A Problem Course in Mathematical Logic

Volume I
Propositional and First-Order Logic

Stefan Bilaniuk

Author address:

Department of Mathematics
Trent University
Peterborough, Ontario
Canada K9J 7B8
E-mail address: sbilaniuk@trentu.ca

1991 *Mathematics Subject Classification.* 03.

A Problem Course In Mathematical Logic
Volume I: Propositional and First-Order Logic
VERSION 1.3

ABSTRACT. This is Volume I of a text for a problem-oriented undergraduate course in mathematical logic. It covers the basics of propositional and first-order logic through the Soundness, Completeness, and Compactness Theorems, and could be used for a one-semester course on these topics. Volume II, *Computability and Incompleteness*, covers the basics of these topics using Turing machines and recursive functions.

Information on availabality and the conditions under which this book may be used and reproduced are given in the preface.

This book was typeset using LaTeX, using the AMS-LaTeX and AMSFonts packages of the American Mathematical Society.

Contents

Preface

This book is intended to be the basis for a problem-oriented full-year course in mathematical logic for students with a modicum of mathematical sophistication. Volume I covers the basics of propositional and first-order logic through the Soundness, Completeness, and Compactness Theorems, plus some material on applications of the Compactness Theorem. It could easily be used for a one-semester course on these topics. Volume II covers the basics of computability, using Turing machines and recursive functions, and incompleteness; it could also be used as for a one-semester course on these topics.

In keeping with the modified Moore-method, this book supplies definitions, problems, and statements of results, along with some explanations, examples, and hints. The intent is for the students, individually or in groups, to learn the material by solving the problems and proving the results for themselves. Besides constructive criticism, it will probably be necessary for the instructor to supply further hints or direct the students to other sources from time to time. Just how this text is used will, of course, depend on the instructor and students in question. However, it is probably *not* appropriate for a conventional lecture-based course nor for a really large class.

The material presented in this volume is somewhat stripped-down. Various concepts and topics that are often covered in introductory mathematical logic courses are given very short shrift or omitted entirely, among them normal forms, definability, and model theory.[1] Instructors might consider having students do projects on additional material if they wish to to cover it.

Acknowledgements. Various people and institutions deserve the credit for this text: All the people who developed the subject. My teachers and colleagues, especially Gregory H. Moore, whose mathematical logic course convinced me that I wanted to do the stuff.

[1]Future versions of both volumes may include more – or less! – material. Feel free to send suggestions, corrections, criticisms, and the like — I'll feel free to ignore them or use them.

The students at Trent University who suffered, suffer, and will suffer through assorted versions of this text. Trent University and the taxpayers of Ontario, who paid my salary. Ohio University, where I spent my sabbatical in 1995–96. All the people and organizations who developed the software and hardware with which this book was prepared. Anyone else I've missed.

Any blame properly accrues to the author.

Conditions. This book may be freely transmitted, stored, copied, and used until 31 December, 1998, subject to the following restrictions:[2]

1. It may not be modified in any way without the express written permission of the author.
2. It may not be sold at a profit, but only to recover the cost of reproduction.
3. After 31 December, 1998, it may no longer be reproduced, stored, or transmitted in any form without the express written permission of the author, except that printed copies existing as of 31 December, 1998, may be retained and used after this date.

The reason for the time-limit is that I hope to improve this book and make a new version available.[3]

Availability. The URL of the home page for *A Problem Course In Mathematical Logic*, with links to LaTeX and PostScript source files of the latest release of both volumes, is:

- `http://www.trentu.ca/academic/math/sb/misc/pcml.html`

A text-only information file and LaTeX and PostScript source files of the latest release of both volumes can be accessed by anonymous ftp in the directory:

- `ftp://ftp.trentu.ca/pub/sbilaniuk/pcml/`

Please note that in addition to LaTeX you will need the AMS-LaTeX and AMSFonts packages to typeset and print either volume.

If you have any problems, feel free to contact the author at the addresses given on the title page, preferably by e-mail, for assistance or even to ask for a paper copy.

Author's Opinion. It's not great, but the price is right!

[2]If you violate these restrictions, I will be flattered, but you will still be in the wrong.

[3]Who knows, it might even find a publisher ...

Introduction

Mathematical Logic. What sets mathematics aside from other disciplines is its reliance on proof as the principal technique for determining truth, where science, for example, relies on (carefully analyzed) experience. So what is a proof? Practically speaking, a proof is any reasoned argument accepted as such by other mathematicians.[4] A more precise definition is needed, however, if one wishes to discover what mathematical reasoning can accomplish in principle. This is one of the reasons for studying mathematical logic, which is also pursued for its own sake and finding new tools to use in the rest of mathematics and in related fields.

In any case, mathematical logic is concerned with formalizing and analyzing the kinds of reasoning used in the rest of mathematics. The point of mathematical logic is not to try to do mathematics *per se* completely formally — the practical problems involved in doing so are usually such as to make this an exercise in frustration — but to study formal logical systems as mathematical objects in their own right in order to (informally!) prove things about them. For this reason, the formal systems developed in this book are optimized to be easy to prove things about, rather than to be easy to use. Natural deductive systems such as those developed by philosophers to formalize logical reasoning are equally capable in principle and much easier to actually use, but harder to prove things about.

Part of the problem with formalizing mathematical reasoning is the necessity of precisely specifying the language(s) in which it is to be done. The natural languages spoken by humans won't do: they are so complex and continually changing as to be impossible to pin down completely. By contrast, the languages which underly formal logical systems are, like programming languages, much simpler and less flexible than natural languages but rigidly defined. A formal logical system also requires the careful specification of the allowable rules of reasoning, plus some notion of how to interpret statements in the underlying language

[4] If you are not a mathematician, gentle reader, you are hereby temporarily promoted.

and determine their truth. The real fun lies in the relationship between interpretation of statements, truth, and reasoning.

This volume develops the basics of two kinds of formal logical systems, propositional logic and first-order logic. Propositional logic attempts to make precise the relationships that certain connectives like *not*, *and*, *or*, and *if ... then* are used to express in English. While it has uses, propositional logic is not powerful enough to formalize most mathematical discourse. For one thing, it cannot handle the concepts expressed by *all* and *there is*. First-order logic adds *all* and *there is* to those which propositional logic could handle, and suffices, in principle, to formalize most mathematical reasoning. To be sure, it will not handle concepts which arise outside of mathematics, such as *possible* and *relevant*, among many others. (Trying to incorporate such concepts into systems extending first-order logic is a substantial industry in philosophy, but of marginal interest in mathematics.) Propositional logic, which is much simpler, will be dealt with first in order to gain some experience in dealing with formal systems before tackling first-order logic. Besides, some of the results about propositional logic carry over to first-order logic with little change.

Approach. This book supplies definitions and statements of results, plus some explanations and a number of problems and examples, but no proofs of the results. The hope is that you, gentle reader, will learn the material presented here by solving the problems and proving the results for yourself. Brief hints are supplied for almost all of the problems and results, but if these do not suffice, you should consult your peers, your instructor, or other texts.

Prerequisites. In principle, not much is needed by way of prior mathematical knowledge to define and prove the basic facts about propositional and first-order logic. Some knowledge of the natural numbers and a little set theory suffices; the former will be assumed and the latter is very briefly summarized in Appendix A. What really *is* needed to get anywhere with the material developed here is competence in handling abstraction and proofs, especially proofs by induction. The experience provided by a rigorous introductory course in algebra, analysis, or discrete mathematics ought to be sufficient. Some problems and examples draw on concepts from other parts of mathematics; students who are not already familiar with these should consult texts in the appropriate subjects for the necessary definitions.

Other Sources and Further Reading. [**4**], [**8**], and [**9**] are texts which go over similar ground (and much more), while [**1**] and [**3**] are good references for more advanced material. Entertaining accounts of some related topics may be found in [**7**] and [**10**]. Those interested in natural deductive systems might try [**2**], which has a very clean presentation.

Credit. Almost no attempt has been made to give due credit to those who developed and refined the ideas, results, and proofs mentioned in this work. In mitigation, it would often be difficult to assign credit fairly because many people were involved, frequently having interacted in complicated ways. (Which really means that I'm too lazy to do it. I apologize to those who have been hurt by this.) Those interested in who did what should start by consulting other texts or reference works covering similar material.

Propositional Logic

CHAPTER 1

Language

Propositional logic (sometimes called sentential or predicate logic) attempts to formalize the reasoning that can be done with connectives like *not*, *and*, *or*, and *if ... then*. We will define the formal language of propositional logic, $\mathcal{L}_P$, by specifying its symbols and rules for assembling these symbols into the formulas of the language.

DEFINITION 1.1. The *symbols* of $\mathcal{L}_P$ are:

1. Parentheses: (and).
2. Connectives: $\neg$ and $\to$.
3. Atomic formulas: A_0, A_1, A_2, ..., A_n, ...

We still need to specify the ways in which the symbols of $\mathcal{L}_P$ can be put together.

DEFINITION 1.2. The *formulas* of $\mathcal{L}_P$ are those finite sequences or strings of the symbols given in Definition 1.1 which satisfy the following rules:

1. Every atomic formula is a formula.
2. If α is a formula, then $(\neg\alpha)$ is a formula.
3. If α and β are formulas, then $(\alpha \to \beta)$ is a formula.
4. No other sequence of symbols is a formula.

We will often use lower-case Greek characters to represent formulas, as we did in the definition above, and upper-case Greek characters to represent sets of formulas.[1] All formulas in Chapters 1–4 will be assumed to be formulas of $\mathcal{L}_P$ unless stated otherwise.

What do these definitions mean? The parentheses are just punctuation: their only purpose is to group other symbols together. (One could get by without them; see Problem 1.6.) $\neg$ and $\to$ are supposed to represent the connectives *not* and *if ... then* respectively. The atomic formulas, A_0, A_1, ..., are meant to represent statements that cannot be broken down any further using our connectives, such as "The moon is made of cheese." Thus, one might translate the the English sentence "If the moon is red, it is not made of cheese" into the formula

[1]The Greek alphabet is given in Appendix B.

$(A_0 \to (\neg A_1))$ of $\mathcal{L}_P$ by using A_0 to represent "The moon is red" and A_1 to represent "The moon is made of cheese." Note that the truth of the formula depends on the interpretation of the atomic sentences which appear in it. Using the interpretations just given of A_0 and A_1, the formula $(A_0 \to (\neg A_1))$ is true, but if we instead use A_0 and A_1 to interpret "My telephone is ringing" and "Someone is calling me", respectively, $(A_0 \to (\neg A_1))$ is false.

Definition 1.2 says that that every atomic formula is a formula and every other formula is built from shorter formulas using the connectives and parentheses in particular ways. For example, A_{1123}, $(A_2 \to (\neg A_0))$, and $(((\neg A_1) \to (A_1 \to A_7)) \to A_7)$ are all formulas, but X_3, (A_5), $()\neg A_{41}$, $A_5 \to A_7$, and $(A_2 \to (\neg A_0)$ are not.

Problem 1.1. *Why are the following* not *formulas of $\mathcal{L}_P$? There might be more than one reason ...*

1. A_{-56}
2. $(Y \to A)$
3. $(A_7 \leftarrow A_4)$
4. $A_7 \to (\neg A_5))$
5. $(A_8 A_9 \to A_{1043998}$
6. $(((\neg A_1) \to (A_\ell \to A_7) \to A_7)$

Problem 1.2. *Show that every formula of $\mathcal{L}_P$ has the same number of left parentheses as it has of right parentheses.*

Problem 1.3. *Suppose α is any formula of $\mathcal{L}_P$. Let $\ell(\alpha)$ be the length of α as a sequence of symbols and let $p(\alpha)$ be the number of parentheses (counting both left and right parentheses) in α. What are the minimum and maximum values of $p(\alpha)/\ell(\alpha)$?*

Problem 1.4. *Suppose α is any formula of $\mathcal{L}_P$. Let $s(\alpha)$ be the number of atomic formulas in α (counting repetitions) and let $c(\alpha)$ be the number of occurrences of $\to$ in α. Show that $s(\alpha) = c(\alpha) + 1$.*

Problem 1.5. *What are the possible lengths of formulas of $\mathcal{L}_P$? Prove it.*

Problem 1.6. *Find a way for doing without parentheses or other punctuation symbols in defining a formal language for propositional logic.*

Proposition 1.7. *Show that the set of formulas of $\mathcal{L}_P$ is countable.*

Informal Conventions. At first glance, $\mathcal{L}_P$ may not seem capable of breaking down English sentences with connectives other than *not* and *if ... then*. However, the sense of many other connectives can be captured by these two by using suitable circumlocutions. We will use the symbols $\wedge$, $\vee$, and $\leftrightarrow$ to represent *and*, *or*,[2] and *if and only if* respectively. Since they are not among the symbols of $\mathcal{L}_P$, we will use them as abbreviations for certain constructions involving only $\neg$ and $\rightarrow$. Namely,

- $(\alpha \wedge \beta)$ is short for $(\neg(\alpha \rightarrow (\neg\beta)))$,
- $(\alpha \vee \beta)$ is short for $((\neg\alpha) \rightarrow \beta)$, and
- $(\alpha \leftrightarrow \beta)$ is short for $((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$.

Interpreting A_0 and A_1 as before, for example, one could translate the English sentence "The moon is red and made of cheese" as $(A_0 \wedge A_1)$. (Of course this is really $(\neg(A_0 \rightarrow (\neg A_1)))$, *i.e.* "It is not the case that if the moon is green, it is not made of cheese.") $\wedge$, $\vee$, and $\leftrightarrow$ were not included among the official symbols of $\mathcal{L}_P$ partly because we can get by without them and partly because leaving them out makes it easier to prove things about $\mathcal{L}_P$.

PROBLEM 1.8. *Take a couple of English sentences with several connectives and translate them into formulas of $\mathcal{L}_P$. You may use $\wedge$, $\vee$, and $\leftrightarrow$ if appropriate.*

PROBLEM 1.9. *Write out $((\alpha \vee \beta) \wedge (\beta \rightarrow \alpha))$ using only $\neg$ and $\rightarrow$.*

For the sake of readability, we will occasionally use some informal conventions that let us get away with writing fewer parentheses:

- We will usually drop the outermost parentheses in a formula, writing $\alpha \rightarrow \beta$ instead of $(\alpha \rightarrow \beta)$ and $\neg\alpha$ instead of $(\neg\alpha)$.
- We will let $\neg$ take precedence over $\rightarrow$ when parentheses are missing, so $\neg\alpha \rightarrow \beta$ is short for $((\neg\alpha) \rightarrow \beta)$, and fit the informal connectives into this scheme by letting the order of precedence be $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$.
- Finally, we will group repetitions of $\rightarrow$, $\vee$, $\wedge$, or $\leftrightarrow$ to the right when parentheses are missing, so $\alpha \rightarrow \beta \rightarrow \gamma$ is short for $(\alpha \rightarrow (\beta \rightarrow \gamma))$.

Just like formulas using $\vee$, $\wedge$, or $\neg$, formulas in which parentheses have been omitted as above are not official formulas of $\mathcal{L}_P$, they are convenient abbreviations for official formulas of $\mathcal{L}_P$. Note that a precedent for the precedence convention can be found in the way that $\cdot$ commonly takes precedence over $+$ in writing arithmetic formulas.

PROBLEM 1.10. *Write out $\neg(\alpha \leftrightarrow \neg\delta) \wedge \beta \rightarrow \neg\alpha \rightarrow \gamma$ first with the missing parentheses included and then as an official formula of $\mathcal{L}_P$.*

[2] We will use *or* inclusively, so that "A or B" is still true if both of A and B are true.

The following notion will be needed later on.

DEFINITION 1.3. Suppose φ is a formula of $\mathcal{L}_P$. The set of *subformulas* of φ, $\mathcal{S}(\varphi)$, is defined as follows.

1. If φ is an atomic formula, then $\mathcal{S}(\varphi) = \{\varphi\}$.
2. If φ is $(\neg\alpha)$, then $\mathcal{S}(\varphi) = \mathcal{S}(\alpha) \cup \{(\neg\alpha)\}$.
3. If φ is $(\alpha \to \beta)$, then $\mathcal{S}(\varphi) = \mathcal{S}(\alpha) \cup \mathcal{S}(\beta) \cup \{(\alpha \to \beta)\}$.

For example, if φ is $(((\neg A_1) \to A_7) \to (A_8 \to A_1))$, then $\mathcal{S}(\varphi)$ includes A_1, A_7, A_8, $(\neg A_1)$, $(A_8 \to A_1)$, $((\neg A_1) \to A_7)$, and $(((\neg A_1) \to A_7) \to (A_8 \to A_1))$ itself.

Note that if you write out a formula with all the official parentheses, then the subformulas are just the parts of the formula enclosed by matching parentheses, plus the atomic formulas. In particular, every formula is a subformula of itself. Note that some subformulas of formulas involving our informal abbreviations $\vee$, $\wedge$, or $\leftrightarrow$ will be most conveniently written using these abbreviations. For example, if ψ is $A_4 \to A_1 \vee A_4$, then

$$\mathcal{S}(\psi) = \{\, A_1,\ A_4,\ (\neg A_1),\ (A_1 \vee A_4),\ (A_4 \to (A_1 \vee A_4)) \,\}\,.$$

(As an exercise, where did $(\neg A_1)$ come from?)

