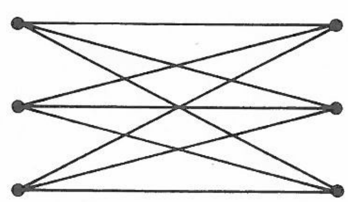
Except where we say otherwise, any of the sets (1.1)–(1.4) may be empty. As mentioned in the chapter introduction, the constant, relation and function 'symbols' can be any mathematical objects, not necessarily written symbols;

but for peace of mind one normally assumes that, for instance, a 3-ary relation symbol doesn't also appear as a 3-ary function symbol or a 2-ary relation symbol. We shall use capital letters A, B, C, . . . for structures.

Sequences of elements of a structure are written $\bar{a}$, $\bar{b}$ etc. A **tuple in** A (or **from** A) is a finite sequence of elements of A; it is an n-**tuple** if it has length n. Usually we leave it to the context to determine the length of a sequence or tuple.

This concludes the definition of 'structure'.

Example 1: ***Graphs.*** A **graph** consists of a set V (the set of **vertices**) and a set E (the set of **edges**), where each edge is a set of two distinct vertices. An edge $\{v, w\}$ is said to **join** the two vertices v and w. We can picture a finite graph by putting dots for the vertices and joining two vertices v, w by a line when $\{v, w\}$ is an edge:



One natural way to make a graph G into a structure is as follows. The elements of G are the vertices. There is one binary relation R^G; the ordered pair (v, w) lies in R^G if and only if there is an edge joining v to w.

Example 2: ***Linear orderings.*** Suppose $\leqslant$ linearly orders a set X. Then we can make $(X, \leqslant)$ into a structure A as follows. The domain of A is the set X. There is one binary relation symbol R, and its interpretation R^A is the ordering $\leqslant$. (In practice we would usually write the relation symbol as $\leqslant$ rather than R.)

Example 3: ***Groups.*** We can think of a group as a structure G with one constant 1 naming the identity 1^G, one binary function symbol $\cdot$ naming the group product operation $\cdot^G$, and one unary function symbol $^{-1}$ naming the inverse operation $^{(-1)^G}$. Another group H will have the same symbols 1, $\cdot$, $^{-1}$; then 1^H is the identity element of H, $\cdot^H$ is the product operation of H, and so on.

Example 4: ***Vector spaces.*** There are several ways to make a vector space into a structure, but here is the most convenient. Suppose V is a vector space over a field of scalars K. Take the domain of V to be the set of vectors of V. There is one constant element 0^V, the origin of the vector space. There is one binary operation, $+^V$, which is addition of vectors. There is a 1-ary operation $-^V$ for additive inverse; and for every scalar k there is a 1-ary operation k^V to represent multiplying a vector by k. Thus each scalar serves as a 1-ary function symbol. (In fact the symbol '$-$' is redundant, because $-^V$ is the same operation as $(-1)^V$.)

When we speak of vector spaces below, we shall assume that they are structures of this form (unless anything is said to the contrary). The same goes for modules, replacing the field K by a ring.

Two questions spring to mind. First, aren't these examples a little arbitrary? For example, why did we give the group structure a symbol for the multiplicative inverse $^{-1}$, but not a symbol for the commutator $[\,,\,]$? Why did we put into the linear ordering structure a symbol for the ordering $\leqslant$, but not one for the corresponding strict ordering $<$?

The answer is yes; these choices were arbitrary. But some choices are more sensible than others. We shall come back to this in the next section.

And second, *exactly* what is a structure? Our definition said nothing about the way in which the ingredients (1.1)–(1.4) are packed into a single entity.

True again. But this was a deliberate oversight – the packing arrangements will never matter to us. Some writers define A to be an ordered pair $\langle \mathrm{dom}(A), f\rangle$ where f is a function taking each symbol S to the corresponding item S^A. The important thing is to know what the symbols and the ingredients are, and this can be indicated in any reasonable way.

For example a model theorist may refer to the structure

$$\langle \mathbb{R}, +, -, \cdot, 0, 1, \leqslant\rangle.$$

With some common sense the reader can guess that this means the structure whose domain is the set of real numbers, with constants 0 and 1 naming the numbers 0 and 1, a 2-ary relation symbol $\leqslant$ naming the relation $\leqslant$, 2-ary function symbols $+$ and $\cdot$ naming addition and multiplication respectively, and a 1-ary function symbol naming minus.

