Introduction to Model-Theory and Mathematical Logic

David Pierce

2006.01.03

Preface

These notes started out as transcriptions of lectures given for Math 406, 'Introduction to Mathematical Logic and Model-theory', at METU in 2004. I have expanded on some points and rearranging some topics.

I assume that the reader already knows something of certain topics, as covered in Math 111, 'Fundamentals of Mathematics':

(∗) formal logic (predicate and first-order);

(†) sets, relations, and functions;

(‡) induction and recursion on the set of natural numbers.

In writing these notes, I attempt to distinguish notationally between **constants** and **variables.** However, what is a constant in one context is a variable in another.

In a tradition at least as old as Descartes's *Geometry* [4], originally published in French in 1637), letters from the beginning of the Latin alphabet stand for known quantities; letters from the end, unknown. Hence, if we are asked to solve the equation

$$ax^2 + bx + c = 0, \tag{1}$$

we know that we are expected to come up with the equation

$$x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a},$$

rather than, say,

$$a = -\frac{bx + c}{x^2}.$$

In Equation (1), the letters a, b, and c, are understood to stand for particular numbers; the letter x does not stand for a particular number, but for the 'possibility' of a number. Grammatically, Equation (1) is 'incomplete' or 'elliptical'. The equation might stand for the instructions,

> Find every number d such that Equation (1) becomes a true statement when d replaces x.

Alternatively, the equation stands for the set

$$\{x : ax^2 + bx + c = 0\}, \tag{2}$$

whose members would be obtained by following the instructions.

The expression on Line (2) is grammatically a noun. It represents a sort of 'completion' of Equation (1) by means of an additional use of the letter x. A different letter, such as z, could be used in place of x in Line (2) without changing the set indicated. Such an observation identifies the letter x in Equation (1) as a **variable.**

Contents

Chapter 1

Introduction

{ch:intro}

1.0 Building-blocks

This first section reviews some basic definitions and conventions to be followed in these notes.

An **ordered pair** is a set

$$\{\{a\},\{a,b\}\},$$

which is denoted

$$(a,b).$$

The sole purpose of the definition is to ensure that

$$(a,b)=(x,y) \iff a=x \;\&\; b=y.$$

The **Cartesian product** of sets A and B is the set

$$\{(x,y) : x \in A \wedge y \in B\},$$

denoted

$$A \times B.$$

To express that some set f is a **function** from A to B, we can just write

$$f : A \longrightarrow B.$$

This means f is a subset of $A \times B$ with a certain property, namely, for every a in A, there is a unique b in B such that $(a,b) \in f$; then we write

$$f(a)=b.$$

The function f can also be written as

$$x \longmapsto f(x).$$

The set of all functions from A to B can be denoted

$$B^A. \tag{1.1}$$

(Some people write ${}^{A}B$ instead.)

Let ω be the set of **natural numbers**:

$$\omega = \{0, 1, 2, 3, \dots\} = \{0, 0', 0'', 0''', \dots\}.$$

It is notationally convenient to treat 0 as $\varnothing$, and n' as $n \cup \{n\}$. Then

$$n = \{0, \dots, n-1\}$$

for all n in ω. Under this understanding of the natural numbers, the n**th Cartesian power** of A is precisely

$$A^n,$$

in the notation introduced on Line (1.1) above. Thus, the nth Cartesian power of A is the set of functions from n to A. An element of A^n can be written as any one of

$$(a_0, \dots, a_{n-1}), \qquad i \longmapsto a_i, \qquad \vec{a}\,;$$

it can be called an **(ordered)** n**-tuple** from A. Note well that

$$A^0 = \{\varnothing\} = \{0\} = 1;$$

this is true even if A is empty. Also, every element of A^1 is $\{(0, a)\}$ for some a in A. So we have a bijection

$$x \longmapsto \{(0, x)\} \tag{1.2}$$

from A to A^1. We may sometimes treat this bijection as an **identification:** that is, we may neglect to distinguish between a and $\{(0, a)\}$.

For any m and n in ω, we have a bijection

$$(\vec{x}\,, \vec{y}\,) \longmapsto \vec{x}\,{}^\frown\vec{y} \tag{1.3}$$

from $A^m \times A^n$ to A^{m+n}. In this notation, $\vec{a}\,{}^\frown\vec{b}$ is the $(m+n)$-tuple

$$(a_0, \dots, a_{m-1}, b_0, \dots, b_{n-1});$$

this is the $(m+n)$-tuple $\vec{c}$ such that

$$c_k = \begin{cases} a_k, & \text{if } k < m; \\ b_{k-m}, & \text{if } m \leqslant k < m+n. \end{cases}$$

We shall always treat the bijection on Line (1.3) as an identification; in particular, we shall always write $(\vec{a}\,, \vec{b}\,)$ instead of $\vec{a}\,{}^\frown\vec{b}$.

An n**-ary operation on** A is a function from A^n to A. The set of these can be denoted

$$A^{A^n}.$$

In particular, a 0-ary or **nullary** operation on A is an element of A^1; by the bijection in Line (1.2) then, we may identify a nullary operation on A with an element of A.

An n**-ary relation on** A is a subset of A^n; the set of these is

$$\mathcal{P}(A^n).$$

In particular, a nullary operation is a subset of A^0, that is, of 1 (or $\{0\}$); so the nullary operation is 0 or 1.

An n-ary operation on A is then a (certain kind of) subset of $A^n \times A$, and this product can be identified with $A^n \times A^1$ and hence with A^{n+1}; so an n-ary operation on A can be thought of as an $(n+1)$-ary relation on A. More precisely, if $f : A^n \to A$, then one may refer to the $(n+1)$-ary relation

$$\{(\vec{x}, f(\vec{x})) : \vec{x} \in A^n\}$$

as the **graph** of f; but there is a bijection between graphs in this sense and functions.

1.1 Structures

{S:structures}

Our fundamental object of study will be *structures*. The notion of a structure provides a way to unify the treatment of many mathematical ideas. By our official definition, a **structure** is an ordered pair $(A, \mathfrak{I})$, also referred to as $\mathfrak{A}$, where:

(∗) A is a non-empty set, called the **universe** of the structure;

(†) $\mathfrak{I}$ is a function, written also
$$s \longmapsto s^{\mathfrak{A}},$$
whose domain $\mathcal{L}$ is called the **signature** of the structure;

(‡) $s^{\mathfrak{A}}$ is either an element of A or an n-ary operation or relation on A for some positive n, for each s in $\mathcal{L}$.

If $\mathcal{L} = \{s_0, s_1, \dots\}$, then $\mathfrak{A}$ can be written

$$(A, s_0^{\mathfrak{A}}, s_1^{\mathfrak{A}}, \dots).$$

Examples 1.1.1. The following are structures: {examples:structures}

(1) $(\omega, {}', 0)$;
(2) a group G, or $(G, \cdot, {}^{-1}, 1)$;
(3) an *abelian* group G, or $(G, +, -, 0)$;
(4) a unital ring R, or $(R, +, -, \cdot, 0, 1)$;
(5) the ring $\mathbb{Z}$, or $(\mathbb{Z}, +, -, \cdot, 0, 1)$;
(6) the field $\mathbb{R}$, or $(\mathbb{R}, +, -, \cdot, 0, 1)$;
(7) the two-element field $\mathbb{F}_2$, or $(\mathbb{F}_2, +, -, \cdot, 0, 1)$; see § 1.2;
(8) a partial order $(\Omega, \leqslant)$;
(9) the *ordered* field $\mathbb{R}$, or $(\mathbb{R}, +, -, \cdot, 0, 1, \leqslant)$;
(10) a vector-space V over a field K; here the signature of V is
$$\{+, -, 0\} \cup \{a\cdot\ : a \in K\},$$
where $a\,\cdot$ is the singulary operation of multiplying by a;

(11) the **power-set** structure on a non-empty set Ω, namely

$$(\mathcal{P}(\Omega), \cap, \cup, {}^{\mathrm{c}}, \varnothing, \Omega, \subseteq);$$

(12) the **truth-**structure

$$(\mathbb{B}, \wedge, \vee, \neg, 0, 1, \vDash),$$

where $\mathbb{B} = \{0, 1\}$, and $\vDash$ is the binary relation $\{(0,0), (0,1), (1,1)\}$. (The name 'truth-structure' is not standard, as far as I know.) •

Note well that $\mathbb{B} = \{\varnothing, \{\varnothing\}\} = \mathcal{P}(\{\varnothing\}) = \mathcal{P}(1)$, and the truth-structure *is* the power-set structure on 1. **Propositional logic** studies the truth-structure; **model-theory** studies *all* structures.

With $\mathfrak{I}$ as above in the structure $(A, \mathfrak{I})$:

($*$) $s^{\mathfrak{A}}$ is the **interpretation** in $\mathfrak{A}$ of s;

(†) s is a **symbol** for $s^{\mathfrak{A}}$.

So s is one of the following:

($*$) a **constant**;

(†) an n**-ary function-symbol** for some positive n in ω;

(‡) an n**-ary predicate**[1] for some positive n in ω.

Since nullary operations on A can be considered as elements of A, a constant can be considered as a nullary function-symbol.

Here are some observations about our definition of *structure*:

($*$) I am following the old convention (used for example in [2]) of denoting the universe of a structure by a Roman letter, and the structure itself by the corresponding Fraktur letter. Recent writers (as in [9] or [12]) use 'calligraphic' letters, not Fraktur:

For a structure with universe:	A	B	C	$\dots$	M	N	$\dots$
I write:	$\mathfrak{A}$	$\mathfrak{B}$	$\mathfrak{C}$	$\dots$	$\mathfrak{M}$	$\mathfrak{N}$	$\dots$
Others may write:	$\mathcal{A}$	$\mathcal{B}$	$\mathcal{C}$	$\dots$	$\mathcal{M}$	$\mathcal{N}$	$\dots$

Another option (taken in [7]) is to use an ordinary letter like A for a structure, and then $\mathrm{dom}(A)$ for its universe. (Here 'dom' stands for *domain*.) Finally, one might not bother to make a typographical distinction between a structure and its universe. Indeed, as suggested in the examples, the distinction is not easy to make with standard structures like $\mathbb{Z}$ or $\mathbb{R}$.

(†) Similarly, it is not always easy or convenient to distinguish between a symbol and its interpretation. A **homomorphism** from a group G to a group H is usually described as a function f from G to H such that

$$f(g_0 \cdot g_1) = f(g_0) \cdot f(g_1)$$

for all g_e in G. If we are trying to be precise, we should call the groups $\mathfrak{G}$ and $\mathfrak{H}$, with group-operations $\cdot^{\mathfrak{G}}$ and $\cdot^{\mathfrak{H}}$ respectively, and we should say that f is such that

$$f(g_0 \cdot^{\mathfrak{G}} g_1) = f(g_0) \cdot^{\mathfrak{H}} f(g_1)$$

[1] Or *relation-symbol*.

for all g_0 and g_1 in G. But writing this way soon becomes tedious.

(‡) In a structure $(A, \mathfrak{I})$, the universe A and the **interpretation**-function $\mathfrak{I}$ work together to provide interpretations of the symbols in $\mathcal{L}$ *as* elements of, or operations or relations on, a certain set, namely A itself. That's all a structure is: something that provides a mathematical interpretation for certain symbols. We shall develop this idea later. What makes model-theory interesting is that the same symbols can have different interpretions. For example, $\cdot$ in $\mathbb{Z}$ has different properties from $\cdot$ in $\mathbb{R}$. Here {syntax} begins the distinction between **syntax** (abstract rules for working with symbols) and **semantics** (the meaning of the symbols; see also Chapter 2).

1.2 Propositional logic

{sect:prop}

This section reviews propositional logic; but the subject will be treated more deeply in Chapter 2.

Of the so-called *truth-structure* given among the Examples 1.1.1, the signature is $\{\wedge, \vee, \neg, 0, 1, \vDash\}$. The symbol $\vDash$ is here a binary predicate (later it will also have other uses). The other symbols are function-symbols; we shall call them **propositional connectives.**[2] We may use additional propositional connectives. For example:

(0) 0 and 1 are nullary connectives;

(1) $\neg$ is a singulary[3] connective;

(2) $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$, and $\nleftrightarrow$ are binary connectives.

Each of these has a standard interpretation as an operation on $\mathbb{B}$. These interpretations can be given by **truth-tables:**

P	Q	0	1	$\neg P$	$P \wedge Q$	$P \vee Q$	$P \rightarrow Q$	$P \leftrightarrow Q$	$P \nleftrightarrow Q$
0	0	0	1	1	0	0	1	1	0
1	0			0	0	1	0	0	1
0	1				0	1	1	0	1
1	1				1	1	1	1	0

Alternatively, we can first understand $\mathbb{B}$ as a two-element unital ring with addition- and multiplication-tables

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

Then $\mathbb{B}$ is the ring sometimes denoted $\mathbb{Z}_2$; it is a *field,* and as such can be denoted $\mathbb{F}_2$. Then

($*$) 0 and 1 are symbols for themselves;

(†) $\nleftrightarrow$ is another symbol for addition;

[2]Or *Boolean connectives.*

[3]This word is more etymologically correct than the more common ***unary.***

(‡) $\wedge$ is another symbol for multiplication;

(§) the remaining connectives are thus:

symbol	interpretation
$\neg$	$x \mapsto x+1$
$\vee$	$(x,y) \mapsto x \cdot y + x + y$
$\to$	$(x,y) \mapsto x \cdot y + x + 1$
$\leftrightarrow$	$(x,y) \mapsto x + y + 1$

In general, a **signature** for propositional logic is a set of propositional connectives. Let $\mathcal{L}$ be such. The **(propositional) formulas** of $\mathcal{L}$ are certain strings composed of:

($*$) symbols from $\mathcal{L}$;

(†) **propositional variables** P_0, P_1, P_2, ...

In particular:

($*$) each variable is a formula.

(†) $*\mathbf{F}_0 \cdots \mathbf{F}_{n-1}$ is a formula, if $*$ is an n-ary connective from $\mathcal{L}$, and the $\mathbf{F}_i$ are formulas. (If $n = 0$, then $*$ by itself is a formula.)

(Why we *can* define a set of propositional formulas this way will be indicated in § 2.0.) Usually, if $*$ is binary, then, instead of $*\mathbf{F}_0\mathbf{F}_1$, we write

$$(\mathbf{F}_0 * \mathbf{F}_1) \tag{1.4}$$

(thus introducing new symbols: the two parentheses; but we usually do not write the outermost set of parentheses in a formula). A formula is **n-ary** if[4] its variables belong to the set $\{P_0, \dots, P_{n-1}\}$. If a formula is n-ary, then its **arity** is n. An n-ary formula $\mathbf{F}$ can be written as

$$\mathbf{F}(P_0, \dots, P_{n-1}).$$

As each propositional connective has a standard interpretation as an operation on $\mathbb{B}$, so every n-ary formula has a standard interpretation as such an operation, in an obvious way. We can say then that the n-ary formula **represents** the n-ary operation that is its standard interpretation. Notationally, an n-ary formula $\mathbf{F}$ will represent the function

$$\vec{x} \longmapsto \widehat{\mathbf{F}}(\vec{x})$$

from $\mathbb{B}^n$ to $\mathbb{B}$. Then the standard interpretations of formulas can be defined as follows:

($*$) If $k < n$, then the formula P_k is an n-ary formula and, as such, represents the operation

$$\vec{x} \longmapsto x_k$$

from $\mathbb{B}^n$ to $\mathbb{B}$. (This operation can be denoted $\widehat{P}_k$.)

(†) If $\{\mathbf{F}_0, \dots, \mathbf{F}_{n-1}\}$ is a set of n formulas, each of them k-ary, and if $*$ is an n-ary propositional connective in $\mathcal{L}$, then the formula $*\mathbf{F}_0 \cdots \mathbf{F}_{n-1}$ represents the function

$$\vec{x} \longmapsto g(\widehat{\mathbf{F}}_0(\vec{x}), \dots, \widehat{\mathbf{F}}_{n-1}(\vec{x}))$$

[4] Alternatively, one may want to refer to a formula as n-ary if it contains no more than n distinct variables, without worrying about which variables those are.

from $\mathbb{B}^k$ to $\mathbb{B}$, where g is the standard interpretation of $*$.

In particular, if $*$ is n-ary, then its standard interpretation is $\widehat{\mathbf{G}}$, where $\mathbf{G}$ is the formula $*P_0 \cdots P_{n-1}$. When formulas are written with the notation of Line (1.4), then their interpretations can be given by truth-tables in the style shown in the proof of Theorem 1.2.2 below.

The notion that a propositional formula represents an operation will be developed further in the next chapter in case $\mathcal{L}$ is $\{\neg, \to\}$. We shall be able to restrict ourselves to this signature, because it is *adequate*. In general, a signature for propositional logic is **adequate** if, for each n-ary operation g on $\mathbb{B}$, there is an $(n+k)$-ary formula $\mathbf{F}$ of $\mathcal{L}$ (for some k) such that

$$g(\vec{e}) = \widehat{\mathbf{F}}(\vec{e}, \vec{f})$$

for all $\vec{e}$ in $\mathbb{B}^n$ and $\vec{f}$ in $\mathbb{B}^k$: that is, every operation on $\mathbb{B}$ is **represented** in $\mathcal{L}$ by some formula. We allow the arity of $\mathbf{F}$ here to be larger than that of g, since we want it to be possible for signatures without nullary connectives to be adequate.

The following basic tool for establishing adequacy of a signature was proved by Emil Post in 1921 [11]:

{lem:Post}

Lemma 1.2.1. *A signature of propositional logic is adequate, provided that, in this signature, the following operations are represented:*

($*$) *the constant functions* 0 *and* 1*;*

($\dagger$) *the ternary function* f *given by*

e_0	e_1	e_2	$f(\vec{e})$
0	0	0	0
1	0	0	1
0	1	0	0
1	1	0	1
0	0	1	0
1	0	1	0
0	1	1	1
1	1	1	1

.

Proof. We use induction on the arity of operations. The nullary operations are represented in the signature by assumption. Suppose all n-ary operations are represented, and g is $(n+1)$-ary. If $e \in \mathbb{B}$, let h_e be the n-ary operation $\vec{x} \mapsto g(\vec{x}, e)$. By definition,

$$f(e_0, e_1, e_2) = \begin{cases} e_0, & \text{if } e_2 = 0; \\ e_1, & \text{if } e_2 = 1. \end{cases}$$

Then for all $\vec{d}$ in $\mathbb{B}^n$, we have

$$g(\vec{d}, e) = h_e(\vec{d}) = f(h_0(\vec{d}), h_1(\vec{d}), e).$$

Thus the function g is

$$(\vec{x}, y) \longmapsto f(h_0(\vec{x}), h_1(\vec{x}), y).$$

By inductive hypothesis, each of the functions h_e is represented by some formula

$$\mathbf{H}_e(P_0, \dots, P_{n-1}, \dots);$$

by assumption, f is represented by some function $\mathbf{F}(P_0, P_1, P_2, \dots)$. Hence g is represented by

$$\mathbf{F}(\mathbf{H}_0(P_0, \dots, P_{n-1}, \dots), \mathbf{H}_1(P_0, \dots, P_{n-1}, \dots), P_n, \dots).$$

By induction, the operations of all arities are represented. □

{thm:to-not}

Theorem 1.2.2. *The propositional signature* $\{\to, \neg\}$ *is adequate.*

Proof. By the lemma, it is enough to observe:

(∗) $P_0 \to P_0$ represents 1;

(†) $\neg(P_0 \to P_0)$ represents 0;

(‡) the operation f as in the lemma is represented by the formula

$$\neg((\neg P_2 \to P_0) \to \neg(P_2 \to P_1)),$$

since its truth-table is

$\neg$	$(($$\neg$	P_2	$\to$	$P_0)$	$\to$	$\neg$	$(P_2$	$\to$	$P_1))$
0	1	0	0	0	1	0	0	1	0
1	1	0	1	1	0	0	0	1	0
0	1	0	0	0	1	0	0	1	1
1	1	0	1	1	0	0	0	1	1
0	0	1	1	0	1	1	1	0	0
0	0	1	1	1	1	1	1	0	0
1	0	1	1	0	0	0	1	1	1
1	0	1	1	1	0	0	1	1	1

.

(Note that the last formula is **equivalent** to $(\neg P_2 \to P_0) \wedge (P_2 \to P_1)$.) □

1.3 Syntax and semantics

We introduce propositional connectives as a way to understand, and to make precise, certains parts of ordinary language: namely, conjunctions and other 'structural' words like ***and, or, not,*** and ***if. . . then.*** For example, we interpret the connectives $\neg$ and $\to$ as in the truth-tables

$\neg$	P_0
1	0
0	1

P_0	$\to$	P_1
0	1	0
1	0	0
0	1	1
1	1	1

because:

(∗) we think[5] of 0 as falsity and 1 as truth;

[5]It is possible to think the other way, where 0 is truth and 1 is falsity; this is done, for example, in [14, Ch. 4, Exercise 3.7, p. 178].

(†) we take $\neg$ to stand for a word like ***not*** that *negates* sentences, and we take $\rightarrow$ to stand for the locution ***if. . . then;***

(‡) in our mathematical writing at any rate,

- a claim will be true if and only if its negation is false, and
- an implication ***If*** A***, then*** B will be false if and only if A is true, but B is false.

A *tautology* will be a propositional formula that is 'always true'; that is, the interpretation of an n-ary tautology will be the constant n-ary function $\vec{x} \mapsto 1$. In particular, the propositional formula $P_0 \rightarrow P_0$ will be a tautology. This will be so, because the sentence ***If*** A***, then*** A is always true; also, from the truth-table for $P_0 \rightarrow P_1$ given above, we can construct the truth-table

P_0	$\rightarrow$	P_0
0	1	0
1	1	1

In short, tautology is a **semantic** notion: it concerns the 'meaning' of formulas. At least, the notion of tautology concerns the meaning of formulas when they are thought of as forms of sentences of ordinary language. (The etymology of ***semantic*** is discussed below). To express that a formula is a tautology, we shall write in front of it the symbol

$$\vDash .$$

Whenever a notion is based directly on truth-tables, we shall consider it to be semantic.

Gottlob Frege is credited with the first precise formulation of an alternative to the truth-table method for establishing tautologies. This formulation is given in the *Begriffsschift* [15] of 1879. A bit of Frege's peculiar notation (discussed below in § 2.4) survives: To express that a formula is a tautology found by Frege's method, we shall write in front of it the symbol

$$\vdash .$$

We shall refer to this method as **syntactic,** because it directly involves the way that symbols are arranged into formulas, without consideration of their possible interpretations. (Again, etymology is discussed below.) Forty-two years later, in 1921, in the same paper cited in the last section as the source of Lemma 1.2.1, Emil Post published a proof [11, p. 169] that the syntactical method can establish *all* tautologies:

$$\vdash \mathbf{F} \iff \vDash \mathbf{F}.$$

The syntactical method is the method of **formal proof.** It is a mechanical method, in the sense that a machine can recognize when the method has been applied successfully.

The method of formal proof does have its foundation in semantic considerations, if only because it is designed to establish semantic facts. Also, the truth-table method itself is mechanical, so it too seems to have a right to be called syntactic. Thus, our distinction between the syntactic and the semantic is somewhat arbitrary. The distinction will be more profound in the context of first-order logic.

The remainder of this section treats[6] the Greek origins of our words ***syntax*** and ***semantics.***

The Greek etymon for ***syntax,*** namely s ntaxij, -ewj, refers originally to an *arranging,* a *putting together in order,* especially of soldiers. In one passage of Plato's *Republic* [10, 591d], it is *wealth* that may be arranged. In that passage, the character of Socrates describes the wise man:

> O ko n, e pon, ka t n nt t n crhm twn kt sei s ntax n te ka sumfwn an; ka t n gkon to pl qouj o k kpl htt menoj p to t n poll n makarismo peiron a x sei, p ranta kak cwn;
>
> And will it not also be so, I said, with the arranging and harmonizing of his possessions? He will not let himself be dazzled by the felicitations of the multitude and pile up the mass of his wealth without measure, involving himself in measureless ills, will he?[7]

The arranging implied by s ntaxij can also be *grammatical,* a putting together of *words.*

The source of ***semantics*** is the Greek adjective shmantik j, - , - n, meaning *significant* or *meaningful.* Related words include the verb shma nw (signify) and the noun t shme on (sign). In *On Interpretation* [1, 16a19, b5], Aristotle defines nouns and verbs:

> Onoma m n o n st fwn shmantik kat sunq khn neu cr nou, j mhd n mroj st shmantik n kecwrism non:
>
> 'R ma d sti t pross hma non cr non, o m roj o d n shma nei cwr j, ka stin e t n kaq' t rou l egom nwn shme on.
>
> A noun is a sound, meaningful by convention, without [grammatical] tense, of which no part separately is meaningful.
>
> A verb is [a sound] signifying a tense besides; no part of it is meaningful separately; it is always a sign of *things said of* something.

The more basic t s ma, -atoj, meaning *sign, mark, token,* appears in Homer (*Iliad,* X.465–468):

> Wj r' f nhs en, ka p qen y s' e raj
> q ken n mur khn: d el on d/ p s m t' qhke
> summ ryaj d nakaj mur khj t' riqhl aj zouj,
> m l qoi a tij nte qo n di n kt a m l ainan.
>
> With these words, he took the spoils and set them upon a tamarisk tree, and they make a mark at the place by pulling up reeds and gathering boughs of tamarisk, that they might not miss it as they came back through the fleeting hours of darkness.[8]

[6] With the help of the Liddell–Scott lexicon [8].
[7] The translation is adapted from Shorey's [10].
[8] Text and Samuel Butler's translation are from `http://www.perseus.tufts.edu`.

Chapter 2

Propositional model-theory

{ch:prop}

2.0 Propositional formulas

{sect:prop-form}

This chapter is inspired in part by Chang and Kiesler [2, § 1.2], who describe the subject to be discussed here as '"toy" model theory'.

Usually, the term *model-theory* refers to first-order model-theory, because the logic it uses is first-order logic. The notion of *structure* defined above in § 1.1 is the notion as used in first-order model-theory. A structure provides an interpretation for certain symbols; also, as we shall see, a structure can be a *model* for sets of *sentences*.

The concepts of structure, interpretation, model, and sentence, have analogues in the simpler context of propositional logic. In this logic, a *truth-assignment* will take the place of a structure. A truth-assignment will provide an interpretation for *propositional formulas*, and will serve as a model for sets of propositional formulas.

Our official signature for propositional logic will be

$$\{\rightarrow, \neg\},$$

although we may introduce other connectives as abbreviations. With the elements of our signature, along with parentheses, we shall build up **propositional formulas** from a countably infinite set

$$V$$

of **propositional variables**. For us, this set will be

$$\{P_k : k \in \omega\};$$

however, we establish:

Notational Convention 2.0.1. *Bold-face letters* $\mathbf{P}$*,* $\mathbf{Q}$*, and* $\mathbf{R}$*, will stand for members of the set* V*.*

The set of propositional formulas will be called

$$\mathrm{PF};$$

we shall give a set-theoretic definition this set.

A formula is a certain *string* of symbols. A **string** is just a function on some n in ω. So, a string is just an n-tuple; but we shall generally write a string as

$$s_0 s_1 \cdots s_{n-1}$$

instead of $(s_0, s_1, \dots, s_{n-1})$. Each of the expressions s_k here stands for an **entry** in the string. The **length** of the string is n: we may write

$$\ell\mathrm{n}(s_0 s_1 \cdots s_{n-1}) = n.$$

The string **begins** with s_0 and **ends** with s_{n-1}. (The unique string of length 0 has no entries, and no beginning or ending.) If the string $s_0 s_1 \cdots s_{n-1}$ is in PF, then each entry s_i in the string will be:

($*$) an element of the set V of variables, or

(†) one of the connectives $\rightarrow$ and $\neg$, or

(‡) one of the parentheses (or).

We shall want to refer to strings by single letters, such as **A** and **B**. As such, these letters are strings of length 1; but they *stand for* strings of other lengths. I am writing the letters in boldface as a reminder that they are not literally strings of symbols in our propositional logic.[1] For the same reason, **P**, **Q**, and **R**, are written in bold face; they are not variables in PF, but they *stand for* variables.

{strings} For the moment, let S be the set of *all* strings of entries from the set

$$V \cup \{\rightarrow, \neg, (,)\}.$$

That is, let

$$S = \bigcup_{n\in\omega} (V \cup \{\rightarrow, \neg, (,)\})^n.$$

We shall define PF as a subset of S. To do so, we let $\mathcal{U}$ be the set of all subsets N of S such that:

($*$) each variable (considered as a string of length 1) is in N;

(†) if $\mathbf{A}$ is in N, then $\neg\mathbf{A}$ is in N;

(‡) if $\mathbf{A}$ and $\mathbf{B}$ are in N, then $(\mathbf{A} \rightarrow \mathbf{B})$ is in N.

To put this another way,[2] for the moment, let f be the singulary operation such that $f(\mathbf{A})$ is the string $\neg\mathbf{A}$, and let g be the binary operation such that $g(\mathbf{A}, \mathbf{B})$ is the string $(\mathbf{A} \rightarrow \mathbf{B})$, for all $\mathbf{A}$ and $\mathbf{B}$ in S. Then

$$\begin{aligned}\mathcal{U} = \{\Xi \in \mathcal{P}(S) : V \subseteq \Xi \wedge \forall X\ (X \in \Xi \rightarrow f(X) \in \Xi) \wedge \\ \wedge \forall X\ \forall Y\ (X \in \Xi \wedge Y \in \Xi \rightarrow g(X, Y) \in \Xi)\}.\end{aligned}$$

[1] In technical terms, they are *syntactical variables:* they are certain symbols of the *syntax language.* The latter is the language that we our using to talk about the *object language,* which in this case is the language of propositional logic, which uses the symbols just listed above. See [3, § 8].

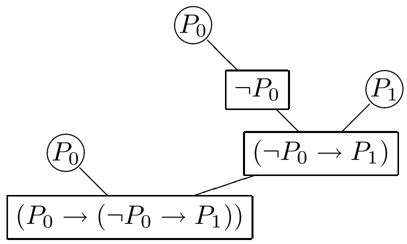
[2] And in a way that uses the formal symbolism of propositional and first-order logic!

Now we can define

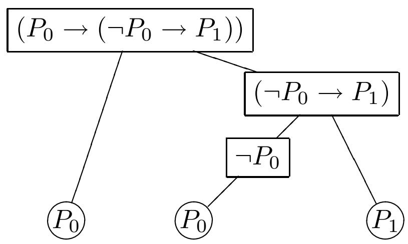
$$\mathrm{PF} = \bigcap \mathcal{U}.$$

This is an *inductive* definition, as will be discussed further in the next section. In particular, it means that every element of PF can be displayed as the 'trunk' or 'root' of a *tree* whose 'leaves' are variables: {example:tree}

Example 2.0.2. The string $(P_0 \to (\neg P_0 \to P_1))$ belongs to PF, because it is constructed from the variables P_0 and P_1 in a way permitted by the conditions for being an element of PF. This is suggested by the following picture:



One might draw this picture upside down:



Thus one can depict a formula as being built *up* from the variables, rather than *down* from them. •

In this context, a **tree** is a set T equipped with a partial ordering $\leqslant$ such that, for each a in T, the set $\{x : x \leqslant a\}$ is finite and is totally ordered by $\leqslant$. (Here a partial ordering is just a reflexive, antisymmetric, and transitive relation. In a more general definition of tree, each set $\{x : x \leqslant a\}$ is only required to be *well-ordered* by $\leqslant$; the set may be infinite.)

Exercises

(1) As in Example 2.0.2, draw trees for some formulas, such as

 (∗) $(P_0 \to (P_1 \to P_0))$,
 (†) $((P_0 \to (P_1 \to P_2)) \to ((P_0 \to P_1) \to (P_0 \to P_2)))$,
 (‡) $((\neg P_0 \to \neg P_1) \to (P_1 \to P_0))$.

(2) Show that the binary relation

$$\{(\mathbf{A}, \mathbf{B}) \in \mathrm{PF}^2 : \ell\mathrm{n}(\mathbf{A}) \leqslant \ell\mathrm{n}(\mathbf{B})\}$$

on PF is a partial ordering R such that

- $\mathbf{A}\ R\ \neg\mathbf{A}$,
- $\mathbf{A}\ R\ (\mathbf{A} \to \mathbf{B})$, and
- $\mathbf{B}\ R\ (\mathbf{A} \to \mathbf{B})$,

for all $\mathbf{A}$ and $\mathbf{B}$ in PF.

{exercise:sub-f}

(3) The **sub-formulas** of a formula are the formulas that appear in the tree for that formula. Give a precise definition of the notion of sub-formula: that is, define a partial ordering $\leqslant$ of PF such that $\mathbf{A} \leqslant \mathbf{B}$ if and only if $\mathbf{A}$ is a sub-formula of $\mathbf{B}$. (Your answer may be that $\leqslant$ is the intersection of a family of partial orderings of PF.)