PROBLEM 1.11. *Find all the subformulas of each of the following formulas.*

1. $(\neg((\neg A_{56}) \to A_{56}))$
2. $A_9 \to A_8 \to \neg(A_{78} \to \neg\neg A_0)$
3. $\neg A_0 \wedge \neg A_1 \leftrightarrow \neg(A_0 \vee A_1)$

Unique Readability. The slightly paranoid — er, truly rigorous — might ask whether Definitions 1.1 and 1.2 actually ensure that the formulas of $\mathcal{L}_P$ are unambiguous, *i.e.* can be read in only one way according to the rules given in Definition 1.2. To actually prove this one must add to Definition 1.1 the requirement that all the symbols of $\mathcal{L}_P$ are distinct and that no symbol is a subsequence of any other symbol. With this addition, one can prove the following:

THEOREM 1.12 (Unique Readability Theorem). *A formula of $\mathcal{L}_P$ must satisfy exactly one of conditions 1–3 in Definition 1.2.*

CHAPTER 2

Truth Assignments

Whether a given formula φ of $\mathcal{L}_P$ is true or false usually depends on how we interpret the atomic formulas which appear in φ. For example, if φ is the atomic formula A_2 and we interpret it as "2+2 = 4", it is true, but if we interpret it as "The moon is made of cheese", it is false. Since we don't want to commit ourselves to a single interpretation — after all, we're really interested in general logical relationships — we will define how any assignment of *truth values* T ("true") and F ("false") to atomic formulas of $\mathcal{L}_P$ can be extended to all other formulas. We will also get a reasonable definition of what it means for a formula of $\mathcal{L}_P$ to follow logically from other formulas.

DEFINITION 2.1. A *truth assignment* is a function v whose domain is the set of all formulas of $\mathcal{L}_P$ and whose range is the set $\{T, F\}$ of truth values, such that:

1. $v(A_n)$ is defined for every atomic formula A_n.
2. For any formula α,

$$v(\,(\neg\alpha)\,) = \begin{cases} T & \text{if } v(\alpha) = F \\ F & \text{if } v(\alpha) = T. \end{cases}$$

3. For any formulas α and β,

$$v(\,(\alpha \to \beta)\,) = \begin{cases} F & \text{if } v(\alpha) = T \text{ and } v(\beta) = F \\ T & \text{otherwise.} \end{cases}$$

Given interpretations of all the atomic formulas of $\mathcal{L}_P$, the corresponding truth assignment would give each atomic formula representing a true statement the value T and every atomic formula representing a false statement the value F. Note that we have not defined how to handle any truth values besides T and F in $\mathcal{L}_P$. Logics with other truth values have uses, but are not relevant in most of mathematics.

For an example of how non-atomic formulas are given truth values on the basis of the truth values given to their components, suppose v is a truth assignment such that $v(A_0) = T$ and $v(A_1) = F$. Then $v(\,((\neg A_1) \to (A_0 \to A_1))\,)$ is determined from $v(\,(\neg A_1)\,)$ and $v(\,(A_0 \to$

$A_1)$) according to clause 3 of Definition 2.1. In turn, $v(\,(\neg A_1)\,)$ is determined from of $v(A_1)$ according to clause 2 and $v(\,(A_0 \to A_1)\,)$ is determined from $v(A_1)$ and $v(A_0)$ according to clause 3. Finally, by clause 1, our truth assignment must be defined for all atomic formulas to begin with; in this case, $v(A_0) = T$ and $v(A_1) = F$. Thus $v(\,(\neg A_1)\,) = T$ and $v(\,(A_0 \to A_1)\,) = F$, so $v(\,((\neg A_1) \to (A_0 \to A_1))\,) = F$.

A convenient way to write out the determination of the truth value of a formula on a given truth assignment is to use a *truth table*: list all the subformulas of the given formula across the top in order of length and then fill in their truth values on the bottom from left to right. Except for the atomic formulas at the extreme left, the truth value of each subformula will depend on the truth values of the subformulas to its left. For the example above, one gets something like:

A_0	A_1	$(\neg A_1)$	$(A_0 \to A_1)$	$(\neg A_1) \to (A_0 \to A_1))$
T	F	T	F	F

Problem 2.1. *Suppose v is a truth assignment such that $v(A_0) = v(A_2) = T$ and $v(A_1) = v(A_3) = F$. Find $v(\alpha)$ if α is:*

1. $\neg A_2 \to \neg A_3$
2. $\neg A_2 \to A_3$
3. $\neg(\neg A_0 \to A_1)$
4. $A_0 \vee A_1$
5. $A_0 \wedge A_1$

The use of finite truth tables to determine what truth value a particular truth assignment gives a particular formula is justified by the following proposition, which asserts that only the truth values of the atomic sentences in the formula matter.

Proposition 2.2. *Suppose δ is any formula and u and v are truth assignments such that $u(A_n) = v(A_n)$ for all atomic formulas A_n which occur in δ. Then $u(\delta) = v(\delta)$.*

Corollary 2.3. *Suppose u and v are truth assignments such that $u(A_n) = v(A_n)$ for every atomic formula A_n. Then $u = v$,* i.e. *$u(\varphi) = v(\varphi)$ for every formula φ.*

Proposition 2.4. *If α and β are formulas and v is a truth assignment, then:*

1. *$v(\neg\alpha) = T$ if and only if $v(\alpha) = F$.*
2. *$v(\alpha \to \beta) = T$ if and only if $v(\beta) = T$ whenever $v(\alpha) = T$;*
3. *$v(\alpha \wedge \beta) = T$ if and only if $v(\alpha) = T$ and $v(\beta) = T$;*
4. *$v(\alpha \vee \beta) = T$ if and only if $v(\alpha) = T$ or $v(\beta) = T$; and*
5. *$v(\alpha \leftrightarrow \beta) = T$ if and only if $v(\alpha) = v(\beta)$.*

Truth tables are often used even when the formula in question is not broken down all the way into atomic formulas. For example, if α and β are any formulas and we know that α is true but β is false, then the truth of $(\alpha \to (\neg\beta))$ can be determined by means of the following table:

α	β	$(\neg\beta)$	$(\alpha \to (\neg\beta))$
T	F	T	T

Definition 2.2. If v is a truth assignment and φ is a formula, we will often say that v *satisfies* φ if $v(\varphi) = T$. Similarly, if Σ is a set of formulas, we will often say that v satisfies Σ if $v(\sigma) = T$ for every $\sigma \in \Sigma$. We will say that φ (respectively, Σ) is *satisfiable* if there is some truth assignment which satisfies it.

Definition 2.3. A formula φ is a *tautology* if it is satisfied by every truth assignment. A formula ψ is a *contradiction* if there is no truth assignment which satisfies it.

For example, $(A_4 \to A_4)$ is a tautology while $(\neg(A_4 \to A_4))$ is a contradiction, and A_4 is a formula which is neither. One can check whether a given formula is a tautology, contradiction, or neither, by grinding out a complete truth table for it, with a separate line for each possible assignment of truth values to the atomic subformulas of the formula. For $A_3 \to (A_4 \to A_3)$ this gives

A_3	A_4	$A_4 \to A_3$	$A_3 \to (A_4 \to A_3)$
T	T	T	T
T	F	T	T
F	T	F	T
F	F	T	T

so $A_3 \to (A_4 \to A_3)$ is a tautology. Note that, by Proposition 2.2, we need only consider the possible truth values of the atomic sentences which actually occur in a given formula.

One can often use truth tables to determine whether a given formula is a tautology or a contradiction even when it is not broken down all the way into atomic formulas. For example, if α is any formula, then the table

α	$(\alpha \to \alpha)$	$(\neg(\alpha \to \alpha))$
T	T	F
F	T	F

demonstrates that $(\neg(\alpha \to \alpha))$ is a contradiction, no matter which formula of $\mathcal{L}_P$ α actually is.

PROPOSITION 2.5. *If α is any formula, then $((\neg\alpha) \vee \alpha)$ is a tautology and $((\neg\alpha) \wedge \alpha)$ is a contradiction.*

PROPOSITION 2.6. *A formula β is a tautology if and only if $\neg\beta$ is a contradiction.*

After all this warmup, we are finally in a position to define what it means for one formula to follow logically from other formulas.

DEFINITION 2.4. A set of formulas Σ *implies* a formula φ, written as $\Sigma \models \varphi$, if every truth assignment v which satisfies Σ also satisfies φ. We will often write $\Sigma \not\models \varphi$ if it is not the case that $\Sigma \models \varphi$. In the case where Σ is empty, we will usually write $\models \varphi$ instead of $\emptyset \models \varphi$.

Similarly, if Δ and Γ are sets of formulas, then Δ *implies* Γ, written as $\Delta \models \Gamma$, if every truth assignment v which satisfies Δ also satisfies Γ.

For example, $\{ A_3, (A_3 \to \neg A_7) \} \models \neg A_7$, but $\{ A_8, (A_5 \to A_8) \} \not\models A_5$. (There is a truth assignment which makes A_8 and $A_5 \to A_8$ true, but A_5 false.) Note that a formula φ is a tautology if and only if $\models \varphi$, and a contradiction if and only if $\models (\neg\varphi)$.

PROPOSITION 2.7. *If Γ and Σ are sets of formulas such that $\Gamma \subseteq \Sigma$, then $\Gamma \models \Sigma$.*

PROBLEM 2.8. *How can one check whether or not $\Sigma \models \varphi$ for a formula φ and a finite set of formulas Σ?*

PROPOSITION 2.9. *Suppose Σ is a set of formulas and ψ and ρ are formulas. Then $\Sigma \cup \{\psi\} \models \rho$ if and only if $\Sigma \models \psi \to \rho$.*

PROPOSITION 2.10. *A set of formulas Σ is satisfiable if and only if there is no contradiction χ such that $\Sigma \models \chi$.*

CHAPTER 3

Deductions

In this chapter we develop a way of defining logical implication that does not rely on any notion of truth, but only on manipulating sequences of formulas, namely formal proofs or deductions. (Of course, any way of defining logical implication had better be compatible with that given in Chapter 2.) To define these, we first specify a suitable set of formulas which we can use freely as premisses in deductions.

DEFINITION 3.1. The three *axiom schema* of $\mathcal{L}_P$ are:

A1: $(\alpha \to (\beta \to \alpha))$
A2: $((\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma)))$
A3: $(((\neg\beta) \to (\neg\alpha)) \to (((\neg\beta) \to \alpha) \to \beta))$.

Replacing α, β, and γ by particular formulas of $\mathcal{L}_P$ in any one of the schemas A1, A2, or A3 gives an *axiom* of $\mathcal{L}_P$.

For example, $(A_1 \to (A_4 \to A_1))$ is an axiom, being an instance of axiom schema A1, but $(A_9 \to (\neg A_0))$ is not an axiom as it is not the instance of any of the schema. As had better be the case, every axiom is always true:

PROPOSITION 3.1. *Every axiom of $\mathcal{L}_P$ is a tautology.*

Second, we specify our one (and only!) rule of inference.[1]

DEFINITION 3.2 (Modus Ponens). Given the formulas φ and $(\varphi \to \psi)$, one may infer ψ.

We will usually refer to Modus Ponens by its initials, MP. Like any rule of inference worth its salt, MP preserves truth.

PROPOSITION 3.2. *Suppose φ and ψ are formulas. Then $\{\varphi, (\varphi \to \psi)\} \models \psi$.*

With axioms and a rule of inference in hand, we can execute formal proofs in $\mathcal{L}_P$.

[1]Natural deductive systems, which are usually more convenient to actually execute deductions in than the system being developed here, compensate for having few or no axioms by having many rules of inference.

DEFINITION 3.3. Let Σ be a set of formulas. A *deduction* or *proof* from Σ in $\mathcal{L}_P$ is a finite sequence $\varphi_1\varphi_2\dots\varphi_n$ of formulas such that for each $k \leq n$,

1. φ_k is an axiom, or
2. $\varphi_k \in \Sigma$, or
3. there are $i, j < k$ such that φ_k follows from φ_i and φ_j by MP.

A formula of Σ appearing in the deduction is called a *premiss*. Σ *proves* a formula α, written as $\Sigma \vdash \alpha$, if α is the last formula of a deduction from Σ. We'll usually write $\vdash \alpha$ for $\emptyset \vdash \alpha$, and take $\Sigma \vdash \Delta$ to mean that $\Sigma \vdash \delta$ for every formula $\delta \in \Delta$.

It's worth noticing that if $\varphi_1\dots\varphi_k\dots\varphi_n$ is a deduction from Σ, then so is $\varphi_1\dots\varphi_k$ for every $k \leq n$. In order to make it easier to verify that an alleged deduction really is one, we will number the formulas in a deduction, write them out in order on separate lines, and give a justification for each formula. Like the additional connectives and conventions for dropping parentheses in Chapter 1, this is not officially a part of the definition of a deduction.

EXAMPLE 3.1. Let us show that $\vdash \varphi \to \varphi$.

1. $(\varphi \to ((\varphi \to \varphi) \to \varphi)) \to ((\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi))$	A2
2. $\varphi \to ((\varphi \to \varphi) \to \varphi)$	A1
3. $(\varphi \to (\varphi \to \varphi)) \to (\varphi \to \varphi)$	1,2 MP
4. $\varphi \to (\varphi \to \varphi)$	A1
5. $\varphi \to \varphi$	3,4 MP

Hence $\vdash \varphi \to \varphi$, as desired. Note that indication of the formulas from which formulas 3 and 5 beside the mentions of MP.

EXAMPLE 3.2. Let us show that $\{\alpha \to \beta, \beta \to \gamma\} \vdash \alpha \to \gamma$.

1. $(\beta \to \gamma) \to (\alpha \to (\beta \to \gamma))$	A1
2. $\beta \to \gamma$	Premiss
3. $\alpha \to (\beta \to \gamma)$	1,2 MP
4. $(\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma))$	A2
5. $(\alpha \to \beta) \to (\alpha \to \gamma)$	4,3 MP
6. $\alpha \to \beta$	Premiss
7. $\alpha \to \gamma$	5,6 MP

Hence $\{\alpha \to \beta, \beta \to \gamma\} \vdash \alpha \to \gamma$, as desired.

It is frequently convenient to save time and effort by simply referring to a deduction one has already done instead of writing it again as part of another deduction. If you do so, please make sure you appeal only to deductions that have already been carried out.

EXAMPLE 3.3. Let us show that $\vdash (\neg\alpha \to \alpha) \to \alpha$.

1. $(\neg\alpha \to \neg\alpha) \to ((\neg\alpha \to \alpha) \to \alpha)$ A3
2. $\neg\alpha \to \neg\alpha$ Example 3.1
3. $(\neg\alpha \to \alpha) \to \alpha$ 1,2 MP

Hence $\vdash (\neg\alpha \to \alpha) \to \alpha$, as desired. To be completely formal, one would have to insert the deduction given in Example 3.1 (with φ replaced by $\neg\alpha$ throughout) in place of line 2 above and renumber the old line 3.

PROBLEM 3.3. *Show that if α, β, and γ are formulas, then*

1. $\{\, \alpha \to (\beta \to \gamma),\, \beta \,\} \vdash \alpha \to \gamma$
2. $\vdash \alpha \lor \neg\alpha$

EXAMPLE 3.4. Let us show that $\vdash \neg\neg\beta \to \beta$.

1. $(\neg\beta \to \neg\neg\beta) \to ((\neg\beta \to \neg\beta) \to \beta)$ A3
2. $\neg\neg\beta \to (\neg\beta \to \neg\neg\beta)$ A1
3. $\neg\neg\beta \to ((\neg\beta \to \neg\beta) \to \beta)$ 1,2 Example 3.2
4. $\neg\beta \to \neg\beta$ Example 3.1
5. $\neg\neg\beta \to \beta$ 3,4 Problem 3.3.1

Hence $\vdash \neg\neg\beta \to \beta$, as desired.

Certain general facts are sometimes handy:

PROPOSITION 3.4. *If $\Gamma \vdash \delta$ and $\Gamma \vdash \delta \to \beta$, then $\Gamma \vdash \beta$.*

PROPOSITION 3.5. *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \alpha$, then $\Delta \vdash \alpha$.*

PROPOSITION 3.6. *If $\Gamma \vdash \Delta$ and $\Delta \vdash \sigma$, then $\Gamma \vdash \sigma$.*

The following theorem often lets one take substantial shortcuts when trying to show that certain deductions exist in $\mathcal{L}_P$, even though it doesn't give us the deductions explicitly.

THEOREM 3.7 (Deduction Theorem). *If Σ is any set of formulas and α and β are any formulas, then $\Sigma \vdash \alpha \to \beta$ if and only if $\Sigma \cup \{\alpha\} \vdash \beta$.*

EXAMPLE 3.5. Let us show that $\vdash \varphi \to \varphi$. By the Deduction Theorem it is enough to show that $\{\varphi\} \vdash \varphi$, which is trivial:

1. φ Premiss

Compare this to the deduction in Example 3.1.