Signatures

The **signature** of a structure A is specified by giving

(1.5) the set of constants of A, and for each separate $n > 0$, the set of n-ary relation symbols and the set of n-ary function symbols of A.

We shall assume that the signature of a structure can be read off uniquely from the structure.

f V to be the set of vectors of V.
gin of the vector space. There is one
f vectors. There is a 1-ary operation
alar k there is a 1-ary operation k^V
Thus each scalar serves as a 1-ary
' is redundant, because $-^V$ is the

ow, we shall assume that they are
is said to the contrary). The same
y a ring.

st, aren't these examples a little
the group structure a symbol for the
l for the commutator [,]? Why did
a symbol for the ordering $\leqslant$, but not
$<$?
arbitrary. But some choices are more
to this in the next section.
e? Our definition said nothing about
.4) are packed into a single entity.
oversight – the packing arrangements
define A to be an ordered pair
g each symbol S to the correspond-
o know what the symbols and the
in any reasonable way.
r to the structure
$0, 1, \leqslant\rangle$.
guess that this means the structure
, with constants 0 and 1 naming the
ol $\leqslant$ naming the relation $\leqslant$, 2-ary
tion and multiplication respectively,
s.

res

d by giving
each separate $n > 0$, the set
the set of n-ary function

a structure can be read off uniquely

The symbol L will be used to stand for signatures. Later it will also stand for languages – think of the signature of A as a kind of rudimentary language for talking about A. If A has signature L, we say A is an **L-structure**.

A signature L with no constants or function symbols is called a **relational signature**, and an L-structure is then said to be a **relational structure**. A signature with no relation symbols is sometimes called an **algebraic signature**.

Exercises for section 1.1

1. According to Thomas Aquinas, God is a structure G with three elements '*pater*', '*filius*' and '*spiritus sanctus*', in a signature consisting of one asymmetric binary relation ('*relatio opposita*') R, read as '*relatio originis*'. Aquinas asserts also that the three elements can be uniquely identified in terms of R^G. Deduce – as Aquinas did – that if the pairs (*pater, filius*) and (*pater, spiritus sanctus*) lie in R^G, then exactly one of the pairs (*filius, spiritus sanctus*) and (*spiritus sanctus, filius*) lies in R^G.

2. Let X be a set and L a signature; write $\kappa(X, L)$ for the number of distinct L-structures which have domain X. Show that if X is a finite set then $\kappa(X, L)$ is either finite or at least 2^ω.

1.2 Homomorphisms and substructures

The following definition is meant to take in, with one grand sweep of the arm, virtually all the things that are called 'homomorphism' in any branch of algebra.

Let L be a signature and let A and B be L-structures. By a **homomorphism** f from A to B, in symbols $f: A \to B$, we shall mean a function f from $\mathrm{dom}(A)$ to $\mathrm{dom}(B)$ with the following three properties.

(2.1) For each constant c of L, $f(c^A) = c^B$.

(2.2) For each $n > 0$ and each n-ary relation symbol R of L and n-tuple $\bar{a}$ from A, if $\bar{a} \in R^A$ then $f\bar{a} \in R^B$.

(2.3) For each $n > 0$ and each n-ary function symbol F of L and n-tuple $\bar{a}$ from A, $f(F^A(\bar{a})) = F^B(f\bar{a})$.

(If $\bar{a}$ is $(a_0, \ldots, a_{n-1})$ then $f\bar{a}$ means $(fa_0, \ldots, fa_{n-1})$; cf. Note on notation.) By an **embedding** of A into B we mean a homomorphism $f: A \to B$ which is injective and satisfies the following stronger version of (2.2).

(2.4) For each $n > 0$, each n-ary relation symbol R of L and each n-tuple $\bar{a}$ from A, $\bar{a} \in R^A \Leftrightarrow f\bar{a} \in R^B$.

An **isomorphism** is a surjective embedding. Homomorphisms $f: A \to A$ are called **endomorphisms** of A. Isomorphisms $f: A \to A$ are called **automorphisms** of A.