2.1 Induction

{sect:induction}

The definition of PF is **inductive,** because it makes the next theorem possible. Recall that, to prove by **induction** that a certain set A of natural numbers contains *all* natural numbers, one proves two things:

($*$) that A contains 0, and

(†) that if A contains n, then A contains $n+1$ (no matter which natural number n is).

With the following theorem, a similar method will be available for showing that a certain set of propositional formulas contains *all* propositional formulas. This method can be called **induction on (the complexity of) formulas.** The proof that the method works is almost immediate from the definition of PF:

{thm:induction}

Theorem 2.1.1 (Induction on Formulas)**.** *Suppose N is a set of propositional formulas such that:*

($*$) *each variable (considered as a string of length* 1*) is in N;*

(†) *if $\mathbf{F}$ is in N, then $\neg\mathbf{F}$ is in N;*

(‡) *if $\mathbf{F}$ and $\mathbf{G}$ are in N, then $(\mathbf{F} \to \mathbf{G})$ is in N.*

Then $N =$ PF.

Proof. Let N be as supposed. Then, in particular, $N \subseteq$ PF. Moreover, N is a member of the set $\mathcal{U}$ given in the definition of PF; so $\bigcap\mathcal{U} \subseteq N$. But PF $= \bigcap\mathcal{U}$. Therefore $N \subseteq$ PF and PF $\subseteq N$; so $N =$ PF. □

Notational Convention 2.1.2. *Bold-face letters $\mathbf{F}$, $\mathbf{G}$, and $\mathbf{H}$, (and variants like $\mathbf{F}'$ and $\mathbf{G}_k$,) will always stand for formulas (that is, elements of* PF*).*

Suppose $g : V \to$ PF. Then g determines a certain function, to be denoted

$$\mathbf{F} \longmapsto \mathbf{F}(g),$$

with domain PF. This function can be called **substitution with respect to** g. The reason for the unusual way of denoting the function will become clear. Suppose $g(P_k)$ is $\mathbf{G}_k$ for each k in ω. If exactly m entries in $\mathbf{F}$ are variables, so that $\mathbf{F}$ can be written as

$$\dots P_{k_0} \dots P_{k_1} \dots \cdots \dots P_{k_{m-1}} \dots,$$

then the formula $\mathbf{F}(g)$ is

$$\ldots \mathbf{G}_{k_0} \ldots \mathbf{G}_{k_1} \ldots \cdots \ldots \mathbf{G}_{k_{m-1}} \ldots$$

If the variables that appear in $\mathbf{F}$ belong to the set $\{P_0, \ldots, P_{n-1}\}$, then $\mathbf{F}$ can be denoted

$$\mathbf{F}(P_0, \ldots, P_{n-1}),$$

and $\mathbf{F}(g)$ can be denoted

$$\mathbf{F}(\mathbf{G}_0, \ldots, \mathbf{G}_{n-1}).$$

It would be a tedious exercise, but a practicable one, to write down an expression for the kth entry in $\mathbf{F}(\mathbf{G}_0, \ldots, \mathbf{G}_{n-1})$, for arbitrary k. That $\mathbf{F}(\mathbf{G}_0, \ldots, \mathbf{G}_{n-1})$ is actually *in* PF is a consequence of the following:

Theorem 2.1.3. *Let* $g : V \to \mathrm{PF}$*. Then the function* $\mathbf{F} \mapsto \mathbf{F}(g)$ *on* PF *has co-domain* PF*.*

Proof. We use induction on formulas to prove

$$\{\mathbf{F} : \mathbf{F}(g) \in \mathrm{PF}\} = \mathrm{PF}.$$

The argument has three parts:

- $(*)$ By assumption, $\mathbf{P}(g)$, which is $g(\mathbf{P})$, is in PF.
- $(\dagger)$ Suppose $\mathbf{F}(g)$ is a formula $\mathbf{H}$. Then substitution with respect to g in $\neg\mathbf{F}$ results in $\neg\mathbf{H}$, which is in PF by definition (of PF).
- $(\ddagger)$ Suppose $\mathbf{F}(g)$ and $\mathbf{G}(g)$ are formulas $\mathbf{H}$ and $\mathbf{H}'$ respectively. Then $(\mathbf{F} \to \mathbf{G})(g)$ is $(\mathbf{H} \to \mathbf{H}')$, which again is in PF by definition.

This completes the induction and the proof. □

If the foregoing discussion of substitution seems too informal or imprecise, let it be noted that the operation $\mathbf{F} \mapsto \mathbf{F}(g)$ can be defined *recursively*, by means of Theorem 2.2.5 below. However, substitution makes sense for sets of strings that do *not* admit definition by recursion or even proof by induction.

Exercises

(1) Prove by induction that every formula has as many left as right parentheses.

(2) Prove by induction that an entry $\neg$ is never preceded by a variable in any formula.

(3) For each k in ω, there is a function h_k from PF to ω such that $h_k(\mathbf{F})$ is the number of times that P_k appears in $\mathbf{F}$. Using this, find the length of $\mathbf{F}(\mathbf{G}_0, \ldots, \mathbf{G}_{n-1})$ in terms of $\ell\mathrm{n}(\mathbf{F})$ and the $\ell\mathrm{n}(\mathbf{G}_k)$.

(4) Supposing $\mathbf{F}(\mathbf{G}_0, \ldots, \mathbf{G}_{n-1})$ is $s_0 s_1 \cdots s_{M-1}$, what is s_k?

2.2 Recursion

Recall that functions from ω into an arbitrary set B can be defined by **recursion:** If

($*$) $c \in B$, and

(†) $f : B \to B$,

then there is exactly one function g from ω to B such that

($*$) $g(0) = c$, and

(†) $g(n+1) = f(g(n))$ for all n in ω.

That there is *exactly* one such function g means:

($*$) there is at most one such function, and

(†) there is at least one such function.

The first of these claims can be proved by induction; the second requires more. Similarly, induction on formulas gives us the following:

{lem:uniqueness}

Lemma 2.2.1. *Suppose A is a set, and*

($*$) $h_0 : V \to A$,

(†) h_1 *is a singulary operation on A, and*

(‡) h_2 *a binary operation on A.*

Then there is at most one function h on PF *such that*

($*$) h *agrees with* h_0 *on* V*;*

(†) $h(\neg\mathbf{F}) = h_1(h(\mathbf{F}))$*, for all* $\mathbf{F}$*;*

(‡) $h((\mathbf{F} \to \mathbf{G})) = h_2(h(\mathbf{F}), h(\mathbf{G}))$*, for all* $\mathbf{F}$ *and* $\mathbf{G}$*.*

Proof. Suppose h' and h'' are such functions h. Let D consist of those formulas $\mathbf{F}$ such that $h'(\mathbf{F}) = h''(\mathbf{F})$. Then D contains the variables; and if D contains $\mathbf{F}$ and $\mathbf{G}$, then D contains $\neg\mathbf{F}$ and $(\mathbf{F} \to \mathbf{G})$. By induction, D is PF, so $h' = h''$. □

If there *is* a function h as described in the Lemma, then it is said to be defined by **recursion.** To prove that recursively defined functions exist at all, we shall use the following.

{lem:formulas}

Lemma 2.2.2. *Every formula in* PF *meets the following conditions:*

($*$) *it has positive length;*

(†) *if it has length* 1*, then it is a variable;*

(‡) *if it has length greater than* 1*, then it begins with* $\neg$ *or* (*;*

(§) *if it begins with* $\neg$*, then it is* $\neg\mathbf{F}$ *for some* formula $\mathbf{F}$*;*

(¶) *if it begins with* (*, then it is* $(\mathbf{F} \to \mathbf{G})$ *for some* formulas $\mathbf{F}$ *and* $\mathbf{G}$*.*

Proof. We use induction on formulas. Let N be the subset of PF containing every formula that meets the given conditions.

We first show that N contains the variables. Let $\mathbf{P}$ be a variable. Then $\mathbf{P}$ has length 1, which is positive; so $\mathbf{P}$ meets the first two conditions. Since $\mathbf{P}$ does not have length greater than 1, it trivially meets the third condition. Since $\mathbf{P}$ begins with a variable, it does not begin with (or $\neg$; hence $\mathbf{P}$ trivially meets the remaining two conditions. Therefore $\mathbf{P}$ is in N.

Suppose N contains $\mathbf{F}$. Then $\mathbf{F}$ has positive length, so $\neg\mathbf{F}$ is a formula that has length greater than 1 and satisfies the five conditions. Hence $\neg\mathbf{F}$ is in N.

Suppose finally that N contains $\mathbf{F}$ and $\mathbf{G}$. Then $(\mathbf{F} \to \mathbf{G})$ is a formula that meets the five conditions, so it is in N.

Hence $N = \mathrm{PF}$ by Theorem 2.1.1. □

So that we can talk clearly about formulas, we declare that:

(∗) every formula $\neg\mathbf{F}$ is a **negation;**

(†) every formula $(\mathbf{F} \to \mathbf{G})$ is an **implication,** with **antecedent F** and **consequent G**.

The last lemma formalizes the straightforward observation that every formula is a variable, a negation, or an implication, and is *only* one of these. To prove that recursively defined functions on PF exist, we shall need to know that every implication has a *unique* antecedent and consequent. To prove *this,* it will be useful to have the following definitions. An **initial segment** of the string $s_0 s_1 \cdots s_{n-1}$ is one of the strings $s_0 s_1 \cdots s_{k-1}$, where $k \leqslant n$. This initial segment is **proper** if $k < n$.

The proof of the following lemma requires *strong induction* on the *lengths* of formulas. Recall that, to prove by **strong induction** that a subset A of ω *is* ω, one proves that, for all n in ω, if A contains every natural number that is *less* than n, then A contains n.

{lem:no-pis}

Lemma 2.2.3. *No proper initial segment of a formula is a formula.*

Proof. Let N be the set of formulas of which no proper initial segment is a formula. We shall prove by strong induction on the lengths of formulas that $N = \mathrm{PF}$. Suppose N contains all formulas *shorter* than a formula $\mathbf{F}$. By Lemma 2.2.2, we know that $\mathbf{F}$ is a variable $\mathbf{P}$ or a formula $\neg\mathbf{G}$ or $(\mathbf{G} \to \mathbf{H})$. The only proper initial segment of $\mathbf{P}$ is the empty string, which is not a formula. Any proper initial segment of $\neg\mathbf{G}$ is $\neg\mathbf{A}$ for some proper initial segment $\mathbf{A}$ of $\mathbf{G}$; so $\mathbf{A}$ is not a formula, by our strong inductive hypothesis; hence $\neg\mathbf{A}$ is not a formula, again by Lemma 2.2.2. Finally, say $\mathbf{F}$ is $(\mathbf{G} \to \mathbf{H})$. Any initial segment of $\mathbf{F}$ that is a formula is $(\mathbf{G}' \to \mathbf{H}')$ for some formulas $\mathbf{G}'$ and $\mathbf{H}'$ (by Lemma 2.2.2). Then one of $\mathbf{G}$ and $\mathbf{G}'$ is an initial segment of the other. But each one is shorter than $\mathbf{F}$; so by strong inductive hypothesis, $\mathbf{G}$ and $\mathbf{G}'$ are the same formula. Then $\mathbf{H}'$ is an initial segment of $\mathbf{H}$; so *these* formulas must be the same. Thus, in all cases, $\mathbf{F}$ is in N. By strong induction, $N = \mathrm{PF}$. □

{thm:urf}

Theorem 2.2.4 (Unique Readability of Formulas)**.** *Every formula is a variable, a negation, or an implication, but only one of these; and every implication has a unique antecedent and consequent.*

Proof. We know the first part by Lemma 2.2.2. For the second part, suppose $(\mathbf{F} \to \mathbf{G})$ and $(\mathbf{F}' \to \mathbf{G}')$ are the same formula. Then one of $\mathbf{F}$ and $\mathbf{F}'$ is an initial segment of the other, so they are the same by Lemma 2.2.3; hence $\mathbf{G}$ and $\mathbf{G}'$ are the same. □

Now we can prove that recursively defined functions on PF exist. Note carefully how the proof relies on the previous theorem.

{thm:recursion}

Theorem 2.2.5 (Recursion on Formulas)**.** *Suppose A is a set, and*

($*$) $h_0 : V \to A$,

(†) h_1 *is a singulary operation on* A, *and*

(‡) h_2 *a binary operation on* A.

Then there is a unique function h on PF *such that*

($*$) h *agrees with* h_0 *on* V*;*

(†) $h(\neg\mathbf{F}) = h_1(h(\mathbf{F}))$, *for all* $\mathbf{F}$*;*

(‡) $h((\mathbf{F} \to \mathbf{G})) = h_2(h(\mathbf{F}), h(\mathbf{G}))$, *for all* $\mathbf{F}$ *and* $\mathbf{G}$.

Proof. By Lemma 2.2.1, we now need only prove that a function h *does* exist as desired. Let $\mathcal{U}$ be the set of all subsets R of PF $\times A$ such that:

($*$) $(\mathbf{P}, h_0(\mathbf{P})) \in R$ for all variables $\mathbf{P}$;

(†) if $(\mathbf{F}, b) \in R$, then $(\neg\mathbf{F}, h_1(b)) \in R$;

(‡) if $(\mathbf{F}, b) \in R$ and $(\mathbf{G}, c) \in R$, then $((\mathbf{F} \to \mathbf{G}), h_2(b, c)) \in R$.

Let $S = \bigcap \mathcal{U}$. Then $S \in \mathcal{U}$, so S has the properties desired of h, except perhaps for being a function. To prove that S *is* a function on PF, let D be the set of formulas $\mathbf{F}$ for which there is a unique b in A such that $(\mathbf{F}, b) \in S$. We proceed again by induction on formulas:

($*$) By its definition, S contains every ordered pair $(\mathbf{P}, h_0(\mathbf{P}))$. If $b \neq h_0(\mathbf{P})$, then $S \smallsetminus \{(\mathbf{P}, b)\} \in \mathcal{U}$, so $S \subseteq S \smallsetminus \{(\mathbf{P}, b)\}$, which means $(\mathbf{P}, b) \notin S$. Hence $\mathbf{P} \in D$.

(†) Suppose $\mathbf{F} \in D$. Then $(\mathbf{F}, b) \in S$ for some unique b. Hence S contains $(\neg\mathbf{F}, h_1(b))$, but if $c \neq h_1(b)$, then $S \smallsetminus \{(\neg\mathbf{F}, c)\} \in \mathcal{U}$, so $(\neg\mathbf{F}, c) \notin S$. Therefore $\neg\mathbf{F} \in D$.

(‡) Suppose finally that $\mathbf{F}$ and $\mathbf{G}$ are in D, and $(\mathbf{F}, b)$ and $(\mathbf{G}, c)$ are in S. By Theorem 2.2.4, if $(\mathbf{F}' \to \mathbf{G}')$ is the same formula as $(\mathbf{F} \to \mathbf{G})$, then $\mathbf{F}'$ is $\mathbf{F}$, and $\mathbf{G}'$ is $\mathbf{G}$. Hence, if $d \neq h_2(b, c)$, then $S \smallsetminus \{(\mathbf{F} \to \mathbf{G}), d)\} \in \mathcal{U}$, so $((\mathbf{F} \to \mathbf{G}), d) \notin S$. Therefore $(\mathbf{F} \to \mathbf{G}) \in D$.

We can conclude that $D = $ PF, so S is a function h as desired. □

Example 2.2.6. There is a unique singulary operation $\mathbf{F} \mapsto \mathbf{F}^*$ on PF such that

($*$) $\mathbf{P}^* = \mathbf{P}$ for each $\mathbf{P}$;

(†) $(\neg\mathbf{F})^* = \neg\mathbf{G}$, where $\mathbf{G}$ is $\mathbf{F}^*$, for each formula $\mathbf{F}$;

(‡) $(\mathbf{F} \to \mathbf{G})^* = \neg(\mathbf{G}^* \to \mathbf{F}^*)$ for all formulas $\mathbf{F}$ and $\mathbf{G}$.

In the notation of the Recursion Theorem, $\mathbf{F} \mapsto \mathbf{F}^*$ is the function h when h_0 is id_V, and h_1 is $\mathbf{F} \mapsto \neg\mathbf{F}$, and h_2 is $(\mathbf{F}, \mathbf{G}) \mapsto \neg(\mathbf{G} \to \mathbf{F})$. Hence, for example,

$$\begin{aligned}(\mathbf{P} \to (\neg\mathbf{Q} \to \mathbf{R}))^* &= \neg((\neg\mathbf{Q} \to \mathbf{R})^* \to \mathbf{P}^*)\\ &= \neg(\neg(\mathbf{R}^* \to \neg\mathbf{Q}^*) \to \mathbf{P})\\ &= \neg(\neg(\mathbf{R} \to \neg\mathbf{Q}) \to \mathbf{P}),\end{aligned}$$

whereas $((\mathbf{P} \to \neg\mathbf{Q}) \to \mathbf{R})^*$ is the formula $\neg(\mathbf{R} \to \neg(\neg\mathbf{Q} \to \mathbf{P}))$. •

The proof of the Recursion Theorem can be modified so as to yield the following more general result. (The reader should check the details.) {por:recursion}

Porism 2.2.7. *Suppose*

- $h_0 : V \to A$,
- $h_1 : \mathrm{PF} \times A \to A$, *and*
- $h_2 : (\mathrm{PF} \times A)^2 \to A$.

Then there is a unique function h on PF *such that*

($*$) *h agrees with h_0 on V;*

(†) *$h(\neg\mathbf{F}) = h_1(\mathbf{F}, h(\mathbf{F}))$ for all $\mathbf{F}$;*

(‡) *$h((\mathbf{F} \to \mathbf{G})) = h_2((\mathbf{F}, h(\mathbf{F})), (\mathbf{G}, h(\mathbf{G})))$ for all $\mathbf{F}$ and $\mathbf{G}$.*

We used Unique Readability (Theorem 2.2.4) to prove the Recursion Theorem, 2.2.5; conversely, Unique Readability follows from Porism 2.2.7. Indeed, using the notation of the Porism, let A be PF, let h_0 and h_1 be chosen arbitrarily, and let h_2 be

$$((\mathbf{F}, \mathbf{F}'), (\mathbf{G}, \mathbf{G}')) \longmapsto (\mathbf{F}, \mathbf{G}).$$

Let h be the function guaranteed by the Porism. Then $h((\mathbf{F} \to \mathbf{G})) = (\mathbf{F}, \mathbf{G})$. Thus h selects, from an implication, its antecedent and consequent. Since h is a function, the antecedent and consequent are unique.

Note well that the Recursion Theorem is *not* a consequence of the Induction Theorem: {example:no-recursion}

Example 2.2.8. Suppose we define PF without using parentheses. We shall still be able to use induction, but if we are not careful, we shall not have definitions by recursion. Indeed, say we define nPF (for 'not PF') so that:

($*$) each variable is in nPF;

(†) if $\mathbf{A}$ is in nPF, then so is $\neg\mathbf{A}$;

(‡) if $\mathbf{A}$ and $\mathbf{B}$ are in nPF, then so is $\mathbf{A} \to \mathbf{B}$.

Then proof by induction in nPF is possible. However, suppose we try to define a function f from nPF into PF so as to send every element of the former to its 'equivalent' in the latter:

($*$) $f(\mathbf{P}) = \mathbf{P}$;

(†) $f(\neg\mathbf{F}) = \neg f(\mathbf{F})$;

(‡) $f(\mathbf{F} \to \mathbf{G}) = (f(\mathbf{F}) \to f(\mathbf{G}))$.

Then $f(\mathbf{P} \to \mathbf{Q}) = (\mathbf{P} \to \mathbf{Q})$ for all variables $\mathbf{P}$ and $\mathbf{Q}$; but $f(P_0 \to P_1 \to P_2)$ must be both $(P_0 \to (P_1 \to P_2))$ and $((P_0 \to P_1) \to P_2)$, which is absurd, since these are different formulas, and f is a function.

A correct way to avoid using parentheses is to use *Łukasiewicz-* or *Polish notation*, writing $\to \mathbf{F}\ \mathbf{G}$ instead of $(\mathbf{F} \to \mathbf{G})$. Details are left to the reader. See also § 2.4 below. •

Exercises

(1) Give a recursive definition of the set of sub-formulas of a formula. (See § 2.0, Exercise 3.)

(2) Prove Porism 2.2.7.

(3) Prove the Recursion Theorem, 2.2.5, in case all formulas are written in Łukasiewicz-notation (see Example 2.2.8).

2.3 Syntactic entailment

{sect:syn-entail}

Now suppose Σ is a subset of PF. We shall define, on PF, a singulary relation called **syntactic entailment by** Σ. We shall denote the relation by

$$\Sigma \vdash .$$

If $\mathbf{F}$ satisfies the relation, we shall write

$$\Sigma \vdash \mathbf{F}.$$

This can be read as Σ ***syntactically entails*** $\mathbf{F}$, or $\mathbf{F}$ ***is a syntactic consequence of*** Σ**.** The relation is defined by the following conditions:

(∗) $\Sigma \vdash (\mathbf{F} \to (\mathbf{G} \to \mathbf{F}))$;

(†) $\Sigma \vdash ((\mathbf{F} \to (\mathbf{G} \to \mathbf{H})) \to ((\mathbf{F} \to \mathbf{G}) \to (\mathbf{F} \to \mathbf{H})))$;

(‡) $\Sigma \vdash ((\neg\mathbf{F} \to \neg\mathbf{G}) \to (\mathbf{G} \to \mathbf{F}))$;

(§) if $\mathbf{F} \in \Sigma$, then $\Sigma \vdash \mathbf{F}$;

(¶) if $\Sigma \vdash \mathbf{F}$, and $\Sigma \vdash (\mathbf{F} \to \mathbf{G})$, then $\Sigma \vdash \mathbf{G}$.

Thus the set of syntactic consequences of Σ is defined inductively; that is, a claim analogous to Theorem 2.1.1 can be proved (just as easily). Looking ahead to their semantics, we can name the three families of formulas just given:

(∗) $(\mathbf{F} \to (\mathbf{G} \to \mathbf{F}))$ is **Affirmation of the Consequent;**

(†) $((\mathbf{F} \to (\mathbf{G} \to \mathbf{H})) \to ((\mathbf{F} \to \mathbf{G}) \to (\mathbf{F} \to \mathbf{H})))$ is **Self-Distribution of Implication;**

(‡) $((\neg\mathbf{F} \to \neg\mathbf{G}) \to (\mathbf{G} \to \mathbf{F}))$ is **Contraposition.**

Every such formula will be called an **Axiom.** The rule that $\Sigma \vdash \mathbf{G}$, if $\Sigma \vdash \mathbf{F}$ and $\Sigma \vdash (\mathbf{F} \to \mathbf{G})$, is the **Rule of Detachment** (or ***Modus Ponens***); in particular, $\mathbf{G}$ can be 'detached' from $(\mathbf{F} \to \mathbf{G})$ by means of $\mathbf{F}$. So the set of syntactic consequences of Σ is the smallest set of formulas that contains the elements of Σ and the Axioms and is *closed* under application of the Rule of Detachment.

{lem:immediate}

Lemma 2.3.1. *If* $\Sigma \vdash \mathbf{F}$, *and* $\Sigma \subseteq \mathrm{T}$, *then* $\mathrm{T} \vdash \mathbf{F}$.

Proof. Immediate.[3] □

Syntactic entailment of a formula is established fundamentally by the presence of the formula at the root of an appropriate *tree;* but the essential information in such a tree can be expressed as a *deduction* or *formal proof.* First, an example of a syntactic entailment:

{lem:FtoF}

Lemma 2.3.2. $\varnothing \vdash (\mathbf{F} \to \mathbf{F})$.

Proof. We have the following sequence of observations:

(1) $\varnothing \vdash (\mathbf{F} \to ((\mathbf{F} \to \mathbf{F}) \to \mathbf{F}))$ [by Affirmation of the Consequent];

(2) $\varnothing \vdash ((\mathbf{F} \to ((\mathbf{F} \to \mathbf{F}) \to \mathbf{F})) \to ((\mathbf{F} \to (\mathbf{F} \to \mathbf{F})) \to (\mathbf{F} \to \mathbf{F})))$ [by Self-Distribution of Implication];

(3) $\varnothing \vdash ((\mathbf{F} \to (\mathbf{F} \to \mathbf{F})) \to (\mathbf{F} \to \mathbf{F}))$ [by Detachment from (2) by (1)];

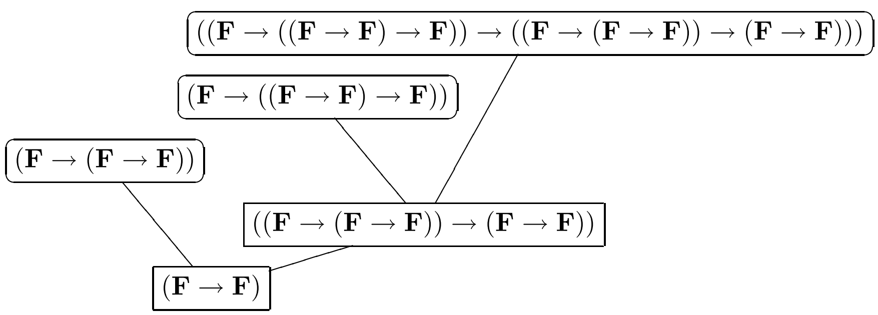
(4) $\varnothing \vdash (\mathbf{F} \to (\mathbf{F} \to \mathbf{F}))$ [by Affirmation of the Consequent];

(5) $\varnothing \vdash (\mathbf{F} \to \mathbf{F})$ [by Detachment from (3) by (4)].

This completes the proof. □

By the last two lemmas, we have $\Sigma \vdash (\mathbf{F} \to \mathbf{F})$ always.

The preceding proof can be written as a tree, thus:



Stripped further of explanatory details, the proof can be written as the following string of length 5 (the entries of the string are themselves strings, namely, formulas):

$$(\mathbf{F} \to ((\mathbf{F} \to \mathbf{F}) \to \mathbf{F})), \tag{2.1}$$

$$((\mathbf{F} \to ((\mathbf{F} \to \mathbf{F}) \to \mathbf{F})) \to ((\mathbf{F} \to (\mathbf{F} \to \mathbf{F})) \to (\mathbf{F} \to \mathbf{F}))), \tag{2.2}$$

$$((\mathbf{F} \to (\mathbf{F} \to \mathbf{F})) \to (\mathbf{F} \to \mathbf{F})), \tag{2.3}$$

$$(\mathbf{F} \to (\mathbf{F} \to \mathbf{F})), \tag{2.4}$$

$$(\mathbf{F} \to \mathbf{F}). \tag{2.5}$$

This string is a *deduction* of $(\mathbf{F} \to \mathbf{F})$ from $\varnothing$ (or any other set of formulas).

[3]For the typographically minded: The letter T in the statement of the theorem is a Greek capital *tau,* in upright font, rather than a Latin *T*, in italic font.

By definition, a **(formal) proof** or **deduction of H from** Σ is a string of formulas, ending with $\mathbf{H}$, such that each entry $\mathbf{G}$ in the string

($*$) is an Axiom, or

(†) is an element of Σ, or

(‡) is preceded (somewhere in the string) by formulas $\mathbf{F}$ and $(\mathbf{F} \to \mathbf{G})$.

{lem:pis-proof}

Lemma 2.3.3. *Every initial segment of a formal proof from a set of formulas is itself a formal proof from that set.*

Proof. Immediate. □

{thm:proves}

Theorem 2.3.4. *A formula* $\mathbf{G}$ *has a formal proof from* Σ *if and only if* $\Sigma \vdash \mathbf{G}$.

Proof. By strong induction on the lengths of formal proofs, we first show that every formula with a formal proof from Σ is a syntactic consequence of Σ. Suppose the claim is true for all formulas with proofs of length less than n, and now $\mathbf{G}$ has a formal proof of length n. If $\mathbf{G}$ is an Axiom or an element of Σ, then $\Sigma \vdash \mathbf{G}$. If, in its proof, $\mathbf{G}$ is preceded by $\mathbf{F}$ and $(\mathbf{F} \to \mathbf{G})$, then, by inductive hypothesis and Lemma 2.3.3, $\Sigma \vdash \mathbf{F}$ and $\Sigma \vdash (\mathbf{F} \to \mathbf{G})$, hence $\Sigma \vdash \mathbf{G}$. Hence $\Sigma \vdash \mathbf{H}$ for all formulas $\mathbf{H}$ that are formally provable from Σ.

We prove the converse by induction on syntactic consequences. Every Axiom, and every element of Σ, is a formal proof of itself from Σ. Suppose $\mathbf{F}$ has formal proof $\mathbf{A}$, and $(\mathbf{F} \to \mathbf{G})$ has formal proof $\mathbf{B}$, from Σ. Then $\mathbf{A}\ \mathbf{B}\ \mathbf{G}$ is a formal proof of $\mathbf{G}$ from Σ. Hence every syntactic consequence of Σ is formally provable from it. □

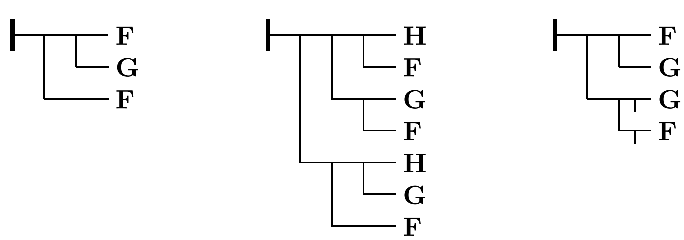
Establishing syntactic entailment by means of the original definition or by formal proof is usually quite tedious. We shall develop short-cuts. First, let us develop a simpler notation for formulas, in the next section.

2.4 Notation

{sect:notation}

We have chosen a signature for our propositional formulas, namely $\{\to, \neg\}$. We have also chosen a 'style' of notation, namely **infix** notation. There are alternatives (as mentioned in Example 2.2.8).

What we are calling ***syntactic consequence*** seems to have its origin in the *Begriffsschrift* [15] of Gottlob Frege, published in 1879. (The title can be rendered as 'ideography' or 'concept writing'). In Frege's work, what we call formulas appear not as strings, but as two-dimensional figures. For example, our three Axioms correspond to Frege's Judgments (1), (2), and—almost—(28); he writes them as follows:



This style of writing formulas never caught on, except in the following sense: To assert a ***judgment*** whose ***content*** is A, Frege writes

$$\vdash\!\!\!-\!\!\!-\!\!\!-\, A$$

The vertical bar here is the ***judgment stroke,*** while the horizontal is merely the ***content stroke.*** Frege's notation appears to be the origin of our own symbol $\vdash$.

I propose to modify our own style of writing parentheses by removing excess parentheses. When this is done, for example, our three Axioms become

$$\begin{gathered}\mathbf{F} \to \mathbf{G} \to \mathbf{F},\\ (\mathbf{F} \to \mathbf{G} \to \mathbf{H}) \to (\mathbf{F} \to \mathbf{G}) \to \mathbf{F} \to \mathbf{H},\\ (\neg\mathbf{F} \to \neg\mathbf{G}) \to \mathbf{G} \to \mathbf{F}.\end{gathered}$$

In this abbreviated system, we can again define formulas inductively, albeit in a more complicated way. The set of these formulas can be called PF'. Every formula in PF' will be a variable, a negation, or an implication. Then:

($*$) V is the set of variables in PF'.

(†) If $\mathbf{F}$ is a variable or a negation in PF', then $\neg\mathbf{F}$ is a negation in PF'.

(‡) If $\mathbf{F}$ is an implication in PF', then $\neg(\mathbf{F})$ is a negation in PF.

(§) If $\mathbf{F}$ is a variable or a negation in PF', and $\mathbf{G}$ is in PF', then $\mathbf{F} \to \mathbf{G}$ is an implication in PF'.

(¶) If $\mathbf{F}$ is an implication in PF', and $\mathbf{G}$ is in PF', then $(\mathbf{F}) \to \mathbf{G}$ is an implication in PF'.