PROBLEM 3.8. *Appealing to previous deductions and the Deduction Theorem if you wish, show that:*

1. $\{\delta, \neg\delta\} \vdash \gamma$
2. $\vdash \varphi \to \neg\neg\varphi$
3. $\vdash (\neg\beta \to \neg\alpha) \to (\alpha \to \beta)$

4. $\vdash (\alpha \to \beta) \to (\neg\beta \to \neg\alpha)$
5. $\vdash (\beta \to \neg\alpha) \to (\alpha \to \neg\beta)$
6. $\vdash (\neg\beta \to \alpha) \to (\neg\alpha \to \beta)$
7. $\vdash \sigma \to (\sigma \vee \tau)$
8. $\{\alpha \wedge \beta\} \vdash \beta$
9. $\{\alpha \wedge \beta\} \vdash \alpha$

CHAPTER 4

Soundness and Completeness

How are deduction and implication related, given that they were defined in completely different ways? We have some evidence that they behave alike; compare, for example, Proposition 2.9 and the Deduction Theorem. It had better be the case that if there is a deduction of a formula φ from a set of premisses Σ, then φ is implied by Σ. (Otherwise, what's the point of defining deductions?) It would also be nice for the converse to hold: whenever φ is implied by Σ, there is a deduction of φ from Σ. (So anything which is true can be proved.) The Soundness and Completeness Theorems say that both ways do hold, so $\Sigma \vdash \varphi$ if and only if $\Sigma \models \varphi$, *i.e.* $\vdash$ and $\models$ are equivalent for propositional logic. One direction is relatively straightforward to prove ...

THEOREM 4.1 (Soundness Theorem). *If Δ is a set of formulas and α is a formula such that $\Delta \vdash \alpha$, then $\Delta \models \alpha$.*

... but for the other direction we need some additional concepts.

DEFINITION 4.1. A set of formulas Γ is *inconsistent* if $\Gamma \vdash \neg(\alpha \to \alpha)$ for some formula α, and *consistent* if it is not inconsistent.

For example, $\{A_{41}\}$ is consistent by Proposition 4.2, but it follows from Problem 3.8 that $\{A_{13}, \neg A_{13}\}$ is inconsistent.

PROPOSITION 4.2. *If a set of formulas is satisfiable, then it is consistent.*

PROPOSITION 4.3. *Suppose Δ is an inconsistent set of formulas. Then $\Delta \vdash \psi$ for any formula ψ.*

PROPOSITION 4.4. *Suppose Σ is an inconsistent set of formulas. Then there is a finite subset Δ of Σ such that Δ is inconsistent.*

COROLLARY 4.5. *A set of formulas Γ is consistent if and only if every finite subset of Γ is consistent.*

To obtain the Completeness Theorem requires one more definition.

DEFINITION 4.2. A set of formulas Σ is *maximally consistent* if Σ is consistent but $\Sigma \cup \{\varphi\}$ is inconsistent for any $\varphi \notin \Sigma$.

That is, a set of formulas is maximally consistent if it is consistent, but there is no way to add any other formula to it and keep it consistent.

PROBLEM 4.6. *Suppose v is a truth assignment. Show that $\Sigma = \{\, \varphi \mid v(\varphi) = T \,\}$ is maximally consistent.*

We will need some facts concerning maximally consistent theories.

PROPOSITION 4.7. *If Σ is a maximally consistent set of formulas, φ is a formula, and $\Sigma \vdash \varphi$, then $\varphi \in \Sigma$.*

PROPOSITION 4.8. *Suppose Σ is a maximally consistent set of formulas and φ is a formula. Then $\neg\varphi \in \Sigma$ if and only if $\varphi \notin \Sigma$.*

PROPOSITION 4.9. *Suppose Σ is a maximally consistent set of formulas and φ and ψ are formulas. Then $\varphi \to \psi \in \Sigma$ if and only if $\varphi \notin \Sigma$ or $\psi \in \Sigma$.*

It is important to know that any consistent set of formulas can be expanded to a maximally consistent set.

THEOREM 4.10. *Suppose Γ is a consistent set of formulas. Then there is a maximally consistent set of formulas Σ such that $\Gamma \subseteq \Sigma$.*

Now for the main event!

THEOREM 4.11. *A set of formulas is consistent if and only if it is satisfiable.*

Theorem 4.11 gives the equivalence between $\vdash$ and $\models$ in slightly disguised form.

THEOREM 4.12 (Completeness Theorem). *If Δ is a set of formulas and α is a formula such that $\Delta \models \alpha$, then $\Delta \vdash \alpha$.*

It follows that anything provable from a given set of premisses must be true if the premisses are, and *vice versa*. The fact that $\vdash$ and $\models$ are actually equivalent can be very convenient in situations where one is easier to use than the other. For example, most parts of Problems 3.3 and 3.8 are much easier to do with truth tables instead of deductions, even if one makes use of the Deduction Theorem.

Finally, one more consequence of Theorem 4.11.

THEOREM 4.13 (Compactness Theorem). *A set of formulas Γ is satisfiable if and only if every finite subset of Γ is satisfiable.*

We will not look at any uses of the Compactness Theorem now, but we will consider a few applications of its counterpart for first-order logic in Chapter 9.

First-Order Logic

CHAPTER 5

Languages

As noted in the Introduction, propositional logic has obvious deficiencies as a tool for mathematical reasoning. First-order logic remedies enough of these to be adequate for formalizing most ordinary mathematics. It does have enough in common with propositional logic to let us recycle some of the material in Chapters 1–4.

A few informal words about how first-order languages work are in order. In mathematics one often deals with structures consisting of a set of elements plus various operations on them or relations among them. To cite three common examples, a group is a set of elements plus a binary operation on these elements satisfying certain conditions, a field is a set of elements plus two binary operations on these elements satisfying certain conditions, and a graph is a set of elements plus a binary relation with certain properties. In most such cases, one frequently uses symbols naming the operations or relations in question, symbols for variables which range over the set of elements, symbols for logical connectives such as *not* and *for all*, plus auxiliary symbols such as parentheses, to write formulas which express some fact about the structure in question. For example, if $(G, \cdot)$ is a group, one might express the associative law by writing something like

$$\forall x \, \forall y \, \forall z \; x \cdot (y \cdot z) = (x \cdot y) \cdot z \,,$$

it being understood that the variables range over the set G of group elements. A formal language to do as much will require some or all of these: symbols for various logical notions and for variables, some for functions or relations, plus auxiliary symbols. It will also be necessary to specify rules for putting the symbols together to make formulas, for interpreting the meaning and determining the truth of these formulas, and for making inferences in deductions.

For a concrete example, consider elementary number theory. The set of elements under discussion is the set of natural numbers $\mathbb{N} = \{0, 1, 2, 3, 4, \dots\}$. One might need symbols or names for certain interesting numbers, say 0 and 1; for variables over $\mathbb{N}$ such as n and x; for functions on $\mathbb{N}$, say $\cdot$ and $+$; and for relations, say $=$, $<$, and $|$. In addition, one is likely to need symbols for punctuation, such as (and

); for logical connectives, such as $\neg$ and $\rightarrow$; and for quantifiers, such as $\forall$ ("for all") and $\exists$ ("there exists"). A statement of mathematical English such as "For all n and m, if n divides m, then n is less than or equal to m" can then be written as a cool formula like

$$\forall n \forall m \, (n \mid m \rightarrow (n < m \land n = m)) \, .$$

The extra power of first-order logic comes at a price: greater complexity. First, there are many first-order languages one might wish to use, practically one for each subject, or even problem, in mathematics.[1] We will set up our definitions and general results, however, to apply to a wide range of them.[2]

As with $\mathcal{L}_P$, our formal language for propositional logic, first-order languages are defined by specifying their symbols and how these may be assembled into formulas.

DEFINITION 5.1. The *symbols* of a first-order language $\mathcal{L}$ include:

1. Parentheses: (and).
2. Connectives: $\neg$ and $\rightarrow$.
3. Quantifier: $\forall$.
4. Variables: v_0, v_1, v_2, ... , v_n, ...
5. Equality: $=$.
6. A (possibly empty) set of *constant* symbols.
7. For each $k \geq 1$, a (possibly empty) set of *k-place function* symbols.
8. For each $k \geq 1$, a (possibly empty) set of *k-place relation* (or *predicate*) symbols.

The symbols described in parts 1–5 are the *logical* symbols of $\mathcal{L}$, shared by every first-order language, and the rest are the *non-logical* symbols of $\mathcal{L}$, which usually depend on what the language's intended use.

NOTE. It is possible to define first-order languages without $=$, so $=$ is considered a non-logical symbol by many authors. While such languages have some uses, they are uncommon in ordinary mathematics.

Observe that any first-order language $\mathcal{L}$ has countably many logical symbols. It may have uncountably many symbols if it has uncountably many non-logical symbols. *Unless explicitly stated otherwise, we will*

[1]It is possible to formalize almost all of mathematics in a single first-order language, like that of set theory or category theory. However, trying to actually do most mathematics in such a language is so hard as to be pointless.

[2]Specifically, to countable one-sorted first-order languages with equality.

assume that every first-order language we encounter has only countably many non-logical symbols. Most of the results we will prove actually hold for countable and uncountable first-order languages alike, but some require heavier machinery to do in the uncountable case.

Just as in $\mathcal{L}_P$, the parentheses are just punctuation while the connectives, $\neg$ and $\rightarrow$, are intended to express *not* and *if ... then*. However, the rest of the symbols are new and are intended to express ideas that cannot be handled by $\mathcal{L}_P$. The quantifier symbol, $\forall$, is meant to represent *for all*, and is intended to be used with the variable symbols, *e.g.* $\forall v_4$. The constant symbols are meant to be names for particular elements of the structure under discussion. k-place function symbols are meant to name particular functions which map k-tuples of elements of the structure to elements of the structure. k-place relation symbols are intended to name particular k-place relations among elements of the structure.[3] Finally, $=$ is a special binary relation symbol intended to represent equality.

EXAMPLE 5.1. Since the logical symbols are always the same, first-order languages are usually defined by specifying the non-logical symbols. A formal language for elementary number theory like that unofficially described above, call it $\mathcal{L}_{NT}$, can be defined as follows.

- Constant symbols: 0 and 1
- Two 2-place function symbols: $+$ and $\cdot$
- Two binary relation symbols: $<$ and $|$

Each of these symbols is intended to represent the same thing it does in informal mathematical usage: 0 and 1 are intended to be names for the numbers zero and one, $+$ and $\cdot$ names for the operations of addition and multiplications, and $<$ and $|$ names for the relations "less than" and "divides". (Note that we could, in principle, interpret things completely differently – let 0 represent the number forty-one, $+$ the operation of exponentiation, and so on – or even use the language to talk about a different structure – say the real numbers, $\mathbb{R}$, with 0, 1, $+$, $\cdot$, and $<$ representing what they usually do and, just for fun, $|$ interpreted as "is not equal to". More on this in Chapter 6.) We will usually use the same symbols in our formal languages that we use informally for various common mathematical objects. This convention

[3]Intuitively, a relation or predicate expresses some (possibly arbitrary) relationship among one or more objects. For example, "n is prime" is a 1-place relation on the natural numbers, $<$ is a 2-place or binary relation on the rationals, and $\vec{a} \times (\vec{b} \times \vec{c}) = \vec{0}$ is a 3-place relation on $\mathbb{R}^3$. Formally, a k-place relation on a set X is just a subset of X^k, *i.e.* the collection of sequences of length k of elements of X for which the relation is true.

can occasionally cause confusion if it is not clear whether an expression involving these symbols is supposed to be an expression in a formal language or not.

EXAMPLE 5.2. Here are some other first-order languages. Recall that we need only specify the non-logical symbols in each case and note that some parts of Definitions 5.2 and 5.3 may be irrelevant for a given language if it is missing the appropriate sorts of non-logical symbols.

1. The language of pure equality, $\mathcal{L}_=$:
   - No non-logical symbols at all.
2. A language for fields, $\mathcal{L}_F$:
   - Constant symbols: 0, 1
   - 2-place function symbols: $+$, $\cdot$
3. A language for set theory, $\mathcal{L}_S$:
   - 2-place relation symbol: $\in$
4. A language for linear orders, $\mathcal{L}_O$:
   - 2-place relation symbol: $<$
5. Another language for elementary number theory, $\mathcal{L}_N$:
   - Constant symbol: 0
   - 1-place function symbol: S
   - 2-place function symbols: $+$, $\cdot$, E

   Here 0 is intended to represent zero, S the successor function, *i.e.* $S(n) = n+1$, and E the exponential function, *i.e.* $E(n,m) = n^m$.
6. A "worst-case" countable language, $\mathcal{L}_1$:
   - Constant symbols: c_1, c_2, c_3, $\dots$
   - For each $k \geq 1$, k-place function symbols: f_1^k, f_2^k, f_3^k, $\dots$
   - For each $k \geq 1$, k-place relation symbols: P_1^k, P_2^k, P_3^k, $\dots$

   This language has no use except as an abstract example.

It remains to specify how to form valid formulas from the symbols of a first-order language $\mathcal{L}$. This will be more complicated than it was for $\mathcal{L}_P$. In fact, we first need to define a type of expression in $\mathcal{L}$ which has no counterpart in propositional logic.

DEFINITION 5.2. The *terms* of a first-order language $\mathcal{L}$ are those finite sequences of symbols of $\mathcal{L}$ which satisfy the following rules:

1. Every variable symbol v_n is a term.
2. Every constant symbol c is a term.
3. If f is a k-place function symbol and $t_1, \dots, t_k$ are terms, then $ft_1 \dots t_k$ is also a term.
4. Nothing else is a term.

That is, a term is an expression which represents some (possibly indeterminate) element of the structure under discussion. For example, in $\mathcal{L}_{NT}$ or $\mathcal{L}_N$, $+v_0v_1$ (informally, $v_0 + v_1$) is a term, though precisely which natural number it represents depends on what values are assigned to the variables v_0 and v_1.

PROBLEM 5.1. *Which of the following are terms of one of the languages defined in Examples 5.1 and 5.2? If so, which of these language(s) are they terms of; if not, why not?*

1. $\cdot v_2$
2. $+0 \cdot +v_6 11$
3. $|1 + v_3 0$
4. $(< E101 \to +11)$
5. $+ + \cdot + 00000$
6. $f_4^3 f_7^2 c_4 v_9 c_1 v_4$
7. $\cdot v_5(+1v_8)$
8. $< v_6 v_2$
9. $1 + 0$

Note that in languages with no function symbols all terms have length one.

PROBLEM 5.2. *Choose one of the languages defined in Examples 5.1 and 5.2 which has terms of length greater than one and determine the possible lengths of terms of this language.*

PROPOSITION 5.3. *The set of terms of a countable first-order language $\mathcal{L}$ is countable.*

Having defined terms, we can finally define first-order formulas.

DEFINITION 5.3. The *formulas* of a first-order language $\mathcal{L}$ are the finite sequences of the symbols of $\mathcal{L}$ satisfying the following rules:

1. If P is a k-place relation symbol and $t_1, \dots, t_k$ are terms, then $Pt_1 \dots t_k$ is a formula.
2. If t_1 and t_2 are terms, then $= t_1t_2$ is a formula.
3. If α is a formula, then $(\neg\alpha)$ is a formula.
4. If α and β are formulas, then $(\alpha \to \beta)$ is a formula.
5. If φ is a formula and v_n is a variable, then $\forall v_n\varphi$ is a formula.
6. Nothing else is a formula.

Formulas of form 1 or 2 will often be referred to as the *atomic formulas* of $\mathcal{L}$.

Note that three of the conditions in Definition 5.3 are borrowed directy from propositional logic. As before, we will exploit the way formulas are built up in making definitions and in proving results by induction on the length of a formula. We will also recycle the use of lower-case Greek characters to refer to formulas and of upper-case Greek characters to refer to sets of formulas.

Problem 5.4. *Which of the following are formulas of one of the languages defined in Examples 5.1 and 5.2? If so, which of these language(s) are they formulas of; if not, why not?*

1. $= 0 + v_7 \cdot 1 v_3$
2. $(\neg = v_1 v_1)$
3. $(|v_2 0 \to \cdot 01)$
4. $(\neg \forall v_5 (= v_5 v_5))$
5. $< +01 | v_1 v_3$
6. $(v_3 = v_3 \to \forall v_5\, v_3 = v_5)$
7. $\forall v_6 (= v_6 0 \to \forall v_9 (\neg | v_9 v_6))$
8. $\forall v_8 < +11 v_4$

Problem 5.5. *Show that every formula of a first-order language has the same number of left parentheses as of right parentheses.*

Problem 5.6. *Choose one of the languages defined in Examples 5.1 and 5.2 and determine the possible lengths of formulas of this language.*

Proposition 5.7. *A countable first-order language $\mathcal{L}$ has countably many formulas.*

In practice, devising a formal language intended to deal with a particular (kind of) structure isn't the end of the job: one must also specify axioms in the language that the structure(s) one wishes to study should satisfy. Defining satisfaction is officially done in the next chapter, but it is usually straightforward to unofficially figure out what a formula in the language is supposed to mean.

Problem 5.8. *In each case, write down a formula of the given language expressing the given informal statement.*

1. *"Addition is associative" in $\mathcal{L}_F$.*
2. *"There is an empty set" in $\mathcal{L}_S$.*
3. *"Between any two distinct elements there is a third element" in $\mathcal{L}_O$.*
4. *"$n^0 = 1$ for every n different from 0" in $\mathcal{L}_N$.*
5. *"There is only one thing" in $\mathcal{L}_=$.*

Problem 5.9. *Define first-order languages to deal with the following structures and, in each case, an appropriate set of axioms in your language:*

1. *Groups.*
2. *Graphs.*
3. *Vector spaces.*

We will need a few additional concepts and facts about formulas of first-order logic later on. First, what are the subformulas of a formula?

PROBLEM 5.10. *Define the set of subformulas of a formula φ of a first-order language $\mathcal{L}$.*

For example, if φ is

$$(((\neg\forall v_1\,(\neg = v_1c_7)) \to P_3^2v_5v_8) \to \forall v_8(= v_8f_5^3c_0v_1v_5 \to P_2^1v_8))$$

in the language $\mathcal{L}_1$, then the set of subformulas of φ, $\mathcal{S}(\varphi)$, ought to include

- $= v_1c_7$, $P_3^2v_5v_8$, $= v_8f_5^3c_0v_1v_5$, $P_2^1v_8$,
- $(\neg = v_1c_7)$, $(= v_8f_5^3c_0v_1v_5 \to P_2^1v_8)$,
- $\forall v_1\,(\neg = v_1c_7)$, $\forall v_8(= v_8f_5^3c_0v_1v_5 \to P_2^1v_8)$,
- $(\neg\forall v_1\,(\neg = v_1c_7))$, and
- $(((\neg\forall v_1\,(\neg = v_1c_7)) \to P_3^2v_5v_8) \to \forall v_8(= v_8f_5^3c_0v_1v_5 \to P_2^1v_8))$ itself.

Second, we will need a concept that has no counterpart in propositional logic.

DEFINITION 5.4. Suppose x is a variable of a first-order language $\mathcal{L}$. Then x to *occurs free* in a formula φ of $\mathcal{L}$ is defined as follows:

1. If φ is atomic, then x occurs free in φ if and only if x occurs in φ.
2. If φ is $(\neg\alpha)$, then x occurs free in φ if and only if x occurs free in α.
3. If φ is $(\beta \to \delta)$, then x occurs free in φ if and only if x occurs free in β or in δ.
4. If φ is $\forall v_k\,\psi$, then x occurs free in φ if and only if x is different from v_k and x occurs free in ψ.

An occurrence of x in φ which is not free is said to be *bound*. A formula σ of $\mathcal{L}$ in which no variable occurs free is said to be a *sentence*.

Part 4 is the key: it asserts that an occurrence of a variable x is bound instead of free if it is in the "scope" of an occurrence of $\forall x$. For example, v_7 is free in $\forall v_5 = v_5v_7$, but v_5 is not. Different occurences of a given variable in a formula may be free or bound, depending on where they are; *e.g.* v_6 occurs both free and bound in $\forall v_0\,(= v_0f_3^1v_6 \to (\neg\forall v_6\,P_9^1v_6))$.

PROBLEM 5.11. *Give a precise definition of the scope of a quantifier.*

Note the distinction between sentences and ordinary formulas introduced in the last part of Definition 5.4. As we shall see, sentences are often more tractable and useful theoretically than ordinary formulas.

PROBLEM 5.12. *Which of the formulas you gave in solving Problem 5.8 are sentences?*

Finally, we will eventually need to consider a relationship between first-order languages.

DEFINITION 5.5. A first-order language $\mathcal{L}'$ is an *extension* of a first-order language $\mathcal{L}$, sometimes written as $\mathcal{L} \subseteq \mathcal{L}'$, if every non-logical symbol of $\mathcal{L}$ is a non-logical symbol of the same kind of $\mathcal{L}'$.

For example, every first-order language is an extension of $\mathcal{L}_=$.

PROBLEM 5.13. *Which of the languages given in Example 5.2 are extensions of other languages given in Example 5.2?*

PROPOSITION 5.14. *Suppose $\mathcal{L}$ is a first-order language and $\mathcal{L}'$ is an extension of $\mathcal{L}$. Then every formula φ of $\mathcal{L}$ is a a formula of $\mathcal{L}'$.*

Common Conventions. As with propositional logic, we will often use abbreviations and informal conventions to simplify the writing of formulas in first-order languages. In particular, we will use the same additional connectives we used in propositional logic, plus an additional quantifier, $\exists$ ("there exists"):

- $(\alpha \wedge \beta)$ is short for $(\neg(\alpha \rightarrow (\neg\beta)))$.
- $(\alpha \vee \beta)$ is short for $((\neg\alpha) \rightarrow \beta)$.
- $(\alpha \leftrightarrow \beta)$ is short for $((\alpha \rightarrow \beta) \wedge (\beta \rightarrow \alpha))$.
- $\exists v_k \varphi$ is short for $(\neg\forall v_k(\neg\varphi))$.

($\forall$ is often called the universal quantifier and $\exists$ is often called the existential quantifier.)

Parentheses will often be omitted in formulas according to the same conventions we used in propositional logic, with the modification that $\forall$ and $\exists$ take precedence over all the logical connectives:

- We will usually drop the outermost parentheses in a formula, writing $\alpha \rightarrow \beta$ instead of $(\alpha \rightarrow \beta)$ and $\neg\alpha$ instead of $(\neg\alpha)$.
- We will let $\forall$ take precedence over $\neg$, and $\neg$ take precedence over $\rightarrow$ when parentheses are missing, and fit the informal abbreviations into this scheme by letting the order of precedence be $\forall$, $\exists$, $\neg$, $\wedge$, $\vee$, $\rightarrow$, and $\leftrightarrow$.

- Finally, we will group repetitions of $\to$, $\lor$, $\land$, or $\leftrightarrow$ to the right when parentheses are missing, so $\alpha \to \beta \to \gamma$ is short for $(\alpha \to (\beta \to \gamma))$.

For example, $\exists v_k \neg\alpha \to \forall v_n \beta$ is short for $((\neg\forall v_k(\neg(\neg\alpha))) \to \forall v_n\beta)$. On the other hand, we will sometimes add parentheses and arrange things in unofficial ways to make terms and formulas easier to read. In particular we will often write

1. $f(t_1, \dots, t_k)$ for $ft_1 \dots t_k$ if f is a k-place function symbol and $t_1, \dots, t_k$ are terms,
2. $s \circ t$ for $\circ st$ if $\circ$ is a 2-place function symbol and s and t are terms,
3. $P(t_1, \dots, t_k)$ for $Pt_1 \dots t_k$ if P is a k-place relation symbol and $t_1, \dots, t_k$ are terms,
4. $s \bullet t$ for $\bullet st$ if $\bullet$ is a 2-place relation symbol and s and t are terms, and
5. $s = t$ for $= st$ if s and t are terms, and
6. enclose terms in parentheses to group them.

Thus, we could write the formula $= +1 \cdot 0 v_6 \cdot 11$ of $\mathcal{L}_{NT}$ as $1 + (0 \cdot v_6) = 1 \cdot 1$.

As was observed in Example 5.1, it is customary in devising a formal language to recycle the same symbols used informally for the given objects. In situations where we want to talk about symbols without committing ourselves to a particular one, such as when talking about first-order languages in general, we will often use "generic" choices:

- a, b, c, ... for constant symbols;
- x, y, z, ... for variable symbols;
- f, g, h, ... for function symbols;
- P, Q, R, ... for relation symbols; and
- r, s, t, ... for generic terms.

These can be thought of as variables in the metalanguage[4] ranging over different kinds objects of first-order logic, much as we're already using lower-case Greek characters as variables which range over formulas. (In fact, we have already used some of these conventions in this chapter ...)

Unique Readability. The slightly paranoid might ask whether Definitions 5.1, 5.2 and 5.3 actually ensure that the terms and formulas

[4]The metalanguage is the language, mathematical English in this case, in which we talk *about* a language. The theorems we prove about formal logic are, strictly speaking, metatheorems, as opposed to the theorems proved within a formal logical system. For more of this kind of stuff, read some philosophy ...

of a first-order language $\mathcal{L}$ are unambiguous, *i.e.* cannot be read in more than one way. As with $\mathcal{L}_P$, to actually prove this one must assume that all the symbols of $\mathcal{L}$ are distinct and that no symbol is a subsequence of any other symbol. It then follows that:

THEOREM 5.15. *Any term of a first-order language $\mathcal{L}$ satisfies exactly one of conditions 1–3 in Definition 5.2.*

THEOREM 5.16 (Unique Readability Theorem). *Any formula of a first-order language satisfies exactly one of conditions 1–5 in Definition 5.3.*

CHAPTER 6

Structures and Models

Defining truth and implication in first-order logic is a lot harder than it was in propositional logic. First-order languages are intended to deal with mathematical objects like groups or linear orders, so it makes little sense to speak of the truth of a formula without specifying a context. For example, one can write down a formula expressing the commutative law in a language for group theory, $\forall x\,\forall y\,x \cdot y = y \cdot x$, but whether it is true or not depends on which group we're dealing with. It follows that we need to make precise which mathematical objects or structures a given first-order language can be used to discuss and how, given a suitable structure, formulas in the language are to be interpreted. Such a structure for a given language should supply most of the ingredients needed to interpret formulas of the language. Throughout this chapter, let $\mathcal{L}$ be an arbitrary fixed countable first-order language. All formulas will be assumed to be formulas of $\mathcal{L}$ unless stated otherwise.

DEFINITION 6.1. A *structure* $\mathfrak{M}$ for $\mathcal{L}$ consists of the following:

1. A non-empty set M, often written as $|\mathfrak{M}|$, called the *universe* of $\mathfrak{M}$.
2. For each constant symbol c of $\mathcal{L}$, an element $c^{\mathfrak{M}}$ of M.
3. For each k-place function symbol f of $\mathcal{L}$, a function $f^{\mathfrak{M}} : M^k \to M$, *i.e.* a k-place function on M.
4. For each k-place relation symbol P of $\mathcal{L}$, a relation $P^{\mathfrak{M}} \subseteq M^k$, *i.e.* a k-place relation on M.

That is, a structure supplies an underlying set of elements plus interpretations for the various non-logical symbols of the language: constant symbols are interpreted by particular elements of the underlying set, function symbols by functions on this set, and relation symbols by relations among elements of this set.

It is customary to use upper-case "gothic" characters such as $\mathfrak{M}$ and $\mathfrak{N}$ for structures.

For example, consider $\mathfrak{Q} = (\mathbb{Q}, <)$, where $<$ is the usual "less than" relation on the rationals. This is a structure for $\mathcal{L}_O$, the language for linear orders defined in Example 5.2; it supplies a 2-place relation to interpret the language's 2-place relation symbol. $\mathfrak{Q}$ is *not* the only possible structure for $\mathcal{L}_O$: $(\mathbb{R}, <)$, $(\{0\}, \emptyset)$, and $(\mathbb{N}, \mathbb{N}^2)$ are three others among infinitely many more. (Note that in these cases the relation symbol $<$ is interpreted by relations on the universe which are not linear orders. One can ensure that a structure satisfy various conditions beyond what Definition 6.1 guarantees by requiring appropriate formulas to be true when interpreted in the structure.) On the other hand, $(\mathbb{R})$ is not a structure for $\mathcal{L}_O$ because it lacks a binary relation to interpret the symbol $<$ by, while $(\mathbb{N}, 0, 1, +, \cdot, |, <)$ is not a structure for $\mathcal{L}_O$ because it has two binary relations where $\mathcal{L}_O$ has a symbol only for one, plus constants and functions for which $\mathcal{L}_O$ lacks symbols.

Problem 6.1. *The first-order languages referred to below were all defined in Example 5.2.*

1. *Is* $(\emptyset)$ *a structure for* $\mathcal{L}_=$*?*
2. *Determine whether* $\mathfrak{Q} = (\mathbb{Q}, <)$ *is a structure for each of* $\mathcal{L}_=$*,* $\mathcal{L}_F$*, and* $\mathcal{L}_S$*.*
3. *Give three different structures for* $\mathcal{L}_F$ *which are not fields.*

To determine what it means for a given formula to be true in a structure for the corresponding language, we will also need to specify how to interpret the variables when they occur free. (Bound variables have the associated quantifier to tell us what to do.)

Definition 6.2. Let $V = \{ v_0, v_1, v_2, \dots \}$ be the set of all variable symbols of $\mathcal{L}$ and suppose $\mathfrak{M}$ is a structure for $\mathcal{L}$. A function $s : V \to |\mathfrak{M}|$ is said to be an *assignment* for $\mathfrak{M}$.

Note that these are *not* truth assignments like those for $\mathcal{L}_P$. An assignment just interprets each variable in the language by an element of the universe of the structure. Also, as long as the universe of the structure has more than one element, any variable can be interpreted in more than one way. Hence there are usually many different possible assignments for a given structure.

Example 6.1. Consider the structure $\mathfrak{R} = (\mathbb{R}, 0, 1, +, \cdot)$ for $\mathcal{L}_F$. Each of the following functions $V \to \mathbb{R}$ is an assignment for $\mathfrak{R}$:

1. $p(v_n) = \pi$ for each n,
2. $r(v_n) = e^n$ for each n, and
3. $s(v_n) = n + 1$ for each n.

In fact, *every* function $V \to \mathbb{R}$ is an assignment for $\mathfrak{R}$.

In order to use assignments to determine whether formulas are true in a structure, we need to know how to use an assignment to interpret each term of the language as an element of the universe.

DEFINITION 6.3. Suppose $\mathfrak{M}$ is a structure for $\mathcal{L}$ and $s\colon V \to |\mathfrak{M}|$ is an assignment for $\mathfrak{M}$. Let T be the set of all terms of $\mathcal{L}$. Then the *extended assignment* $\mathbf{s}\colon T \to |\mathfrak{M}|$ is defined inductively as follows:

1. For each variable x, $\mathbf{s}(x) = s(x)$.
2. For each constant symbol c, $\mathbf{s}(c) = c^{\mathfrak{M}}$.
3. For every k-place function symbol f and terms $t_1, \dots, t_k$,

$$\mathbf{s}(ft_1 \dots t_k) = f^{\mathfrak{M}}(\mathbf{s}(t_1), \dots, \mathbf{s}(t_k)).$$

EXAMPLE 6.2. Let $\mathfrak{R}$ be the structure for $\mathcal{L}_F$ given in Example 6.1, and let $\mathbf{p}$, $\mathbf{r}$, and $\mathbf{s}$ be the extended assignments corresponding to the assignments p, r, and s defined in Example 6.1. Consider the term $+ \cdot v_6 v_0 + 0 v_3$ of $\mathcal{L}_F$. Then:

1. $\mathbf{p}(+ \cdot v_6 v_0 + 0 v_3) = \pi^2 + \pi$,
2. $\mathbf{r}(+ \cdot v_6 v_0 + 0 v_3) = e^6 + e^3$, and
3. $\mathbf{s}(+ \cdot v_6 v_0 + 0 v_3) = 11$.

Here's why for the last one: since $s(v_6) = 7$, $s(v_0) = 1$, $s(v_3) = 4$, and $\mathbf{s}(0) = 0$ (by part 2 of Definition 6.3), it follows from part 3 of Definition 6.3 that $\mathbf{s}(+ \cdot v_6 v_0 + 0 v_3) = (7 \cdot 1) + (0 + 4) = 7 + 4 = 11$.

PROBLEM 6.2. *$\mathfrak{N} = (\mathbb{N}, 0, 1, S, +, \cdot, E)$ is a structure for $\mathcal{L}_N$. Let $s\colon V \to \mathbb{N}$ be the assignment defined by $s(v_k) = k + 1$. What are $\mathbf{s}(E + v_{19} v_1 \cdot 0 v_{45})$ and $\mathbf{s}(SSS + E1 v_6 v_7)$?*

PROPOSITION 6.3. $\mathbf{s}$ *is unique,* i.e. *given an assignment s, no other function $T \to |\mathfrak{M}|$ satisfies conditions 1–3 in Definition 6.3.*

With Definitions 6.2 and 6.3 in hand, we can take our first cut at defining what it means for a first-order formula to be true.