Thus, no formula by itself will be enclosed in parentheses; but an implication must be so enclosed when it is negated or used as the *antecedent* of another implication. It is left to the reader to formulate PF' as an intersection of sets, so that the analogue of Theorem 2.1.1 follows. Written as a string of elements of PF', the deduction given earlier as Lines (2.1–2.5) becomes

$$\begin{gathered}\mathbf{F} \to (\mathbf{F} \to \mathbf{F}) \to \mathbf{F},\\ (\mathbf{F} \to (\mathbf{F} \to \mathbf{F}) \to \mathbf{F}) \to (\mathbf{F} \to \mathbf{F} \to \mathbf{F}) \to \mathbf{F} \to \mathbf{F},\\ (\mathbf{F} \to \mathbf{F} \to \mathbf{F}) \to \mathbf{F} \to \mathbf{F},\\ \mathbf{F} \to \mathbf{F} \to \mathbf{F},\\ \mathbf{F} \to \mathbf{F}.\end{gathered}$$

It is also left to the reader to formulate and prove an analogue of Theorem 2.2.5, so that the following can then be proved: {thm:simpler}

Theorem 2.4.1. *There is a unique bijection* $\mathbf{F} \mapsto \overline{\mathbf{F}}$ *from* PF *to* PF' *such that*

($*$) $\overline{\mathbf{P}} = \mathbf{P}$ *for all variables* $\mathbf{P}$*;*

(†) $\overline{\neg\mathbf{F}} = \begin{cases}\neg\overline{\mathbf{F}}, & \textit{if } \mathbf{F} \textit{ is a variable or negation;}\\ \neg(\overline{\mathbf{F}}), & \textit{if } \mathbf{F} \textit{ is an implication;}\end{cases}$

(‡) $\overline{(\mathbf{F} \to \mathbf{G})} = \begin{cases}\overline{\mathbf{F}} \to \overline{\mathbf{G}}, & \textit{if } \mathbf{F} \textit{ is a variable or negation;}\\ (\overline{\mathbf{F}}) \to \overline{\mathbf{G}}, & \textit{if } \mathbf{F} \textit{ is an implication.}\end{cases}$

The inverse of this map is a function $\mathbf{F} \mapsto \overline{\mathbf{F}}$ *from* PF' *to* PF *such that*

(∗) $\overline{\mathbf{P}} = \mathbf{P}$ *for all variables* $\mathbf{P}$*;*

(†) $\overline{\neg\mathbf{F}} = \neg\overline{\mathbf{F}}$*;*

(‡) $\overline{\neg(\mathbf{F})} = \neg\overline{\mathbf{F}}$*;*

(§) $\overline{\mathbf{F} \to \mathbf{G}} = (\overline{\mathbf{F}} \to \overline{\mathbf{G}})$*;*

(¶) $\overline{(\mathbf{F}) \to \mathbf{G}} = (\overline{\mathbf{F}} \to \overline{\mathbf{G}})$.

Proof. In the notation of Porism 2.2.7, let A be the set of strings of the symbols in $V \cup \{\to, \neg, (,)\}$, let h_0 be the inclusion of V in A, and let

$$h_1(\mathbf{F}, \mathbf{A}) = \begin{cases} \neg\mathbf{A}, & \text{if } \mathbf{F} \text{ is a variable or negation;} \\ \neg(\mathbf{A}), & \text{if } \mathbf{F} \text{ is an implication;} \end{cases}$$

$$h_2((\mathbf{F}, \mathbf{A}), (\mathbf{G}, \mathbf{B})) = \begin{cases} \mathbf{A} \to \mathbf{B}, & \text{if } \mathbf{F} \text{ is a variable or negation;} \\ (\mathbf{A}) \to \mathbf{B}, & \text{if } \mathbf{F} \text{ is an implication.} \end{cases}$$

Then the function from PF to PF′ exists uniquely as desired, by the Porism. This function is bijective, with inverse as claimed (details are left to the reader). □

Exercises

(1) Give a precise definition of PF′. (One way to proceed might be as follows: Let S be as defined on p. 12, and let $\mathcal{U}$ comprise the subsets N of $S \times 2$ such that

  (∗) $(\mathbf{P}, 0) \in N$;

  (†) if $(\mathbf{A}, 0) \in N$, then $(\neg\mathbf{A}, 0) \in N$;

  (‡) if $(\mathbf{A}, 1) \in N$, then $(\neg(\mathbf{A}), 0) \in N$;

  (§) if $(\mathbf{F}, 0) \in N$, and $(\mathbf{G}, e) \in N$, then $(\mathbf{F} \to \mathbf{G}, 1) \in N$;

  (¶) if $(\mathbf{F}, 1) \in N$, and $(\mathbf{G}, e) \in N$, then $((\mathbf{F}) \to \mathbf{G}, 1) \in N$.

  Now extract PF′.)

(2) Complete the proof of Theorem 2.4.1.

2.5 Theorems

Henceforth, let us write propositional formulas in the style of PF′ established in the previous section.

A syntactical consequence of $\varnothing$ can be called a **Theorem.** To express that a formula is a Theorem, we can write

$$\vdash$$

in front of it, instead of $\varnothing \vdash$. Thus, by Lemma 2.3.2, we have

$$\vdash \mathbf{F} \to \mathbf{F}.$$

As promised at the end of § 2.3, we now start to develop more efficient methods of establishing Theorems and other instances of syntactic entailment.

{thm:deduction}

Theorem 2.5.1 (Deduction)**.** $\Sigma \vdash \mathbf{F} \to \mathbf{G} \iff \Sigma \cup \{\mathbf{F}\} \vdash \mathbf{G}$.

Proof. The easy direction is ($\Rightarrow$), which is left to the reader. The other direction uses strong induction on the lengths of formal proofs.

Suppose $\Sigma \cup \{\mathbf{F}\} \vdash \mathbf{G}$, so that, by Theorem 2.3.4, there is a formal proof of $\mathbf{G}$ from $\Sigma \cup \{\mathbf{F}\}$. With respect to this proof, there are three possibilities for $\mathbf{G}$:

If $\mathbf{G}$ is an Axiom, or is one of the formulas in Σ, then $\Sigma \vdash \mathbf{G}$; but $\mathbf{G} \to \mathbf{F} \to \mathbf{G}$ is an Axiom in any case, so also $\Sigma \vdash \mathbf{G} \to \mathbf{F} \to \mathbf{G}$; hence $\Sigma \vdash \mathbf{F} \to \mathbf{G}$ by Detachment.

If $\mathbf{G}$ is $\mathbf{F}$, then $\vdash \mathbf{F} \to \mathbf{G}$ by Lemma 2.3.2, so $\Sigma \vdash \mathbf{F} \to \mathbf{G}$ by Lemma 2.3.1.

Finally, suppose that, in its formal proof, $\mathbf{G}$ is preceded by $\mathbf{H}$ and $\mathbf{H} \to \mathbf{G}$. If $\Sigma \vdash \mathbf{F} \to \mathbf{H}$ and $\Sigma \vdash \mathbf{F} \to \mathbf{H} \to \mathbf{G}$, then by Self-Distributivity of Implication, and the Rule of Detachment, $\Sigma \vdash \mathbf{F} \to \mathbf{G}$, and we are done. Thus, if Σ does *not* syntactically entail $\mathbf{F} \to \mathbf{G}$, then it also fails to entail $\mathbf{F} \to \mathbf{H} \to \mathbf{G}$ or $\mathbf{F} \to \mathbf{H}$. But each of the formulas $\mathbf{H} \to \mathbf{G}$ and $\mathbf{H}$ has a shorter proof from $\Sigma \cup \{\mathbf{F}\}$ than $\mathbf{G}$ does, by Lemma 2.3.3. By strong induction, we are done. □

{lem:several}

Lemma 2.5.2. *The following formulas are Theorems:*

(∗) $\neg\mathbf{G} \to \mathbf{G} \to \mathbf{F}$; {item:contrad}

(†) $\neg\neg\mathbf{F} \to \mathbf{F}$; {item:double-neg}

(‡) $\mathbf{F} \to \neg\neg\mathbf{F}$; {item:other-way}

(§) $(\mathbf{F} \to \mathbf{G}) \to \neg\mathbf{G} \to \neg\mathbf{F}$; {item:other-contrap}

(¶) $\mathbf{F} \to \neg\mathbf{G} \to \neg(\mathbf{F} \to \mathbf{G})$. {item:imp}

(‖) $(\mathbf{F} \to \mathbf{G}) \to (\neg\mathbf{F} \to \mathbf{G}) \to \mathbf{G}$. {item:two-cases}

Proof. The following is a formal proof from $\neg\mathbf{G}$:

$$\neg\mathbf{G},\ \neg\mathbf{G} \to \neg\mathbf{F} \to \neg\mathbf{G},\ \neg\mathbf{F} \to \neg\mathbf{G},\ (\neg\mathbf{F} \to \neg\mathbf{G}) \to \mathbf{G} \to \mathbf{F},\ \mathbf{G} \to \mathbf{F}.$$

So $\neg\mathbf{G} \vdash \mathbf{G} \to \mathbf{F}$. By the Deduction Theorem, $\vdash \neg\mathbf{G} \to \mathbf{G} \to \mathbf{F}$.

As a special case of what we have just shown, we have $\neg\neg\mathbf{F} \vdash \neg\mathbf{F} \to \neg\neg\neg\mathbf{F}$. From Contraposition, we get $\neg\neg\mathbf{F} \vdash \neg\neg\mathbf{F} \to \mathbf{F}$; then, by both directions of the Deduction Theorem, we get $\neg\neg\mathbf{F} \vdash \mathbf{F}$, then $\vdash \neg\neg\mathbf{F} \to \mathbf{F}$.

The remaining parts are left to the reader. □

Exercises

(1) Prove the easy direction of the Deduction Theorem, 2.5.1, and supply missing details of the proof of the other direction.

(2) Prove the remainder of Lemma 2.5.2.

2.6 Logical entailment

{sect:logical}

A **truth-assignment** is a function from V to 2. Let ε be such a function. It determines a substitution $\mathbf{F} \mapsto \mathbf{F}(\varepsilon)$ as in § 2.1, although 0 and 1 are not formulas in PF. By recursion, the truth-assignment ε uniquely determines a function $\mathbf{F} \mapsto \widehat{\mathbf{F}}(\varepsilon)$ as follows (where $+$ and $\cdot$ are as in $\mathbb{F}_2$):

$$\widehat{\mathbf{F}}(\varepsilon) = \begin{cases} \varepsilon(\mathbf{P}), & \text{if } \mathbf{F} \text{ is } \mathbf{P}; \\ 1 + \widehat{\mathbf{G}}(\varepsilon), & \text{if } \mathbf{F} \text{ is } \neg\mathbf{G}; \\ 1 + \widehat{\mathbf{G}}(\varepsilon) + \widehat{\mathbf{G}}(\varepsilon) \cdot \widehat{\mathbf{H}}(\varepsilon), & \text{if } \mathbf{F} \text{ is } \mathbf{G} \to \mathbf{H}. \end{cases}$$

A formula $\mathbf{F}$ can be called n**-ary** if each variable that is an entry in $\mathbf{F}$ belongs to the set $\{P_k : k < n\}$. In this case, $\widehat{\mathbf{F}}(\varepsilon)$ depends only on the n-tuple $(\varepsilon(P_0), \dots, \varepsilon(P_{n-1}))$. (This is obvious, but can be confirmed by induction on formulas.) Denoting this n-tuple more briefly by $\vec{e}$, we may write

$$\widehat{\mathbf{F}}(\vec{e})$$

instead of $\widehat{\mathbf{F}}(\varepsilon)$. We may then refer to $\vec{e}$ as an n**-ary truth-assignment.** The number $\widehat{\mathbf{F}}(\vec{e})$ is the **truth-value** of $\mathbf{F}$ with respect to ε or $\vec{e}$. In particular, $\mathbf{F}$ is **true in** ε (or $\vec{e}$) if $\widehat{\mathbf{F}}(\varepsilon) = 1$; otherwise, $\mathbf{F}$ is **false in** ε.

The truth-values of $\mathbf{F}$ with respect to all truth-assignments can be given in a **truth-table** with 2^n rows.

{thm:associativity} The following may seem obvious, once it is understood:

Theorem 2.6.1 (Associativity)**.** *Suppose* $\mathbf{F}$ *is an* n*-ary formula, and* $\mathbf{H}$ *is a formula* $\mathbf{F}(\mathbf{G}_0, \dots, \mathbf{G}_{n-1})$*, and* $\vec{e}$ *and* $\vec{f}$ *are truth-assignments (of appropriate arity) such that*

$$\widehat{\mathbf{G}}_k(\vec{e}) = f_k$$

for each k *in* n*. Then*

$$\widehat{\mathbf{F}}(\vec{f}) = \widehat{\mathbf{H}}(\vec{e}).$$

Proof. We use induction on $\mathbf{F}$. If $\mathbf{F}$ is a variable, then it is P_k for some k in n, so $\mathbf{H}$ is $\mathbf{G}_k$, and

$$\widehat{\mathbf{H}}(\vec{e}) = \widehat{\mathbf{G}}_k(\vec{e}) = f_k = \widehat{P}_k(\vec{f}) = \widehat{\mathbf{F}}(\vec{f}).$$

Suppose the claim is true when $\mathbf{F}$ is $\mathbf{F}_0$ or $\mathbf{F}_1$. If now $\mathbf{F}$ is $\neg\mathbf{F}_0$, then $\mathbf{H}$ is $\neg\mathbf{F}_0(\mathbf{G}_0, \dots, \mathbf{G}_{n-1})$, which we can write as $\neg\mathbf{H}_0$, so that

$$\begin{aligned} \widehat{\mathbf{H}}(e) &= 1 + \widehat{\mathbf{H}}_0(\vec{e}) \\ &= 1 + \widehat{\mathbf{F}}_0(\vec{f}) \qquad \text{[by inductive hypothesis]} \\ &= \widehat{\mathbf{F}}(\vec{f}). \end{aligned}$$

The remaining case, where $\mathbf{F}$ is $(\mathbf{F}_0 \to \mathbf{F}_1)$, is left to the reader. □

In the present context, we can think of truth as a *relation* from 2^V to PF, namely the relation

$$\{(\xi, X) \in 2^V \times \mathrm{PF} : \widehat{X}(\xi) = 1\}.$$

We may denote this relation by

$$\models .$$

Hence, instead of $\widehat{\mathbf{F}}(\varepsilon) = 1$, we may write

$$\varepsilon \models \mathbf{F}. \tag{2.6}$$

The complement of the truth-relation can be denoted

$$\nvDash .$$

Hence we can express a fundamental fact as follows: {lem:tf}

Lemma 2.6.2. *For all truth-assignments ε and formulas $\mathbf{F}$, we have*

$$\varepsilon \models \mathbf{F} \iff \varepsilon \nvDash \neg\mathbf{F}; \tag{2.7}$$

likewise,

$$\varepsilon \nvDash \mathbf{F} \iff \varepsilon \models \neg\mathbf{F}. \tag{2.8}$$

Proof. Suppose $e \in 2$. Then $e = 1 \iff e \neq 0 \iff e + 1 = 0$. Let $\mathbf{G}$ be $\neg\mathbf{F}$. Then

$$\begin{aligned} \varepsilon \models \mathbf{F} &\iff \widehat{\mathbf{F}}(\varepsilon) = 1 \\ &\iff 1 + \widehat{\mathbf{F}}(\varepsilon) = 0 \\ &\iff \widehat{\mathbf{G}}(\varepsilon) = 0 \\ &\iff \widehat{\mathbf{G}}(\varepsilon) \neq 1 \\ &\iff \varepsilon \nvDash \mathbf{G} \\ &\iff \varepsilon \nvDash \neg\mathbf{F}. \end{aligned}$$

The other equivalence follows immediately. □

From the truth-relation, we obtain three new functions, as follows.

(∗) A **model** of a set of formulas is a truth-assignment in which every element of the set is true. If Σ is a set of formulas, let

$$\mathrm{Mod}(\Sigma)$$

be the set of its models. This is the set

$$\bigcap_{X \in \Sigma} \{\xi \in 2^V : \xi \models X\}.$$

We now have a function $\Xi \mapsto \mathrm{Mod}(\Xi)$ from $\mathcal{P}(\mathrm{PF})$ to $\mathcal{P}(2^V)$.

(†) The **theory** of a set of truth-assignments is the set of formulas that are true in all of the truth-assignments. If A is a set of truth-assignments, let

$$\mathrm{Th}(A)$$

be its theory. This is the set

$$\bigcap_{\xi \in A} \{X \in \mathrm{PF} : \xi \models X\}.$$

So we have a function $\Xi \mapsto \mathrm{Th}(\Xi)$ from $\mathcal{P}(2^V)$ to $\mathcal{P}(\mathrm{PF})$.

(‡) The **logical consequences** of a set of formulas are the formulas that are true in every model of the original set. The logical consequences of Σ compose a set

$$\mathrm{Con}(\Sigma).$$

This is the set $\bigcap_{\xi\in\mathrm{Mod}(\Sigma)}\{X\in\mathrm{PF}:\xi\vDash X\}$, which is

$$\mathrm{Th}(\mathrm{Mod}(\Sigma)).$$

So we have a singulary operation $\Xi\mapsto\mathrm{Con}(\Xi)$ on $\mathcal{P}(\mathrm{PF})$.

We shall not study the operation $\Xi\mapsto\mathrm{Mod}(\mathrm{Th}(\Xi))$ on $\mathcal{P}(2^V)$.

If T is a set of formulas that is the theory of *some* set of truth-assignments, then T can be called a **theory,** simply.

If $\mathbf{F}$ is a logical consequence of Σ, we may say also that Σ **logically entails F**. So we have several ways of saying the same thing:

(∗) $\mathbf{F}$ is a logical consequence of Σ;

(†) Σ logically entails $\mathbf{F}$;

(‡) $\mathbf{F}\in\mathrm{Con}(\Sigma)$.

A fourth way is $\Sigma\vDash\mathbf{F}$; but I shall avoid this notation, lest it be confused with the notation introduced on Line (2.6), which has a different meaning.

The logical consequences of $\varnothing$ are called **tautologies;** these are the formulas that are true in *every* truth-assignment.

Note well that the definition of logical entailment is not inductive: there is (at the moment) no obvious way to prove by induction that a given set of formulas contains all logical consequences of Σ (or even all tautologies).

{lem:reversing}

Lemma 2.6.3. *The operations* $\Xi\mapsto\mathrm{Mod}(\Xi)$ *and* $\Xi\mapsto\mathrm{Th}(\Xi)$ *are inclusion-reversing, that is,*

(∗) $\Sigma\subseteq\mathrm{T}\implies\mathrm{Mod}(\mathrm{T})\subseteq\mathrm{Mod}(\Sigma)$, *and*

(†) $A\subseteq B\implies\mathrm{Th}(B)\subseteq\mathrm{Th}(A)$,

for all sets Σ *and* T *of formulas, and all sets* A *and* B *of truth-assignments.*

Proof. This is a purely set-theoretic fact, as the reader should check. □

{thm:closure}

Theorem 2.6.4. *Let* Σ *and* T *be subsets of* PF.

(∗) $\Sigma\subseteq\mathrm{Con}(\Sigma)$.

(†) *If* $\Sigma\subseteq\mathrm{T}$, *then* $\mathrm{Con}(\Sigma)\subseteq\mathrm{Con}(\mathrm{T})$.

(‡) $\mathrm{Con}(\mathrm{Con}(\Sigma))=\mathrm{Con}(\Sigma)$.

Proof. In a model of Σ, every element of Σ is true by definition. This proves the first claim.

For the second claim, use Lemma 2.6.3 twice.

For the last claim, we have $\mathrm{Con}(\Sigma)\subseteq\mathrm{Con}(\mathrm{Con}(\Sigma))$ by the first claim. Suppose now $\mathbf{F}\in\mathrm{Con}(\mathrm{Con}(\Sigma))$, and ε is a model of Σ. Then ε is a model of $\mathrm{Con}(\Sigma)$, so $\varepsilon\vDash\mathbf{F}$. Thus $\mathbf{F}\in\mathrm{Con}(\Sigma)$. □

Exercises

(1) Complete the proof of Theorem 2.6.1.

(2) Use Lemma 2.6.2 to show $\varepsilon \vDash \mathbf{F} \iff \varepsilon \vDash \neg\neg\mathbf{F}$.

(3) Prove Lemma 2.6.3.

(4) Prove that Σ is a theory if and only if $\mathrm{Con}(\Sigma) = \Sigma$.

(5) Can you find a formula $\mathbf{F}$ such that $\mathrm{Con}(\{\mathbf{F}\}) = \mathrm{PF}$?

(6) Can you find a formula $\mathbf{G}$ such that $\mathrm{Con}(\{\mathbf{G}\}) = \varnothing$?

(7) Suppose $\mathbf{H} \in \mathrm{Con}(\{\mathbf{F} \to \mathbf{G}\})$. Does it follow that $\mathbf{H}$ is a logical consequence of $\neg\mathbf{F}$ or of $\mathbf{G}$?

(8) Suppose $\mathbf{H}$ logically entails either $\neg\mathbf{F}$ or $\mathbf{G}$; does it entail $\mathbf{F} \to \mathbf{G}$?

(9) If $\mathbf{H} \in \mathrm{Con}(\{\neg\mathbf{F}\}) \cup \mathrm{Con}(\{\mathbf{G}\})$, must $\mathbf{H} \in \mathrm{Con}(\{\mathbf{F} \to \mathbf{G}\})$?

(10) If $\mathbf{H} \in \mathrm{Con}(\{\neg\mathbf{F}\}) \cap \mathrm{Con}(\{\mathbf{G}\})$, must $\mathbf{H} \in \mathrm{Con}(\{\mathbf{F} \to \mathbf{G}\})$? {exercise:caps}

(11) Show that $\mathrm{Mod}(\bigcup_{i\in I} \Sigma_i) = \bigcap_{i\in I} \mathrm{Mod}(\Sigma_i)$. {exercise:cups}

(12) Show that $\mathrm{Mod}(\{\mathbf{F}\}) \cup \mathrm{Mod}(\{\mathbf{G}\}) = \mathrm{Mod}(\{\neg\mathbf{F} \to \mathbf{G}\})$.

(13) Show that $\mathrm{Mod}(\{\mathbf{F}\})^{\mathrm{c}} = \mathrm{Mod}(\{\neg\mathbf{F}\})$.

2.7 Compactness

A set of formulas with a model can be called **satisfiable.** {lem:sat}

Lemma 2.7.1. *Σ logically entails $\mathbf{F}$ if and only if $\Sigma \cup \{\neg\mathbf{F}\}$ is not satisfiable.*

Proof. Suppose Σ does *not* logically entail $\mathbf{F}$. Then Σ has a model ε in which $\mathbf{F}$ is false. Hence $\varepsilon \vDash \neg\mathbf{F}$ by Lemma 2.6.2, (2.8), so ε is a model of $\Sigma \cup \{\neg\mathbf{F}\}$.

Suppose conversely that $\Sigma \cup \{\neg\mathbf{F}\}$ has a model. Then $\mathbf{F}$ is false in this model, again by Lemma 2.6.2, (2.8), so $\mathbf{F}$ is not a logical consequence of Σ. □

A set of formulas whose every *finite* subset has a model can be called **finitely satisfiable.**

Lemma 2.7.2. *If Σ is finitely satisfiable, then the same is true of $\Sigma \cup \{\mathbf{F}\}$ or $\Sigma \cup \{\neg\mathbf{F}\}$.*

Proof. Suppose neither $\Sigma \cup \{\mathbf{F}\}$ nor $\Sigma \cup \{\neg\mathbf{F}\}$ is finitely satisfiable. Then Σ has finite subsets T_0 and T_1 such that neither $\mathrm{T}_0 \cup \{\mathbf{F}\}$ nor $\mathrm{T}_1 \cup \{\neg\mathbf{F}\}$ is satisfiable. Then also $\mathrm{T}_0 \cup \{\neg\neg\mathbf{F}\}$ is not satisfiable (why?); hence $\neg\mathbf{F} \in \mathrm{Con}(\mathrm{T}_0)$ and $\mathbf{F} \in \mathrm{Con}(\mathrm{T}_1)$, by Lemma 2.7.1. Therefore, every model ε of $\mathrm{T}_0 \cup \mathrm{T}_1$ is a model of T_0 and T_1 (by Lemma 2.6.3), hence ε is a model of $\{\mathbf{F}, \neg\mathbf{F}\}$. There can be no such models ε (why not?); so $\mathrm{T}_0 \cup \mathrm{T}_1$ is not satisfiable. But this is a finite subset of Σ; hence Σ is not finitely satisfiable. □

Theorem 2.7.3 (Compactness)**.** *Every finitely satisfiable set of formulas is satisfiable.*

Proof. Let Σ be finitely satisfiable. By strong recursion, we first define a function $n \mapsto \mathbf{F}_n$ from ω into PF. Suppose $\{\mathbf{F}_k : k < n\}$ has been defined. We then let $\mathbf{F}_n$ be P_n, if $\Sigma \cup \{\mathbf{F}_k : k < n\} \cup \{P_n\}$ is finitely satisfiable; otherwise, $\mathbf{F}_n$ is $\neg P_n$. This completes the recursive definition.

We now observe by induction that every set $\Sigma \cup \{\mathbf{F}_k : k < n\}$ is finitely satisfiable. Indeed, it is true by assumption when $n = 0$; and if it is true when $n = m$, then it is true when $n = m + 1$, by the last lemma and the definition of the $\mathbf{F}_k$.

Every finite subset of $\Sigma \cup \{\mathbf{F}_k : k \in \omega\}$ is a finite subset of $\Sigma \cup \{\mathbf{F}_k : k < n\}$ for some n. We have just seen that $\Sigma \cup \{\mathbf{F}_k : k < n\}$ is finitely satisfiable; therefore the whole set $\Sigma \cup \{\mathbf{F}_k : k \in \omega\}$ is finitely satisfiable.

Now let ε be the truth-assignment given by

$$\varepsilon(P_k) = \begin{cases} 1, & \text{if } \mathbf{F}_k = P_k; \\ 0, & \text{if } \mathbf{F}_k = \neg P_k. \end{cases} \tag{2.9}$$

This is a model of Σ. Indeed, suppose $\mathbf{G} \in \Sigma$. Then $\mathbf{G}$ is n-ary for some n. The finite set $\{\mathbf{G}\} \cup \{\mathbf{F}_k : k < n\}$ has a model ζ. In particular, ζ must agree with ε on $\{P_k : k < n\}$ (why?); so $\varepsilon \vDash \mathbf{G}$. □

There are sets Σ of formulas such that *every* finite subset of Σ has a model that is not a model of Σ itself.

{example:compactness}

Example 2.7.4. Let Σ_n comprise the formulas

$$P_0 \to P_1 \to \cdots \to P_k,$$

where $k < n$. So Σ_0 is empty, and $\Sigma_1 = \{P_0\}$, and we have a chain

$$\Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \subseteq \cdots .$$

Let $\Sigma = \bigcup_{n\in\omega} \Sigma_n$. Then every finite subset of Σ is a subset of some Σ_n. Let ε_n be the truth-assignment such that

$$\varepsilon_n(P_k) = 1 \iff k < n.$$

Then ε_n is a model of Σ_n, but not of Σ_{n+1} (why?), hence not of Σ. •

If a set A is a *finite* subset of a set B, we may denote this by

$$A \subseteq_{\mathrm{f}} B.$$

{cor:finitary}

Now one consequence of the Compactness Theorem can be expressed as follows:

Corollary 2.7.5. $\mathrm{Con}(\Sigma) = \displaystyle\bigcup_{\Xi \subseteq_{\mathrm{f}} \Sigma} \mathrm{Con}(\Xi)$.

Proof. By Theorem 2.6.4, it is enough to show that

$$\mathrm{Con}(\Sigma) \subseteq \bigcup_{\Xi \subseteq_{\mathrm{f}} \Sigma} \mathrm{Con}(\Xi).$$

Suppose $\mathbf{F}$ is *not* a member of the union. Then, for each finite subset T of Σ, the set $\mathrm{Con}(\mathrm{T})$ does not contain $\mathbf{F}$, and so the set $\mathrm{T} \cup \{\neg\mathbf{F}\}$ is satisfiable, by Lemma 2.7.1. This means $\Sigma \cup \{\neg\mathbf{F}\}$ is finitely satisfiable; so it is satisfiable, by the Compactness Theorem. Therefore $\neg\mathbf{F} \notin \mathrm{Con}(\Sigma)$, again by Lemma 2.7.1. □

Exercises

(1) If $\mathrm{T} \cup \{\mathbf{F}\}$ is not satisfiable, why is $\mathrm{T} \cup \{\neg\neg\mathbf{F}\}$ not satisfiable?

(2) Why has the set $\{\mathbf{F}, \neg\mathbf{F}\}$ no models?

(3) In the proof of the Compactness Theorem, why does ζ agree with ε on $\{P_k : k < n\}$?

(4) In Example 2.7.4, prove by induction that $\varepsilon_n \in \mathrm{Mod}(\Sigma_n) \smallsetminus \mathrm{Mod}(\Sigma_{n+1})$.

(5) Suppose I is a set, and there is a function $i \mapsto \mathbf{F}_i$ from I into PF, such that
$$\bigcup_{i \in I} \mathrm{Mod}(\{\mathbf{F}_i\}) = 2^V.$$
Prove that I has a finite subset J such that $\bigcup_{i \in J} \mathrm{Mod}(\{\mathbf{F}_i\}) = 2^V$.

2.8 Generalizations

The concepts of the previous section are instances of more general concepts.

For an arbitrary set Ω, a singulary operation $X \mapsto \mathrm{cl}(X)$ on $\mathcal{P}(\Omega)$ is called a **closure-operator** on Ω if it is:

($*$) **increasing** (that is, $A \subseteq \mathrm{cl}(A)$ for all subsets A of Ω);

(†) **monotone** (that is, $\mathrm{cl}(A) \subseteq \mathrm{cl}(B)$ whenever $A \subseteq B \subseteq \Omega$); and

(‡) **idempotent** (that is, $\mathrm{cl}(\mathrm{cl}(A)) = \mathrm{cl}(A)$ for all A in $\mathcal{P}(\Omega)$).

The terminology is potentially confusing: a closure-operator on Ω is an operation on $\mathcal{P}(\Omega)$ (not on Ω).

The closure-operator $X \mapsto \mathrm{cl}(X)$ is called **finitary** if

$$\mathrm{cl}(A) = \bigcup_{X \subseteq_{\mathrm{f}} A} \mathrm{cl}(X)$$

for all A in $\mathcal{P}(\Omega)$.

Examples 2.8.1.

(1) If Ω is a topological space, the function taking a subset of Ω to its topological closure is a closure-operator on Ω (usually not finitary).

(2) If G is a group, the function $X \mapsto \langle X \rangle$ taking a subset of G to the group that it generates is a finitary closure-operator on G.

(3) On PF, the function $\Xi \mapsto \mathrm{Con}(\Xi)$ is a closure-operator, by Theorem 2.6.4; it is finitary, by Corollary 2.7.5.

(4) On any set, the identity-function $X \mapsto X$ is trivially a finitary closure-operator. •

Closure-operators can arise from a *Galois correspondence* between two sets. Suppose A and B are sets, and R is a relation from A to B. If $C \subseteq A$, and $D \subseteq B$, let

$$C' = \bigcap_{x \in C} \{y \in B : x \mathrel{R} y\} = \{y \in B : \forall x\ (x \in C \to x \mathrel{R} y)\}$$
$$D' = \bigcap_{y \in D} \{x \in A : x \mathrel{R} y\} = \{x \in A : \forall y\ (y \in D \to x \mathrel{R} y)\}.$$

So we have functions $X \mapsto X'$ from $\mathcal{P}(A)$ to $\mathcal{P}(B)$, and from $\mathcal{P}(B)$ to $\mathcal{P}(A)$. These functions are inclusion-reversing; so the functions $X \mapsto X''$ are inclusion-preserving (monotone). Moreover,

$$\begin{aligned} C'' &= \{x \in A : \forall y\ (y \in C' \to x \mathrel{R} y)\} \\ &= \{x \in A : \forall y\ (\forall z\ (z \in C \to z \mathrel{R} y) \to x \mathrel{R} y)\}, \end{aligned}$$

so $C \subseteq C''$; similarly, $D \subseteq D''$. Replacing C with D', we get $D' \subseteq D'''$; but since $D \subseteq D''$, we get also $D''' \subseteq D'$; thus, $D' = D'''$. Similarly, $C' = C'''$. Therefore the functions $X \mapsto X''$ are closure-operators on A and B. Also, the functions $X \mapsto X'$ are bijections—each the inverse of the other—between $\{X' : X \subseteq B\}$ and $\{X' : X \subseteq A\}$; the existence of such functions is a **Galois correspondence.**

Exercises 11 and 12 in § 2.6 show that the sets $\mathrm{Mod}(\Sigma)$ are the **closed** sets in a **topology** on 2^V. Then the Compactness Theorem can be understood as the topological statement that this topology is compact.

2.9 Completeness

An arbitrary singulary operation $\Xi \mapsto \Pi(\Xi)$ on $\mathcal{P}(\mathrm{PF})$ can be called

(∗) **sound,** if always $\Pi(\Sigma) \subseteq \mathrm{Con}(\Sigma)$;

(†) **complete,** if always $\mathrm{Con}(\Sigma) \subseteq \Pi(\Sigma)$.

We shall show that logical entailment is the same relation as syntactic entailment; that is, the operation

$$\Xi \longmapsto \{X \in \mathrm{PF} : \Xi \vdash X\}$$

on $\mathcal{P}(\mathrm{PF})$ is sound and complete.

Theorem 2.9.1 (Soundness)**.** *If* $\Sigma \vdash \mathbf{F}$*, then* $\mathbf{F} \in \mathrm{Con}(\Sigma)$.

Proof. We use induction on the set of syntactic consequences of Σ to show that it is a subset of $\mathrm{Con}(\Sigma)$. All axioms are tautologies; hence they are logical consequences of Σ by Theorem 2.6.4. By the same theorem, all elements of Σ are logical consequences of Σ. Finally, suppose $\mathbf{F}$ and $\mathbf{F} \to \mathbf{G}$ are logical consequences of Σ, and ε is a model of Σ. Then $\widehat{\mathbf{F}}(\varepsilon) = 1$. Also, writing $\mathbf{H}$ for $\mathbf{F} \to \mathbf{G}$, we have

$$1 = \widehat{\mathbf{H}}(\varepsilon) = 1 + \widehat{\mathbf{F}}(\varepsilon) + \widehat{\mathbf{F}}(\varepsilon) \cdot \widehat{\mathbf{G}}(\varepsilon) = 1 + 1 + 1 \cdot \widehat{\mathbf{G}}(\varepsilon) = \widehat{\mathbf{G}}(\varepsilon),$$

so $\varepsilon \vDash \mathbf{G}$. This completes the induction and the proof. □

Proving completeness will take more work.[4]

{lem:eval}

Lemma 2.9.2. *Suppose $\vec{e}$ is an n-ary truth-assignment, and suppose Σ is a set of formulas that, for each k in n, contains* $\begin{cases} P_k, & \text{if } e_k = 1; \\ \neg P_k, & \text{if } e_k = 0. \end{cases}$ *Then*

$$\Sigma \vdash \mathbf{F}'$$

for all n-ary formulas $\mathbf{F}$, where $\mathbf{F}'$ is the formula $\begin{cases} \mathbf{F}, & \text{if } \widehat{\mathbf{F}}(\vec{e}) = 1; \\ \neg\mathbf{F}, & \text{if } \widehat{\mathbf{F}}(\vec{e}) = 0. \end{cases}$

Proof. When $k < n$, let $\mathbf{P}'_k = \begin{cases} P_k, & \text{if } e_k = 1; \\ \neg P_k, & \text{if } e_k = 0. \end{cases}$ Suppose $\{\mathbf{P}'_k : k < n\} \subseteq \Sigma$. Let $\mathbf{F}'$ be defined as in the statement of the theorem. We proceed by induction on n-ary formulas.

If $\mathbf{F}$ is a variable P_k, where $k < n$, then $\mathbf{F}'$ is $\mathbf{P}'_k$, which is in Σ, so $\Sigma \vdash \mathbf{F}'$.

Suppose $\Sigma \vdash \mathbf{G}'$, and $\mathbf{F}$ is $\neg\mathbf{G}$. There are two cases to consider.

(∗) If $\widehat{\mathbf{F}}(\vec{e}) = 1$, then $\widehat{\mathbf{G}}(\vec{e}) = 0$, so $\mathbf{F}'$ is $\mathbf{F}$, but $\mathbf{G}'$ is $\neg\mathbf{G}$, which is $\mathbf{F}$, that is, $\mathbf{F}'$.

(†) If $\widehat{\mathbf{F}}(\vec{e}) = 0$, then $\widehat{\mathbf{G}}(\vec{e}) = 1$, so $\mathbf{G}'$ is $\mathbf{G}$, but $\mathbf{F}'$ is $\neg\mathbf{F}$, which is $\neg\neg\mathbf{G}$, that is, $\neg\neg\mathbf{G}'$.

In either case, we have $\vdash \mathbf{G}' \to \mathbf{F}'$, by Lemmas 2.3.2 and 2.5.2; hence $\Sigma \vdash \mathbf{F}'$ by inductive hypothesis and Detachment.

Suppose finally that $\Sigma \vdash \mathbf{G}'$ and $\Sigma \vdash \mathbf{H}'$, and $\mathbf{F}$ is $\mathbf{G} \to \mathbf{H}$. There are three cases to consider:

(∗) $\widehat{\mathbf{G}}(\vec{e}) = 0$;

(†) $\widehat{\mathbf{H}}(\vec{e}) = 1$;

(‡) $\widehat{\mathbf{G}}(\vec{e}) = 1$ and $\widehat{\mathbf{H}}(\varepsilon) = 0$.

Details are left to the reader. This completes the proof. □

Theorem 2.9.3 (Completeness)**.** *If* $\mathbf{F} \in \mathrm{Con}(\Sigma)$*, then* $\Sigma \vdash \mathbf{F}$.

[4] The following lemma corresponds to one found in Church [3, *151, p. 98]; the origin is not clear.

Proof. Suppose $\mathbf{F} \in \mathrm{Con}(\Sigma)$. By Compactness (rather, Corollary 2.7.5), Σ has a finite subset T such that $\mathbf{F} \in \mathrm{Con}(\Sigma)$. Write T as $\{\mathbf{F}_0, \dots, \mathbf{F}_{m-1}\}$, and $\mathbf{F}$ as $\mathbf{F}_m$. Then the formula

$$\mathbf{F}_0 \to \cdots \to \mathbf{F}_m$$

is a tautology (the proof of this is left to the reader). Call this tautology $\mathbf{G}$, and suppose it is n-ary. We shall show by induction on n that $\mathbf{G}$ is a Theorem.

Let $\mathbf{P}'_k \in \{P_k, \neg P_k\}$ for each k in n. By the previous lemma, we have

$$\mathbf{P}'_0 \dots, \mathbf{P}'_{n-1} \vdash \mathbf{G}. \tag{2.10}$$

If $n = 0$, we are done. (However, there are no nullary formulas.) Suppose that $\mathbf{G}$ is a Theorem under the assumption that $n = \ell$; but now suppose $n = \ell+1$. From Entailment (2.10) in this case, by the Deduction Theorem (2.5.1), remembering that $\mathbf{P}'_\ell$ can be either P_ℓ or $\neg P_\ell$, we have

$$\begin{aligned} &\mathbf{P}'_0 \dots, \mathbf{P}'_{\ell-1} \vdash P_\ell \to \mathbf{G}, \\ &\mathbf{P}'_0 \dots, \mathbf{P}'_{\ell-1} \vdash \neg P_\ell \to \mathbf{G}. \end{aligned}$$

so $\mathbf{P}'_0 \dots, \mathbf{P}'_{\ell-1} \vdash \mathbf{G}$ by Lemma 2.5.2. By inductive hypothesis, $\mathbf{G}$ is a Theorem. □

Exercises

(1) Complete the proof of Lemma 2.9.2.

(2) Prove by induction that, if $\mathbf{F}_m \in \mathrm{Con}(\{\mathbf{F}_0, \dots, \mathbf{F}_{m-1}\})$, then the formula $\mathbf{F}_0 \to \cdots \to \mathbf{F}_m$ is a tautology.

Chapter 3

First-order logic

Recall from § 1.1 the definitions and examples involving *structures*; these are the kinds of structures that we shall now be dealing with.

Throughout this chapter, we shall let $\mathfrak{A}$ stand for an arbitrary structure; its signature will be $\mathcal{L}$. So $\mathfrak{A}$ has universe A, which is just a non-empty set. We shall use c, R and f to stand for arbitrary constants, predicates and relations of $\mathcal{L}$, respectively. We shall use n for the arity of R and f.

Instead of propositional variables, we shall use a set of **individual variables,** namely

$$\{x_k : k \in \omega\}.$$

The structure $\mathfrak{A}$, as such, comes equipped with the operations $f^{\mathfrak{A}}$ and the relations $R^{\mathfrak{A}}$. We can combine these, in the ways to be described below, so as to obtain new operations and relations. These new operations and relations will be symbolized by certain strings:

(∗) operations will be symbolized by *terms;*

(†) relations will be symbolized by *formulas.*

3.1 Terms

If $k < n$, then there is an n-ary operation

$$\vec{a} \longmapsto a_k \tag{3.1}$$

on A. This operation is **projection** onto the kth coordinate.

Each element b in A determines, for each positive n, the constant n-ary operation

$$\vec{a} \longmapsto b. \tag{3.2}$$

If b is $c^{\mathfrak{A}}$, then we have the n-ary operation $\vec{a} \mapsto c^{\mathfrak{A}}$.

More generally, if α is an n-ary operation on A, then there is an $(n+k)$-ary operation on A, namely

$$(\vec{x}\,, \vec{y}\,) \longmapsto \alpha(\vec{x}\,).$$

All operations on A can be composed with one another: If α is an n-ary operation on A, and $(\beta_k : k < n)$ is an n-tuple of

, and with projections, to give other operations on A. The **terms** of $\mathcal{L}$ symbolize these possibilities. The symbols used in terms of $\mathcal{L}$ are:

(∗) the function-symbols f of $\mathcal{L}$;

(†) the constants c of $\mathcal{L}$;

(‡) **(individual) variables**, say from the set $\{x_k : k \in \omega\}$; these will symbolize the projections.

Then the terms of $\mathcal{L}$ are defined inductively thus:

(∗) Each individual variable is a term of $\mathcal{L}$.

(†) Each constant in $\mathcal{L}$ is a term of $\mathcal{L}$.

(‡) If f is an n-ary function-symbol of $\mathcal{L}$, and $t_0, \dots, t_{n-1}$ are terms of $\mathcal{L}$, then the string

$$ft_0 \cdots f_{n-1}$$

is a term of $\mathcal{L}$. (This is not generally a string of length $n+1$; it is a string whose length is 1 more than the sum of the lengths of the strings t_k. If f is binary, then we may unofficially write the term as $(t_0\ f\ t_1)$ instead of ft_0t_1.)

Let the set of terms of $\mathcal{L}$ be denoted

$$\mathrm{Tm}_{\mathcal{L}}\,.$$

As in propositional logic, so here, definition by recursion is possible, because of the following:

Theorem 3.1.1 (Unique Readability). *Every term of $\mathcal{L}$ is* uniquely

$$st_0 \cdots t_{n-1}$$

for some n in ω, some terms t_k of $\mathcal{L}$ (if $n \neq 0$), and some s in $\mathcal{L}$. If $n \neq 0$, then s is an n-ary function-symbol of $\mathcal{L}$; if $n = 0$, then s is a constant of $\mathcal{L}$ or a variable.

Proof. **Exercise.** (The proof can be developed as for [a Theorem].) □

Note well that, by our definition, none of the symbols used in terms is a bracket.

If the variables in a term t come from $\{x_k : k < n\}$, then t is n**-ary**; the set of n-ary terms of $\mathcal{L}$ can be denoted

$$\mathrm{Tm}^n_{\mathcal{L}}\,.$$

Note then

$$\mathrm{Tm}^0_{\mathcal{L}} \subseteq \mathrm{Tm}^1_{\mathcal{L}} \subseteq \mathrm{Tm}^2_{\mathcal{L}} \subseteq \cdots\,.$$

An n-ary term t of $\mathcal{L}$ determines an n-ary operation $t^{\mathfrak{A}}$ on A. The formal definition is recursive:

(∗) ${x_k}^{\mathfrak{A}}$ is $\vec{a} \mapsto a_k$, if $k < n$ (as in (3.1)).

(†) $c^{\mathfrak{A}}$ is $\vec{a} \mapsto c^{\mathfrak{A}}$ (as in (3.2); here c is understood respectively as term and constant).

(‡) $(ft_0 \cdots t_{n-1})^{\mathfrak{A}}$ is

$$\vec{a} \longmapsto f^{\mathfrak{A}}({t_0}^{\mathfrak{A}}(\vec{a}\,), \dots, {t_{k-1}}^{\mathfrak{A}}(\vec{a}\,)),$$

that is, $f^{\mathfrak{A}} \circ ({t_0}^{\mathfrak{A}}, \dots, {t_{k-1}}^{\mathfrak{A}})$.

We have just extended the interpretation-function $\mathfrak{I}$ of $\mathfrak{A}$ so as to include a function

$$t \longmapsto t^{\mathfrak{A}} : \mathrm{Tm}^n_{\mathcal{L}} \longrightarrow A^{A^n}. \tag{3.3}$$

If $t \in \mathrm{Tm}^0_{\mathcal{L}}$, then $t^{\mathfrak{A}} = \{(0, a)\}$ for some a in A; but (as in ch. 1) we can then identify $t^{\mathfrak{A}}$ with a, and we can call t a **constant term**.

Suppose $\mathcal{L} \subseteq \mathcal{L}'$. An **expansion** of $\mathfrak{A}$ to $\mathcal{L}'$ is a structure $\mathfrak{A}'$ whose signature is $\mathcal{L}'$, and whose universe is A, such that

$$s^{\mathfrak{A}'} = s^{\mathfrak{A}}$$

for all s in $\mathcal{L}$. Then $\mathfrak{A}$ is the **reduct** of $\mathfrak{A}'$ to $\mathcal{L}$.

Example 3.1.2. The ring $(\mathbb{Z}, +, -, \cdot, 0, 1)$ is an expansion of the abelian group $(\mathbb{Z}, +, -, 0)$; the latter is a reduct of the former. •

We can treat the elements of A as new constants (not belonging to $\mathcal{L}$); adding these to $\mathcal{L}$ gives the signature $\mathcal{L}(A)$. Then $\mathfrak{A}$ has a natural expansion $\mathfrak{A}_A$ to this signature, so that

$$a^{\mathfrak{A}_A} = a$$

for all a in A. (Some writers prefer to define $\mathcal{L}(A)$ as $\mathcal{L} \cup \{c_a : a \in A\}$, and then to define ${c_a}^{\mathfrak{A}_A} = a$.)

In fact, when it comes to interpreting terms (and, later, formulas), we always treat $\mathfrak{A}$ as if it were $\mathfrak{A}_A$. This means that every n-ary term t of $\mathcal{L}(A)$ has an interpretation $t^{\mathfrak{A}}$ in $\mathfrak{A}$ according to the definition above, provided we understand $a^{\mathfrak{A}}$ as a itself when $a \in A$. In other contexts, however, it will be important to distinguish clearly between $\mathfrak{A}$ and $\mathfrak{A}_A$. We shall also want to speak of expansions $\mathfrak{A}_X$ of $\mathfrak{A}$, where X is an arbitrary subset of A.

If t is an n-ary term of $\mathcal{L}$ (or $\mathcal{L}(A)$), and $\vec{a} \in A^n$, then the result of replacing each x_k in t with a_k, for each k in n, can be written

$$t(\vec{a}\,);$$

this is a constant term of $\mathcal{L}(A)$. For a recursive definition, we have that $t(\vec{a}\,)$ is:

(∗) a_k, if t is x_k;

(†) c, if t is c;

(‡) $ft_0(\vec{a}\,) \cdots t_{k-1}(\vec{a}\,)$, if t is $ft_0 \cdots f_{k-1}$.

Thus we have defined a function

$$t \longmapsto t(\vec{a}\,) : \mathrm{Tm}^n_{\mathcal{L}} \longrightarrow \mathrm{Tm}^0_{\mathcal{L}(A)}\,. \tag{3.4}$$

The tuple $\vec{a}$ also determines the function

$$g \longmapsto g(\vec{a}\,) : A^{A^n} \longrightarrow A. \tag{3.5}$$

We now have several functions, in (3.3), (3.4) and (3.5), fitting together into a square:

$$\begin{tikzcd}
\mathrm{Tm}^n_{\mathcal{L}} \arrow[r, "\vec{a}"] \arrow[d, "\mathfrak{I}"'] & \mathrm{Tm}^0_{\mathcal{L}(A)} \arrow[d, "\mathfrak{I}"] \\
A^{A^n} \arrow[r, "\vec{a}"'] & A
\end{tikzcd}.$$

It doesn't matter which way you go around:

Lemma 3.1.3. *$t^{\mathfrak{A}}(\vec{a}\,) = t(\vec{a}\,)^{\mathfrak{A}_A}$ for all n-ary terms of $\mathcal{L}$, all $\mathcal{L}$-structures $\mathfrak{A}$, and all n-tuples $\vec{a}$ from A.*

Proof. The claim is perhaps obvious; but there *is* a proof by induction:

If t is x_k, then $t^{\mathfrak{A}}(\vec{a}\,) = a_k$, and $t(\vec{a}\,)^{\mathfrak{A}_A} = {a_k}^{\mathfrak{A}_A} = a_k$.

If t is c, then $t^{\mathfrak{A}}(\vec{a}\,) = c^{\mathfrak{A}}$, while $t(\vec{a}\,)^{\mathfrak{A}_A} = c^{\mathfrak{A}_A} = c^{\mathfrak{A}}$.

Finally, if the claim holds when t is any of terms t_i, and now t is $ft_0 \cdots f_{k-1}$, then we have

$$\begin{aligned}
t^{\mathfrak{A}}(\vec{a}\,) &= f^{\mathfrak{A}}({t_0}^{\mathfrak{A}}(\vec{a}\,), \dots, {t_{k-1}}^{\mathfrak{A}}(\vec{a}\,)) \\
&= f^{\mathfrak{A}}(t_0(\vec{a}\,)^{\mathfrak{A}_A}, \dots, t_{k-1}(\vec{a}\,)^{\mathfrak{A}_A}) \\
&= (ft_0(\vec{a}\,) \cdots t_{k-1}(\vec{a}\,))^{\mathfrak{A}_A} \\
&= t(\vec{a}\,)^{\mathfrak{A}_A}.
\end{aligned}$$

This completes the induction. □

As an **exercise**, you can give a recursive definition of

$$t(u_0, \dots, u_{n-1}),$$

where t is an n-ary term, and the u_k are terms. What is the **arity** of the resulting term? Show that

$$t(u_0, \dots, u_{n-1})^{\mathfrak{A}}(\vec{a}\,) = t^{\mathfrak{A}}({u_0}^{\mathfrak{A}}(\vec{a}\,), \dots, {u_{n-1}}^{\mathfrak{A}}(\vec{a}\,)).$$

Note then that, if t is n-ary, then t is precisely the term denoted

$$t(x_0, \dots, x_{n-1}).$$

Example 3.1.4. Let $\mathcal{L}$ be the signature of rings (with identity), and let $\mathfrak{A}$ be $\mathbb{Z}$ (or $\mathbb{Q}$ or $\mathbb{R}$ or $\mathbb{C}$ or some other infinite integral domain or field). If t is a term of $\mathcal{L}(A)$, then $t^{\mathfrak{A}}$ is a **polynomial** over A. What if $\mathfrak{A}$ is finite, say the 2-element field $\mathbb{F}_2$? In that case, if t is $x_0 \cdot (x_0 + 1)$ or 0, then $t^{\mathfrak{A}}(a) = 0$ for all a in A. However, $x_0 \cdot (x_0 + 1)$ and 0 do not represent the same polynomial, since they have different interpretations in fields (like $\mathbb{F}_4$) that properly include $\mathbb{F}_2$. (Here, $\mathbb{F}_4$ can be defined as $\mathbb{F}_2[X]/(X^2 + 1)$.) •

3.2 Formulas

As terms symbolize operations, so *formulas* will symbolize relations. Each formula φ of $\mathcal{L}$ will have an *interpretation* $\varphi^{\mathfrak{A}}$ that is a relation on A. When this relation is nullary and is in fact $\{\varnothing\}$, that is, 1, then φ will be called *true* in $\mathfrak{A}$, and we shall write

$$\mathfrak{A} \models \varphi.$$

Conversely, it is possible to define *truth* in structures first, and then interpretations. We shall look at both approaches.

So-called polynomial equations are examples of *atomic formulas*, which are the first kinds of formulas to be defined. From these, we shall define *open formulas*, and then arbitrary *formulas*.

Atomic formulas and their interpretations

The **atomic formulas** of $\mathcal{L}$ are of two kinds:

($*$) If t_0 and t_1 are terms of $\mathcal{L}$, then $t_0 = t_1$ is an atomic formula of $\mathcal{L}$. (Some writers prefer to use a symbol like $\equiv$ instead of $=$.)

($\dagger$) If R is an n-ary predicate of $\mathcal{L}$, and t_0, …, t_{n-1} are terms of $\mathcal{L}$, then $Rt_0 \cdots t_{n-1}$ is an atomic formula of $\mathcal{L}$. (If R is binary, then we may unofficially write $(t_0\ R\ t_1)$ instead of Rt_0t_1.)

An atomic formula α can be called **k-ary** if the terms it is made from are k-ary.

A polynomial equation in two variables over $\mathbb{R}$ has a solution-set, which can be considered as the *interpretation* of the equation. Likewise, arbitrary atomic formulas have solution-sets, which are their interpretations: If α is a k-ary atomic formula of $\mathcal{L}$, then the **interpretation** in $\mathfrak{A}$ of α is the k-ary relation $\alpha^{\mathfrak{A}}$ on A defined as follows. (Strictly, the validity of the definition depends on Theorem 3.2.1 below.)

$$(t_0 = t_1)^{\mathfrak{A}} = \{\vec{a} \in A^k : t_0{}^{\mathfrak{A}}(\vec{a}\,) = t_1{}^{\mathfrak{A}}(\vec{a}\,)\}; \tag{3.6}$$

$$(Rt_0 \cdots R_{n-1})^{\mathfrak{A}} = \{\vec{a} \in A^k : (t_0{}^{\mathfrak{A}}(\vec{a}\,), \dots, t_{n-1}{}^{\mathfrak{A}}(\vec{a}\,)) \in R^{\mathfrak{A}}\}. \tag{3.7}$$

As a special case, if $k = 0$, we have

$$(t_0 = t_1)^{\mathfrak{A}} = 1 \iff t_0{}^{\mathfrak{A}} = t_1{}^{\mathfrak{A}}; \tag{3.8}$$

$$(Rt_0 \cdots t_{n-1})^{\mathfrak{A}} = 1 \iff (t_0{}^{\mathfrak{A}}, \dots, t_{n-1}{}^{\mathfrak{A}}) \in R^{\mathfrak{A}}. \tag{3.9}$$

Note that the atomic formula $t_0 = t_1$ can be considered as the special case of $Rt_0 \cdots t_{n-1}$ when $n = 2$ and R is $=$. We treat the special case separately because we consider the equals-sign to be *always* available for use in formulas, and we *always* interpret it as true equality.

Open formulas and their interpretations

We can treat atomic formulas as propositional variables, combining them to get **open** (or **quantifier-free**) **formulas**:

(∗) atomic formulas are open formulas;

(†) if φ and χ are open formulas, then so are $\neg\varphi$ and $(\varphi \to \chi)$.

As with atomic formulas, so with arbitrary open formulas: they are k-ary if the terms they are built up from are k-ary. Hence, if φ and χ are k-ary open formulas, then so are $\neg\varphi$ and $(\varphi \to \chi)$. We can now define interpretations of k-ary open formulas by adding to (3.6) and (3.7) the following rules (again, Theorem 3.2.1 is required):

$$(\neg\varphi)^{\mathfrak{A}} = A^k \smallsetminus \varphi^{\mathfrak{A}} = (\varphi^{\mathfrak{A}})^{\mathrm{c}}; \tag{3.10}$$

$$(\varphi \to \chi)^{\mathfrak{A}} = A^k \smallsetminus (\varphi^{\mathfrak{A}} \smallsetminus \chi^{\mathfrak{A}}) = (\varphi^{\mathfrak{A}} \smallsetminus \chi^{\mathfrak{A}})^{\mathrm{c}}. \tag{3.11}$$

In particular, if $k = 0$, then:

$$(\neg\varphi)^{\mathfrak{A}} = 1 \iff \varphi^{\mathfrak{A}} = 0;$$
$$(\varphi \to \chi)^{\mathfrak{A}} = 0 \iff \varphi^{\mathfrak{A}} = 1 \ \&\ \chi^{\mathfrak{A}} = 0.$$

Formulas in general

Formulas in general may contain the **existential quantifier** $\exists$. The inductive definition of **formula** is:

(∗) atomic formulas are formulas;

(†) if φ and χ are formulas, then so are $\neg\varphi$ and $(\varphi \to \chi)$;

(‡) if φ is a formula, and x is a variable, then $\exists x\ \varphi$ is a formula.

The possibility of defining the foregoing interpretations of open formulas depends on the following:

{thm:UR-formulas}

Theorem 3.2.1 (Unique Readability)**.** *Every formula of $\mathcal{L}$ is* uniquely *one of the following:*

(∗) *an equation $t_0 = t_1$, for some terms t_e of $\mathcal{L}$;*

(†) *a relational formula $Rt_0 \cdots t_{n-1}$ for some terms t_k and n-ary predicate R of $\mathcal{L}$, for some positive n;*

(‡) *a negation $\neg\varphi$ for some formula φ;*

(§) *an implication $(\varphi \to \chi)$ for some formulas φ and χ;*

(¶) *an existential formula $\exists x\ \varphi$ for some formula φ and some variable x.*

Proof. **Exercise.** □

Towards interpretations in general

In order to define interpretations of arbitrary formulas, we can still use (3.10) and (3.11) above to define $(\neg\varphi)^{\mathfrak{A}}$ and $(\varphi \to \chi)^{\mathfrak{A}}$ in terms of $\varphi^{\mathfrak{A}}$ and $\chi^{\mathfrak{A}}$. However, we also must define $(\exists x\ \varphi)^{\mathfrak{A}}$ in terms of $\varphi^{\mathfrak{A}}$; and we must first define the **arity** $\exists x\ \varphi$ in terms of the arity of φ. This is not quite so easy. We shall do it presently. When we are done, then, for every n-ary formula φ of $\mathcal{L}$, there will be an n-ary relation $\varphi^{\mathfrak{A}}$ on A; this relation is **defined** by φ, and the relation can be called a 0-**definable** relation of $\mathfrak{A}$. The **definable** relations are those defined by formulas of $\mathcal{L}(A)$; more generally, if $X \subseteq A$, then the X-definable relations are those defined by formulas of $\mathcal{L}(X)$. (Singulary definable relations can just be called definable **sets**.)

If X and Y are k-ary definable relations of $\mathfrak{A}$, then so are X^{c}, $X \cap Y$, $X \cup Y$, &c. In short, all **Boolean combinations** of definable relations are definable, since $\{\neg, \to\}$ is an adequate signature for propositional logic.

Now, if φ is an n-ary formula, defining as such the n-ary relation X, then we can also treat φ as $(n+1)$-ary, defining the relation $X \times A$ on A. This relation is the set

$$\{(\vec{a}\,, b) \in A^{n+1} : \vec{a} \in X\}.$$

This set is also $\pi^{-1}(X)$, where π is the function

$$(\vec{a}\,, b) \longmapsto \vec{a} : A^{n+1} \longrightarrow A^n; \tag{3.12}$$

this map is **projection** onto the first n coordinates. In short then, *inverse images* of definable sets under projections are definable. Using the quantifier $\exists$ in formulas will allow *images* under projections to be definable.

Indeed, suppose φ is an $(n+1)$-ary formula. Then we can define $(\exists x_n\ \varphi)^{\mathfrak{A}}$ to consist of those $\vec{a}$ in A^n such that there *exists* b in A such that $(\vec{a}\,, b) \in \varphi^{\mathfrak{A}}$. Hence $(\exists x_n\ \varphi)^{\mathfrak{A}}$ is $\pi''(\varphi^{\mathfrak{A}})$, the image of $\varphi^{\mathfrak{A}}$, where π is the projection in (3.12).

But what is $(\exists x_i\ \varphi)^{\mathfrak{A}}$ here, if $i < n$? Defining this takes a bit more work; see Remark 3.2.4 below. Meanwhile, we can give an alternative approach to interpreting formulas:

Truth

Let $\mathrm{Fm}\,\mathcal{L}$ be the set of formulas of $\mathcal{L}$. We recursively define a function

$$\varphi \longmapsto \mathrm{fv}(\varphi) : \mathrm{Fm}\,\mathcal{L} \longrightarrow \mathcal{P}(\{x_k : k \in \omega\})$$

as follows:

- $(*)$ $\mathrm{fv}(\alpha)$ is the set of variables in α, if α is atomic (for an **exercise**, this can be given a recursive definition);
- $(\dagger)$ $\mathrm{fv}(\varphi \to \chi) = \mathrm{fv}(\varphi) \cup \mathrm{fv}(\chi)$;
- $(\ddagger)$ $\mathrm{fv}(\exists x\ \varphi) = \mathrm{fv}(\varphi) \smallsetminus \{x\}$.

Then $\mathrm{fv}(\varphi)$ is the set of **free variables** of φ.

If $\mathrm{fv}(\varphi) = \varnothing$, then φ is a **sentence**. So an atomic sentence α is a nullary atomic formula; in this case, we can define

$$\mathfrak{A} \vDash \alpha \iff \alpha^{\mathfrak{A}} = 1; \tag{3.13}$$

in either case, α is **true in** $\mathfrak{A}$. Otherwise, α is **false** in $\mathfrak{A}$, and we can write

$$\mathfrak{A} \nvDash \alpha.$$

We can also define

$$\mathfrak{A} \vDash \neg\sigma \iff \mathfrak{A} \nvDash \sigma; \tag{3.14}$$

$$\mathfrak{A} \nvDash (\sigma \to \tau) \iff \mathfrak{A} \vDash \sigma \;\&\; \mathfrak{A} \vDash \neg\tau; \tag{3.15}$$

provided σ and τ are sentences for which truth and falsity in $\mathfrak{A}$ have been defined. To define $\mathfrak{A} \vDash \exists v\ \varphi$, we should assume that we have been working with formulas of $\mathcal{L}(A)$ all along, and we should define a kind of *substitution*:

For formulas φ, if x is a variable and t is a term, we define the formula

$$\varphi^x_t$$

recursively:

(∗) If α is atomic, then α^x_t is the result of replacing each occurrence of x in α with t (as an **exercise**, you can define this recursively);

(†) $(\neg\varphi)^x_t$ is $\neg(\varphi^x_t)$;

(‡) $(\varphi \to \chi)^x_t$ is $(\varphi^x_t \to \chi^x_t)$;

(§) $(\exists x\ \varphi)^x_t$ is $\exists x\ \varphi$ (no change);

(¶) $(\exists u\ \varphi)^x_t$ is $\exists u\ \varphi^x_t$, if u is not x.

Then φ^x_t is the result of replacing each *free* instance of x in φ with t. Now we can define

$$\mathfrak{A} \vDash \exists x\ \varphi \iff \mathfrak{A} \vDash \varphi^x_a \text{ for some } a \text{ in } A. \tag{3.16}$$

We have now completed the definition of truth; it is expressed by lines (3.8), (3.9), (3.13), (3.14), (3.15) and (3.16).

Interpretations

If $\mathrm{fv}(\varphi) \subseteq \{x_k : k < n\}$, then φ can be called n-ary, and we can write φ as

$$\varphi(x_0, \dots, x_{n-1}).$$

Then, instead of $\varphi^{x_0}_{a_0} \cdots ^{x_{n-1}}_{a_{n-1}}$, we can write

$$\varphi(a_0, \dots, a_{n-1})$$

or $\varphi(\vec{a}\,)$. (Here, $\vec{a}$ is a tuple of *constants*. We could let it be a tuple $(t_0, \dots, t_{n-1})$ of arbitrary terms; but then we should have to ensure that $\varphi(t_0, \dots, t_{n-1})$ is the result of *simultaneously* substituting each t_k for the free instances of the corresponding variable x_k.)

Lemma 3.2.2. *Let φ be an n-ary formula of $\mathcal{L}$.*

(∗) *If φ is atomic, then $\varphi^{\mathfrak{A}} = \{\vec{a} \in A^n : \mathfrak{A} \vDash \varphi(\vec{a}\,)\}$.*

(†) *If φ is $\neg\chi$, then $\{\vec{a} \in A^n : \mathfrak{A} \vDash \varphi(\vec{a}\,)\} = \{\vec{a} \in A^n : \mathfrak{A} \vDash \chi(\vec{a}\,)\}^{\mathrm{c}}$.*

(‡) *If φ is $(\chi \to \psi)$, then*

$$\{\vec{a} \in A^n : \mathfrak{A} \vDash \varphi(\vec{a}\,)\} = \{\vec{a} \in A^n : \mathfrak{A} \vDash \chi(\vec{a}\,)\}^{\mathrm{c}} \cup \{\vec{a} \in A^n : \mathfrak{A} \vDash \psi(\vec{a}\,)\}.$$

(§) *If* φ *is* $\exists x_n\ \chi$, *then*

$$\{\vec{a} \in A^n : \mathfrak{A} \vDash \varphi(\vec{a})\} = \pi''(\{(\vec{a}, b) \in A^{n+1} : \mathfrak{A} \vDash \chi(\vec{a}, b)\}),$$

where π *(as in* (3.12)*) is projection onto the first* n *coordinates.*

Proof. **Exercise.** □

Now we can define

$$\varphi^{\mathfrak{A}} = \{\vec{a} \in A^n : \mathfrak{A} \vDash \varphi(\vec{a})\}$$

for all formulas φ.