DEFINITION 6.4. Suppose $\mathfrak{M}$ is a structure for $\mathcal{L}$, s is an assignment for $\mathfrak{M}$, and φ is a formula of $\mathcal{L}$. Then $\mathfrak{M} \models \varphi[s]$ is defined as follows:

1. If φ is $t_1 = t_2$ for some terms t_1 and t_2, then $\mathfrak{M} \models \varphi[s]$ if and only if $\mathbf{s}(t_1) = \mathbf{s}(t_2)$.
2. If φ is $Pt_1 \dots t_k$ for some k-place relation symbol P and terms $t_1, \dots, t_k$, then $\mathfrak{M} \models \varphi[s]$ if and only if $(\mathbf{s}(t_1), \dots, \mathbf{s}(t_k)) \in P^{\mathfrak{M}}$, *i.e.* $P^{\mathfrak{M}}$ is true of $(\mathbf{s}(t_1), \dots, \mathbf{s}(t_k))$.
3. If φ is $(\neg\psi)$ for some formula ψ, then $\mathfrak{M} \models \varphi[s]$ if and only if it is not the case that $\mathfrak{M} \models \psi[s]$.
4. If φ is $(\alpha \to \beta)$, then $\mathfrak{M} \models \varphi[s]$ if and only if $\mathfrak{M} \models \beta[s]$ whenever $\mathfrak{M} \models \alpha[s]$, *i.e.* unless $\mathfrak{M} \models \alpha[s]$ but not $\mathfrak{M} \models \beta[s]$.
5. If φ is $\forall x\, \delta$ for some variable x, then $\mathfrak{M} \models \varphi[s]$ if and only if $\mathfrak{M} \models \delta[s(x|m)]$ for all $m \in |\mathfrak{M}|$, where $s(x|m)$ is the assignment

given by

$$s(x|m)(v_k) = \begin{cases} s(v_k) & \text{if } v_k \text{ is different from } x \\ m & \text{if } v_k \text{ is } x. \end{cases}$$

If $\mathfrak{M} \models \varphi[s]$, we shall say that $\mathfrak{M}$ *satisfies* φ *on assignment* s or that φ *is true in* $\mathfrak{M}$ *on assignment* s. We will often write $\mathfrak{M} \nvDash \varphi[s]$ if it is not the case that $\mathfrak{M} \models \varphi[s]$. Also, if Γ is a set of formulas of $\mathcal{L}$, we shall take $\mathfrak{M} \models \Gamma[s]$ to mean that $\mathfrak{M} \models \gamma[s]$ for every formula γ in Γ and say that $\mathfrak{M}$ *satisfies* Γ *on assignment* s. Similarly, we shall take $\mathfrak{M} \nvDash \Gamma[s]$ to mean that $\mathfrak{M} \nvDash \gamma[s]$ for *some* formula γ in Γ.

Clauses 1 and 2 are pretty straightforward and clauses 3 and 4 are essentially identical to the corresponding parts of Definition 2.1. The key clause is 5, which says that $\forall$ should be interpreted as "for all elements of the universe".

EXAMPLE 6.3. Let $\mathfrak{R}$ be the structure for $\mathcal{L}_F$ and s the assignment for $\mathfrak{R}$ given in Example 6.1, and consider the formula $\forall v_1 (= v_3 \cdot 0 v_1 \to = v_1 0)$ of $\mathcal{L}_F$. We can verify that $\mathfrak{R} \models \forall v_1 (= v_3 \cdot 0 v_1 \to = v_3 0)\, [s]$ as follows:

$$\begin{aligned} & \mathfrak{R} \models \forall v_1 (= v_3 \cdot 0 v_1 \to = v_3 0)\, [s] \\ \iff & \mathfrak{R} \models (= v_3 \cdot 0 v_1 \to = v_3 0)\, [s(v_1|a)] \text{ for all } a \in |\mathfrak{R}| \\ \iff & \mathfrak{R} \models = v_3 0\, [s(v_1|a)] \text{ if } \mathfrak{R} \models = v_3 \cdot 0 v_1\, [s(v_1|a)] \text{ for all } a \in |\mathfrak{R}| \\ \iff & \mathbf{s}(v_1|a)(v_3) = \mathbf{s}(v_1|a)(0) \text{ if } \mathbf{s}(v_1|a)(v_3) = \mathbf{s}(v_1|a)(\cdot 0 v_1) \text{ for all } a \in |\mathfrak{R}| \\ \iff & s(v_3) = 0 \text{ if } s(v_3) = \mathbf{s}(v_1|a)(0) \cdot mathbfs(v_1|a)(v_1) \text{ for all } a \in |\mathfrak{R}| \\ \iff & 4 = 0 \text{ if } 4 = 0 \cdot a \text{ for all } a \in |\mathfrak{R}| \\ \iff & 4 = 0 \text{ if } 4 = 0 \text{ for all } a \in |\mathfrak{R}|. \end{aligned}$$

PROBLEM 6.4. *Let* $\mathfrak{N}$ *be the structure for* $\mathcal{L}_N$ *given in Problem 6.2. Let* $p : V \to \mathbb{N}$ *be defined by* $p(v_{2k}) = k$ *and* $p(v_{2k+1}) = k$. *Verify that*

1. $\mathfrak{N} \models \forall w\, (\neg S w = 0)\, [p]$ *and*
2. $\mathfrak{N} \nvDash \forall x \exists y\, x + y = 0\, [p]$.

PROPOSITION 6.5. *Suppose* $\mathfrak{M}$ *is a structure for* $\mathcal{L}$, s *is an assignment for* $\mathfrak{M}$, x *is a variable, and* φ *is a formula of a first-order language* $\mathcal{L}$. *Then* $\mathfrak{M} \models \exists x\, \varphi[s]$ *if and only if* $\mathfrak{M} \models \varphi[s(x|m)]$ *for some* $m \in |\mathfrak{M}|$.

Working with particular assignments is difficult but, while sometimes unavoidable, not always necessary.

DEFINITION 6.5. Suppose $\mathfrak{M}$ is a structure for $\mathcal{L}$, and φ a formula of $\mathcal{L}$. Then $\mathfrak{M} \models \varphi$ if and only if $\mathfrak{M} \models \varphi[s]$ for every assignment $s : V \to |\mathfrak{M}|$ for $\mathfrak{M}$. $\mathfrak{M}$ is a *model* of φ or that φ is *true* in $\mathfrak{M}$ if $\mathfrak{M} \models \varphi$. We will often write $\mathfrak{M} \nvDash \psi$ if it is not the case that $\mathfrak{M} \models \psi$

Similarly, if Γ is a set of formulas, we will write $\mathfrak{M} \models \Gamma$ if $\mathfrak{M} \models \gamma$ for every formula $\gamma \in \Gamma$, and say that $\mathfrak{M}$ is a *model* of Γ or that $\mathfrak{M}$ *satisfies* Γ. A formula or set of formulas is *satisfiable* if there is some structure $\mathfrak{M}$ which satisfies it. We will often write $\mathfrak{M} \not\models \Gamma$ if it is not the case that $\mathfrak{M} \models \Gamma$.

PROBLEM 6.6. *$\mathfrak{Q} = (\mathbb{Q}, <)$ is a structure for $\mathcal{L}_O$. For each of the following formulas φ of $\mathcal{L}_O$, determine whether or not $\mathfrak{Q} \models \varphi$.*

1. $\forall v_0 \, \exists v_2 \, v_0 < v_2$
2. $\exists v_1 \, \forall v_3 \, (v_1 < v_3 \to v_1 = v_3)$
3. $\forall v_4 \, \forall v_5 \, \forall v_6 (v_4 < v_5 \to (v_5 < v_6 \to v_4 < v_6))$

The following facts are counterparts of sorts for Proposition 2.2. Their point is that what a given assignment does with a given term or formula depends only on the assignment's values on the (free) variables of the term or formula.

LEMMA 6.7. *Suppose $\mathfrak{M}$ is a structure for $\mathcal{L}$, t is a term of $\mathcal{L}$, and r and s are assignments for $\mathfrak{M}$ such that $r(x) = s(x)$ for every variable x which occurs in t. Then $\mathbf{r}(t) = \mathbf{s}(t)$.*

PROPOSITION 6.8. *Suppose $\mathfrak{M}$ is a structure for $\mathcal{L}$, φ is a formula of $\mathcal{L}$, and r and s are assignments for $\mathfrak{M}$ such that $r(x) = s(x)$ for every variable x which occurs free in φ. Then $\mathfrak{M} \models \varphi[r]$ if and only if $\mathfrak{M} \models \varphi[s]$.*

COROLLARY 6.9. *Suppose $\mathfrak{M}$ is a structure for $\mathcal{L}$ and σ is a sentence of $\mathcal{L}$. Then $\mathfrak{M} \models \sigma$ if and only if there is some assignment $s : V \to |\mathfrak{M}|$ for $\mathfrak{M}$ such that $\mathfrak{M} \models \sigma[s]$.*

Thus sentences are true or false in a structure independently of any particular assignment. This does not necessarily make life easier when trying to verify whether a sentence is true in a structure – try doing Problem 6.6 again with the above results in hand – but it does let us simplify things on occasion when proving things about sentences rather than formulas.

We recycle a sense in which we used $\models$ in propositional logic.

DEFINITION 6.6. Suppose Γ is a set of formulas of $\mathcal{L}$ and ψ is a formula of $\mathcal{L}$. Then Γ *implies* ψ, written as $\Gamma \models \psi$, if $\mathfrak{M} \models \psi$ whenever $\mathfrak{M} \models \Gamma$ for every structure $\mathfrak{M}$ for $\mathcal{L}$.

Similarly, if Γ and Δ are sets of formulas of $\mathcal{L}$, then Γ *implies* Δ, written as $\Gamma \models \Delta$, if $\mathfrak{M} \models \Delta$ whenever $\mathfrak{M} \models \Gamma$ for every structure $\mathfrak{M}$ for $\mathcal{L}$.

We will usually write $\models \dots$ for $\emptyset \models \dots$.

DEFINITION 6.7. A formula ψ of $\mathcal{L}$ is a *tautology* if it is true in every structure, *i.e.* if $\models \psi$. ψ is a *contradiction* if there is no structure in which it is true, *i.e.* $\mathfrak{M} \not\models \psi$ for every structure $\mathfrak{M}$ for $\mathcal{L}$.

For some trivial examples, let φ be a formula of $\mathcal{L}$ and $\mathfrak{M}$ a structure for $\mathcal{L}$. Then $\mathfrak{M} \models \{\varphi\}$ if and only if $\mathfrak{M} \models \varphi$, so it must be the case that $\{\varphi\} \models \varphi$. It is also easy to check that $\varphi \to \varphi$ is a tautology and $\neg(\varphi \to \varphi)$ is a contradiction.

PROPOSITION 6.10. *Suppose α and β are formulas of some first-order language. Then $\{ (\alpha \to \beta), \alpha \} \models \beta$.*

PROBLEM 6.11. *Show that $\forall y\, y = y$ is a tautology and that $\exists y\, \neg y = y$ is a contradiction.*

PROPOSITION 6.12. *Suppose Σ is a set of formulas and ψ and ρ are formulas of some first-order language. Then $\Sigma \cup \{\psi\} \models \rho$ if and only if $\Sigma \models (\psi \to \rho)$.*

PROBLEM 6.13. *Show that a set of formulas Σ is satisfiable if and only if there is no contradiction χ such that $\Sigma \models \chi$.*

The following fact is a counterpart of Proposition 2.4.

PROPOSITION 6.14. *Suppose $\mathfrak{M}$ is a structure for $\mathcal{L}$ and α and β are formulas of $\mathcal{L}$. Then:*

1. $\mathfrak{M} \models \neg\alpha$ *if and only if* $\mathfrak{M} \not\models \alpha$.
2. $\mathfrak{M} \models \alpha \to \beta$ *if and only if* $\mathfrak{M} \models \beta$ *whenever* $\mathfrak{M} \models \alpha$.
3. $\mathfrak{M} \models \alpha \vee \beta$ *if and only if* $\mathfrak{M} \models \alpha$ *or* $\mathfrak{M} \models \beta$.
4. $\mathfrak{M} \models \alpha \wedge \beta$ *if and only if* $\mathfrak{M} \models \alpha$ *and* $\mathfrak{M} \models \beta$.
5. $\mathfrak{M} \models \alpha \leftrightarrow \beta$ *if and only if* $\mathfrak{M} \models \alpha$ *exactly when* $\mathfrak{M} \models \beta$.
6. $\mathfrak{M} \models \forall x\, \alpha$ *if and only if* $\mathfrak{M} \models \alpha$.
7. $\mathfrak{M} \models \exists x\, \alpha$ *if and only if there is some* $m \in |\mathfrak{M}|$ *so that* $\mathfrak{M} \models \alpha\,[s(x|m)]$ *for every assignment* s *for* $\mathfrak{M}$.

Recall that by Proposition 5.14 a formula of a first-order language is also a formula of any extension of the language. The following relationship between extension languages and satisfiability will be needed later on.

PROPOSITION 6.15. *Suppose $\mathcal{L}$ is a first-order language, $\mathcal{L}'$ is an extension of $\mathcal{L}$, and Γ is a set of formulas of $\mathcal{L}$. Then Γ is satisfiable in a structure for $\mathcal{L}$ if and only if Γ is satisfiable in a structure for $\mathcal{L}'$.*

One last bit of terminology ...

DEFINITION 6.8. If $\mathfrak{M}$ is a structure for $\mathcal{L}$, then the *theory* of $\mathfrak{M}$ is just the set of all sentences of $\mathcal{L}$ true in $\mathfrak{M}$, *i.e.*

$$\mathrm{Th}(\mathfrak{M}) = \{\, \tau \mid \tau \text{ is a sentence and } \mathfrak{M} \models \tau \,\}.$$

If Δ is a set of sentences and $\mathcal{S}$ is a collection of structures, then Δ is a set of (non-logical) *axioms* for $\mathcal{S}$ if for every structure $\mathfrak{M}$, $\mathfrak{M} \in \mathcal{S}$ if and only if $\mathfrak{M} \models \Delta$.

EXAMPLE 6.4. Consider the sentence $\exists x\, \exists y\, ((\neg x = y) \land \forall z\, (z = x \lor z = y))$ of $\mathcal{L}_=$. Every structure of $\mathcal{L}_=$ satisfying this sentence must have exactly two elements in its universe, so $\{\, \exists x\, \exists y\, ((\neg x = y) \land \forall z\, (z = x \lor z = y))\,\}$ is a set of non-logical axioms for the collection of sets of cardinality 2:

$$\{\, \mathfrak{M} \mid \mathfrak{M} \text{ is a structure for } \mathcal{L}_= \text{ with exactly 2 elements} \,\}.$$

PROBLEM 6.16. *In each case, find a suitable language and a set of axioms in it for the given collection of structures.*

1. *Sets of size 3.*
2. *Bipartite graphs.*
3. *Commutative groups.*
4. *Fields of characteristic 5.*

CHAPTER 7

Deductions

Deductions in first-order logic are not unlike deductions in propositional logic. Of course, some changes are necessary to handle the various additional features of propositional logic, especially quantifiers. In particular, one of the new axioms requires a tricky preliminary definition. Roughly, the problem is that we need to know when we can replace occurrences of a variable in a formula by a term without letting any variable in the term get captured by a quantifier.

Throughout this chapter, let $\mathcal{L}$ be a fixed arbitrary first-order language. Unless stated otherwise, all formulas will be assumed to be formulas of $\mathcal{L}$.

Definition 7.1. Suppose x is a variable, t is a term, and φ is a formula. Then *t is substitutable for x in φ* is defined as follows:

1. If φ is atomic, then t is substitutable for x in φ.
2. If φ is $(\neg\psi)$, then t is substitutable for x in φ if and only if t is substitutable for x in ψ.
3. If φ is $(\alpha \to \beta)$, then t is substitutable for x in φ if and only if t is substitutable for x in α and t is substitutable for x in β.
4. If φ is $\forall y\, \delta$, then t is substitutable for x in φ if and only if either
   (a) x does not occur free in φ, or
   (b) if y does not occur in t and t is substitutable for x in δ.

If t is substitutable for x in φ, then φ^x_t (*i.e.* φ with t substituted for x) is defined as follows:

1. If φ is atomic, then φ^x_t is the formula obtained by replacing each occurrence of x in φ by t.
2. If φ is $(\neg\psi)$, then φ^x_t is the formula $(\neg\psi^x_t)$.
3. If φ is $(\alpha \to \beta)$, then φ^x_t is the formula $(\alpha^x_t \to \beta^x_t)$.
4. If φ is $\forall y\, \delta$, then φ^x_t is the formula
   (a) $\forall y\, \delta$ if x is y, and
   (b) $\forall y\, \delta^x_t$ if x isn't y.

For example, x is always substitutable for itself in any formula φ and φ^x_x is just φ (Problem 7.1). For another, y is not substitutable for x in $\forall y\, x = y$ because if x were to be replaced by y, the new instance of

y would be bound by the quantifier $\forall y$. Note that the truth of $\forall y\, x = y$ depends on the structure in which it is interpreted — it's true if the universe has only one element and false otherwise — but that $\forall y\, y = y$ is a tautology by Problem 6.11. Problems of this kind make it necessary to be careful when substituting for variables.

PROBLEM 7.1. 1. *Is x substitutable for z in ψ if ψ is $z = x \to \forall z\, z = x$? If so, what is ψ^z_x?*
2. *Show that if t is any term and σ is a sentence, then t is substitutable in σ for any variable x. What is σ^x_t?*
3. *Show that if t is a term in which no variable occurs that occurs in the formula φ, then t is substitutable in φ for any variable x.*
4. *Show that x is substitutable for x in φ for any variable x and any formula φ, and that φ^x_x is just φ.*

Along with the notion of substitutability, we need an additional notion in order to define the logical axioms of $\mathcal{L}$.