In a formula of $\mathcal{L}(A)$, any constants from A can be called **parameters**. So the definable relations of $\mathfrak{A}$ are, more fully, the relations definable *with parameters.*

Example 3.2.3. Algebraic geometry studies the definable relations of $\mathbb{C}$ and of other algebraically closed fields. It can be shown that, on $\mathbb{C}$, all definable relations are definable by *open* formulas. The model-theoretic expression for this fact is that the *theory* of algebraically closed fields admits *elimination of quantifiers.* •

As an **exercise**, you can think about what are the definable sets of

(1) the field $\mathbb{C}$;
(2) $(\omega, <, 0)$;
(3) $(\omega, <)$;
(4) (ω, s), if s is $x \mapsto x + 1$;
(5) a set (that is, a structure in the empty signature).

You probably will not be able to *prove* your answers at this point. {rem:def}

Remark 3.2.4. To complete our first approach to definable sets, let us ignore the ordering of ω. If I is a finite subset of ω, and if $\{i : x_i \in \mathrm{fv}(\varphi)\} \subseteq I$, let us say that φ is I-ary. Let A^I be the set of functions from I to A, a typical such function being denoted

$$(a_i : i \in I).$$

The definition of $\varphi^{\mathfrak{A}}$ as a subset of A^I starts out as before. To define $(\exists x_j\ \varphi)^{\mathfrak{A}}$, let π^I_j be the function

$$(x_i : i \in I) \longmapsto (x_i : i \in I \smallsetminus \{j\}) : A^I \longrightarrow A^{I \smallsetminus \{j\}}.$$

Now we can define

$$(\exists x_j\ \varphi)^{\mathfrak{A}} = (\pi^I_j)''(\varphi^{\mathfrak{A}}).$$

But this doesn't allow $\exists v\ \varphi$ to be treated as J-ary when J contains j. So we should say in addition that if φ is I-ary, and J is any finite subset of ω, then the set

$$\varphi^{\mathfrak{A}} \times A^{J \smallsetminus I}$$

is the interpretation of φ when considered as $(I \cup J)$-ary. Also, suppose $\{i : x_i \in \mathrm{fv}(\exists x_j\ \varphi)\} \subseteq J$. Then φ is $(J \cup \{j\})$-ary, and we can define

$$(\exists x_j\ \varphi)^{\mathfrak{A}} = (\pi^{J \cup \{j\}}_j)''(\varphi^{\mathfrak{A}}) \times A^{\{j\} \cap J}.$$

This formulation of definable relations is rather complicated to be useful; the main point is that a geometric characterization of definable relations is possible:

Theorem 3.2.5. *The family of* 0*-definable relations of a structure* $\mathfrak{A}$ *of* $\mathcal{L}$ *is the smallest family of relations on* A *that is closed under Boolean operations, Cartesian products, projections and permutations of coordinates; that contains the diagonal* $\{(a,a) : a \in A\}$*; and that contains the sets* $\{c^{\mathfrak{A}}\}$*,* $R^{\mathfrak{A}}$ *and* $\{(a_0,\dots,a_n) : f^{\mathfrak{A}}(a_0,\dots,a_{n-1}) = a_n\}$*.*

3.3 Logical consequence

Having defined truth, we can define *logical consequence*. Let $\mathrm{Sn}_{\mathcal{L}}$ be the set of sentences of $\mathcal{L}$. The $\mathcal{L}$-structure $\mathfrak{A}$ is a **model** of a subset Σ of $\mathrm{Sn}_{\mathcal{L}}$ if each sentence in Σ is true in $\mathfrak{A}$; then we can write

$$\mathfrak{A} \models \Sigma.$$

If a sentence σ is true in every model of Σ, then σ is a **(logical) consequence** of Σ, and we can write

$$\Sigma \models \sigma.$$

If $\varnothing \models \sigma$, then we can write just

$$\models \sigma;$$

in this case, σ is a **validity**.

Two sentences are **(logically) equivalent** if each is a logical consequence of the other.

Lemma 3.3.1. *Let* σ *and* τ *be sentences of* $\mathcal{L}$*.*

(∗) $\sigma \models \tau$ *if and only if* $\models (\sigma \to \tau)$*, for all* σ *and* τ *in* $\mathrm{Sn}_{\mathcal{L}}$*.*

(†) σ *and* τ *are equivalent if and only if* $\models (\sigma \to \tau) \land (\tau \to \sigma)$*.*

(‡) *Logical equivalence is an equivalence-relation on* $\mathrm{Sn}_{\mathcal{L}}$*.*

Proof. **Exercise**. □

Instead of the formula $(\varphi \to \chi) \land (\chi \to \varphi)$, let us write

$$\varphi \leftrightarrow \chi.$$

By the lemma, σ and τ are logically equivalent if and only if $(\sigma \leftrightarrow \tau)$ is a validity. We may blur the distinction between logically equivalent sentences, identifying σ with $\neg\neg\sigma$ for example.

Instead of $\neg\exists v\ \neg\varphi$, we may write

$$\forall v\ \varphi.$$

Then $\neg\forall v\ \varphi$ is (equivalent to) $\exists v\ \neg\varphi$.

{example:entailment}

Example 3.3.2. The sentence

$$(\forall x\ (Px \to Qx) \to (\forall x\ Px \to \forall x\ Qx))$$

is a validity, where P and Q are unary predicates. To prove this, note that, by (3.15), it is enough to show that $\mathfrak{A} \models (\forall x\ Px \to \forall x\ Qx)$ whenever $\mathfrak{A} \models \forall x\ (Px \to Qx)$. So suppose

$$\mathfrak{A} \models \forall x\ (Px \to Qx). \tag{3.17}$$

It is now enough to show that, if also $\mathfrak{A} \models \forall x\ Px$, then $\mathfrak{A} \models \forall x\ Qx$. So suppose

$$\mathfrak{A} \models \forall x\ Px. \tag{3.18}$$

Let $a \in A$. Then $\mathfrak{A} \models Pa$, by (3.18). But $\mathfrak{A} \models (Pa \to Qa)$, by (3.17). Hence $\mathfrak{A} \models Qa$. Since a was arbitrary, we have $\mathfrak{A} \models \forall x\ Qx$. •

If $\mathrm{fv}(\varphi) = \{u_0, \dots, u_{n-1}\}$, and $\mathfrak{A} \models \forall u_0\ \cdots \forall u_{n-1}\ \varphi$, we may write just

$$\mathfrak{A} \models \varphi.$$

Here, the sentence $\forall u_0\ \cdots \forall u_{n-1}\ \varphi$ is the **(universal) generalization** of φ. Now we can define $\Sigma \models \varphi$ for arbitrary formulas φ (although Σ should still be a set of *sentences*); we can also say that arbitrary formulas φ and χ are **(logically) equivalent** if

$$\models (\varphi \leftrightarrow \chi).$$

For the formula φ with free variables $x_0, \dots, x_{n-1}$, if we have

$$\mathfrak{A} \models \exists u_0\ \cdots \exists u_{n-1}\ \varphi,$$

then we can say that φ is **satisfied** in $\mathfrak{A}$.

It can happen then that $\mathfrak{A} \not\models \varphi$ and $\mathfrak{A} \not\models \neg\varphi$. However, if σ is a *sentence*, then either σ or $\neg\sigma$ is true in $\mathfrak{A}$.

Example 3.3.3. Each of the following formulas is true in every group:

$$\begin{gathered} x \cdot (y \cdot z) = (x \cdot y) \cdot z, \\ x \cdot 1 = x, \qquad x \cdot x^{-1} = 1, \\ 1 \cdot x = x, \qquad x^{-1} \cdot x = 1. \end{gathered}$$

•

If $\Sigma \subseteq \mathrm{Sn}_{\mathcal{L}}$, let

$$\mathrm{Con}_{\mathcal{L}}(\Sigma) = \{\sigma \in \mathrm{Sn}_{\mathcal{L}} : \Sigma \models \sigma\}.$$

Lemma 3.3.4. $\mathrm{Con}_{\mathcal{L}}(\mathrm{Con}_{\mathcal{L}}(\Sigma)) = \mathrm{Con}_{\mathcal{L}}(\Sigma)$.

Proof. Since $\Sigma \subseteq \mathrm{Con}_{\mathcal{L}}(\Sigma)$, we have $\mathrm{Con}_{\mathcal{L}}(\Sigma) \subseteq \mathrm{Con}_{\mathcal{L}}(\mathrm{Con}_{\mathcal{L}}(\Sigma))$. Suppose $\sigma \in \mathrm{Con}_{\mathcal{L}}(\mathrm{Con}_{\mathcal{L}}(\Sigma))$. Then $\mathrm{Con}_{\mathcal{L}}(\Sigma) \models \sigma$. But if $\mathfrak{A} \models \Sigma$, then $\mathfrak{A} \models \mathrm{Con}_{\mathcal{L}}(\Sigma)$, so in this case $\mathfrak{A} \models \sigma$. Thus $\sigma \in \mathrm{Con}_{\mathcal{L}}(\Sigma)$. □

A subset T of $\mathrm{Sn}_{\mathcal{L}}$ is a **theory** of $\mathcal{L}$ if $\mathrm{Con}_{\mathcal{L}}(T) = T$. A subset Σ of a theory T is a set of **axioms** for T if

$$T = \mathrm{Con}_{\mathcal{L}}(\Sigma);$$

we may also say then that Σ **axiomatizes** T.

{examp:groups}

Example 3.3.5. The theory of groups is axiomatized by

$$\begin{gathered}\forall x\, \forall y\, \forall z\ x \cdot (y \cdot z) = (x \cdot y) \cdot z, \\ \forall x\ x \cdot 1 = x, \qquad \forall x\ x \cdot x^{-1} = 1, \\ \forall x\ 1 \cdot x = x, \qquad \forall x\ x^{-1} \cdot x = 1.\end{gathered}$$

•

If $\mathfrak{A}$ is an $\mathcal{L}$-structure, let

$$\operatorname{Th}(\mathfrak{A}) = \{\sigma \in \operatorname{Sn}_{\mathcal{L}} : \mathfrak{A} \models \sigma\}.$$

Lemma 3.3.6. $\operatorname{Th}(\mathfrak{A})$ *is a theory.*

Proof. Say $\operatorname{Th}(\mathfrak{A}) \models \sigma$. Since $\mathfrak{A} \models \operatorname{Th}(\mathfrak{A})$, we have $\mathfrak{A} \models \sigma$, so $\sigma \in \operatorname{Th}(\mathfrak{A})$. □

We can now call $\operatorname{Th}(\mathfrak{A})$ the **theory of** $\mathfrak{A}$. Note that, if T is $\operatorname{Th}(\mathfrak{A})$, then

$$T \models \sigma \iff T \not\models \neg\sigma$$

for all sentences σ. An arbitrary theory T need not have this property; if it does, then T is **complete**. So, the theory *of* a structure is always complete. The converse holds, by the next lemma; also, the set $\operatorname{Sn}_{\mathcal{L}}$ is a theory, but it is not complete:

{lem:complete}

Lemma 3.3.7. *Let T be a theory of $\mathcal{L}$.*

(∗) *If T has no model, then T is $\operatorname{Sn}_{\mathcal{L}}$ itself.*

(†) *If T is complete, then T is $\operatorname{Th}(\mathfrak{A})$ for some structure $\mathfrak{A}$, which is a model of T.*

(‡) *If T has a model $\mathfrak{A}$, then T is included in $\operatorname{Th}(\mathfrak{A})$, which is a complete theory: in particular*

$$T \models \sigma \implies T \not\models \neg\sigma$$

for all σ in $\operatorname{Sn}_{\mathcal{L}}$.

(§) *Hence, to prove that T is complete, it is enough to show that T has models and*

$$T \not\models \sigma \implies T \models \neg\sigma$$

for all σ in $\operatorname{Sn}_{\mathcal{L}}$.

Proof. Consider the points in order:

(∗) If T is a theory with no models, and σ is a sentence, then σ is true in every model of T, so $T \models \sigma$, whence $\sigma \in T$.

(†) If T is complete, then by definition it cannot contain all sentences, so it must have a model $\mathfrak{A}$. Then $T \subseteq \operatorname{Th}(\mathfrak{A})$. By this and completeness of T, we have

$$T \models \sigma \implies \alpha \models \sigma \implies \alpha \not\models \neg\sigma \implies T \not\models \neg\sigma \implies T \models \sigma$$

for all σ in $\operatorname{Sn}_{\mathcal{L}}$. In short, $T \models \sigma \iff \mathfrak{A} \models \sigma$, so $T = \operatorname{Th}(\mathfrak{A})$.

(‡) The set $\{\sigma, \neg\sigma\}$ has no models.

(§) Obvious.

This completes the proof. □

We can also speak of the theory of a *class* of $\mathcal{L}$-structures. If K is such a class, then $\mathrm{Th}(K)$ is the set of sentences of $\mathcal{L}$ that are true in *every* structure in K.

In particular, if $\Sigma \subseteq \mathrm{Sn}_{\mathcal{L}}$, then we can define

$$\mathrm{Mod}(\Sigma)$$

to be the class of all models of Σ. Then

$$\mathrm{Th}(\mathrm{Mod}(\Sigma)) = \mathrm{Con}_{\mathcal{L}}(\Sigma).$$

Example 3.3.8. By definition, a group is just a model of the theory of groups, as axiomatized in Example 3.3.5. Hence this theory is $\mathrm{Th}(K)$, where K is the class of all groups. •

3.4 Additional exercises

(1) Letting P and Q be unary predicates, determine, from the definition of $\models$, whether the following hold. (A method is shown in Example 3.3.2.)

 (∗) $(\exists x\ Px \to \exists x\ Qx) \models \forall x\ (Px \to Qx)$;
 (†) $(\forall x\ Px \to \exists x\ Qx) \models \exists x\ (Px \to Qx)$;
 (‡) $\exists x\ (Px \to Qx) \models (\forall x\ Px \to \exists x\ Qx)$;
 (§) $\{\exists x\ Px,\ \exists x\ Qx\} \models \exists x\ (Px \land Qx)$;
 (¶) $\exists x\ Px \to \exists y\ Qy \models \forall x\ \exists y\ (Px \to Qy)$.

(2) Let $\mathcal{L} = \{R\}$, where R is a binary predicate, and let $\mathfrak{A}$ be the $\mathcal{L}$-structure $(\mathbb{Z}, \leqslant)$. Determine $\varphi^{\mathfrak{A}}$ if φ is:

 (∗) $\forall x_1\ (Rx_1x_0 \to Rx_0x_1)$;
 (†) $\forall x_2\ (Rx_2x_0 \lor Rx_1x_2)$.

(3) Let $\mathcal{L}$ be $\{S, P\}$, where S and P are binary function-symbols. Then $(\mathbb{R}, +, \cdot)$ is an $\mathcal{L}$-structure. Show that the following sets and relations are definable in this structure:

 (∗) $\{0\}$;
 (†) $\{1\}$;
 (‡) $\{a \in \mathbb{R} : 0 < a\}$;
 (§) $\{(a, b) \in \mathbb{R}^2 : a < b\}$.

(4) Show that the following sets are definable in $(\omega, +, \cdot, \leqslant, 0, 1)$:

 (∗) the set of even numbers;
 (†) the set of prime numbers.

(5) Let R be the binary relation

$$\{(x, x+1) : x \in \mathbb{Z}\}$$

on $\mathbb{Z}$. Show that R is 0-definable in the structure $(\mathbb{Z}, <)$; that is, find a binary formula φ in the signature $\{<\}$ such that $\varphi^{(\mathbb{Z},<)} = R$.

Chapter 4

Quantifier-elimination

In general, if we have some sentences, how might we show that the theory that they axiomatize is complete? If the theory is *not* complete, this is easy to show:

Example 4.0.1. The theory of groups is not complete, since the sentence

$$\forall x \, \forall y \; xy = yx$$

is true (by definition) only in abelian groups, but there are non-abelian groups (such as the group of permutations of three objects). The theory of abelian groups is not complete either, since (in the signature $\{+, -, 0\}$) the sentence

$$\forall x \, (x + x = 0 \to x = 0)$$

is true in $(\mathbb{Z}, +, -, 0)$, but false in $(\mathbb{Z}/2\mathbb{Z}, +, -, 0)$. •

Let TO be the theory of *strict* total orders; this is axiomatized by the universal generalizations of:

$$\begin{gathered}\neg(x < x),\\ x < y \to \neg(y < x),\\ x < y \wedge y < z \to x < z,\\ x < y \vee y < x \vee x = y.\end{gathered}$$

This theory is not complete, since $(\omega, <)$ and $(\mathbb{Z}, <)$ are models of TO with different complete theories (**exercise**).

Let TO^* be the theory of **dense total orders without endpoints**, namely, TO^* has the axioms of TO, along with the universal generalizations of:

$$\begin{gathered}\exists z \, (x < z \wedge z < y),\\ \exists y \; y < x,\\ \exists y \; x < y.\end{gathered}$$

The theory TO^* has a model, namely $(\mathbb{Q}, <)$. We shall show that TO^* is complete. In order to do this, we shall first show that the theory admits *(full) elimination of quantifiers*.

An arbitrary theory T admits **(full) elimination of quantifiers** if, for every formula φ of $\mathcal{L}$, there is an *open* formula χ of $\mathcal{L}$ such that

$$T \vDash (\varphi \leftrightarrow \chi)$$

—in words, φ **is equivalent to** χ ***modulo*** T.

{lem:QE}

Lemma 4.0.2. *An $\mathcal{L}$-theory T admits quantifier-elimination, provided that, if φ is an open formula, and v is a variable, then $\exists v\ \varphi$ is equivalent* modulo T *to an open formula.*

Proof. Use induction on formulas. Specifically:

Every atomic formula is equivalent *modulo* T to an open formula, namely itself.

Suppose φ is equivalent *modulo* T to an open formula α. Then $T \vDash (\neg\varphi \leftrightarrow \neg\alpha)$; but $\neg\alpha$ is open.

Suppose also χ is equivalent *modulo* T to an open formula β. Then

$$T \vDash ((\varphi \to \chi) \leftrightarrow (\alpha \to \beta));$$

but $(\alpha \to \beta)$ is open.

Finally, $T \vDash (\exists v\ \varphi \leftrightarrow \exists v\ \alpha)$ (**exercise**); but by assumption, $\exists v\ \alpha$ is equivalent to an open formula γ; so $T \vDash (\exists v\ \varphi \leftrightarrow \gamma)$ (**exercise**). This completes the induction. □

The lemma can be improved slightly. Every open formula is logically equivalent to a formula in *disjunctive normal form*:

$$\bigvee_{i<m} \bigwedge_{j<n} \alpha_i^{(j)},$$

where each $\alpha_i^{(j)}$ is either an atomic or a negated atomic formula. (See § 2.6 of this year's notes for Math 111.) This formula in disjunctive normal form can also be written

$$\bigvee_{i<m} \bigwedge \Sigma_i$$

where $\Sigma_i = \{a_i^{(j)} : j < n\}$. Note that

$$\vDash (\exists v \bigvee_{i<m} \bigwedge \Sigma_i \leftrightarrow \bigvee_{i<m} \exists v\ \bigwedge \Sigma_i) \tag{4.1}$$

(**exercise**). The formulas $\exists v\ \bigwedge \Sigma_i$ are said to be *primitive*. In general, a **primitive** formula is a formula

$$\exists u_0\ \cdots \exists u_{n-1}\ \bigwedge \Sigma,$$

where Σ is a *finite* non-empty set of atomic and negated atomic formulas. (Remember that $\bigwedge \Sigma$ is just an abbreviation for $\varphi_0 \wedge \ldots \wedge \varphi_{n-1}$, where the formulas φ_i compose Σ; so Σ must be finite since formulas must have finite length. Also, formulas have *positive* length, so Σ must be non-empty. However, the notation $\bigwedge \varnothing$ could be understood to stand for a validity.)

Using (4.1), we can adjust the induction above to show that T *admits quantifier-elimination, provided that every primitive formula with one (existential) quantifier is equivalent* modulo T *to an open formula.*

Henceforth suppose $\mathcal{L}$ is $\{<\}$, and $\mathrm{TO} \subseteq T$; so T is *a* theory of total orders. Then we can improve Lemma 4.0.2 even more. Indeed, the atomic formulas of $\mathcal{L}$ now are $x = y$ and $x < y$, where x and y are variables. Moreover,

$$\mathrm{TO} \vDash (\neg(x < y) \leftrightarrow (x = y \lor y < x)),$$
$$\mathrm{TO} \vDash (\neg(x = y) \leftrightarrow (x < y \lor y < x)).$$

Hence, in $\mathcal{L}$, any formula is equivalent, *modulo* TO, to the result of replacing each negated atomic sub-formula with the appropriate disjunction of atomic formulas. If this replacement is done to a formula in disjunctive normal form, then the new formula will have a disjunctive normal form that involves no negations. So T admits quantifier-elimination, provided that every formula

$$\exists v \bigwedge \Sigma$$

is equivalent, *modulo* T, to an open formula, where now Σ is a set of atomic formulas.

{thm:TO-QE} Using this criterion, we shall show that TO^* admits quantifier-elimination:

Theorem 4.0.3. TO^* *admits (full) elimination of quantifiers.*

Proof. Let Σ be a finite, non-empty set of atomic formulas (in the signature $\{<\}$). Let X be the set of variables appearing in formulas in Σ; that is,

$$X = \bigcup_{\alpha \in \Sigma} \mathrm{fv}(\alpha).$$

Then X is a finite non-empty set; say

$$X = \{x_0, \dots, x_n\}.$$

Suppose $\mathfrak{A}$ is an $\mathcal{L}$-structure, and $\vec{a} \in A^{n+1}$. If α is an atomic formula of $\mathcal{L}$ with variables from X, we can let $\alpha(\vec{a}\,)$ be the result of replacing each x_i in α with a_i. Then we can let

$$\Sigma(\vec{a}\,) = \{\alpha(\vec{a}\,) : \alpha \in \Sigma\}.$$

Suppose in fact

$$\mathfrak{A} \vDash \mathrm{TO} \cup \{\bigwedge \Sigma(\vec{a}\,)\}.$$

Let us define $\Sigma_{(\mathfrak{A},\vec{a}\,)}$ as the set of atomic formulas α such that $\mathrm{fv}(\alpha) \subseteq X$ and $\mathfrak{A} \vDash \alpha(\vec{a}\,)$. Then

$$\Sigma \subseteq \Sigma_{(\mathfrak{A},\vec{a}\,)}.$$

Moreover, once Σ has been chosen, *there are only finitely many possibilities for the set* $\Sigma_{(\mathfrak{A},\vec{a}\,)}$. Let us list these possibilities as

$$\Sigma_0, \dots, \Sigma_{m-1}.$$

Now, possibly $m = 0$ here. In this case,

$$\mathrm{TO} \vDash (\exists v \ \bigwedge \Sigma \leftrightarrow v \neq v),$$

so we are done. Henceforth we may assume $m > 0$. If $\mathfrak{B} \vDash \mathrm{TO} \cup \{\bigwedge \Sigma(\vec{b}\,)\}$, then

$$\mathfrak{B} \vDash \bigwedge \Sigma_i(\vec{b}\,)$$

for some i in m. Therefore

$$\mathrm{TO} \vDash (\bigwedge \Sigma \leftrightarrow \bigvee_{i<m} \bigwedge \Sigma_i),$$

and hence

$$\mathrm{TO} \vDash (\exists v \ \bigwedge \Sigma \leftrightarrow \bigvee_{i<m} \exists v \ \bigwedge \Sigma_i).$$

Therefore, for our proof of quantifier-elimination, we may assume that Σ *is* one of the sets $\Sigma_{(\mathfrak{A},\vec{a}\,)}$ (so that, in particular, $m = 1$).

Now partition Σ as $\Gamma \cup \Delta$, where no formula in Γ, but every formula in Δ, contains v. There are two extreme possibilities:

($*$) Suppose $\Gamma = \varnothing$. Then $X = \{v\}$ (since if $x \in X \smallsetminus \{v\}$, then $(x = x) \in \Gamma$). Also, $\Sigma = \Delta = \{v = v\}$, so

$$\vDash (\exists v \ \bigwedge \Sigma \leftrightarrow v = v),$$

and we are done in this case.

($\dagger$) Suppose $\Delta = \varnothing$. Then $v \notin X$, and

$$\vDash (\exists v \ \bigwedge \Sigma \leftrightarrow \bigwedge \Sigma),$$

so we are done in *this* case.

Henceforth, suppose neither Γ nor Δ is empty. Then

$$\vDash (\exists v \ \bigwedge \Sigma \leftrightarrow \bigwedge \Gamma \wedge \exists v \ \bigwedge \Delta).$$

We shall show that

$$\mathrm{TO}^* \vDash (\exists v \ \bigwedge \Sigma \leftrightarrow \bigwedge \Gamma), \tag{4.2}$$

which will complete the proof. To show (4.2), it is enough to show

$$\mathrm{TO}^* \vDash (\bigwedge \Gamma \to \exists v \ \bigwedge \Delta).$$

But this follows from the definition of TO^*:

Indeed, remember that Σ is $\Sigma_{(\mathfrak{A},\vec{a}\,)}$. Hence, for all i and j in $n+1$, we have

$$a_i < a_j \iff (x_i < x_j) \in \Sigma;$$
$$a_i = a_j \iff (x_i = x_j) \in \Sigma.$$

We have $v \in X$. We can relabel the elements of X as necessary so that v is x_n and

$$a_0 \leqslant \ldots \leqslant a_{n-1}.$$

(Here, $a_i \leqslant a_{i+1}$ means $a_i < a_{i+1}$ or $a_i = a_{i+1}$ as usual.) Suppose $\mathfrak{B} \vDash \mathrm{TO}^*$, and B^n contains $\vec{b}$ such that $\mathfrak{B} \vDash \bigwedge \Gamma(\vec{b})$. We have to show that there is c in B such that $\mathfrak{B} \vDash \bigwedge \Delta(\vec{b}, c)$. Now, for all i and j in n, we have

$$\begin{aligned} b_i < b_j &\iff a_i < a_j; \\ b_i = b_j &\iff a_i = a_j. \end{aligned}$$

Because $\mathfrak{B}$ is a model of TO^* (and not just TO), we can find c as needed according to the relation of a_n with the other a_i:

(∗) If $a_n = a_i$ for some i in n, then let $c = b_i$.

(†) If $a_{n-1} < a_n$, then let c be greater than b_{n-1}.

(‡) If $a_n < a_0$, then let c be less than b_0.

(§) If $a_k < a_n < a_{k+1}$, then we can let c be such that $b_k < c < b_{k+1}$.

This completes the proof that TO^* admits quantifier-elimination. □

We have proved more than quantifier-elimination: we have shown that, *modulo* TO^*, the formula $\exists v \ \bigwedge \Sigma$ is equivalent to $v \neq v$ or $v = v$ or an open formula *with the same free variables as* $\exists v \ \bigwedge \Sigma$. In the proof, we introduced $v \neq v$ simply as a formula φ such that $\mathfrak{A} \nvDash \varphi$ for every structure $\mathfrak{A}$. Such a formula corresponds to a nullary Boolean connective, namely an **absurdity** (the negation of a validity). We used 0 as such a connective; but let us now use $\bot$.

Likewise, instead of $v = v$, we can use, as a validity, the nullary Boolean connective $\top$. From the last proof, therefore, we have:

Porism 4.0.4. *In the signature* $\{<\}$, *with the nullary connectives* $\bot$ *and* $\top$ *allowed, every formula is equivalent* modulo TO^* *to an open formula with the same free variables.*

In a signature of first-order logic without constants, an open *sentence* consists entirely of Boolean connectives, with no propositional variables; so it is either an absurdity or a validity. As a consequence, we have:

Theorem 4.0.5. TO^* *is a complete theory.*

Proof. By the porism, every *sentence* is equivalent to an open *sentence*; as just noted, such a sentence is an absurdity or a validity. Suppose $\mathrm{TO}^* \vDash (\sigma \leftrightarrow \bot)$. But $\vDash (\sigma \leftrightarrow \bot) \leftrightarrow \neg\sigma$; so $\mathrm{TO}^* \vDash \neg\sigma$. Similarly, if $\mathrm{TO}^* \vDash (\sigma \leftrightarrow \top)$, then $\mathrm{TO}^* \vDash \sigma$. Hence, for all sentences σ, if $\mathrm{TO}^* \nvDash \sigma$, then $\mathrm{TO}^* \vDash \neg\sigma$. Therefore TO^* is complete by Lemma 3.3.7. □

Chapter 5

Relations between structures

There are several binary relations on the class of structures in a signature $\mathcal{L}$. Some relations involve universes of structures; others do not.

Let $\mathfrak{A}$ and $\mathfrak{B}$ be $\mathcal{L}$-structures.

5.1 Fundamental definitions

The structure $\mathfrak{A}$ is a **substructure of** $\mathfrak{B}$, or $\mathfrak{B}$ is an **extension of** $\mathfrak{A}$, if $A \subseteq B$ and

($*$) $c^{\mathfrak{A}} = c^{\mathfrak{B}}$ for all constants c of $\mathcal{L}$;

(†) $R^{\mathfrak{A}} = A^n \cap R^{\mathfrak{B}}$ for all n-ary predicates R of $\mathcal{L}$, for all positive n in ω;

(‡) $f^{\mathfrak{A}} = f^{\mathfrak{B}} \circ \mathsf{id}_{A^n}$ for all n-ary function-symbols f of $\mathcal{L}$, for all positive n in ω.

In this case, we write

$$\mathfrak{A} \subseteq \mathfrak{B}.$$

Immediately, $\mathfrak{A} \subseteq \mathfrak{B}$ if and only if $A \subseteq B$ and

$$\mathfrak{A} \vDash \sigma \iff \mathfrak{B} \vDash \sigma \tag{5.1}$$

for all atomic sentences σ of $\mathcal{L}(A)$ of one of the forms

$$\begin{gathered} a_0 = c, \\ Ra_0 \cdots a_{n-1}, \\ fa_0 \cdots a_{n-1} = a_n. \end{gathered}$$

The two structures $\mathfrak{A}$ and $\mathfrak{B}$ are called **elementarily equivalent** if (5.1) holds for *all* sentences σ of $\mathcal{L}$ (not $\mathcal{L}(A)$). In this case, we write

$$\mathfrak{A} \equiv \mathfrak{B}.$$

Then the relation $\equiv$ of **elementary equivalence** is in fact the equivalence-relation induced on the class of $\mathcal{L}$-structures by the function $\mathfrak{M} \mapsto \mathrm{Th}(\mathfrak{M})$; that is,

$$\mathfrak{A} \equiv \mathfrak{B} \iff \mathrm{Th}(\mathfrak{A}) = \mathrm{Th}(\mathfrak{B}).$$

All models of a complete theory are elementarily equivalent, and first-order logic provides no means to distinguish between elementarily equivalent structures. We shall see *other* possible ways to distinguish between them.

5.2 Additional definitions

The structure $\mathfrak{A}$ is an **elementary substructure of** $\mathfrak{B}$, and $\mathfrak{B}$ is an **elementary extension of** $\mathfrak{A}$, if $A \subseteq B$ and $\mathfrak{A}_A \equiv \mathfrak{B}_A$. Then we write

$$\mathfrak{A} \preccurlyeq \mathfrak{B}.$$

(Some people prefer just to write $\mathfrak{A} \prec \mathfrak{B}$.) Note here that $\mathfrak{A}_A \equiv \mathfrak{B}_A$ if and only if (5.1) holds for all sentences σ of $\mathcal{L}(A)$. In particular, elementary substructures *are* substructures.

Various functions between (universes of) structures are possible. To describe them, it is convenient to use the following convention. If h is a function from A to B, we also understand h as the function from A^n to B^n given by

$$h(\vec{a}\,) = h(a_0, \dots, a_{n-1}) = (h(a_0), \dots, h(a_{n-1})), \tag{5.2}$$

for each n in ω. In particular, as a function from A^0 to B^0, h is $\{(0,0)\}$.

The structure $\mathfrak{A}$ **embeds** in $\mathfrak{B}$ if there is an *injection* h from A to B such that:

- ($*$) $h(c^{\mathfrak{A}}) = c^{\mathfrak{B}}$ for all constants c in $\mathcal{L}$;
- (†) $h''(R^{\mathfrak{A}}) = h''(A^n) \cap R^{\mathfrak{B}}$ for all n-ary predicates R in $\mathcal{L}$, for all positive n in ω;
- (‡) $h \circ f^{\mathfrak{A}} = f^{\mathfrak{B}} \circ h$ for all n-ary function-symbols f in $\mathcal{L}$, for all positive n in ω.