DEFINITION 7.2. If φ is any formula and $x_1, \dots, x_n$ are any variables, then $\forall x_1 \dots \forall x_n\, \varphi$ is said to be a *generalization* of φ.

For example, $\forall y\, \forall x\, (x = y \to fx = fy)$ and $\forall z\, (x = y \to fx = fy)$ are (different) generalizations of $x = y \to fx = fy$, but $\forall x\, \exists y\, (x = y \to fx = fy)$ is not. Note that the variables being quantified don't have to occur in the formula being generalized.

LEMMA 7.2. *Any generalization of a tautology is a tautology.*

DEFINITION 7.3. Every first-order language $\mathcal{L}$ has eight *logical axiom schema*:

A1: $(\alpha \to (\beta \to \alpha))$
A2: $((\alpha \to (\beta \to \gamma)) \to ((\alpha \to \beta) \to (\alpha \to \gamma)))$
A3: $(((\neg\beta) \to (\neg\alpha)) \to (((\neg\beta) \to \alpha) \to \beta))$
A4: $(\forall x\, \alpha \to \alpha^x_t)$, if t is substitutable for x in α.
A5: $(\forall x\, (\alpha \to \beta) \to (\forall x\, \alpha \to \forall x\, \beta))$
A6: $(\alpha \to \forall x\, \alpha)$, if x does not occur free in α.
A7: $x = x$
A8: $(x = y \to (\alpha \to \beta))$, if α is atomic and β is obtained from α by replacing some occurrences (possibly all or none) of x in α by y.

Plugging in any particular formulas of $\mathcal{L}$ for α, β, and γ, and any particular variables for x and y, in any of A1–A8 gives a *logical axiom* of $\mathcal{L}$. In addition, any generalization of a logical axiom of $\mathcal{L}$ is also a logical axiom of $\mathcal{L}$.

The reason for calling the instances of A1–A8 the logical axioms, instead of just axioms, is to avoid conflict with Definition 6.8.

PROBLEM 7.3. *Determine whether or not each of the following formulas is a logical axiom.*

1. $\forall x\, \forall z\, (x = y \to (x = c \to x = y))$
2. $x = y \to (y = z \to z = x)$
3. $\forall z\, (x = y \to (x = c \to y = c))$
4. $\forall w\, \exists x\, (Pwx \to Pww) \to \exists x\, (Pxx \to Pxx)$
5. $\forall x\, (\forall x\, c = fxc \to \forall x\, \forall x\, c = fxc)$
6. $(\exists x\, Px \to \exists y\, \forall z\, Qzfy) \to ((\exists x\, Px \to \forall y\, \neg \forall z\, Qzfy) \to \forall x\, \neg Px)$

PROPOSITION 7.4. *Every logical axiom is a tautology.*

Note that we have recycled our axiom schemas A1—A3 from propositional logic. We will also recycle MP as the sole rule of inference for first-order logic.

DEFINITION 7.4 (Modus Ponens). Given the formulas φ and $(\varphi \to \psi)$, one may infer ψ.

As in propositional logic, we will usually refer to Modus Ponens by its initials, MP. That MP preserves truth in the sense of Chapter 6 follows from Problem 6.10. Using the logical axioms and MP, we can execute deductions in first-order logic just as we did in propositional logic.

DEFINITION 7.5. Let Δ be a set of formulas of the first-order language $\mathcal{L}$. A *deduction* or *proof* from Δ in $\mathcal{L}$ is a finite sequence $\varphi_1\varphi_2\ldots\varphi_n$ of formulas of $\mathcal{L}$ such that for each $k \leq n$,

1. φ_k is a logical axiom, or
2. $\varphi_k \in \Delta$, or
3. there are $i, j < k$ such that φ_k follows from φ_i and φ_j by MP.

A formula of Δ appearing in the deduction is usually referred to as a *premiss* of the deduction. Δ *proves* a formula α, written as $\Delta \vdash \alpha$, if α is the last formula of a deduction from Δ. We'll usually write $\vdash \alpha$ instead of $\emptyset \vdash \alpha$. Finally, if Γ and Δ are sets of formulas, we'll take $\Gamma \vdash \Delta$ to mean that $\Gamma \vdash \delta$ for every formula $\delta \in \Delta$.

NOTE. We have reused the axiom schema, rule of inference, and definition of deduction from propositional logic. It follows that any deduction of propositional logic can be converted into a deduction of first-order logic simply by replacing the formulas of $\mathcal{L}_P$ occurring in the deduction by first-order formulas. In fact, feel free to appeal to the deductions you did in Chapter 3.

EXAMPLE 7.1. We'll show that $\{\alpha\} \vdash \exists x\, \alpha$ for any first-order formula α and any variable x.

1.	$(\forall x\, \neg\alpha \to \neg\alpha) \to (\alpha \to \neg\forall x\, \neg\alpha)$	Problem 3.8.5
2.	$\forall x\, \neg\alpha \to \neg\alpha$	A4
3.	$\alpha \to \neg\forall x\, \neg\alpha$	1,2 MP
4.	α	Premiss
5.	$\neg\forall x\, \neg\alpha$	3,4 MP
6.	$\exists x\, \alpha$	Definition of $\exists$

Strictly speaking, the last line is just for our convenience, like $\exists$ itself.

PROBLEM 7.5. *Show that:*

1. $\vdash \forall x\, \varphi \to \forall y\, \varphi^x_y$, *if y does not occur at all in φ.*
2. $\vdash \alpha \vee \neg\alpha$.
3. $\{c = d\} \vdash \forall z\, Qazc \to Qazd$.
4. $\vdash x = y \to y = x$.
5. $\{\exists x\, \alpha\} \vdash \alpha$ *if x does not occur free in α.*

Many general facts about deductions can be recycled from propositional logic, including the Deduction Theorem.

PROPOSITION 7.6. *If $\Gamma \vdash \delta$ and $\Gamma \vdash \delta \to \beta$, then $\Gamma \vdash \beta$.*

PROPOSITION 7.7. *If $\Gamma \subseteq \Delta$ and $\Gamma \vdash \alpha$, then $\Delta \vdash \alpha$.*

PROPOSITION 7.8. *Then if $\Gamma \vdash \Delta$ and $\Delta \vdash \sigma$, then $\Gamma \vdash \sigma$.*

THEOREM 7.9 (Deduction Theorem). *If Σ is any set of formulas and α and β are any formulas, then $\Sigma \vdash \alpha \to \beta$ if and only if $\Sigma \cup \{\alpha\} \vdash \beta$.*

Just as in propositional logic, the Deduction Theorem is useful because it often lets us take shortcuts when trying to show that deductions exist. There is also another result about first-order deductions which often supplies useful shortcuts.

THEOREM 7.10 (Generalization Theorem). *Suppose x is a variable, Γ is a set of formulas in which x does not occur free, and φ is a formula such that $\Gamma \vdash \varphi$. Then $\Gamma \vdash \forall x\, \varphi$.*

THEOREM 7.11 (Generalization On Constants). *Suppose that c is a constant symbol, Γ is a set of formulas in which c does not occur, and φ is a formula such that $\Gamma \vdash \varphi$. Then there is a variable x which does not occur in φ such that $\Gamma \vdash \forall x\, \varphi^c_x$.*[1] *Moreover, there is a deduction of $\forall x\, \varphi^c_x$ from Γ in which c does not occur.*

[1] φ^c_x is φ with every occurence of the constant c replaced by x.

EXAMPLE 7.2. We'll show that if φ and ψ are any formulas, x is any variable, and $\vdash \varphi \to \psi$, then $\vdash \forall x\, \varphi \to \forall x\, \psi$.

Since x does not occur free in any formula of $\emptyset$, it follows from $\vdash \varphi \to \psi$ by the Generalization Theorem that $\vdash \forall x\, (\varphi \to \psi)$. But then

1. $\forall x\, (\varphi \to \psi)$	above
2. $\forall x\, (\varphi \to \psi) \to (\forall x\, \varphi \to \forall x\, \psi)$	A5
3. $\forall x\, \varphi \to \forall x\, \psi$	1,2 MP

is the tail end of a deduction of $\forall x\, \varphi \to \forall x\, \psi$ from $\emptyset$.

PROBLEM 7.12. *Show that:*

1. $\vdash \forall x\, \forall y\, \forall z\, (x = y \to (y = z \to x = z))$.
2. $\vdash \forall x\, \alpha \to \exists x\, \alpha$.
3. $\vdash \exists y\, (Py \to \forall z\, Pz)$.
4. $\vdash \exists x\, \gamma \to \forall x\, \gamma$ *if* x *does not occur free in* γ.
5. $\{\forall x\, \varphi\} \vdash \forall y\, \varphi^x_y$ *if* y *is substitutable for* x *in* φ.

We conclude with a bit of terminology.

DEFINITION 7.6. If Σ is a set of sentences, then the *theory* of Σ is

$$\mathrm{Th}(\Sigma) = \{\, \tau \mid \tau \text{ is a sentence and } \Sigma \vdash \tau \,\}.$$

That is, the theory of Σ is just the collection of all sentences which can be proved from Σ.

CHAPTER 8

Soundness and Completeness

As with propositional logic, first-order logic had better satisfy the Soundness Theorem and it is desirable that it satisfy the Completeness Theorem. These theorems do hold for first-order logic. The Soundness Theorem is proved in a way similar to its counterpart for propositional logic, but the Completeness Theorem will require a fair bit of additional work.[1] It is in this extra work that the distinction between formulas and sentences becomes useful.

Let $\mathcal{L}$ be a fixed countable first-order language throughout this chapter. All formulas will be assumed to be formulas of $\mathcal{L}$ unless stated otherwise.

First, we rehash many of the definitions and facts we proved for propositional logic in Chapter 4 for first-order logic.

THEOREM 8.1 (Soundness Theorem). *If α is a sentence and Δ is a set of sentences such that $\Delta \vdash \alpha$, then $\Delta \models \alpha$.*

DEFINITION 8.1. A set of sentences Γ is *inconsistent* if $\Gamma \vdash \neg(\psi \to \psi)$ for some formula ψ, and is *consistent* if it is not inconsistent.

Recall that a set of sentences Γ is satisfiable if $\mathfrak{M} \models \Gamma$ for some structure $\mathfrak{M}$.

PROPOSITION 8.2. *If a set of sentences Γ is satisfiable, then it is consistent.*

PROPOSITION 8.3. *Suppose Δ is an inconsistent set of sentences. Then $\Delta \vdash \psi$ for any formula ψ.*

PROPOSITION 8.4. *Suppose Σ is an inconsistent set of sentences. Then there is a finite subset Δ of Σ such that Δ is inconsistent.*

COROLLARY 8.5. *A set of sentences Γ is consistent if and only if every finite subset of Γ is consistent.*

[1]This is not too surprising because of the greater complexity of first-order logic. Also, it turns out that first-order logic is about as powerful as a logic can get and still have the Completeness Theorem hold.

DEFINITION 8.2. A set of sentences Σ is *maximally consistent* if Σ is consistent but $\Sigma \cup \{\tau\}$ is inconsistent whenever τ is a sentence such that $\tau \notin \Sigma$.

PROBLEM 8.6. *Suppose $\mathfrak{M}$ is a structure. Show that $Th(\mathfrak{M})$ is a maximally consistent set of sentences.*

PROBLEM 8.7. *Give an explicit example of a set of sentences Σ in a first-order language such that $Th(\Sigma)$ is maximally consistent.*

PROPOSITION 8.8. *If Σ is a maximally consistent set of sentences, τ is a sentence, and $\Sigma \vdash \tau$, then $\tau \in \Sigma$.*

PROPOSITION 8.9. *Suppose Σ is a maximally consistent set of sentences and τ is a sentence. Then $\neg\tau \in \Sigma$ if and only if $\tau \notin \Sigma$.*

PROPOSITION 8.10. *Suppose Σ is a maximally consistent set of sentences and φ and ψ are any sentences. Then $\varphi \to \psi \in \Sigma$ if and only if $\varphi \notin \Sigma$ or $\psi \in \Sigma$.*

THEOREM 8.11. *Suppose Γ is a consistent set of sentences. Then there is a maximally consistent set of sentences Σ with $\Gamma \subseteq \Sigma$.*

The counterparts of these notions and facts for propositional logic sufficed to prove the Completeness Theorem, but here we will need some additional tools. The basic problem is that instead of defining a suitable truth assignment from a maximally consistent set of formulas, we need to construct a suitable structure from a maximally consistent set of sentences. Unfortunately, structures for first-order languages are usually more complex than truth assignments for propositional logic. The following definition supplies the key new idea we will use to prove the Completeness Theorem.

DEFINITION 8.3. Suppose Σ is a set of sentences and C is a set of (some of the) constant symbols of $\mathcal{L}$. Then C is a *set of witnesses* for Σ in $\mathcal{L}$ if for every formula φ of $\mathcal{L}$ with at most one free variable x, there is a constant symbol $c \in C$ such that $\Sigma \vdash \exists x\, \varphi \to \varphi^x_c$.

The idea is that every element of the universe which Σ proves must exist is named, or "witnessed", by a constant symbol in C. Note that if $\Sigma \vdash \neg\exists x\, \varphi$, then $\Sigma \vdash \exists x\, \varphi \to \varphi^x_c$ for any constant symbol c.

PROPOSITION 8.12. *Suppose Γ and Σ are sets of sentences of $\mathcal{L}$, $\Gamma \subseteq \Sigma$, and C is a set of witnesses for Γ in $\mathcal{L}$. Then C is a set of witnesses for Σ in $\mathcal{L}$.*

EXAMPLE 8.1. Let $\mathcal{L}'_O$ be the first-order language with a single 2-place relation symbol, $<$, and countably many constant symbols, c_q for each $q \in \mathbb{Q}$. Let Σ include all the sentences

1. $c_p < c_q$, for every $p, q \in \mathbb{Q}$ such that $p < q$,
2. $\forall x\,(\neg x < x)$,
3. $\forall x\,\forall y\,(x < y \lor x = y \lor y < x)$,
4. $\forall x\,\forall y\,\forall z\,(x < y \to (y < z \to x < z))$,
5. $\forall x\,\forall y\,(x < y \to \exists z\,(x < z \land z < y))$,
6. $\forall x\,\exists y\,(x < y)$, and
7. $\forall x\,\exists y\,(y < x)$.

In effect, Σ asserts that $<$ is a linear order on the universe (2–4) which is dense (4) and has no endpoints (5–6), and which has an isomorphic copy of $\mathbb{Q}$ as a sub-order (1). Then $C = \{\, c_q \mid q \in \mathbb{Q}\,\}$ is a set of witnesses for Σ in $\mathcal{L}'_O$.

In the example above, one can "reverse-engineer" a model for the set of sentences in question from the set of witnesses simply by letting the universe of the structure *be* the set of witnesses. One can also define the necessary relation interpreting $<$ in a pretty obvious way from Σ.[2] This example is obviously contrived: there are no constant symbols around which are not witnesses, Σ proves that distinct constant symbols aren't equal to to each other, there is little by way of non-logical symbols needing interpretation, and Σ explicitly includes everything we need to know about $<$.

In general, trying to build a model for a set of sentences Σ in this way runs into a number of problems. First, how do we know whether Σ has a set of witnesses at all? Many first-order languages have few or no constant symbols, after all. Second, if Σ has a set of witnesses C, it's unlikely that we'll be able to get away with just letting the universe of the model be C. What if $\Sigma \vdash c = d$ for some distinct witnesses c and d? Third, how do we handle interpreting constant symbols which are not in C? Fourth, what if Σ doesn't prove enough about whatever relation and function symbols exist to let us define interpretations of them in the structure under construction? (Imagine, if you like, that someone hands you a copy of Joyce's *Ulysses* and asks you to produce a complete road map of Dublin on the basis of the book. Even if it has no geographic contradictions, you are unlikely to find all the information in the novel needed to do the job.) Finally, even if Σ does prove all we need to define functions and relations on the universe to interpret the function and relation symbols, just how do we do it? Getting around all these difficulties requires a fair bit of work. One can get around

[2]Note, however, that an isomorphic copy of $\mathbb{Q}$ is not the only structure for $\mathcal{L}'_O$ satisfying Σ. For example, $\mathfrak{R} = (\mathbb{R}, <, q+\pi : q \in \mathbb{Q})$ will also satisfy Σ if we intepret c_q by $q + \pi$.

many by sticking to maximally consistent sets of sentences in suitable languages.

LEMMA 8.13. *Suppose* Σ *is a set of sentences,* φ *is any formula, and* x *is any variable. Then* $\Sigma \vdash \varphi$ *if and only if* $\Sigma \vdash \forall x\, \varphi$.