Then h is an **embedding** of $\mathfrak{A}$ in $\mathfrak{B}$; to express this, we can write

$$h : \mathfrak{A} \longrightarrow \mathfrak{B}.$$

Immediately, $h : \mathfrak{A} \to \mathfrak{B}$ if and only if $h : A \to B$ and

$$\mathfrak{A} \vDash \varphi(\vec{a}\,) \iff \mathfrak{B} \vDash \varphi(h(\vec{a}\,)), \qquad \text{for all } \vec{a} \text{ from } A, \tag{5.3}$$

for all atomic formulas φ of $\mathcal{L}$ of one of the forms

$$\begin{gathered} x_0 = x_1, \\ x_0 = c, \\ Rx_0 \cdots x_{n-1}, \\ fx_0 \cdots x_{n-1} = x_n. \end{gathered}$$

If (5.3) holds for *all* formulas φ of $\mathcal{L}$, then h is an **elementary embedding** of $\mathfrak{A}$ in $\mathfrak{B}$, and we can write

$$h : \mathfrak{A} \xrightarrow{\equiv} \mathfrak{B}.$$

Example 5.2.1. The map $x \mapsto x/1$ is an embedding of the ring $\mathbb{Z}$ in the field $\mathbb{Q}$, but not an elementary embedding, since $\mathbb{Z} \vDash \varphi(1)$, but $\mathbb{Q} \nvDash \varphi(1/1)$, where φ is $\neg\exists y\ y + y = x$. •

If $h : \mathfrak{A} \to \mathfrak{B}$ and h is a *surjection* onto B, then h is called an **isomorphism** from $\mathfrak{A}$ to $\mathfrak{B}$, and we can write

$$h : \mathfrak{A} \xrightarrow{\cong} \mathfrak{B}.$$

If an isomorphism from $\mathfrak{A}$ to $\mathfrak{B}$ exists, then $\mathfrak{A}$ is **isomorphic** to $\mathfrak{B}$, and we can write

$$\mathfrak{A} \cong \mathfrak{B};$$

the relation $\cong$ can be called **isomorphism**.

5.3 Implications

Lemma 5.3.1. *Isomorphism is an equivalence-relation. If* $h : \mathfrak{A} \xrightarrow{\cong} \mathfrak{B}$*, then* $h^{-1} : \mathfrak{B} \xrightarrow{\cong} \mathfrak{A}$.

Proof. **Exercise.** □

Isomorphic structures are practically the same. One way to make this precise is by means of the following:

{lem:fundamental}

Lemma 5.3.2. *Suppose* $h : \mathfrak{A} \to \mathfrak{B}$*. Then* (5.3) *holds for all atomic formulas* φ *of* $\mathcal{L}$*. If also* h *is* onto B*, then* (5.3) *holds for* all *formulas* φ *of* $\mathcal{L}$.

Proof. Note that (5.3) can be re-formulated in other ways, according to taste:

$$\vec{a} \in \varphi^{\mathfrak{A}} \iff h(\vec{a}\,) \in \varphi^{\mathfrak{B}}, \text{ for all } n\text{-tuples } \vec{a} \text{ from } A, \tag{5.4}$$

or more simply

$$h''(\varphi^{\mathfrak{A}}) = h''(A^n) \cap \varphi^{\mathfrak{B}}.$$

To prove it, assuming $h : \mathfrak{A} \to \mathfrak{B}$, we first establish by induction that

$$h \circ t^{\mathfrak{A}} = t^{\mathfrak{B}} \circ h \tag{5.5}$$

for all terms t of $\mathcal{L}$:

(∗) (5.5) is true by definition if t is a constant or variable;

(†) if (5.5) is true when $t \in \{u_0, \dots, u_{n-1}\}$, and now t is $fu_0 \cdots u_{n-1}$, then

$$\begin{aligned}
h \circ t^{\mathfrak{A}} &= h \circ f^{\mathfrak{A}} \circ ({u_0}^{\mathfrak{A}}, \dots, {u_{n-1}}^{\mathfrak{A}}) && \text{[by def'n of } t^{\mathfrak{A}}\text{]}\\
&= f^{\mathfrak{B}} \circ h \circ ({u_0}^{\mathfrak{A}}, \dots, {u_{n-1}}^{\mathfrak{A}}) && \text{[by def'n of } \subseteq\text{]}\\
&= f^{\mathfrak{B}} \circ (h \circ {u_0}^{\mathfrak{A}}, \dots, h \circ {u_{n-1}}^{\mathfrak{A}}) && \text{[by (5.2)]}\\
&= f^{\mathfrak{B}} \circ ({u_0}^{\mathfrak{B}} \circ h, \dots, {u_{n-1}}^{\mathfrak{B}} \circ h) && \text{[by inductive hyp.]}\\
&= f^{\mathfrak{B}} \circ ({u_0}^{\mathfrak{B}}, \dots, {u_{n-1}}^{\mathfrak{B}}) \circ h\\
&= t^{\mathfrak{B}} \circ h. && \text{[by def'n of } t^{\mathfrak{A}}\text{]}
\end{aligned}$$

Therefore (5.5) holds for all t. Now we turn to (5.4). To prove it for open formulas, we observe:

($*$) If φ is $t_0 = t_1$ for some terms t_i, then

$$\begin{aligned}
\vec{a} \in \varphi^{\mathfrak{A}} &\iff {t_0}^{\mathfrak{A}}(\vec{a}\,) = {t_1}^{\mathfrak{A}}(\vec{a}\,) && \text{[by definition of } \varphi^{\mathfrak{A}}] \\
&\iff h({t_0}^{\mathfrak{A}}(\vec{a}\,)) = h({t_1}^{\mathfrak{A}}(\vec{a}\,)) && \text{[since } h \text{ is injective]} \\
&\iff {t_0}^{\mathfrak{B}}(h(\vec{a}\,)) = {t_1}^{\mathfrak{B}}(h(\vec{a}\,)) && \text{[by (5.5)]} \\
&\iff h(\vec{a}\,) \in \varphi^{\mathfrak{B}}. && \text{[by definition of } \varphi^{\mathfrak{B}}]
\end{aligned}$$

($\dagger$) If φ is $Rt_0 \cdots t_{n-1}$ for some terms t_i and predicate R, then:

$$\begin{aligned}
\vec{a} \in \varphi^{\mathfrak{A}} &\iff ({t_0}^{\mathfrak{A}}(\vec{a}\,), \dots, {t_{n-1}}^{\mathfrak{A}}(\vec{a}\,)) \in R^{\mathfrak{A}} && \text{[by def'n of } \varphi^{\mathfrak{A}}] \\
&\iff h({t_0}^{\mathfrak{A}}(\vec{a}\,), \dots, {t_{n-1}}^{\mathfrak{A}}(\vec{a}\,)) \in R^{\mathfrak{B}} && \text{[by def'n of isom.]} \\
&\iff ({t_0}^{\mathfrak{B}}(h(\vec{a}\,)), \dots, {t_{n-1}}^{\mathfrak{B}}(h(\vec{a}\,))) \in R^{\mathfrak{B}} && \text{[by (5.5)]} \\
&\iff h(\vec{a}\,) \in \varphi^{\mathfrak{B}}. && \text{[by def'n of } \varphi^{\mathfrak{B}}]
\end{aligned}$$

($\ddagger$) If (5.4) holds when φ is χ, and now φ is $\neg\chi$, then:

$$\begin{aligned}
\vec{a} \in \varphi^{\mathfrak{A}} &\iff \vec{a} \notin \chi^{\mathfrak{A}} && \text{[by def'n of } \varphi^{\mathfrak{A}}] \\
&\iff h(\vec{a}\,) \notin \chi^{\mathfrak{B}} && \text{[by inductive hypothesis]} \\
&\iff h(\vec{a}\,) \in \varphi^{\mathfrak{B}}. && \text{[by def'n of } \varphi^{\mathfrak{B}}]
\end{aligned}$$

(§) Similarly, if (5.4) holds when φ is χ or ψ, and now φ is $(\chi \to \psi)$, then:

$$\begin{aligned}
\vec{a} \notin \varphi^{\mathfrak{A}} &\iff \vec{a} \in \chi^{\mathfrak{A}} \ \ \&\ \ \vec{a} \notin \psi^{\mathfrak{A}} && \text{[by def'n of } \varphi^{\mathfrak{A}}] \\
&\iff h(\vec{a}\,) \in \chi^{\mathfrak{B}} \ \ \&\ \ h(\vec{a}\,) \notin \psi^{\mathfrak{B}} && \text{[by inductive hypothesis]} \\
&\iff h(\vec{a}\,) \notin \varphi^{\mathfrak{B}}. && \text{[by def'n of } \varphi^{\mathfrak{B}}]
\end{aligned}$$

Finally, to establish (5.4) in case h is surjective, suppose (5.4) holds when φ is an $(m+1)$-ary formula χ, and now φ is the m-ary $\exists x_m\, \chi$. We have

$$\begin{aligned}
\vec{a} \in \varphi^{\mathfrak{A}} &\iff (\vec{a}\,, b) \in \chi^{\mathfrak{A}} \text{ for some } b \text{ in } A \\
&\iff (h(\vec{a}\,), h(b)) \in \chi^{\mathfrak{B}} \text{ for some } b \text{ in } A \\
&\iff (h(\vec{a}\,), c) \in \chi^{\mathfrak{B}} \text{ for some } c \text{ in } A \\
&\iff h(\vec{a}\,) \in \varphi^{\mathfrak{B}}
\end{aligned}$$

(Note how the surjectivity of h was used.) This completes the proof. □

As an immediate consequence, we have:

Theorem 5.3.3. *If* $\mathfrak{A} \cong \mathfrak{B}$*, then* $\mathfrak{A} \equiv \mathfrak{B}$.

For other consequences, we first observe:

Lemma 5.3.4. *If $h : \mathfrak{A} \to \mathfrak{B}$, then $h(A)$ is the universe of a structure $h(\mathfrak{A})$ such that $h : \mathfrak{A} \xrightarrow{\cong} h(\mathfrak{A})$ and $h(\mathfrak{A}) \subseteq \mathfrak{B}$.*

Proof. **Exercise.** □

Theorem 5.3.5. *Suppose $h : \mathfrak{A} \to \mathfrak{B}$. Then $\mathfrak{A} \xrightarrow{\equiv} \mathfrak{B}$ if and only if $h(\mathfrak{A}) \preccurlyeq \mathfrak{B}$.*

Let the **diagram of** $\mathfrak{A}$ be the set of *open* sentences of $\mathrm{Th}(\mathfrak{A}_A)$; this set can be denoted

$$\operatorname{diag}\mathfrak{A}.$$

Then we can give the following characterization of the relations $\subseteq$ and $\preccurlyeq$: {thm:elsomb}

Theorem 5.3.6. *Suppose $h : A \to B$, and $\mathfrak{B}^*$ is the expansion of $\mathfrak{B}$ to $\mathcal{L}(A)$ such that*

$$a^{\mathfrak{B}^*} = h(a) \tag{5.6}$$

for all a in A. Then

$$\mathfrak{B}^* \vDash \operatorname{diag}\mathfrak{A} \iff h : \mathfrak{A} \to \mathfrak{B}; \tag{5.7}$$

$$\mathfrak{B}^* \vDash \mathrm{Th}(\mathfrak{A}_A) \iff h : \mathfrak{A} \xrightarrow{\equiv} \mathfrak{B}. \tag{5.8}$$

In particular, if $A \subseteq B$, then

$$\mathfrak{B} \vDash \operatorname{diag}\mathfrak{A} \iff \mathfrak{A} \subseteq \mathfrak{B};$$
$$\mathfrak{B} \vDash \mathrm{Th}(\mathfrak{A}_A) \iff \mathfrak{A} \preccurlyeq \mathfrak{B}.$$

Proof. Note that $\mathfrak{B}^* \vDash \varphi(\vec{a}\,) \iff \mathfrak{B} \vDash \varphi(h(\vec{a}\,))$. The points about elementary embeddings and substructures follow from the definitions; about embeddings and substructures, from Lemma 5.3.2. □ {cor:QE-MC}

Corollary 5.3.7. *If T is a theory admitting quantifier-elimination, then all embeddings of models of T are elementary embeddings.*

Proof. If T admits quantifier-elimination and $\mathfrak{A} \vDash T$, then $\operatorname{diag}\mathfrak{A} \vDash \mathrm{Th}(\mathfrak{A}_A)$. □

Model-theory is interesting because not all elementarily equivalent structures are isomorphic:

Example 5.3.8. We know that $\mathrm{Th}(\mathbb{Q}, <) = \mathrm{TO}^*$. Since also $(\mathbb{R}, <) \vDash \mathrm{TO}^*$, we have $(\mathbb{R}, <) \equiv (\mathbb{Q}, <)$; however, $(\mathbb{Q}, <) \not\cong (\mathbb{R}, <)$, simply because $\mathbb{R}$ is uncountable, so there is no bijection at all between $\mathbb{Q}$ and $\mathbb{R}$. •

5.4 Categoricity

The **cardinality** of a structure $\mathfrak{A}$ is the cardinality $|A|$ of its universe A. Let κ be an infinite cardinality. A theory T is called **κ-categorical** if

($*$) T has a model of cardinality κ;

(†) all models of T of cardinality κ are isomorphic (to each other).

{example:TO*}

Example 5.4.1. We shall prove later, in Theorem 8.1.1, that TO^* is ω-categorical. •

A theory is **totally categorical** if it is κ-categorical for each κ.

{example:empty}

Example 5.4.2. In the empty signature, structures are pure sets, and isomorphisms are just bijections. Hence, if $\mathcal{L} = \varnothing$, then $\mathrm{Con}_{\mathcal{L}}(\varnothing)$ is totally categorical. •

There are sentences σ_n (where $n > 0$) in the empty signature such that, for all theories T and structures $\mathfrak{A}$ of some common signature,

$$\mathfrak{A} \vDash T \cup \{\sigma_n : n > 0\} \iff \mathfrak{A} \vDash T \ \& \ |A| \geqslant \omega.$$

Indeed, let σ_n be

$$\exists x_0 \ \cdots \exists x_{n-1} \bigwedge_{i<j<n} x_i \neq x_j.$$

Moreover, for any formula φ with at most one free variable, x, if $n > 1$, we can form the sentence

$$\exists x_0 \ \cdots \exists x_{n-1} \ (\bigwedge_{i<j<n} x_i \neq x_j \wedge \bigwedge_{i<n} \varphi(x_i));$$

this sentence can be abbreviated

$$\exists^{\geqslant n} x \ \ \varphi.$$

Then

$$\mathfrak{A} \vDash \exists^{\geqslant n} x \ \varphi \iff |\varphi^{\mathfrak{A}}| \geqslant n.$$

Example 5.4.3. Suppose $\mathcal{L} = \{E\}$, where E is a binary predicate, and let T be the theory of equivalence-relations with exactly two classes, both infinite. So T has the axioms:

$$\begin{gathered}
\forall x \; x \ E \ x; \\
\forall x \ \forall y \ (x \ E \ y \rightarrow y \ E \ x); \\
\forall x \ \forall y \ \forall z \ (x \ E \ y \wedge y \ E \ z \rightarrow x \ E \ z); \\
\exists x \ \exists y \ \forall z \ (\neg(x \ E \ y) \wedge (x \ E \ z \vee y \ E \ z)) \\
\forall x \ \exists^{\geqslant n} y \ x \ E \ y
\end{gathered}$$

for each n greater than 1. Then T is ω-categorical. However, if κ is an *uncountable* cardinal, then T is not κ-categorical. For example, there is a model in which both E-classes have size ω_1 (that is, $\aleph_1$), and a model in which one class has size ω_1, the other ω. •

In a countable signature, there are at most $|2^\omega|$—that is, **continuum-many**—structures with a given countable universe A, because each symbol in the signature will be interpreted as a subset of some A^n, and there are at most continuum-many of these.

The **spectrum-function** is

$$(T, \kappa) \longmapsto I(T, \kappa),$$

where T is a theory,κ is an infinite cardinal, and $I(T, \kappa)$ is the number of non-isomorphic models of T of size κ. A theory in a countable signature is also called **countable**. If T is countable, then we have

$$1 \leqslant I(T, \omega) \leqslant |2^{\omega}|. \tag{5.9}$$

We've seen in Examples 5.4.1 and 5.4.2 that the lower bound cannot be improved. **Vaught's conjecture** is that

$$I(T, \omega) < |2^{\omega}| \implies I(T, \omega) \leqslant \omega.$$

If the Continuum Hypothesis is accepted, than this implication is trivial; the Conjecture is that the implication holds even if the Continuum Hypothesis is rejected.

The upper bound of (5.9) cannot be improved: {example:binary}

Example 5.4.4. Let $\mathcal{L}$ be $\{P_n : n \in \omega\}$, where each P_n is a unary predicate. Let T have the following axioms, where I and J are finite disjoint subsets of ω:

$$\exists x\ (\bigwedge_{i \in I} P_i x \wedge \bigwedge_{j \in J} \neg P_j x).$$

In the same way that we proved TO* admitted quantifier-elimination and was complete, we can prove that T admits QE and is complete. But T has continuum-many countably infinite models. Indeed, T has a model $\mathfrak{A}$, where $A = 2^{\omega}$, and

$$P_n{}^{\mathfrak{A}} = \{\sigma \in A : s(n) = 1\}.$$

We could replace A with the set A_0 of σ in 2^{ω} such that, for some k, if $n \geqslant k$, then $\sigma(n) = 0$. This A_0 is countable. In fact there is an injection from A_0 into $2^{<\omega}$, where

$$2^{<\omega} = \bigcup_{n \in \omega} 2^n.$$

This set is partially ordered by $\subseteq$ and is a *tree*. A *branch* of this tree is a maximal totally ordered subset; the union of a branch is an element of 2^{ω}. If σ and τ are distinct elements of 2^{ω}, then $\sigma(n) \neq \tau(n)$ for some n in ω, and then

$$\sigma \in P_n{}^{\mathfrak{A}} \iff \tau \notin P_n{}^{\mathfrak{A}}.$$

Hence, if also σ and τ are not in A_0, then $A_0 \cup \{\sigma\}$ and $\tau \cup \{\tau\}$ determine non-isomorphic models of T. Hence T has at least (and therefore exactly) continuum-many countable models, since $|2^{\omega} \smallsetminus A_0| = |2^{\omega}|$. •

For those who know some algebra:

Examples 5.4.5. A

s examples of complete T where $I(T, \omega) = \omega$, we have:

(∗) the theory of torsion-free divisible abelian groups;

(†) ACF_0, the theory of algebraically closed fields of characteristic 0.

•

Chapter 6

Compactness

We now aim to prove *compactness* for first-order logic. A subset Σ of $\mathrm{Sn}_{\mathcal{L}}$ is

($*$) **satisfiable** if it has a model;

($\dagger$) **finitely satisfiable** if every finite subset of Σ has a model.

Compactness is that every finitely satisfiable set is satisfiable.

Lemma 6.0.6. *If Σ is finitely satisfiable, but $\Sigma \cup \{\sigma\}$ is not, then $\Sigma \cup \{\neg\sigma\}$ is.*

Proof. Say Σ_0 is a finite subset of Σ such that $\Sigma_0 \cup \{\sigma\}$ has no model. Then $\Sigma_0 \models \neg\sigma$. Say Σ_1 is another finite subset of Σ. Then $\Sigma_0 \cup \Sigma_1$ has a model in which $\neg\sigma$ is true. $\square$

In proving the Completeness Theorem for propositional logic, we start from a set Σ of propositional formulas from which a formula F cannot be derived. Then $\Sigma \cup \{\neg F\}$ is consistent. We find a *maximal* consistent set Σ^* that includes $\Sigma \cup \{\neg F\}$. From Σ^* we *define* a structure A that is a model of Σ in which F is false.

We can try to do something similar to prove compactness for first-order logic. Suppose Σ is a maximal finitely satisfiable set of first-order formulas in some signature $\mathcal{L}$. (In particular then, $\sigma \in \Sigma \iff \neg\sigma \notin \Sigma$.) We can try to define an $\mathcal{L}$-structure $\mathfrak{A}$ by letting:

($*$) A be the set of constants in $\mathcal{L}$;

($\dagger$) $c^{\mathfrak{A}} = c$ for every constant c in $\mathcal{L}$;

($\ddagger$) $f^{\mathfrak{A}}(c_0, \dots, c_{n-1}) = d \iff (fc_0 \cdots c_{n-1} = d) \in \Sigma$;

($\S$) $(c_0, \dots, c_{n-1}) \in R^{\mathfrak{A}} \iff Rc_0 \cdots c_{n-1} \in \Sigma$.

We want $\mathfrak{A}$ to be a model of Σ. There are three problems:

($*$) The signature $\mathcal{L}$ might not contain any constants.

($\dagger$) Suppose $\mathcal{L}$ does contain constants c and d. We have $\mathfrak{A} \models (c = d) \iff c^{\mathfrak{A}} = d^{\mathfrak{A}} \iff c = d$. So $\mathfrak{A}$ can't be a model of Σ unless either Σ does not contain $(c = d)$, or c and d are the same symbol.

($\ddagger$) If $\mathfrak{A} \models \neg\varphi^x_c$ for every constant c in $\mathcal{L}$, then $\mathfrak{A} \models \neg\exists x\ \varphi$. However, possibly Σ contains all of the formulas $\neg\varphi^x_c$, but also $\exists x\ \varphi$.

The solution to these problems is as follows:

(∗) We expand $\mathcal{L}$ to a signature $\mathcal{L}'$ that contains infinitely many constants. Then we enlarge Σ to a maximal finitely satisfiable subset Σ' of $\mathrm{Sn}_{\mathcal{L}'}$.

(†) Letting C be the set of constants of $\mathcal{L}'$, we define an equivalence-relation E on C by

$$c\ E\ d \iff (c = d) \in \Sigma'.$$

Then we let A be, not C, but C/E.

(‡) In enlarging Σ to Σ', we ensure that, if $\exists x\ \varphi \in \Sigma'$, then $\varphi^x_c \in \Sigma'$ for some c in C.

Theorem 6.0.7 (Compactness for first-order logic)**.** *Every finitely satisfiable set of formulas (in some signature) is satisfiable.*

Proof. Suppose Σ is a finitely satisfiable subset of $\mathrm{Sn}_{\mathcal{L}}$. Let C be a set of new constants (so $\mathcal{L} \cap C = \varnothing$). For any $\mathcal{L}$-structure $\mathfrak{A}$, there is some a in A; so we can expand $\mathfrak{A}$ to an $\mathcal{L} \cup C$-structure $\mathfrak{A}'$ by defining

$$c^{\mathfrak{A}'} = a$$

for all c in C. In particular, Σ is still finitely satisfiable *as a set of sentences of* $\mathcal{L}'$.

We'll assume that $\mathcal{L}$ is countable (although the general case would proceed similarly). So we can enumerate $\mathrm{Sn}_{\mathcal{L}\cup C}$ as $\{\sigma_n : n \in \omega\}$, and C as $\{c_n : n \in \omega\}$. We shall define a chain

$$\Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \subseteq \cdots,$$

where each Σ_k is finitely satisfiable, and only finitely many constants in C appear in formulas in Σ_k. The recursive definition is the following:

(∗) $\Sigma_0 = \Sigma$. (By assumption, Σ_0 is finitely satisfiable, and it contains no constants of C.)

(†) Assume Σ_{2n} has been defined as required. Then define

$$\Sigma_{2n+1} = \begin{cases} \Sigma_{2n} \cup \{\sigma_n\}, & \text{if this is finitely satisfiable;} \\ \Sigma_{2n}, & \text{if not.} \end{cases}$$

Then Σ_{2n+1} is as required.

(‡) Suppose Σ_{2n+1} has been defined as required. Suppose also $\sigma_n \in \Sigma_{2n+1}$, and σ_n is $\exists x\ \varphi$ for some φ. The set of m such that c_m does *not* appear in a formula in Σ_{2n+1} has a least element, k. Then the set $\Sigma_{2n+1} \cup \{\varphi^x_{c_k}\}$ is finitely satisfiable. For, if Γ is a finite subset of Σ_{2n+1}, then it has a model $\mathfrak{A}$. Then $\mathfrak{A} \vDash \varphi^x_a$ for some a in A; so we can expand $\mathfrak{A}$ to a model of $\Sigma_{2n+1} \cup \{\varphi^x_{c_k}\}$ by interpreting c_k as a. In this case we define

$$\Sigma_{2n+2} = \Sigma_{2n+1} \cup \{\varphi^x_{c_k}\};$$

otherwise, let $\Sigma_{2n+2} = \Sigma_{2n+1}$. In either case, Σ_{2n+2} is as desired.

Now we define

$$\Sigma^* = \bigcup_{n\in\omega} \Sigma_n.$$

This is finitely satisfiable, since each finite subset is a subset of some Σ_n. Suppose $\Sigma^* \cup \{\sigma\}$ is finitely satisfiable. But σ is σ_n for some n, and $\Sigma_{2n} \cup \{\sigma\}$ is finitely satisfiable, so $\sigma \in \Sigma_{2n+1}$, and $\sigma \in \Sigma^*$. So Σ^* is a maximal finitely satisfiable set.

We now define a structure $\mathfrak{A}$ of $\mathcal{L} \cup C$ that will turn out to be a model of Σ:

We first define

$$E = \{(c, d) \in C^2 : (c = d) \in \Sigma^*\}.$$

Then E is an equivalence-relation on C (**exercise**). So, we can let

$$A = C/E.$$

Let the E-class of c be denoted $[c]$. We can define

$$c^{\mathfrak{A}} = [c].$$

If R is an n-ary predicate in $\mathcal{L}$, we define

$$R^{\mathfrak{A}} = \{([c_0], \dots, [c_{n-1}]) \in A^n : (Rc_0 \cdots c_{n-1}) \in \Sigma^*\}.$$

This means

$$(Rc_0 \cdots c_{n-1}) \in \Sigma^* \implies ([c_0], \dots, [c_{n-1}]) \in R^{\mathfrak{A}}.$$

In fact the converse holds too; that is,

$$c_0 \mathrel{E} d_0 \;\&\; \dots \;\&\; c_{n-1} \mathrel{E} d_{n-1} \;\&\; (Rc_0 \cdots c_{n-1}) \in \Sigma^* \implies (Rd_0 \cdots d_{n-1}) \in \Sigma^*$$

(**exercise**). If f is an n-ary function-symbol in $\mathcal{L}$, then $(\exists x\; fc_0 \cdots c_{n-1} = x) \in \Sigma^*$ (since the sentence is true in every structure), so $(fc_0 \cdots c_{n-1} = d) \in \Sigma^*$ for some d in C. Moreover,

$$\begin{aligned} c_0 \mathrel{E} c_0' \;\&\; \dots \;\&\; c_{n-1} \mathrel{E} c_{n-1}' \;\&\; (fc_0 \cdots c_{n-1} = d) \in \Sigma^* \;\& \\ (fc_0' \cdots c_{n-1}' = d') \in \Sigma^* \implies d \mathrel{E} d' \end{aligned}$$

(**exercise**). Hence we can define

$$f^{\mathfrak{A}} = \{([c_0], \dots, [c_{n-1}], [d]) \in A : (fc_0 \cdots c_{n-1} = d) \in \Sigma^*\}.$$

Note then

$$f^{\mathfrak{A}}[c_0] \cdots [c_{n-1}] = [d] \iff (fc_0 \cdots c_{n-1} = d) \in \Sigma^*$$

(**exercise**). Finally, if c is a constant of $\mathcal{L}$, we can consider it as a nullary function-symbol, obtaining the interpretation

$$c^{\mathfrak{A}} = [d] \iff (c = d) \in \Sigma^*.$$

So we have $\mathfrak{A}$. It remains to show $\mathfrak{A} \models \Sigma^*$. We shall do this by showing

$$\mathfrak{A} \models \sigma \iff \sigma \in \Sigma^* \tag{6.1}$$

for all sentences σ of $\mathcal{L} \cup C$, by induction on the length of σ.

We need a preliminary observation: If t is a term with no variables, and $c \in C$, then

$$t^{\mathfrak{A}} = [c] \iff (t = c) \in \Sigma^*$$

(**exercise**). Now suppose σ is the atomic sentence $Rt_0 \cdots t_{n-1}$, and ${t_i}^{\mathfrak{A}} = [c_i]$ for each i in n. Then

$$\begin{aligned}
\mathfrak{A} \vDash \sigma &\iff ({t_0}^{\mathfrak{A}}, \dots, {t_{n-1}}^{\mathfrak{A}}) \in R^{\mathfrak{A}} \\
&\iff ([c_0], \dots, [c_{n-1}]) \in R^{\mathfrak{A}} \\
&\iff (Rc_0 \cdots c_{n-1}) \in \Sigma^* \\
&\iff \sigma \in \Sigma^*.
\end{aligned}$$

If instead σ is the equation $t_0 = t_1$, then

$$\begin{aligned}
\mathfrak{A} \vDash \sigma &\iff {t_0}^{\mathfrak{A}} = {t_1}^{\mathfrak{A}} \\
&\iff [c_0] = [c_1] \\
&\iff (c_0 = c_1) \in \Sigma^* \\
&\iff \sigma \in \Sigma^*.
\end{aligned}$$

Now suppose that (6.1) holds when σ has the length of τ, θ or φ:

($*$) If σ is $\neg\tau$, then

$$\mathfrak{A} \vDash \sigma \iff \mathfrak{A} \nvDash \tau \iff \tau \notin \Sigma^* \iff \sigma \in \Sigma^*$$

by maximality of Σ.

(†) If σ is $(\tau \to \theta)$, then

$$\begin{aligned}
\mathfrak{A} \nvDash \sigma &\iff \mathfrak{A} \vDash \tau \ \& \ \mathfrak{A} \vDash \not\theta \\
&\iff \tau \in \Sigma^* \ \& \ \theta \notin \Sigma^* \\
&\iff \sigma \in \Sigma^*
\end{aligned}$$

by maximality of Σ^*.

(‡) If σ is $\exists x\ \varphi$, then

$$\begin{aligned}
\mathfrak{A} \vDash \sigma &\iff \mathfrak{A} \vDash \varphi^x_c \text{ for some } c \text{ in } C \\
&\iff \varphi^x_c \in \Sigma^* \text{ for some } c \text{ in } C \\
&\iff \exists x\ \varphi \in \Sigma^*
\end{aligned}$$

by definition of Σ^*.

By induction, (6.1) holds for all σ, so $\mathfrak{A} \vDash \Sigma^*$. □

In the proof, we introduced a set C of new constants such that $|C| = |\mathrm{Sn}_{\mathcal{L}}|$. We can denote $|\mathrm{Sn}_{\mathcal{L}}|$ by $|\mathcal{L}|$. For the model $\mathfrak{A}$ of Σ produced, we have $|A| \leqslant |C| = |\mathcal{L}|$.

Theorem 6.0.8. *If T is a theory such that, for all n in ω, there is a model of T of size greater than n, then T has an infinite model.*

Proof. For each n in ω, introduce a new constant c_n. Every model of the theory $T \cup \{c_i \neq c_j : i < j < \omega\}$ is infinite. Also this theory *has* models, by Compactness, since the theory is finitely satisfiable. Indeed, every finite subset of the theory is a subset of $T \cup \{c_i < c_j : i < j < n\}$ for some n. We can expand a model of T of size greater than n to a model of the larger theory by interpreting each c_i by a different element of the universe. □

Example 6.0.9. Let **K** be the class of finite fields (considered as structures in the signature $\{+, -, \cdot, 0, 1\}$). Then $\mathrm{Th}(\mathbf{K})$ has infinite models; these are called **pseudo-finite** fields. Every field F has a **characteristic**: If

$$F \vDash \underbrace{1 + \cdots + 1}_{p} = 0$$

for some prime number p, then p is the characteristic of F, or $\operatorname{char} F = p$; if there is no such p, then $\operatorname{char} F = 0$. The field F is **perfect** if either:

(∗) $\operatorname{char} F = 0$; or

(†) $\operatorname{char} F = p$ and every element of F has a p-th root.

Then perfect fields are precisely the fields that satisfy the axioms

$$\forall x\ \exists y\ (\underbrace{1 + \cdots + 1}_{p} = 0 \to y^p = x).$$

Now, if F is finite, then $\operatorname{char} F = p$ for some prime p, and the function $x \mapsto x^p$ is an **automorphism** of F, that is, an isomorphism from F to itself. This shows F is perfect. Therefore the pseudo-finite fields are also perfect. In fact, axioms can be written for the theory of pseudo-finite fields (James Ax, 1968). •

Another field-theoretic application of Compactness is:

Example 6.0.10. An **ordered field** is a structure $\mathfrak{F}$ or $(F, +, -, \cdot, 0, 1, <)$ such that:

(∗) $(F, +, -, \cdot, 0, 1)$ is a field;

(†) $(F, <)$ is a total order;

(‡) $\mathfrak{F} \vDash \forall x\ \forall y\ (0 < x \wedge 0 < y \to 0 < x + y \wedge 0 < x \cdot y)$;

(§) $\mathfrak{F} \vDash \forall x\ (x < 0 \to 0 < -x)$.