THEOREM 8.14. *Suppose* Γ *is a consistent set of sentences of* $\mathcal{L}$. *Let* C *be an infinite countable set of constant symbols which are* not *symbols of* $\mathcal{L}$, *and let* $\mathcal{L}' = \mathcal{L} \cup C$ *be the language obtained by adding the constant symbols in* C *to the symbols of* $\mathcal{L}$. *Then there is a maximally consistent set* Σ *of sentences of* $\mathcal{L}'$ *such that* $\Gamma \subseteq \Sigma$ *and* C *is a set of witnesses for* Σ.

This theorem allows one to use a certain measure of brute force: No set of witnesses? Just add one! The set of sentences doesn't decide enough? Decide *everything* one way or the other!

THEOREM 8.15. *Suppose* Σ *is a maximally consistent set of sentences and* C *is a set of witnesses for* Σ. *Then there is a structure* $\mathfrak{M}$ *such that* $\mathfrak{M} \models \Sigma$.

The important part here is to define $\mathfrak{M}$ — proving that $\mathfrak{M} \models \Sigma$ is tedious but fairly straightforward if you have the right definition. Proposition 6.15 now lets us deduce the fact we really need.

COROLLARY 8.16. *Suppose* Γ *is a consistent set of sentences of a first-order language* $\mathcal{L}$. *Then there is a structure* $\mathfrak{M}$ *for* $\mathcal{L}$ *satisfying* Γ.

With the above facts in hand, we can rejoin our proof of Soundness and Completeness, already in progress:

THEOREM 8.17. *A set of sentences* Σ *in* $\mathcal{L}$ *is consistent if and only if it is satisfiable.*

The rest works just like it did for propositional logic.

THEOREM 8.18 (Completeness Theorem). *If* α *is a sentence and* Δ *is a set of sentences such that* $\Delta \models \alpha$, *then* $\Delta \vdash \alpha$.

It follows that in a first-order logic, as in propositional logic, a sentence is implied by some set of premisses if and only if it has a proof from those premisses.

THEOREM 8.19 (Compactness Theorem). *A set of sentences* Δ *is satisfiable if and only if every finite subset of* Δ *is satisfiable.*

CHAPTER 9

Applications of Compactness

After wading through the preceding chapters, it should be obvious that first-order logic is, in principle, adequate for the job it was originally developed for: the essentially philosophical exercise of formalizing most of mathematics. As something of a bonus, first-order logic can supply useful tools for doing "real" mathematics. The Compactness Theorem is the simplest of these tools and glimpses of two ways of using it are provided below.

From the finite to the infinite. Perhaps the simplest use of the Compactness Theorem is to show that if there exist arbitrarily large finite objects of some type, then there must also be an infinite object of this type.

Example 9.1. We will use the Compactness Theorem to show that there is an infinite commutative group in which every element is of order 2, *i.e.* such that $g \cdot g = e$ for every element g.

Let $\mathcal{L}_G$ be the first-order language with just two non-logical symbols:

- Constant symbol: e
- 2-place function symbol: $\cdot$

Here e is intended to name the group's identity element and $\cdot$ the group operation. Let Σ be the set of sentences of $\mathcal{L}_G$ including:

1. The axioms for a commutative group:
   - $\forall x\, x \cdot e = x$
   - $\forall x\, \exists y\, x \cdot y = e$
   - $\forall x\, \forall y \forall z\, x \cdot (y \cdot z) = (x \cdot y) \cdot z$
   - $\forall x\, \forall y\, y \cdot x = x \cdot y$
2. A sentence which asserts that every element of the universe is of order 2:
   - $\forall x\, x \cdot x = e$
3. For each $n \geq 2$, a sentence, σ_n, which asserts that there are at least n different elements in the universe:
   - $\exists x_1 \ldots \exists x_n\, ((\neg x_1 = x_2) \wedge (\neg x_1 = x_3) \wedge \cdots \wedge (\neg x_{n-1} = x_n))$

We claim that every finite subset of Σ is satisfiable. The most direct way to verify this is to show how, given a finite subset Δ of Σ, to produce a model $\mathfrak{M}$ of Δ. Let n be the largest integer such that $\sigma_n \in \Delta \cup \{\sigma_2\}$ (Why is there such an n?) and choose an integer k such that $2^k \geq n$. Define a structure $(G, \circ)$ for $\mathcal{L}_G$ as follows:

- $G = \{\, \langle a_\ell \mid 1 \leq \ell \leq k \rangle \mid a_\ell = 0 \text{ or } 1 \,\}$
- $\langle a_\ell \mid 1 \leq \ell \leq k \rangle \circ \langle b_\ell \mid 1 \leq \ell \leq k \rangle = \langle a_\ell + b_\ell \pmod 2 \mid 1 \leq \ell \leq k \rangle$

That is, G is the set of binary sequences of length k and $\circ$ is coordinatewise addition modulo 2 of these sequences. It is easy to check that $(G, \circ)$ is a commutative group with 2^k elements in which every element has order 2. Hence $(G, \circ) \models \Delta$, so Δ is satisfiable.

Since every finite subset of Σ is satisfiable, it follows by the Compactness Theorem that Σ is satisfiable. A model of Σ, however, must be an infinite commutative group in which every element is of order 2. (To be sure, it is quite easy to build such a group directly; for example, by using coordinatewise addition modulo 2 of infinite binary sequences.)

Problem 9.1. *Use the Compactness Theorem to show that there is an infinite*

1. *bipartite graph,*
2. *non-commutative group, and*
3. *field of characteristic 3,*

and also give concrete examples of such objects.

Most applications of this method, including the ones above, are not really interesting: it is usually more valuable, and often easier, to directly construct examples of the infinite objects in question rather than just show such must exist. Sometimes, though, the technique can be used to obtain a non-trivial result more easily than by direct methods. We'll use it to prove an important result from graph theory, Ramsey's Theorem. Some definitions first:

Definition 9.1. If X is a set, let the set of unordered pairs of elements of X be $[X]^2 = \{\, \{a, b\} \mid a, b \in X \text{ and } a \neq b \,\}$. (See Definition A.1.)

1. A *graph* is a pair (V, E) such that V is a non-empty set and $E \subseteq [V]^2$. Elements of V are called *vertices* of the graph and elements of E are called *edges*.
2. A *subgraph* of (V, E) is a pair (U, F), where $U \subset V$ and $F = E \cap [U]^2$.
3. A subgraph (U, F) of (V, E) is a *clique* if $F = [U]^2$.
4. A subgraph (U, F) of (V, E) is an *independent set* if $F = \emptyset$.

That is, a graph is some collection of vertices, some of which are joined to one another. A subgraph is just a subset of the vertices, together with all edges joining vertices of this subset in the whole graph. It is a clique if it happens that the original graph joined every vertex in the subgraph to all other vertices in the subgraph, and an independent set if it happens that the original graph joined none of the vertices in the subgraph to each other. The question of when a graph must have a clique or independent set of a given size is of some interest in many applications, especially in dealing with colouring problems.

THEOREM 9.2 (Ramsey's Theorem). *For every $n \geq 1$ there is an integer R_n such that any graph with at least R_n vertices has a clique with n vertices or an independent set with n vertices.*

R_n is the *nth Ramsey number*. It is easy to see that $R_1 = 1$ and $R_2 = 2$, but R_3 is already 6, and R_n grows very quickly as a function of n thereafter. Ramsey's Theorem is fairly hard to prove directly, but the corresponding result for infinite graphs is comparatively straightforward.

LEMMA 9.3. *If (V, E) is a graph with infinitely many vertices, then it has an infinite clique or an infinite independent set.*

A relatively quick way to prove Ramsey's Theorem is to first prove its infinite counterpart, Lemma 9.3, and then get Ramsey's Theorem out of it by way of the Compactness Theorem. (If you're an ambitious minimalist, you can try to do this using the Compactness Theorem for propositional logic instead!)

Elementary equivalence and non-standard models. One of the common uses for the Compactness Theorem is to construct "non-standard" models of the theories satisfied by various standard mathematical structures. Such a model satisfies all the same first-order sentences as the standard model, but differs from it in some way not expressible in the first-order language in question. This brings home one of the intrinsic limitations of first-order logic: it can't always tell essentially different structures apart. Of course, we need to define just what constitutes essential difference.

DEFINITION 9.2. Suppose $\mathcal{L}$ is a first-order language and $\mathfrak{N}$ and $\mathfrak{M}$ are two structures for $\mathcal{L}$. Then $\mathfrak{N}$ and $\mathfrak{M}$ are:

1. *isomorphic*, written as $\mathfrak{N} \cong \mathfrak{M}$, if there is a function $F\colon |\mathfrak{N}| \to |\mathfrak{M}|$ such that
   (a) F is $1-1$ and onto,
   (b) $F(c^{\mathfrak{N}}) = c^{\mathfrak{M}}$ for every constant symbol c of $\mathcal{L}$,

(c) $F(f^{\mathfrak{N}}(a_1, \dots, a_k) = f^{\mathfrak{M}}(F(a_1), \dots, F(a_k))$ for every k-place function symbol f of $\mathcal{L}$ and elements $a_1, \dots, a_k \in |\mathfrak{N}|$, and

(d) $P^{\mathfrak{N}}(a_1, \dots, a_k)$ holds if and only if $P^{\mathfrak{N}}(F(a_1), \dots, F(a_k))$ for every k-place relation symbol of $\mathcal{L}$ and elements a_1, ... , a_k of $|\mathfrak{N}|$;

and

2. *elementarily equivalent*, written as $\mathfrak{N} \equiv \mathfrak{M}$, if $\mathrm{Th}(\mathfrak{N}) = \mathrm{Th}(\mathfrak{M})$, *i.e.* if $\mathfrak{N} \models \sigma$ if and only if $\mathfrak{M} \models \sigma$ for every sentence σ of $\mathcal{L}$.

That is, two structures for a given language are isomorphic if they are structurally identical and elementarily equivalent if no statement in the language can distinguish between them. Isomorphic structures are elementarily equivalent:

PROPOSITION 9.4. *Suppose $\mathcal{L}$ is a first-order language and $\mathfrak{N}$ and $\mathfrak{M}$ are structures for $\mathcal{L}$ such that $\mathfrak{N} \cong \mathfrak{M}$. Then $\mathfrak{N} \equiv \mathfrak{M}$.*

However, as the following application of the Compactness Theorem shows, elementarily equivalent structures need not be isomorphic:

EXAMPLE 9.2. Note that $\mathfrak{C} = (\mathbb{N})$ is an infinite structure for $\mathcal{L}_=$. Expand $\mathcal{L}_=$ to $\mathcal{L}_R$ by adding a constant symbol c_r for every real number r, and let Σ be the set of sentences of $\mathcal{L}_=$ including

- every sentence τ of $\mathrm{Th}(\mathfrak{C})$, *i.e.* such that $\mathfrak{C} \models \tau$, and
- $\neg c_r = c_s$ for every pair of real numbers r and s such that $r \neq s$.

Every finite subset of Σ is satisfiable. (Why?) Thus, by the Compactness Theorem, there is a structure $\mathfrak{U}'$ for $\mathcal{L}_R$ satisfying Σ, and hence $\mathrm{Th}(\mathfrak{C})$. The structure $\mathfrak{U}$ obtained by dropping the interpretations of all the constant symbols c_r from $\mathfrak{U}'$ is then a structure for $\mathcal{L}_=$ which satisfies $\mathrm{Th}(\mathfrak{C})$. Note that $|\mathfrak{U}| = |\mathfrak{U}'|$ is at least large as the set of all real numbers $\mathbb{R}$, since $\mathfrak{U}'$ requires a distinct element of the universe to interpret each constant symbol c_r of $\mathcal{L}_R$.

Since $\mathrm{Th}(\mathfrak{C})$ is a maximally consistent set of sentences of $\mathcal{L}_=$ by Problem 8.6, it follows from the above that $\mathfrak{C} \equiv \mathfrak{U}$. On the other hand, $\mathfrak{C}$ cannot be isomorphic to $\mathfrak{U}$ because there cannot be an onto map between a countable set, such as $\mathbb{N} = |\mathfrak{C}|$, and a set which is at least as large as $\mathbb{R}$, such as $|\mathfrak{U}|$.

In general, the method used above can be used to show that if a set of sentences in a first-order language has an infinite model, it has many different ones. In $\mathcal{L}_=$ that is essentially all that can happen:

PROPOSITION 9.5. *Two structures for $\mathcal{L}_=$ are elementarily equivalent if and only if they are isomorphic or infinite.*

PROBLEM 9.6. *Let* $\mathfrak{N} = (\mathbb{N}, 0, 1, S, +, \cdot, E)$ *be the standard structure for* $\mathcal{L}_N$. *Use the Compactness Theorem to show there is a structure* $\mathfrak{M}$ *for* $\mathcal{L}_N$ *such that* $\mathfrak{N} \equiv \mathfrak{N}$ *but not* $\mathfrak{N} \cong \mathfrak{M}$.

Note that because $\mathfrak{N}$ and $\mathfrak{M}$ both satisfy $\mathrm{Th}(\mathfrak{N})$, which is maximally consistent by Problem 8.6, there is absolutely no way of telling them apart in $\mathcal{L}_N$.

PROPOSITION 9.7. *Every model of* $\mathrm{Th}(\mathfrak{N})$ *which is* not *isomorphic to* $\mathfrak{N}$ *has*

1. *an isomorphic copy of* $\mathfrak{N}$ *embedded in it,*
2. *an infinite number,* i.e. *one larger than all of those in the copy of* $\mathfrak{N}$*, and*
3. *an infinite decreasing sequence.*

The apparent limitation of first-order logic that non-isomorphic structures may be elementarily equivalent can actually be useful. A non-standard model may have features that make it easier to work with than the standard model one is really interested in. Since both structures satisfy exactly the same sentences, if one uses these features to prove that some statement expressible in the given first-order language is true about the non-standard structure, one gets for free that it must be true of the standard structure as well. A prime example of this idea is the use of non-standard models of the real numbers containing infinitesimals (numbers which are infinitely small but different from zero) in some areas of analysis.

THEOREM 9.8. *Let* $\mathfrak{R} = (\mathbb{R}, 0, 1, +, \cdot)$ *be the field of real numbers, considered as a structure for* $\mathcal{L}_F$. *Then there is a model of Th*$(\mathfrak{R})$ *which contains a copy of* $\mathbb{R}$ *and in which there is an infinitesimal.*

The non-standard models of the real numbers actually used in analysis are usually obtained in more sophisticated ways in order to have more information about their internal structure. It is interesting to note that infinitesimals were the intuition behind calculus for Leibniz when it was first invented, but no one was able to put their use on a rigourous footing until Abraham Robinson did so in 1950.

Hints

CHAPTER 1

Hints

1.1. Symbols not in the language, unbalanced parentheses, lack of connectives ...

1.2. Proceed by induction on the length of the formula or on the number of connectives in the formula.

1.3. Compute $p(\alpha)/\ell(\alpha)$ for a number of examples and look for patterns. Getting a minimum value should be pretty easy.

1.4. Proceed by induction on the length of or on the number of connectives in the formula.

1.5. Construct examples of formulas of all the short lengths that you can, and then see how you can make longer formulas out of short ones.

1.6. Hewlett-Packard sells calculators that use such a trick. A similar one is used in Definition 5.2.

1.7. Observe that $\mathcal{L}_P$ has countably many symbols and that every formula is a finite sequence of symbols. The relevant facts from set theory are given in Appendix A.

1.8. Stick several simple statements together with suitable connectives.

1.9. This should be straightforward.

1.10. Ditto.

1.11. To make sure you get all the subformulas, write out the formula in official form with all the parentheses.

1.12. Proceed by induction on the length or number of connectives of the formula.

CHAPTER 2

Hints

2.1. Use truth tables.

2.2. Proceed by induction on the length of δ or on the number of connectives in δ.

2.3. Use Proposition 2.2.

2.4. In each case, unwind Definition 2.1 and the definitions of the abbreviations.

2.5. Use truth tables.

2.6. Use Definition 2.3 and Proposition 2.4.

2.7. If a truth assignment satisfies every formula in Σ and every formula in Γ is also in Σ, then ...

2.8. Grinding out an appropriate truth table will do the job. Why is it important that Σ be finite here?

2.9. Use Definition 2.4 and Proposition 2.4.

2.10. Use Definitions 2.3 and 2.4. If you have trouble trying to prove one of the two directions directly, try proving its contrapositive instead.

CHAPTER 3

Hints

3.1. Truth tables are probably the best way to do this.

3.2. Look up Proposition 2.4.

3.3. There are usually many different deductions with a given conclusion, so you shouldn't take the following hints as gospel.

1. Use A2 and A1.
2. Recall what $\vee$ abbreviates.

3.4. Put together a deduction of β from Γ from the deductions of δ and $\delta \to \beta$ from Γ.

3.5. Examine Definition 3.3 carefully.

3.6. The key idea is similar to that for proving Proposition 3.4.

3.7. One direction follows from Proposition 3.4. For the other direction, proceed by induction on the length of the shortest proof of β from $\Sigma \cup \{\alpha\}$.