An ordered field must have characteristic 0 (why?); hence $\mathbb{Q}$ can be treated as a sub-field of it. In an ordered field, the formula $0 < x$ defines the set of **positive** elements. The ordered field $\mathfrak{F}$ is **Archimedean** if, for all positive a and b in F, there is a natural number n such that

$$\mathfrak{F} \vDash a < \underbrace{b + \cdots + b}_{n}.$$

Then $\mathbb{R}$ is an Archimedean ordered field. However, there is an ordered field $\mathfrak{F}$ such that $\mathfrak{F} \equiv \mathbb{R}$, but $\mathfrak{F}$ is not Archimedean. Indeed, let c be a new constant. Then the theory

$$\mathrm{Th}(\mathbb{R}) \cup \{n < c : n \in \omega\}$$

is finitely satisfiable, since for every finite subset Σ of this theory, $\mathbb{R}$ itself expands to a model of Σ. So the theory has a model $\mathfrak{F}$, by Compactness; but

$$\mathfrak{F} \models \underbrace{1 + \cdots + 1}_{n} < c$$

for all n in ω. •

{thm:LST}

Theorem 6.0.11 (Löwenheim–Skolem–Tarski)**.** *Suppose $\mathfrak{A}$ is an infinite $\mathcal{L}$-structure, and κ is an infinite cardinal such that $|\mathcal{L}| \leqslant \kappa$. Then there is an $\mathcal{L}$-structure $\mathfrak{B}$ such that $|B| = \kappa$ and $\mathfrak{A} \equiv \mathfrak{B}$.*

Proof. Introduce κ-many new constants c_α (where $\alpha < \kappa$). In the Compactness Theorem, let Σ be $\mathrm{Th}(\mathfrak{A}) \cup \{c_\alpha \neq c_\beta : \alpha < \beta < \kappa\}$. This set is finitely satisfiable. Indeed, any finite subset is included in a subset $\mathrm{Th}(\mathfrak{A}) \cup \{c_{\alpha_i} \neq c_{\alpha_j} : i < j < n\}$ for some finite subset $\{\alpha_0, \dots, \alpha_{n-1}\}$ of κ. Then $\mathfrak{A}$ expands to a model of this set of sentences, once we interpret each constant c_{α_i} as a different element of A. (Since A is infinite, we can do this.) Therefore Σ is finitely satisfiable. The *proof* of Compactness now produces a model of Σ of size κ. □

{thm:Vaught}

Theorem 6.0.12 (Vaught)**.** *Suppose T is a finitely satisfiable theory of $\mathcal{L}$, and $|\mathcal{L}| \leqslant \kappa$. Then T is complete, provided:*

(∗) *T has no finite models;*

(†) *T is κ-categorical.*

Proof. Suppose T is finitely satisfiable, but has no finite models, but is not complete. By Compactness, T does have models. Then for some sentence σ, neither σ nor $\neg\sigma$ is a consequence of T. Hence, both $T \cup \{\neg\sigma\}$ and $T \cup \{\sigma\}$ have models. By Löwenheim–Skolem–Tarski, they have models of size κ. These models are not elementarily equivalent, so they are not isomorphic; this means T is not κ-categorical. □

Examples 6.0.13. [

]

(1) To prove that TO^* is complete, it is enough to show that every model is infinite, and that every countable model is isomorphic to $(\mathbb{Q}, <)$.

(2) If a real vector-space V has positive dimension κ, then

$$|V| = \kappa \cdot |2^\omega| = \max(\kappa, |2^\omega|).$$

A space of dimension 0 is the the **trivial** space, namely the space containing only the 0-vector; this space has size 1. Real vector-spaces of the same dimension are isomorphic Hence the theory of real vector-spaces is κ-categorical if $\kappa > |2^\omega|$. Therefore the theory of *non-trivial* real vector-spaces is complete. •

6.1 Additional exercises

(1) Show that every Archimedean ordered field is elementarily equivalent to some *countable, non-Archimedean* ordered field.

(2) Show that every non-Archimedean ordered field contains **infinitesimal** elements, that is, positive elements a that are less than every positive rational number.

(3) Find an example of a non-Archimedean ordered field.

(4) The **order** of an element g of a group is the size of the subgroup $\{g^n : n \in \mathbb{Z}\}$ that g generates. In a **periodic** group, all elements have finite order. Suppose G is a periodic group in which there is no finite upper bound on the orders of elements. Show that $G \equiv H$ for some non-periodic group H.

(5) Suppose $(X, <)$ is an infinite total order in which X is *well-ordered* by $<$. Show that there is a total order $(X^*, <^*)$ such that

$$(X, <) \equiv (X^*, <^*),$$

but X^* is *not* well-ordered by $<^*$.

Chapter 7

Completeness

We now aim to establish a complete proof-system for first-order logic. The result is Theorem 7.6.8 on p. 77. The proof of this theorem follows the pattern of our proof of Compactness.

First-order logic is based on propositional logic. It will be useful to have a general description of *logics* that encompasses both propositional and first-order logic. So, this is where we begin. All sections following § 7.3 concern first-order logic, unless otherwise noted.

There are a few exercises, on pp. 68, 71, 72, 73, 73, 73 and 76.

7.1 Logic in general

A logic has an **alphabet**, which is just a certain non-empty set; the members of this set can be called the **symbols** of the logic. These symbols can be put together to form **strings**. If we want a formal definition, we can say that such a string is a **finite, non-empty sequence** of symbols of the logic; that is, the string is a function $k \mapsto s_k$ from $\{0, 1, \ldots, n\}$ into the alphabet, for some n in ω. We usually write this function as

$$s_0 s_1 \cdots s_n;$$

this the result of **juxtaposing** the symbols s_k in the prescribed order. Such a string has **sub-strings**, namely the strings

$$s_\ell s_{\ell+1} \cdots s_m,$$

where $0 \leqslant \ell \leqslant m \leqslant n$; the sub-string is **proper** if $0 < \ell$ or $m < n$. Certain strings will be *formulas* of the logic. In particular, certain strings will be **atomic** formulas. Some **rules of construction** are specified for converting certain finite sets of strings into other strings. Then a **formula** of the logic is a member of the smallest set X of strings such that:

($*$) all atomic formulas are in X; and

(†) X contains every string that results from applying a rule of construction to a set of elements of X.

Hence properties of all formulas can be proved by *induction*.

Moreoever, it is required that, for every formula that is not atomic, there is exactly one rule of construction and one set of formulas such that the original formula results from applying that rule to that set. This is the principle of **uniquely readability** as formulas; it makes possible the *recursive* definition of functions on the set of formulas.

For any logic, a **proof-system** consists of:

($*$) **axioms**, which are just certain formulas of the logic;

(†) **rules of inference**, that is, ways of inferring certain formulas from certain *finite* sets of formulas.

So the notions of *axiom* and *rule of inference* are parallel to the notions of *atomic formula* and *rule of construction*. However, in a proof-system, there is no requirement corresponding to unique readability.

Let $\mathcal{S}$ be proof-system. A **deduction** or **formal proof in $\mathcal{S}$ of** the formula φ **from** the set Φ of formulas is a sequence

$$\psi_0, \dots, \psi_n$$

of formulas where ψ_n is φ, and for each k such that $k \leqslant n$, one of the following holds:

($*$) $\psi_k \in \Phi$, or

(†) ψ_k is an axiom of $\mathcal{S}$, or

(‡) ψ_k follows from some subset of $\{\psi_j : j < k\}$ by one of the rules of inference of $\mathcal{S}$.

To denote that such a deduction exists, we can write

$$\Phi \vdash_{\mathcal{S}} \varphi.$$

Then we can say that φ is **deducible** from Φ in $\mathcal{S}$. In case Φ is empty, we can just write

$$\vdash_{\mathcal{S}} \varphi,$$

and we can call φ a **theorem** of $\mathcal{S}$.

{lem:gen} Here are some basic facts:

Lemma 7.1.1.

($*$) *Every non-empty initial segment of a deduction is also a deduction;*

(†) *if $\Phi \vdash_{\mathcal{S}} \varphi$ and $\Phi \subseteq \Phi^*$, then $\Phi^* \vdash_{\mathcal{S}} \varphi$;*

(‡) *if $\Phi \vdash_{\mathcal{S}} \varphi$, then $\Phi_0 \vdash_{\mathcal{S}} \varphi$ for some* finite *subset Φ_0 of Φ;*

(§) *if $\Phi \vdash_{\mathcal{S}} \psi$ for each ψ in Ψ, and $\Psi \vdash_{\mathcal{S}} \chi$, then $\Phi \vdash_{\mathcal{S}} \chi$.*

{ex1} *Proof.* **Exercise.** □

7.2 Propositional logic

We shall work here with the propositional logic whose alphabet consists of:

($*$) the propositional variables P_k, where $k \in \omega$;

(†) the connectives $\neg$ and $\rightarrow$;

(‡) the left bracket (and the right bracket).

The atomic formulas are then the propositional variables. There are two rules of construction:

($*$) From the string A, construct $\neg A$.

(†) From the strings A and B, construct $(A \rightarrow B)$.

Note that the same formula might be both $(A \rightarrow B)$ and $(C \rightarrow D)$ for some *strings* A, B, C and D such that A is not C. But if all of these strings are *formulas*, then (as one can prove) A must be C. We use F and G and H as syntactical variables for propositional formulas.

In propositional logic, there is a notion of **truth**, which we can develop as follows. If $S \subseteq \omega$, let 2^S be the set of functions from S to 2. We can consider 2 as the universe of the field $\mathbb{F}_2$; then a ring-structure on 2^S is induced. If F is a propositional formula, and all variables appearing in F are in S, then there is a function $\hat{F}$ from 2^S into 2, as given by the following recursive definition:

($*$) If F is P_k, then $\hat{F}(\alpha) = \alpha(k)$ for all α in 2^ω.

(†) If F is $\neg G$, then $\hat{F} = 1 + \hat{G}$.

(‡) If F is $(G \rightarrow H)$, then $\hat{F} = 1 + \hat{G} \cdot (1 + \hat{H})$.

Suppose S is the set of variables actually appearing in F, and $\hat{F}(\alpha) = 1$ for all α in 2^S; then F is called a **tautology**.

An element α of 2^ω can be called a **structure** for propositional logic. (Alternatively, the set $\{P_n : \alpha(n) = 1\}$ can be called the structure; each one determines the other.) Then a formula F is **true in** α if $\hat{F}(\alpha) = 1$. If every formula in a set Φ of formulas is true in a structure α, then α is a **model** of Φ. If F is true in every model of Φ, then we say that F is a **logical consequence** of Φ, or that Φ **entails** F, and we write

$$\Phi \vDash F.$$

A formula F is **valid**, or is a **validity**, if it is true in all structures; in that case, we write

$$\vDash F.$$

A proof-system $\mathcal{S}$ for propositional logic is called:

($*$) **sound**, if $\Phi \vDash \varphi$ whenever $\Phi \vdash_{\mathcal{S}} \varphi$;

(†) **complete**, if $\Phi \vdash_{\mathcal{S}} \varphi$ whenever $\Phi \vDash \varphi$.

{lem:sound}

Lemma 7.2.1. *Let $\mathcal{S}$ be a proof-system for propositional logic. Then $\mathcal{S}$ is sound if and only if:*

($*$) *each axiom of $\mathcal{S}$ is valid;*

(†) *$\Phi \vDash \varphi$ whenever φ can be inferred from Φ by one of the rules of inference of $\mathcal{S}$.*

Proof. Suppose $\mathcal{S}$ is sound. If φ is an axiom of $\mathcal{S}$, then the one-term sequence φ is a deduction of φ from $\varnothing$, so $\vdash_{\mathcal{S}} \varphi$ and therefore $\vDash \varphi$. Suppose, instead, that φ can be inferred from Φ by one of the rules of inference of $\mathcal{S}$. Then Φ is a finite set $\{\psi_0, \dots, \psi_n\}$, so the sequence

$$\psi_0, \dots, \psi_n, \varphi$$

is a deduction of φ from Φ in $\mathcal{S}$. Hence $\Phi \vdash_{\mathcal{S}} \varphi$, and therefore $\Phi \vDash \varphi$.

The converse is proved by induction on the lengths of deductions. Suppose that each axiom of $\mathcal{S}$ is valid, and $\Phi \vDash \varphi$ whenever φ can be inferred from Φ by one of the rules of inference of $\mathcal{S}$. As an inductive hypothesis, suppose $\Phi \vDash \varphi$ whenever φ has a deduction in $\mathcal{S}$ from Φ of length less than $n+1$. Now say the sequence

$$\psi_0, \dots, \psi_{n-1}, \varphi$$

of length $n+1$ is a deduction in $\mathcal{S}$ from Φ. If $\varphi \in \Phi$, then $\Phi \vDash \varphi$ trivially. If φ is an axiom of $\mathcal{S}$, then $\vDash \varphi$ by assumption, so $\Phi \vDash \varphi$. The remaining possibility is that φ can be inferred from some subset Γ of $\{\psi_k : k < n\}$ by a rule of inference of $\mathcal{S}$. Then $\Gamma \vDash \varphi$ by assumption. Also, $\Phi \vDash \psi_k$ for each ψ_k in Γ by inductive hypothesis, since each ψ_k has a proof from Φ of length $k+1$, namely

$$\psi_0, \dots, \psi_k.$$

Hence every model of Φ is a model of Γ, and so φ is true in this model; that is, $\Phi \vDash \varphi$. □

Let us also note that if a proof-system is complete, then so is every proof-system obtained by addition of new axioms or rules of inference.

In the only proof-system for first-order logic that we shall consider,

- (∗) the axioms are just the tautologies;
- (†) the only rule of inference is *modus ponens*, that is, G can be inferred from $\{F, (F \to G)\}$.

If, in this system, F is deducible from the set Φ of formulas, then we can just write

$$\Phi \vdash F$$

(since we shall consider no other proof-systems for propositional logic). We have proved (in class) that this system is sound and complete.

7.3 First-order logic

{sect:1st}

The foregoing notions in propositional logic generalize to first-order logic. For us, the alphabet for a first-order logic will consist of:

- (∗) the symbols in a signature $\mathcal{L}$ for the logic;
- (†) individual variables v_k, where $k \in \omega$;
- (‡) the Boolean connectives $\neg$ and $\to$;
- (§) the quantifier $\exists$;

(¶) the brackets (and).

The set of formulas of the resulting logic can be denoted

$$\mathrm{Fm}_{\mathcal{L}}\,.$$

Certain formulas are *sentences*; the set of them is

$$\mathrm{Sn}_{\mathcal{L}}\,.$$

We do not have proof by induction on this set, since sentences can be constructed from formulas that are not sentences. However, we can still define proof-systems for $\mathrm{Sn}_{\mathcal{L}}$. (Alternatively, we could define a proof-system for $\mathrm{Fm}_{\mathcal{L}}$.)

There are **$\mathcal{L}$-structures** $\mathfrak{A}$, and then for each sentence σ of $\mathcal{L}$, there is an element $\sigma^{\mathfrak{A}}$ of 2. Then σ is **true in** $\mathfrak{A}$ if $\sigma^{\mathfrak{A}} = 1$. The notions of **model**, **entailment**, **validity**, **soundness** and **completeness** can now be defined as for propositional logic. Hence we have Lemma 7.2.1 for $\mathrm{Sn}_{\mathcal{L}}$ in addition to propositional logic.

To *prove* that a certain proof-system for $\mathrm{Sn}_{\mathcal{L}}$ is complete, we shall use the method first expounded by Leon Henkin, in [5]. (Henkin's proof was a part of his doctoral thesis; see [6]. We have already used Henkin's method to prove Compactness.) The particular treatment in these notes owes something to Shoenfield's in [13]. I introduce the notions of *tautological* and *deductive* completeness merely to make our ultimate proof-system seem natural.

If F is an n-ary formula $F(P_0, \dots, P_{n-1})$ of propositional logic, and $\sigma_k \in \mathrm{Sn}_{\mathcal{L}}$, then by substitution we can form the sentence

$$F(\sigma_0, \dots, \sigma_{n-1})$$

of $\mathcal{L}$. If F is a tautology, then $F(\sigma_0, \dots, \sigma_{n-1})$ can be called a **tautology** of $\mathrm{Sn}_{\mathcal{L}}$. {lem:validities}

Lemma 7.3.1. *Tautologies of* $\mathrm{Sn}_{\mathcal{L}}$ *are validities.*

Proof. We can prove by induction on propositional formulas F that, if F is $F(P_0, \dots, P_{n-1})$, then for all sentences σ_k of $\mathrm{Sn}_{\mathcal{L}}$, and all $\mathcal{L}$-structure $\mathfrak{A}$,

$$F(\sigma_0, \dots, \sigma_{n-1})^{\mathfrak{A}} = \hat{F}({\sigma_0}^{\mathfrak{A}}, \dots, {\sigma_{n-1}}^{\mathfrak{A}}).$$

(Details are an **exercise**.) The claim follows immediately from this. □ {ex2}

7.4 Tautological completeness

Suppose $\mathcal{S}$ is a proof-system for $\mathrm{Sn}_{\mathcal{L}}$ such that, if F_0, ..., F_k are n-ary propositional formulas, and

$$\{F_0, \dots, F_{k-1}\} \vDash F_k, \tag{7.1}$$

and $\sigma_0, \dots, \sigma_{n-1} \in \mathrm{Sn}_{\mathcal{L}}$, then

$$\{F_0(\sigma_0, \dots, \sigma_{n-1}), \dots, F_{k-1}(\sigma_0, \dots, \sigma_{n-1})\} \vdash_{\mathcal{S}} F_k(\sigma_0, \dots, \sigma_{n-1}); \tag{7.2}$$

let us say then that $\mathcal{S}$ is **tautologically complete**.

{lem:1}

Lemma 7.4.1. *Let $\mathcal{S}$ be a proof-system for* $\mathrm{Sn}_{\mathcal{L}}$*. Then $\mathcal{S}$ is tautologically complete if and only if:*

(∗) $\vdash_{\mathcal{S}} \sigma$ *for all tautologies σ of* $\mathrm{Sn}_{\mathcal{L}}$*, and*

(†) $\{\sigma, \sigma \to \tau\} \vdash_{\mathcal{S}} \tau$ *for all σ and τ in* $\mathrm{Sn}_{\mathcal{L}}$*.*

Proof. If $\mathcal{S}$ is tautologically complete, then immediately all tautologies are theorems; the other condition follows since $\{P_0, P_0 \to P_1\} \vDash P_1$.

To prove the converse, we can use our complete proof-system for propositional logic: Suppose we have (7.1) above. Then F_k has a a formal proof from $\{F_0, \dots, F_{k-1}\}$. Say this proof is

$$G_0, \dots, G_m.$$

Then G_m is F_k. We proceed by induction on m. There are three possibilities:

(∗) If $F_k \in \{F_0, \dots, F_{k-1}\}$, then trivially (7.2) follows.

(†) If F_k is a tautology, then $\vdash_{\mathcal{S}} F_k(\vec{\sigma}\,)$ by assumption, so (7.2).

(‡) If G_j is $(G_i \to F_k)$ for some i and j in m, then, by inductive hypothesis, we have

$$\{F_0(\vec{\sigma}\,), \dots, F_{k-1}(\vec{\sigma}\,)\} \vdash_{\mathcal{S}} G_i(\vec{\sigma}\,);$$
$$\{F_0(\vec{\sigma}\,), \dots, F_{k-1}(\vec{\sigma}\,)\} \vdash_{\mathcal{S}} G_j(\vec{\sigma}\,);$$

hence (7.2) by assumption (and Lemma 7.1.1).

In all cases then, (7.2) follows. □

It should be clear that a complete proof-system is tautologically complete. The converse fails:

Example 7.4.2. The proof-system in which all tautologies are axioms and *modus ponens* is the only rule of inference is not complete, since it cannot be used to prove the validity $\exists x\; x = x$. Indeed, the theorems of this proof-system are just the tautologies (as one can show); but $\exists x\; x = x$ is not a tautology. •

Let $\perp$ be the negation of a tautology, say

$$\neg(\exists x\; x = x \to \exists x\; x = x).$$

Henceforth, let $\Sigma \subseteq \mathrm{Sn}_{\mathcal{L}}$ and $\sigma \in \mathrm{Sn}_{\mathcal{L}}$.

{lem:2}

Lemma 7.4.3. *In a tautologically complete proof-system $\mathcal{S}$, the following are equivalent:*

(∗) $\Sigma \vdash \neg\sigma$ *for some σ in Σ;*

(†) $\Sigma \vdash \sigma$ *and* $\Sigma \vdash \neg\sigma$ *for some σ in* $\mathrm{Sn}_{\mathcal{L}}$*;*

(‡) $\Sigma \vdash \sigma$ *for every σ in* $\mathrm{Sn}_{\mathcal{L}}$*;*

(§) $\Sigma \vdash \perp$.

{ex3} *Proof.* **Exercise.** (There is a corresponding lemma for propositional logic.) □

If $\Sigma \vdash_{\mathcal{S}} \bot$, then Σ is **inconsistent** in $\mathcal{S}$; otherwise, it is **consistent**.

Lemma 7.4.4. *In a complete proof-system, every consistent subset of* $\mathrm{Sn}_{\mathcal{L}}$ *has a model.*

Proof. If $\mathcal{S}$ is complete, but Σ has no model, then $\Sigma \vDash \bot$, so $\Sigma \vdash_{\mathcal{S}} \bot$ by completeness, so Σ is inconsistent. □

The converse of the lemma may fail, even if the proof-system is required to be tautologically complete:

Example 7.4.5. Let the axioms of a proof-system $\mathcal{S}$ be the tautologies, and let the rules of inference be *modus ponens*, along with the rule that $\bot$ can be inferred from every finite set that has no model. (Note however that this is not a *syntactical* rule: it is not based directly on the form of sentences.) By the Compactness Theorem of first-order logic, *every* set with no model is inconsistent in this theory; therefore all consistent sets have models. However, the validity $\exists x\ x = x$ is not a theorem of $\mathcal{S}$. (**Exercise:** show this.) • {ex4}

7.5 Deductive completeness

Let a proof-system $\mathcal{S}$ be called **deductively complete** if $\Sigma \vdash_{\mathcal{S}} (\sigma \to \tau)$ whenever $\Sigma \cup \{\sigma\} \vdash_{\mathcal{S}} \tau$.

{lem:4}

Lemma 7.5.1. *A tautologically and deductively complete proof-system in which every consistent set has a model is complete.*

Proof. Suppose $\mathcal{S}$ is such a system, and $\Sigma \cup \{\neg\sigma\}$ is inconsistent in $\mathcal{S}$. Then $\Sigma \cup \{\neg\sigma\} \vdash_{\mathcal{S}} \sigma$ by Lemma 7.4.3, so $\Sigma \vdash_{\mathcal{S}} (\neg\sigma \to \sigma)$ by deductive completeness. But $(\neg\sigma \to \sigma) \to \sigma$ is a tautology, so $\Sigma \vdash_{\mathcal{S}} \sigma$ by tautological completeness.

Therefore, if $\Sigma \nvdash_{\mathcal{S}} \sigma$, then $\Sigma \cup \{\neg\sigma\}$ is consistent, so it has a model by assumption; this shows $\Sigma \nvDash \sigma$. □

{lem:5}

Lemma 7.5.2. *A tautologically complete proof-system whose only rule of inference is* modus ponens *is deductively complete.*

Proof. **Exercise.** (See the Deduction Theorem of propositional logic.) □ {ex5}

Lemma 7.5.3. *Suppose* $\Sigma \subseteq \mathrm{Sn}_{\mathcal{L}}$ *and* Σ *is consistent in a tautologically and deductively complete proof-system. The following are equivalent:*

(∗) *If* $\Sigma \subseteq \Gamma \subseteq \mathrm{Sn}_{\mathcal{L}}$ *and* Γ *is consistent, then* $\Gamma = \Sigma$.

(†) $\neg\sigma \in \Sigma \iff \sigma \notin \Sigma$ *for all* σ *in* $\mathrm{Sn}_{\mathcal{L}}$.

Proof. **Exercise.** □ {ex6}

A set Σ meeting one of the conditions in the lemma can be called **maximally** consistent.

7.6 Completeness

By Lemma 7.4.1, we know of one tautologically complete proof-system, namely, the system whose axioms are the tautologies, and whose rule of inference is *modus ponens*. Let $\mathcal{S}$ be this system. Then $\mathcal{S}$ is deductively complete, by Lemma 7.5.2, and is sound, by Lemmas 7.2.1 and 7.3.1. Moreover, soundness and deductive completeness are preserved if we add new valid axioms to $\mathcal{S}$. Now we shall see which valid axioms we can add in order to ensure that every consistent set has a model; then we shall have a complete system by Lemma 7.5.1.

We follow the proof of the Compactness Theorem, replacing 'finitely satisfiable' with 'consistent'. We assume that $\mathcal{L}$ is countable. Suppose Σ is a consistent subset of $\mathrm{Sn}_{\mathcal{L}}$. We introduce an infinite set C of new constants and enumerate $\mathrm{Sn}_{\mathcal{L}\cup C}$ as $\{\sigma_n : n \in \omega\}$. We construct a chain

$$\Sigma = \Sigma_0 \subseteq \Sigma_1 \subseteq \Sigma_2 \subseteq \cdots$$

where

$$\Sigma_{2n+1} = \begin{cases} \Sigma_{2n} \cup \{\sigma_n\}, & \text{if this is consistent;} \\ \Sigma_{2n}, & \text{otherwise.} \end{cases}$$

If σ_n is $\exists x\ \varphi$, and this is in Σ_{2n+1}, then we want to define Σ_{2n+2} as

$$\Sigma_{2n+1} \cup \{\varphi^x_c\},$$

where c is a variable not used in Σ_{2n+1}. But we need to know that this set is consistent. For this we assume, as axioms of $\mathcal{S}$, the sentences

$$(\varphi^x_c \to \chi) \to \exists x\ \varphi \to \chi, \tag{7.3}$$

where c is a variable not appearing in χ. Note that these axioms are valid. We now have:

Lemma 7.6.1. *If Γ is consistent and contains $\exists x\ \varphi$, and c does not appear in Γ, then $\Gamma \cup \{\varphi^x_c\}$ is consistent.*

Proof. Suppose it's not. Then

$$\{\psi_0, \dots, \psi_{k-1}\} \cup \{\varphi^x_c\} \vdash_{\mathcal{S}} \bot$$

for some ψ_i in Γ. By deductive completeness,

$$\vdash_{\mathcal{S}} \varphi^x_c \to \psi_0 \to \cdots \to \psi_{k-1} \to \bot, \tag{7.4}$$

where the notational convention is that a terminal string $\chi_0 \to \chi_1 \to \chi_2$ stands for the formula $(\chi_0 \to (\chi_1 \to \chi_2))$. We can re-write (7.4) as

$$\vdash_{\mathcal{S}} \varphi^x_c \to \chi, \tag{7.5}$$

where χ is $\psi_0 \to \cdots \to \psi_{k-1} \to \bot$. Then from (7.3) we have

$$\vdash_{\mathcal{S}} \exists x\ \varphi \to \chi$$

by *modus ponens*; that is,

$$\vdash_{\mathcal{S}} \exists x\ \varphi \to \psi_0 \to \cdots \to \psi_{k-1} \to \bot.$$

Then $k+1$ applications of *modus ponens* show

$$\Gamma \vdash_{\mathcal{S}} \bot,$$

which contradicts the assumption that Γ is consistent. $\square$

So now, given a consistent subset Σ of $\mathrm{Sn}_{\mathcal{L}}$, we can construct a consistent subset Σ^* of $\mathrm{Sn}_{\mathcal{L}\cup C}$ such that

($*$) $\Sigma \subseteq \Sigma^*$;

(†) Σ^* is maximally consistent;

(‡) if $(\exists x\ \varphi) \in \Sigma$, then $\varphi^x_c \in \Sigma$ for some c in C, that is, Σ^* **has witnesses**.

As in the proof of Compactness, we want to use Σ^* to define a model $\mathfrak{A}$ of itself.

For the sake of defining the universe of $\mathfrak{A}$, we assume now that $\mathcal{S}$ has the axioms

$$c = c, \tag{7.6}$$
$$c = c' \to d = d' \to c = d \to c' = d', \tag{7.7}$$

where c, c', d and d' range over C. Let E be the relation

$$\{(c, d) \in C^2 : (c = d) \in \Sigma^*\}.$$

We can now show:

Lemma 7.6.2. *The relation E is an equivalence-relation.*

Proof. We first show

$$\vdash_{\mathcal{S}} c = c, \tag{7.8}$$
$$\vdash_{\mathcal{S}} c = d \to d = c, \tag{7.9}$$
$$\vdash_{\mathcal{S}} c = d \to d = e \to c = e \tag{7.10}$$

for all constants c, d and e in C.

Now, we have (7.8) trivially by (7.6). An instance of (7.7) is

$$c = d \to c = c \to c = c \to d = c;$$

then (7.9) follows by tautological completeness. Another instance of (7.7) is

$$c = c \to d = e \to c = d \to c = e;$$

then (7.10) follows by tautological completeness.

By its maximal consistency then, Σ^* contains $c = c$; and if Σ^* contains $c = d$ and $d = e$, then it contains $d = c$ and $c = e$. $\square$

We define A to be C/E. We now define $R^{\mathfrak{A}}$ (for each n-ary predicate R in $\mathcal{L}$) as the set

$$\{([c_0],\cdots,[c_{n-1}])\in A^n : (Rc_0\cdots c_{n-1})\in\Sigma^*\}.$$

Then we have

$$(Rc_0\cdots c_{n-1})\in\Sigma^* \implies ([c_0],\cdots,[c_{n-1}])\in R^{\mathfrak{A}},$$

but perhaps not the converse. Possibly then both $Rc_0\cdots c_{n-1}$ and $\neg Rc'_0\cdots c'_{n-1}$ are in Σ^*, although $(c_k = c'_k)\in\Sigma^*$ in each case. To prevent this, as as axioms of $\mathcal{S}$ we assume

$$c_0 = c'_0 \to \cdots \to c_{n-1} = c'_{n-1} \to Rc_0\cdots c_{n-1} \to Rc'_0\cdots c'_{n-1}. \tag{7.11}$$

We now have:

Lemma 7.6.3. $([c_0],\cdots,[c_{n-1}])\in R^{\mathfrak{A}} \iff (Rc_0\cdots c_{n-1})\in\Sigma^*$.

{ex7} *Proof.* **Exercise.** □

Finally, suppose f is an n-ary function-symbol (where possibly $n=0$, in which case f is a constant.) We want to be able to define $f^{\mathfrak{A}}$. (If $c\in C$, then $c^{\mathfrak{A}}=[c]$; but there might be constants of $\mathcal{L}$ as well.) To define $f^{\mathfrak{A}}$, we first need some lemmas, which are based on another axiom:

$$\varphi^x_t \to \exists x\ \varphi, \tag{7.12}$$

where $\mathrm{fv}(\varphi)\subseteq\{x\}$ and t is a term with no variables. Let us assume that this is an axiom of $\mathcal{S}$. Then we have:

Lemma 7.6.4 (Substitution)**.** *If* $\mathrm{fv}(\varphi)\subseteq\{x\}$*, and the constant c does not appear in φ, then*

$$\vdash_{\mathcal{S}} \varphi^x_c \to \varphi^x_t$$

for all constant terms t.

Proof. We have

$$\begin{aligned}
&\vdash_{\mathcal{S}} \neg\varphi^x_t \to \exists x\ \neg\varphi, && \text{[by (7.12)]}\\
&\vdash_{\mathcal{S}} \neg\exists x\ \neg\varphi \to \varphi^x_t, && \text{[by tautological completeness]}\\
&\vdash_{\mathcal{S}} (\neg\varphi^x_c \to \bot) \to \exists x\ \neg\varphi \to \bot, && \text{[by (7.3)]}\\
&\vdash_{\mathcal{S}} \varphi^x_c \to \neg\exists x\ \neg\varphi, && \text{[by tautological completeness]}
\end{aligned}$$

and hence $\vdash_{\mathcal{S}} \varphi^x_c \to \varphi^x_t$ by *modus ponens.* □

Lemma 7.6.5. $\vdash_{\mathcal{S}} t = t$ *for all terms t.*

Proof. We have

$$\begin{aligned}
&\vdash_{\mathcal{S}} c = c, && \text{[by (7.6)]}\\
&\vdash_{\mathcal{S}} c = c \to t = t, && \text{[by the Substitution Lemma]}
\end{aligned}$$

and hence $\vdash_{\mathcal{S}} t = t$ by *modus ponens.* □

Lemma 7.6.6. $\vdash_{\mathcal{S}} \exists x\ fc_0 \cdots c_{n-1} = x$.

Proof. We have

$$\begin{aligned} &\vdash_{\mathcal{S}} fc_0 \cdots c_{n-1} = fc_0 \cdots c_{n-1}, && \text{[by the last lemma]} \\ &\vdash_{\mathcal{S}} fc_0 \cdots c_{n-1} = fc_0 \cdots c_{n-1} \to \exists x\ fc_0 \cdots c_{n-1} = x, && \text{[by (7.12)]} \end{aligned}$$

hence $\vdash_{\mathcal{S}} \exists x\ fc_0 \cdots c_{n-1} = x$ by *modus ponens.* □

Finally, we assume as axioms of $\mathcal{S}$ the sentences

$$c_0 = c_0' \to \cdots \to c_{n-1} = c_{n-1}' \to fc_0 \cdots c_{n-1} = fc_0' \cdots c_{n-1}'. \tag{7.13}$$

This enables us to define $f^{\mathfrak{A}}$:

Lemma 7.6.7. *For each n-ary function-symbol f, there is an n-ary operation $f^{\mathfrak{A}}$ on A given by*

$$f^{\mathfrak{A}}([c_0], \dots, [c_{n-1}]) = [d] \iff (fc_0 \cdots c_{n-1} = d) \in \Sigma^*. \tag{7.14}$$

Proof. Since Σ^* is maximally consistent, we now have

$$\exists x\ fc_0 \cdots c_{n-1} = x \in \Sigma^*.$$

Since Σ^* has witnesses, we have

$$fc_0 \cdots c_{n-1} = d \in \Sigma^*$$

for some constant d. This gives us a value for $f^{\mathfrak{A}}([c_0], \cdots, [c_{n-1}])$; we have to show that this value is unique. For this, it is enough to show

$$\begin{aligned} \vdash_{\mathcal{S}} c_0 = c_0' \to \cdots \to c_{n-1} = c_{n-1}' \to \\ d = d' \to fc_0 \cdots c_{n-1} = d \to fc_0' \cdots c_{n-1}' = d' \end{aligned}$$

for all c_k and c_k' and d and d' in C. By (7.13) and tautological completeness, it is enough to show

$$\vdash_{\mathcal{S}} fc_0 \cdots c_{n-1} = fc_0' \cdots c_{n-1}' \to d = d' \to fc_0 \cdots c_{n-1} = d \to fc_0' \cdots c_{n-1}' = d'.$$

In the axiom (7.7), we may assume that c is not one of the variables c', d or d'. Then by the Substitution Lemma, we have

$$\vdash_{\mathcal{S}} fc_0 \cdots c_{n-1} = c' \to d = d' \to fc_0 \cdots c_{n-1} = d \to c' = d'.$$

We may also assume that c' is not one of the variables c_k, d or d'. Applying the Substitution Lemma again gives what we want. □

The structure $\mathfrak{A}$ is now determined and is a model of Σ, by the proof of the Compactness Theorem. In sum, what we have shown is:

{thm:completeness}

Theorem 7.6.8 (Completeness for first-order logic)**.** *That proof-system for* $\mathrm{Sn}_{\mathcal{L}}$ *is complete whose only rule of inference is* modus ponens*, and whose axioms are the following:*

($*$) *the tautologies;*

($\dagger$) $(\varphi^x_c \to \chi) \to \exists x\ \varphi \to \chi$, *where* c *does not appear in* χ*;*

($\ddagger$) $c = c$*;*

($\S$) $c = c' \to d = d' \to c = d \to c' = d'$*;*

($\P$) $c_0 = c'_0 \to \dots c_{n-1} = c'_{n-1} \to Rc_0 \cdots c_{n-1} \to Rc'_0 \cdots c'_{n-1}$*;*

($\|$) $\varphi^x_t \to \exists x\ \varphi$*;*

($**$) $c_0 = c'_0 \to \cdots \to c_{n-1} = c'_{n-1} \to fc_0 \cdots c_{n-1} = fc'_0 \cdots c'_{n-1}$.

Here the notation is as follows:

- x *is a variable;*
- φ *is a formula such that* $\mathrm{fv}(\varphi) \subseteq \{x\}$*;*
- χ *is a sentence;*
- t *is a constant term;*
- c, c', c_k, c'_k, d *and* d' *are constants;*
- $n \in \omega$*;*
- R *is an* n*-ary predicate if* $n > 0$*; and*
- f *is an* n*-ary function-symbol (or a constant, if* $n = 0$*).*

Chapter 8

Numbers of countable models

Our ultimate aim is to show that

$$I(T, \omega) \neq 2 \tag{8.1}$$

whenever T is a countable, *complete* theory. The proof will require several interesting general results.

Note that proving (8.1) requires T to be complete:

Example 8.0.9. Let P be a singulary predicate, and in the signature $\{\mathcal{L}\}$, let T be axiomatized by

$$\forall x \, \forall y \, (Px \land Py \to x = y).$$

Then T has non-isomorphic countably infinite models $(\omega, \varnothing)$ and $(\omega, \{0\})$, and every countably infinite model is isomorphic to one of these. •

8.1 Three models

In the signature $\{<\} \cup \{c_n : n \in \omega\}$, let T_3 be the theory axiomatized by

$$\mathrm{TO}^* \cup \{c_{n+1} < c_n : n \in \omega\}.$$

We shall see that T_3 is complete, and $I(T_3, \omega) = 3$. Let

$$\begin{aligned}
A_0 &= \{a \in \mathbb{Q} : 0 < a\} = \mathbb{Q} \cap (0, \infty), \\
A_1 &= \mathbb{Q} \smallsetminus \{0\}, \\
A_2 &= \mathbb{Q}.
\end{aligned}$$

Then each A_k is the universe of a model $\mathfrak{A}_k$ of T_3, where $<^{\mathfrak{A}_k}$ is the usual ordering $<$, and

$${c_n}^{\mathfrak{A}_k} = \frac{1}{n+1}.$$

Then the set $\{{c_n}^{\mathfrak{A}_k} : n \in \omega\}$, in $\mathfrak{A}_k$,

(∗) has no lower bound, if $k = 0$;

(†) has a lower bound, but no infimum, if $k = 1$;

(‡) has an infimum, if $k = 2$.

Hence the three structures are not isomorphic. However, we shall be able to show:

(∗) if $\mathfrak{B} \vDash T_3$ and is countable, then $\mathfrak{B} \cong \mathfrak{A}_k$ for some k in 3;

(†) T_3 is complete.

The proof of the first claim will be by the **back-and-forth** method. The following gives the prototypical example:

{thm:Cantor}

Theorem 8.1.1. TO^* *is ω-categorical.*

Proof. Suppose $\mathfrak{A}, \mathfrak{B} \vDash \mathrm{TO}^*$ and $|A| = \omega = |B|$. We shall show $\mathfrak{A} \cong \mathfrak{B}$.

We can enumerate the universes:

$$A = \{a_n : n \in \omega\}, \qquad B = \{b_n : n \in \omega\}.$$

We shall recursively define an order-preserving bijection h from A to B. In particular, h will be $\bigcup\{h_n : n \in \omega\}$, where, notationally, we shall have

$$h_n = \{(a_k, b'_k) : k < n\} \cup \{(a'_k, b_k) : k < n\}.$$

We let $h_0 = \varnothing$. Suppose we have h_n so that the tuples

$$(a_0, a'_0, \dots, a_{n-1}, a'_{n-1}), \quad \text{and} \quad (b'_0, b_0, \dots, b'_{n-1}, b_{n-1})$$

have the same **order-type**. This means that, if we write these tuples as $(c_0, \dots, c_{2n-1})$ and $(c'_0, \dots, c'_{2n-1})$ respectively, then

$$c_i < c_j \iff c'_i < c'_j$$

for all i and j in $2n$. Since $\mathfrak{B}$ is a dense total order without endpoints, we can chose b'_n so that

$$(a_0, a'_0, \dots, a_{n-1}, a'_{n-1}, a_n) \quad \text{and} \quad (b'_0, b_0, \dots, b'_{n-1}, b_{n-1}, b'_n)$$

have the same order-type. Likewise, we can choose a'_n so that

$$(a_0, a'_0, \dots, a_n, a'_n), \quad \text{and} \quad (b'_0, b_0, \dots, b'_n, b_n)$$

have the same order-type. Now let $h_{n+1} = h_n \cup \{(a_n, b'_n), (a'_n, b_n)\}$. □

{cor:T3-3}

Corollary 8.1.2. $I(T_3, \omega) = 3$.

Proof. Suppose $\mathfrak{B}$ is a countable model of T_3. The interpretation in $\mathfrak{B}$ of each formula

$$c_{n+1} < x \wedge x < c_n$$

is (when equipped with the ordering induced from $\mathfrak{B}$) a countable model of TO^*. The same is true for the formula $c_0 < x$. Finally, the set

$$\bigcap_{n \in \omega} \{b \in B : b < c_n\}$$

is one of the following:

(∗) empty;

(†) a countable model of TO^*;

(‡) a countable dense total order with a greatest point, but no least point.

Then the previous theorem allows us to construct an isomorphism between $\mathfrak{B}$ and $\mathfrak{A}_0$, $\mathfrak{A}_1$ or $\mathfrak{A}_2$ respectively. □

The following is really a corollary of Theorem 4.0.3:

{thm:T3-QE}

Theorem 8.1.3. *T_3 admits elimination of quantifiers.*

Proof. Any formula $\varphi(\vec{x}\,)$ of $\{<, c_0, c_1, \dots\}$ can be considered as

$$\theta(\vec{x}\,, c_0, \dots, c_{n-1})$$

for some formula θ of $\{<\}$. By quantifier-elimination in TO^*, there is an open formula α of $\{<\}$ such that

$$\mathrm{TO}^* \vDash \forall \vec{x}\ \forall \vec{y}\ (\theta(\vec{x}\,, \vec{y}\,) \land \bigwedge_{i<n} y_{i+1} < y_i \leftrightarrow \alpha(\vec{x}\,, \vec{y}\,)).$$

But $T_3 \vDash c_{i+1} < c_i$, and $T_3 \vDash \mathrm{TO}^*$; so

$$T_3 \vDash \forall \vec{x}\ (\theta(\vec{x}\,, \vec{c}\,) \leftrightarrow \alpha(\vec{x}\,, \vec{c}\,)).$$

Thus T_3 admits quantifier-elimination. □

Corollary 8.1.4. *T_3 is complete.*

Proof. The three countable models $\mathfrak{A}_k$ form a chain:

$$\mathfrak{A}_0 \subseteq \mathfrak{A}_1 \subseteq \mathfrak{A}_2.$$

But here $\operatorname{diag}\mathfrak{B} \vDash \mathrm{Th}(\mathfrak{B}_B)$ for all models $\mathfrak{B}$ of T_3, so by Theorem 5.3.6, the chain is elementary:

$$\mathfrak{A}_0 \preccurlyeq \mathfrak{A}_1 \preccurlyeq \mathfrak{A}_2.$$

In particular, the three structures are elementarily equivalent. Now, if $\mathfrak{B}$ is an arbitrary model of T_3, then it is infinite, so $\mathfrak{B} \equiv \mathfrak{C}$ for some countably infinite structure $\mathfrak{C}$ by Theorem 6.0.11. But $\mathfrak{C} \cong \mathfrak{A}_k$ for some k, by Corollary 8.1.2. Hence $\mathfrak{B} \equiv \mathfrak{A}_0$ by Theorem 5.3.6. Thus

$$T_3 \vDash \mathrm{Th}(\mathfrak{A}_0);$$

so T_3 is complete. □

8.2 Omitting types

Since there is a sound, complete proof-system for first-order logic, we may say that a set of sentences **is consistent** to mean that it has a model.

An n**-type** of a signature $\mathcal{L}$ is a set of n-ary formulas of $\mathcal{L}$.

An n-type Φ of $\mathcal{L}$ is **realized** by $\vec{a}$ in an $\mathcal{L}$-structure $\mathfrak{A}$ if

$$\mathfrak{A} \vDash \varphi(\vec{a})$$

for all φ in Φ. A type not realized in a structure is **omitted** by the structure.

If a consistent theory T of $\mathcal{L}$ is specified, then an n**-type of** T is an n-type Φ that is **consistent with** T: This means that Φ is realized in some model of T. Equivalently, it means that, if $\vec{c}$ is an n-tuple of new constants, then the set

$$T \cup \{\varphi(\vec{c}) : \varphi \in \Phi\}$$

is consistent. By Compactness, for Φ to be consistent with T, it is sufficient that

$$T \cup \{\exists \vec{x} \ \bigwedge \Phi_0\}$$

be consistent for all finite subsets Φ_0 of Φ.

By Compactness also, for *any* collection of types consistent with T, there is a model of T in which all of the types are realized.

An n-type Φ of T is **isolated** in T by an n-ary formula ψ if:

($*$) $T \cup \{\exists \vec{x} \ \psi\}$ is consistent;

($\dagger$) $T \vDash \forall \vec{x} \ (\psi \to \varphi)$ for all φ in Φ.

Hence, if ψ is satisfied by $\vec{a}$ in a model of T, then $\vec{a}$ realizes Φ. Also, if T is complete, then $T \vDash \exists \vec{x} \ \psi$, so Φ is realized in *every* model of T.

We can call a theory **countable** if its signature is countable. (A more general definition is possible: T is *countable* if, in its signature, only countably many formulas are inequivalent in T.) It turns out that, in a *countable* theory, being isolated is the only barrier to being omitted by some model:

Theorem 8.2.1 (Omitting Types)**.** *Suppose T is a countable theory, and Φ is a non-isolated* 1*-type of T. Then Φ is omitted by some countable model of T.*

Proof. We adjust our proof of the Compactness Theorem. As there, we introduce a set C of new constants c_n (where $n \in \omega$). We enumerate $\mathrm{Sn}_{\mathcal{L} \cup C}$ as $\{\sigma_n : n \in \omega\}$. We construct a chain

$$T = \Sigma_0 \subseteq \Sigma_1 \subseteq \cdots$$

as follows. Assume Σ_{3n} is consistent. Then let

$$\Sigma_{3n+1} = \begin{cases} \Sigma_{3n} \cup \{\sigma_n\}, & \text{if this is consistent;} \\ \Sigma_{3n}, & \text{otherwise.} \end{cases}$$

Now let

$$\Sigma_{3n+2} = \Sigma_{3n+1} \cup \{\varphi(c_k)\},$$

where k is minimal such that c_k does not appear in Σ_{3n+1}, if $\sigma_n \in \Sigma_{3n+1}$ and σ_n is $\exists x\ \varphi$; otherwise, $\Sigma_{3n+2} = \Sigma_{3n+1}$. Finally, let

$$\Sigma_{3n+3} = \Sigma_{3n+2} \cup \{\neg\psi(c_n)\},$$

where ψ is an element of Φ such that $\Sigma_{3n+2} \cup \{\neg\psi(c_n)\}$ is consistent. But we have to check that there *is* such a formula ψ in Φ. *If* there is, then we can let

$$\Sigma^* = \bigcup_{n\in\omega} \Sigma_n.$$

Then Σ^* has a countable model $\mathfrak{A}$ (as in the proof of Compactness) such that every element of A is $c^{\mathfrak{A}}$ for some c in C. But by construction, no such element can realize Φ; so $\mathfrak{A}$ omits Φ.

Now, in the definition of Σ_{3n+3}, the formula ψ exists as desired because the set $\Sigma_{3n+2} \smallsetminus T$ can be assumed to be *finite*. In particular, the formulas in this set use only finitely many constants from C. We may assume that these constants form a tuple $(c_n, \vec{d}\,)$. Then we can write $\bigwedge \Sigma_{3n+2} \smallsetminus T$ as a sentence

$$\varphi(c_n, \vec{d}\,),$$

where φ is a certain formula of $\mathcal{L}$. Now, if

$$\Sigma_{3n+2} \vDash \psi(c_n)$$

for some formula ψ, then

$$T \vDash (\varphi(c_n, \vec{d}\,) \to \psi(c_n)),$$

hence

$$T \vDash \forall x\ (\exists \vec{y}\ \ \varphi(x, \vec{y}\,) \to \psi(x)).$$

Since Φ is not isolated in T, it is not isolated by $\exists\vec{y}\ \ \varphi$. Therefore the set $\Sigma_{3n+2} \cup \{\neg\psi(c_n)\}$ must be consistent for some ψ in Φ. □

In the proof, it is essential that $\Sigma_n \smallsetminus T$ is finite; the proof can't be generalized to the case where T is uncountable. But the proof *can* be generalized to yield the following:

Porism 8.2.2. *Suppose T is a countable theory, and Φ_k is an n-type of T for some n (depending on k), for each k in ω. Then T has a countable model omitting each Φ_k.*

An n-type Φ of a theory T is called **complete** if

$$\varphi \notin \Phi \iff \neg\varphi \in \Phi$$

for all n-ary formulas φ of $\mathcal{L}$. Any n-tuple $\vec{a}$ of elements of a model $\mathfrak{A}$ of T determines a complete n-type of T, namely

$$\{\varphi : \mathfrak{A} \vDash \varphi(\vec{a}\,)\};$$

this is the **complete type of $\vec{a}$ in $\mathfrak{A}$** and can be denoted

$$\mathrm{tp}_{\mathfrak{A}}(\vec{a}\,).$$

If Φ is an arbitrary n-type of T, then some $\vec{a}$ from some model $\mathfrak{A}$ of T realizes Φ, and therefore

$$\Phi \subseteq \operatorname{tp}_{\mathfrak{A}}(\vec{a}).$$

In particular, every type of T is included in a complete type of T.

The set of complete n-types of T can be denoted

$$\mathrm{S}_n(T);$$

then we can let $\bigcup_{n\in\omega} \mathrm{S}_n(T)$ be denoted

$$\mathrm{S}(T).$$

So the Omitting-Types Theorem gives us that, if T is countable and $|\mathrm{S}(T)| \leqslant \omega$, then T has a countable model that omits *all* non-isolated types of T.

A structure $\mathfrak{A}$ that realizes only *isolated* types of $\mathrm{Th}(\mathfrak{A})$ is called **atomic**.

Examples 8.2.3. [

]

(1) $(\omega, ', 0)$ is atomic, since each element is named by a term. For example, a 1-type realized by 5 is isolated by the formula $x = 0'''''$.

(2) The theory of Example 5.4.4 has *no* atomic models. •

The following lemma hints at the characterization of countable atomic models that we shall see in the next section.

{lem:types-embeddings}

Lemma 8.2.4. *If $\mathfrak{A}$ embeds elementarily in $\mathfrak{B}$, then $\mathfrak{B}$ realizes all types that $\mathfrak{A}$ realizes.*

Proof. Suppose h is an elementary embedding of $\mathfrak{A}$ in $\mathfrak{B}$, and $\vec{a}$ realizes the type Φ in $\mathfrak{A}$. Then

$$\{\varphi(\vec{a}) : \varphi \in \Phi\} \subseteq \mathrm{Th}(\mathfrak{A}_A),$$

so $h(\vec{a})$ realizes Φ in $\mathfrak{B}$ by Theorem 5.3.6. □

8.3 Prime structures

A structure is **prime** if it embeds elementarily in every model of its theory; if that theory is T, then the structure is a **prime model of** T. (Note then that only complete theories can have prime models, simply because the prime model is elementarily equivalent to all other models.)

Examples 8.3.1. [

]

(1) If T admits quantifier-elimination, then by Corollary 5.3.7, all embeddings of models of T are elementary embeddings. Hence, for example, a countably infinite set is a prime model of the theory of infinite sets. Also, $(\mathbb{Q}, <)$ embeds in every model of TO^*, so it is a prime model.

(2) It is possible to show that, if $|\mathcal{L}| \leqslant \kappa \leqslant |B|$, then $\mathfrak{B}$ is an elementary extension of some structure $\mathfrak{A}$ such that $|A| = \kappa$. Hence, a model of a countable theory T is prime, provided it embeds elementarily in all *countable* models of T. In particular then, if T is ω-categorical, then its countable model is prime. •

{thm:prime-atomic}

Theorem 8.3.2. *Suppose T is a countable complete theory. Then the prime models of T are precisely the countable atomic models of T.*

Proof. Suppose $\mathfrak{A} \vDash T$.

($\Rightarrow$) If $\mathfrak{A}$ is not countable, then $\mathfrak{A}$ cannot embed in countable models of T (which must exist, by Theorem 6.0.11), so $\mathfrak{A}$ cannot be prime.

If $\mathfrak{A}$ is not atomic, then $\mathfrak{A}$ realizes some non-isolated type Φ of T. But by the Omitting-Types Theorem, T has a countable model $\mathfrak{B}$ that omits Φ. Then $\mathfrak{A}$ cannot embed elementarily in $\mathfrak{B}$, by Lemma 8.2.4.

($\Leftarrow$) Suppose $\mathfrak{A}$ is countable and atomic, and $\mathfrak{B} \vDash T$. We construct an elementary embedding of $\mathfrak{A}$ in $\mathfrak{B}$ by the back-and-forth method, except that the construction is in only one direction. Write A as $\{a_n : n \in \omega\}$. Then each $\mathrm{tp}_{\mathfrak{A}}(a_0, \ldots, a_{n-1})$ is isolated in T by some formula φ_n. Then we have

($*$) $T \vDash \exists \vec{x}\ \varphi_n$;

($\dagger$) $T \vDash \forall \vec{x}\ (\varphi_n \to \exists x_n\ \varphi_{n+1})$.

Hence we can recursively find b_k in B so that

$$\mathfrak{B} \vDash \varphi_n(b_0, \ldots, b_{n-1})$$

for all n in ω.

Now, every sentence in $\mathrm{Th}(\mathfrak{A}_A)$ is $\theta(a_0, \ldots, a_{n-1})$ for some formula θ of $\mathcal{L}$. Then

$$T \vDash \forall \vec{x}\ (\varphi_n \to \theta),$$

so $\mathfrak{B} \vDash \theta(\vec{b})$. Therefore the map $a_k \mapsto b_k : A \to B$ is an elementary embedding of $\mathfrak{A}$ in $\mathfrak{B}$. □

{por:prime-isom}

Porism 8.3.3. *All prime models of a countable complete theory are isomorphic.*

Proof. In the proof that $\mathfrak{A}$ embeds elementarily in $\mathfrak{B}$, if we assume also that $\mathfrak{B}$ is countable and atomic, then the full back-and-forth method gives an isomorphism between the structures. □

{lem:I-S}

Lemma 8.3.4. *If $I(T, \omega) \leqslant \omega$, then $|\mathrm{S}(T)| \leqslant \omega$.*

Proof. **Exercise.** □

{thm:prime-existence}

Theorem 8.3.5. *Suppose T is a countable complete theory. Then T has a prime model if $\mathrm{S}(T)$ is countable.*

Proof. **Exercise.** □

8.4 Saturated structures

A **saturated** structure is the opposite of an atomic structure. Atomic structures realize as *few* types as possible. Saturated structures realize as *many* types as possible; moreover, these types are allowed to have parameters from the structure.

To be precise, let $\mathfrak{M}$ be an infinite $\mathcal{L}$-structure, and let $A \subseteq M$. In this context, the set $\mathrm{S}_n(\mathrm{Th}(\mathfrak{M}_A))$ can be denoted

$$\mathrm{S}_n(A).$$

Consider the special case where A is M itself. The set $\mathrm{S}_1(M)$, for example, contains types that include the type

$$\{x \neq a : a \in M\}.$$

These types cannot be realized in $\mathfrak{M}$. So we say that $\mathfrak{M}$ is **saturated**, provided that, whenever $A \subseteq M$ and $|A| \leqslant |M|$, each type in $\mathrm{S}(A)$ is realized in $\mathfrak{M}$. (In particular, if $\mathfrak{M}$ is countable here, then the sets A should be finite.)

{thm:saturated} **Theorem 8.4.1.** *Suppose T is countable and complete, and $|\mathrm{S}(T)| \leqslant \omega$. Then T has a countable saturated model.*

Proof. Suppose $\mathfrak{M}$ is a countable model of T. If A is a finite subset $\{a_k : k < n\}$ of M, then each element of $\mathrm{S}_m(A)$ is

$$\{\varphi(x_0, \dots, x_{m-1}, a_0, \dots, a_{n-1}) : \varphi \in p\}$$

for some p in $\mathrm{S}_{m+n}(T)$. Hence $|\mathrm{S}(A)|$ is countable. Therefore the set

$$\bigcup\{\mathrm{S}(A) : A \text{ is a finite subset of } M\}$$

is countable. So all of the types in this set are realized in a countable elementary extension $\mathfrak{M}'$ of $\mathfrak{M}$.

Thus, if $\mathfrak{M}_0$ is a countable model of T, then we can form an **elementary chain**

$$\mathfrak{M}_0 \preccurlyeq \mathfrak{M}_1 \preccurlyeq \mathfrak{M}_2 \preccurlyeq \cdots.$$

It is straightforward then to define the **union** of this chain: this is a structure $\mathfrak{N}$ whose universe N is $\cup_{n\in\omega} M_n$, and that is an elementary extension of each $\mathfrak{M}_n$. Every finite subset of N is a subset of some $\mathfrak{M}_n$, and so the types of $\mathrm{S}(A)$ are realized in $\mathfrak{M}_{n+1}$, hence in $\mathfrak{N}$. So $\mathfrak{N}$ is saturated. □

If A is a finite subset $\{a_k : k < n\}$ of M, and $\vec{a}$ is $(a_0, \dots, a_{n-1})$, we can denote $\mathfrak{M}_A$ by

$$(\mathfrak{A}, \vec{a}\,).$$

If $\mathfrak{M}$ is countable, then $\mathfrak{M}$ is called **homogeneous** if

$$\mathrm{tp}_{\mathfrak{M}}(\vec{a}\,) = \mathrm{tp}_{\mathfrak{M}}(\vec{b}\,) \implies (\mathfrak{M}, \vec{a}\,) \cong (\mathfrak{M}, \vec{b}\,)$$

for all n-tuples $\vec{a}$ and $\vec{b}$ from M, for all n in ω.

{thm:homog} **Theorem 8.4.2.** *Countable saturated structures are homogeneous.*

Proof. The back-and-forth method. □

8.5 One model

For the sake of stating and proving the following theorem more easily, we can use the following notation. Suppose T is a theory of $\mathcal{L}$. Then equivalence in T is an equivalence-relation on the set of n-ary formulas of $\mathcal{L}$. Let the set of corresponding equivalence-classes be denoted

$$\mathrm{B}_n(T).$$

Theorem 8.5.1. *Suppose T is a countable complete theory. The following statements are equivalent:*

(0) $I(T, \omega) = 1$.

(1) *All types of T are isolated.*

(2) *Each set $\mathrm{B}_n(T)$ is finite.*

(3) *Each set $\mathrm{S}_n(T)$ is finite.*

Proof. (0)⇒(1): If $\mathrm{S}(T)$ contains a non-isolated type, then it is realized in some, but not all, countable models of T, so $I(T, \omega) > 1$.

(1)⇒(0): If all types of T are isolated, then all models of T are atomic, so all *countable* models of T are prime and therefore isomorphic.

(2)⇒(3): Immediate.

(3)⇒(1)&(2): Suppose $\mathrm{S}_n(T) = \{p_0, \dots, p_{m-1}\}$. For each i and j in m, if $i \neq j$, then there is a formula φ_{ij} in $p_i \smallsetminus p_j$. Let ψ_i be the formula

$$\bigwedge_{j \in m \smallsetminus \{i\}} \varphi_{ij}.$$

Then ψ_i is in p_j if and only if $j = i$. If $\mathfrak{A} \vDash T$, and $\vec{a}$ is an n-tuple from A, then $\mathfrak{A}$ realizes some unique p_i, and then $\mathfrak{A} \vDash \psi_i(\vec{a})$. Conversely, if $\mathfrak{A} \vDash \psi_i(\vec{a})$, then $\vec{a}$ must realize p_i. Therefore ψ_i isolates p_i.

If χ is an arbitrary n-ary formula, let $I = \{i \in m : \chi \in p_i\}$. Then

$$T \vDash \forall \vec{x}\ (\chi \leftrightarrow \bigvee_{i \in I} \psi_i).$$

There are only finitely many possibilities for I, so $\mathrm{B}_n(T)$ is finite.

(1)⇒(3): Suppose infinitely many complete n-types are isolated in T. Since T is countable, there must be countably many such types. Say they compose the set $\{p_k : k \in \omega\}$, and each p_k is isolated by φ_k. Then the type

$$\{\neg\varphi_k : k \in \omega\}$$

is consistent with T. It is not included in any of the p_k, so it must be included in a non-isolated type. □

8.6 Not two models

Theorem 8.6.1. *Suppose T is a countable complete theory. Then $I(T, \omega) \neq 2$.*

Proof. Suppose if possible that T_2 has just two non-isomorphic countable models. One of them, $\mathfrak{A}$, is prime, by Lemma 8.3.4 and Theorem 8.3.5. The other one, $\mathfrak{B}$, is saturated, by Theorem 8.4.1. Since $\mathfrak{A}$ embeds elementarily in $\mathfrak{B}$, we may assume $\mathfrak{A} \preccurlyeq \mathfrak{B}$.

Since $\mathfrak{A} \ncong \mathfrak{B}$, there is a non-isolated type Φ realized by some $\vec{b}$ in $\mathfrak{B}$, by Theorem 8.3.2 and Porism 8.3.3. Let $T^* = \mathrm{Th}(\mathfrak{B}, \vec{b})$. Suppose $(\mathfrak{C}, \vec{c})$ is a countable model of T^*. Then $\mathfrak{C} \vDash T_2$, so $\mathfrak{C}$ is isomorphic to $\mathfrak{A}$ or $\mathfrak{B}$. In any case, $\mathfrak{A}$ embeds elementarily in $\mathfrak{C}$. But Φ is realized by $\vec{c}$ in $\mathfrak{C}$. Hence $\mathfrak{C} \cong \mathfrak{B}$ by Lemma 8.2.4. Let the isomorphism take $\vec{c}$ to $\vec{a}$. Then it is enough to show $(\mathfrak{B}, \vec{a}) \cong (\mathfrak{B}, \vec{b})$. But this follows from Theorem 8.4.2. $\square$

Bibliography

[1] Aristotle. *Categories, On Interpretation, and Prior Analytics*, volume 325 of *Loeb Classical Library*. Harvard University Press and William Heinemann Ltd, Cambridge, Massachusetts and London, 1973. Translated by H. P. Cooke and H. Tredennick.

[2] C. C. Chang and H. J. Keisler. *Model theory*. North-Holland Publishing Co., Amsterdam, 1973. Studies in Logic and the Foundations of Mathematics, Vol. 73.

[3] Alonzo Church. *Introduction to mathematical logic. Vol. I.* Princeton University Press, Princeton, N. J., 1956.

[4] René Descartes. *The Geometry of René Descartes*. Dover Publications, Inc., New York, 1954. Translated from the French and Latin by David Eugene Smith and Marcia L. Latham, with a facsimile of the first edition.

[5] Leon Henkin. The completeness of the first-order functional calculus. *J. Symbolic Logic*, 14:159–166, 1949.

[6] Leon Henkin. The discovery of my completeness proofs. *Bull. Symbolic Logic*, 2(2):127–158, 1996.

[7] Wilfrid Hodges. *Model theory*, volume 42 of *Encyclopedia of Mathematics and its Applications*. Cambridge University Press, Cambridge, 1993.

[8] Henry George Liddell and Robert Scott. *A Greek-English Lexicon*. Clarendon Press, Oxford, 1940. Revised and augmented throughout by Sir Henry Stuart Jones.

[9] David Marker. *Model theory*, volume 217 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 2002. An introduction.

[10] Plato. *Republic*. Loeb Classical Library. Harvard University Press, Cambridge, Massachusetts, USA, 1980. With an English Translation by Paul Shorey. In two volumes.

[11] Emil L. Post. Introduction to a general theory of elementary propositions. *Amer. J. Math.*, 43(3):163–185, July 1921.

[12] Philipp Rothmaler. *Introduction to model theory*, volume 15 of *Algebra, Logic and Applications*. Gordon and Breach Science Publishers, Amsterdam, 2000. Prepared by Frank Reitmaier, Translated and revised from the 1995 German original by the author.

[13] Joseph R. Shoenfield. *Mathematical logic.* Addison-Wesley Publishing Co., Reading, Mass.-London-Don Mills, Ont., 1967.

[14] Robert R. Stoll. *Set theory and logic.* Dover Publications Inc., New York, 1979. Corrected reprint of the 1963 edition.

[15] Jean van Heijenoort, editor. *Frege and Gödel. Two fundamental texts in mathematical logic.* Harvard University Press, Cambridge, Mass., 1970.