3.8. Again, don't take these hints as gospel. Try using the Deduction Theorem in each case, plus

1. A3.
2. A3 and Problem 3.3.
3. A3.
4. A3, Problem 3.3, and Example 3.2.
5. Some of the above parts and Problem 3.3.
6. Ditto.
7. Use the definition of $\vee$ and one of the above parts.
8. Use the definition of $\wedge$ and one of the above parts.
9. Aim for $\neg\alpha \to (\alpha \to \neg\beta)$ as an intermediate step.

CHAPTER 4

Hints

4.1. Use induction on the length of the deduction and Proposition 3.2.

4.2. Assume, by way of contradiction, that the given set of formulas is inconsistent. Proceed by induction on the length of the relevant deduction to get the necessary contradiction.

4.3. First show that $\{\neg(\alpha \to \alpha)\} \vdash \psi$.

4.4. Note that deductions are finite sequences of formulas.

4.5. Use Proposition 4.4.

4.6. Use Proposition 4.2, the definition of Σ, and Proposition 2.4.

4.7. Assume, by way of contradiction, that $\varphi \notin \Sigma$. Use Definition 4.2 and the Deduction Theorem to show that Σ must be inconsistent.

4.8. Use Definition 4.2 and Problem 3.8.

4.9. Use Definition 4.2 and Proposition 4.8.

4.10. Use Proposition 1.7 and induction on a list of all the formulas of $\mathcal{L}_P$.

4.11. One direction is just Proposition 4.2. For the other, expand the set of formulas in question to a maximally consistent set of formulas Σ using Theorem 4.10, and define a truth assignment v by setting $v(A_n) = T$ if and only if $A_n \in \Sigma$. Now use induction on the length of φ to show that $\varphi \in \Sigma$ if and only if v satisfies φ.

4.12. Prove the contrapositive using Theorem 4.11.

4.13. Put Corollary 4.5 together with Theorem 4.11.

CHAPTER 5

Hints

5.1. Try to disassemble each string using Definition 5.2. Note that some might be valid terms of more than one of the given languages.

5.2. This is similar to Problem 1.5.

5.3. This is similar to Proposition 1.7.

5.4. Try to disassemble each string using Definitions 5.2 and 5.3. Note that some might be valid formulas of more than one of the given languages.

5.5. This is just like Problem 1.2.

5.6. This is similar to Problem 1.5. You may wish to use your solution to Problem 5.2.

5.7. This is similar to Proposition 1.7.

5.8. You might want to rephrase some of the given statements to make them easier to formalize.

1. Look up associativity if you need to.
2. "There is an object such that every object is not in it."
3. This should be easy.
4. Ditto.
5. "Any two things must be the same thing."

5.9. If necessary, don't hesitate to look up the definitions of the given structures.

1. Read the discussion at the beginning of the chapter.
2. You really need only one non-logical symbol.
3. There are two sorts of objects in a vector space, the vectors themselves and the scalars of the field, which you need to be able to tell apart.

5.10. Use Definition 5.3 in the same way that Definition 1.2 was used in Definition 1.3.

5.11. The scope of a quantifier ought to be a certain subformula of the formula in which the quantifier occurs.

5.12. Check to see whether they satisfy Definition 5.4.

5.13. Check to see which pairs satisfy Definition 5.5.

5.14. Proceed by induction on the length of φ using Definition 5.3.

5.15. This is similar to Theorem 1.12.

5.16. This is similar to Theorem 1.12 and uses Theorem 5.15.

CHAPTER 6

Hints

6.1. In each case, apply Definition 6.1.

1. This should be easy.
2. Ditto.
3. Invent objects which are completely different except that they happen to have the right number of the right kind of components.

6.2. Figure out the relevant values of $s(v_n)$ and apply Definition 6.3.

6.3. Suppose $\mathbf{s}$ and $\mathbf{r}$ both extend the assignment s. Show that $\mathbf{s}(t) = \mathbf{r}(t)$ by induction on the length of the term t.

6.4. Unwind the formulas using Definition 6.4 to get informal statements whose truth you can determine.

6.5. Unwind the abbreviation $\exists$ and use Definition 6.4.

6.6. Unwind each of the formulas using Definitions 6.4 and 6.5 to get informal statements whose truth you can determine.

6.7. This is much like Proposition 6.3.

6.8. Proceed by induction on the length of the formula using Definition 6.4 and Lemma 6.7.

6.9. How many free variables does a sentence have?

6.10. Use Definition 6.4.

6.11. Unwind the sentences in question using Definition 6.4.

6.12. Use Definitions 6.4 and 6.5; the proof is similar in form to the proof of Proposition 2.9.

6.13. Use Definitions 6.4 and 6.5; the proof is similar in form to the proof for Problem 2.10.

6.14. Use Definitions 6.4 and 6.5 in each case, plus the meanings of our abbreviations.

6.15. In one direction, you need to add appropriate objects to a structure; in the other, delete them. In both cases, you still have to verify that Γ is still satisfied.

6.16. Here are some appropriate languages.

1. $\mathcal{L}_=$
2. Modify your language for graph theory from Problem 5.9 by adding a 1-place relation symbol.
3. Use your language for group theory from Problem 5.9.
4. $\mathcal{L}_F$

CHAPTER 7

Hints

7.1. 1. Use Definition 7.1.
2. Ditto.
3. Ditto.
4. Proceed by induction on the length of the formula φ.

7.2. Use the definitions and facts about $\models$ from Chapter 6.

7.3. Check each case against the schema in Definition 7.3. Don't forget that any generalization of a logical axiom is also a logical axiom.

7.4. You need to show that any instance of the schemas A1–A8 is a tautology and then apply Lemma 7.2. That each instance of schemas A1–A3 is a tautology follows from Proposition 6.14. For A4–A8 you'll have to use the definitions and facts about $\models$ from Chapter 6.

7.5. You may wish to appeal to the deductions that you made or were given in Chapter 3.
1. Try using A4 and A6.
2. You don't need A4–A8 here.
3. Try using A4 and A8.
4. A8 is the key; you may need it more than once.
5. This is just A6 in disguise.

7.6. This is just like its counterpart for propositional logic.

7.7. Ditto.

7.8. Ditto.

7.9. Ditto.

7.10. Proceed by induction on the length of the shortest proof of φ from Γ.

7.11. Ditto.

7.12. As usual, don't take the following suggestions as gospel.
1. Try using A8.
2. Start with Example 7.1.
3. Start with an instance of A4.

4. Start with part of Problem 7.5.
5. It's not as easy as the similar-seeming part of Problem 7.5 …

CHAPTER 8

Hints

8.1. This is similar to the proof of the Soundness Theorem for propositional logic, using Proposition 6.10 in place of Proposition 3.2.

8.2. This is similar to its counterpart for prpositional logic, Proposition 4.2. Use Proposition 6.10 instead of Proposition 3.2.

8.3. This is just like its counterpart for propositional logic.

8.4. Ditto.

8.5. Ditto.

8.6. This is a counterpart to Problem 4.6; use Proposition 8.2 instead of Proposition 4.2 and Proposition 6.14 instead of Proposition 2.4.

8.7. Let Σ consist of just a sentence of $\mathcal{L}_=$ asserting that there is only thing in the universe, and then try to find a way to apply Proposition 8.6.

8.8. This is just like its counterpart for propositional logic.

8.9. Ditto

8.10. Ditto.

8.11. This is much like its counterpart for propositional logic, Theorem 4.10.

8.12. Use Proposition 7.7.

8.13. Use the Generalization Theorem for the hard direction.

8.14. This is essentially a souped-up version of Theorem 8.11. To ensure that C is a set of witnesses of the maximally consistent set of sentences, enumerate all the formulas φ of $\mathcal{L}'$ with one free variable and take care of one at each step in the inductive construction.

8.15. To construct the required structure, $\mathfrak{M}$, proceed as follows. Define an equivalence relation $\sim$ on C by setting $c \sim d$ if and only if $c = d \in \Sigma$, and let $[c] = \{ a \in C \mid a \sim c \}$ be the equivalence class of $c \in C$. The universe of $\mathfrak{M}$ will be $M = \{ [c] \mid c \in C \}$. For each k-place function symbol f define $f^{\mathfrak{M}}$ by setting $f^{\mathfrak{M}}([a_1], \dots, [a_k]) = [b]$ if and only if $fa_1 \dots a_k = b$ is in Σ. Define the interpretations of constant symbols and relation symbols in a similar way. You need to show that all these things are well-defined, and then show that $\mathfrak{M} \models \Sigma$.

8.16. Expand Γ to a maximally consistent set of sentences with a set of witnesses in a suitable extension of $\mathcal{L}$, apply Theorem 8.15, and then cut down the resulting structure to one for $\mathcal{L}$.

8.17. One direction is just Proposition 8.2. For the other, use Corollary 8.16.

8.18. This follows from Theorem 8.17 in the same way that the Completeness Theorem for propositional logic followed from Theorem 4.11.

8.19. This follows from Theorem 8.17 in the same way that the Compactness Theorem for propositional logic followed from Theorem 4.11.

CHAPTER 9

Hints

9.1. In each case, apply the trick used in Example 9.1. For definitions and the concrete examples, consult texts on combinatorics and abstract algebra.

9.2. Suppose Ramsey's Theorem fails for some n. Use the Compactness Theorem to get a contradiction to Lemma 9.3 by showing there must be an infnite graph with no clique or independent set of size n.

9.3. Inductively define a sequence a_0, a_1, ... , of vertices so that for every n, either it is the case that for all $k \geq n$ there is an edge joining a_n to a_k or it is the case that for all $k \geq n$ there is no edge joining a_n to a_k. There will then be a subsequence of the sequence which is an infinite clique or a subsequence which is an infinite independent set.

9.4. The key is to figure out how, given an assignment for one structure, one should define the corresponding assignment in the other structure. After that, proceed by induction using the definition of satisfaction.

9.5. When are two finite structures for $\mathcal{L}_=$ elementarily equivalent?

9.6. In a suitable expanded language, consider $\mathrm{Th}(\mathfrak{N})$ together with the sentences $\exists x\, 0 + x = c$, $\exists x\, S0 + x = c$, $\exists x\, SS0 + x = c$, ...

9.7. Suppose $\mathfrak{M} \models \mathrm{Th}(\mathfrak{N})$ but is not isomorphic to $\mathfrak{N}$.

1. Consider the subset of $|\mathfrak{M}|$ given by $0^{\mathfrak{M}}$, $S^{\mathfrak{M}}(0^{\mathfrak{M}})$, $S^{\mathfrak{M}}(S^{\mathfrak{M}}(0^{\mathfrak{M}}))$, ...
2. If it didn't have one, it would be a copy of $\mathfrak{N}$.
3. Start with a infinite number and work down.

9.8. Expand $\mathcal{L}_F$ by throwing in a constant symbol for every real number, plus an extra one, and take it from there.

Appendices

APPENDIX A

A Little Set Theory

This apppendix is meant to provide an informal summary of the notation, definitions, and facts about sets needed in Chapters 1–9. For a proper introduction to elementary set theory, try [**5**] or [**6**].

Definition A.1. Suppose X and Y are sets. Then

1. $a \in X$ means that a is an *element* of (*i.e.* a thing in) the set X.
2. X is a subset of Y, written as $X \subseteq Y$, if $a \in Y$ for every $a \in X$.
3. The *union* of X and Y is $X \cup Y = \{ a \mid a \in X \text{ or } a \in Y \}$.
4. The *intersection* of X and Y is $X \cap Y = \{ a \mid a \in X \text{ and } a \in Y \}$.
5. The *complement of Y relative to X* is $X \setminus Y = \{ a \mid a \in X \text{ and } a \notin Y \}$.
6. The *cross product* of X and Y is $X \times Y = \{ (a, b) \mid a \in X \text{ and } b \in Y \}$.
7. The *power set* of X is $\mathcal{P}(X) = \{ Z \mid Z \subseteq X \}$.
8. $[X]^k = \{ Z \mid Z \subseteq X \text{ and } |Z| = k \}$ is the set of subsets of X of size k.

If all the sets being dealt with are all subsets of some fixed set Z, we'll usually take the complement of Y, $\bar{Y}$, to mean the complement of Y relative to Z. It will be necessary to be able to take unions, intersections, and cross products of more than two sets.

Definition A.2. Suppose A is a set and $\mathbf{X} = \{ X_a \mid a \in A \}$ is a family of sets indexed by A. Then

1. The union of $\mathbf{X}$ is the set $\bigcup \mathbf{X} = \{ z \mid \exists a \in A \colon z \in X_a \}$.
2. The intersection of $\mathbf{X}$ is the set $\bigcap \mathbf{X} = \{ z \mid \forall a \in A \colon z \in X_a \}$.
3. The cross product of $\mathbf{X}$ is the set of sequences (indexed by A) $\prod \mathbf{X} = \prod_{a \in A} X_a = \{ (z_a \mid a \in A) \mid \forall a \in A \colon z_a \in X_a \}$. We will denote the cross product of a set X with itself taken n times (*i.e.* the set of all sequences of length n of elements of X) by X^n.

Definition A.3. A set X is *finite* if there is some $n \in \mathbb{N}$ such that X has n elements, and is *infinite* otherwise. X is *countable* if it is infinite and there is a 1-1 onto function $f : \mathbb{N} \to X$, and *uncountable* if it is infinite but not countable.

Various infinite sets occur frequently in mathematics, such as $\mathbb{N}$ (the natural numbers), $\mathbb{Q}$ (the rational numbers), and $\mathbb{R}$ (the real numbers). Many of these are uncountable, such as $\mathbb{R}$. The basic facts about countable sets needed to do the problems are the following.

PROPOSITION A.1.
1. *If X is a countable set and $Y \subseteq X$, then Y is either finite or a countable.*
2. *Suppose $\mathbf{X} = \{ X_n \mid n \in \mathbb{N} \}$ is a countable family of sets such that each X_n is either finite or countable. Then $\bigcup \mathbf{X}$ is also finite or countable.*
3. *If X is a non-empty finite or countable set, then X^n is finite or countable for each $n \geq 1$, and the set of all finite sequences of elements of X, $X^{<\omega} = \bigcup_{n \in \mathbb{N}} X^n$ is countable.*

The properly sceptical reader will note that setting up propositional or first-order logic formally requires that we have some set theory in hand, but formalizing set theory itself requires one to have first-order logic.[1]

[1]Which came first, the chicken or the egg? Since, biblically speaking, "In the beginning was the Word", maybe we ought to plump for alphabetical order. Which begs the question: In which alphabet?

APPENDIX B

The Greek Alphabet

A	α		alpha
B	β		beta
Γ	γ		gamma
Δ	δ		delta
E	ϵ	ε	epsilon
Z	ζ		zeta
H	η		eta
Θ	θ	ϑ	theta
I	ι		iota
K	κ		kappa
Λ	λ		lambda
M	μ		mu
N	ν		nu
O	o		omicron
Ξ	ξ		xi
Π	π	ϖ	pi
P	ρ	ϱ	rho
Σ	σ	ς	sigma
T	τ		tau
Υ	υ		upsilon
Φ	ϕ	φ	phi
X	χ		chi
Ψ	ψ		psi
Ω	ω		omega

APPENDIX C

Logic Limericks

Deduction Theorem

A Theorem fine is Deduction,
For it allows work-reduction:
To show "A implies B",
Assume A and prove B;
Quite often a simpler production.

Generalization Theorem

When in premiss the variable's bound,
To get a "for all" without wound,
Generalization.
Not globalization;
Stay away from that management sound!

Soundness Theorem

It's a critical logical creed:
Always check that it's safe to proceed.
To tell us deductions
Are truthful productions,
It's the Soundness of logic we need.

Completeness Theorem

The Completeness of logics is Gödel's.
'Tis advice for looking for mödels:
They're always existent
For statements consistent,
Most helpful for logical labörs.

Bibliography

[1] Jon Barwise (ed.), *Handbook of Mathematical Logic*, North Holland, Amsterdam, 1977, ISBN 0-7204-2285-X.

[2] Merrie Bergman, James Moor, and Jack Nelson, *The Logic Book*, Random House, NY, 1980, ISBN 0-394-32323-8.

[3] C.C. Chang and H.J. Keisler, *Model Theory*, third ed., North Holland, Amsterdam, 1990.

[4] Herbert B. Enderton, *A Mathematical Introduction to Logic*, Academic Press, New York, 1972.

[5] Paul R. Halmos, *Naive Set Theory*, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1974, ISBN 0-387-90092-6.

[6] James M. Henle, *An Outline of Set Theory*, Problem Books in Mathematics, Springer-Verlag, New York, 1986, ISBN 0-387-96368-5.

[7] Douglas R. Hofstadter, *Gödel, Escher, Bach*, Random House, New York, 1979, ISBN 0-394-74502-7.

[8] Jerome Malitz, *Introduction to Mathematical Logic*, Springer-Verlag, New York, 1979, ISBN 0-387-90346-1.

[9] Yu.I. Manin, *A Course in Mathematical Logic*, Graduate Texts in Mathematics 53, Springer-Verlag, New York, 1977, ISBN 0-387-90243-0.

[10] Roger Penrose, *The Emperor's New Mind*, Oxford University Press, Oxford, 1989.

Index