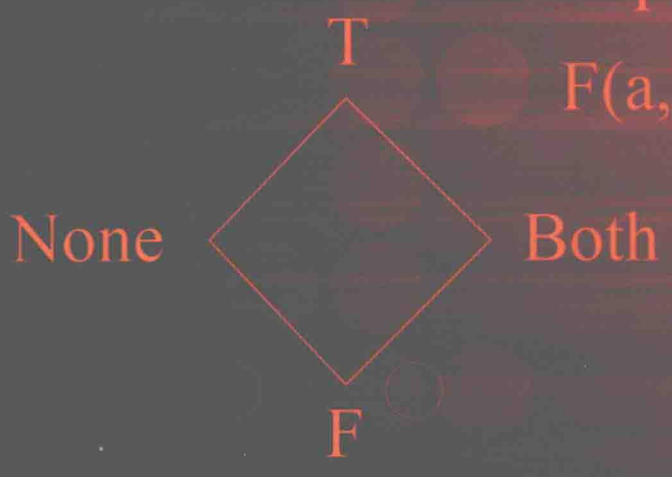
DONALD W. LOVELAND,
RICHARD E. HODEL,
AND S. G. STERRETT

THREE VIEWS OF LOGIC:

Mathematics, Philosophy, and Computer Science

$$F(a,0) = G(a)$$

$$F(a,b+1) = H(a,b,F(a,b))$$



$$\frac{A \vee B, \neg A \vee C}{B \vee C}$$

THREE VIEWS OF LOGIC

Mathematics, Philosophy, and Computer Science

DONALD W. LOVELAND

RICHARD E. HODEL

S. G. STERRETT

PRINCETON UNIVERSITY PRESS

Princeton and Oxford

Published by Princeton University Press, 41 William Street,
Princeton, New Jersey 08540
In the United Kingdom: Princeton University Press,
6 Oxford Street, Woodstock, Oxfordshire OX20 1TW
press.princeton.edu

ISBN 978-0-691-16044-3

Library of Congress Control Number: 2013949122

British Library Cataloging-in-Publication Data is available

This book has been composed in ITC Stone Serif

Printed on acid-free paper ∞

Typeset by S R Nova Pvt Ltd, Bangalore, India
Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

THREE VIEWS OF LOGIC

Preface

This book is based on an interdisciplinary course on logic offered to upper-level undergraduates at Duke University over a period of more than ten years. Why an interdisciplinary course on logic? Although logic has been a discipline of study in philosophy since ancient times, in recent decades it has played an important role in other disciplines as well. For example, logic is at the core of two programming languages, is used in program verification, has enriched philosophy (and computer science) with non-classical logics that can deal constructively with contradictions, and has shaken the foundations of mathematics with insight into non-computable functions and non-provability. Several of these ideas are treated in this book.

We developed a one-semester course suitable for undergraduates that presents some of these more recent, exciting ideas in logic as well as some of the traditional, core ideas. Undergraduate students generally have limited time to pursue logic courses and we found that the course we offered gave them some understanding of both the breadth of logic and the depth of ideas in logic.

This book addresses select topics drawn from three different areas of logic: proof theory, computability theory, and philosophical logic. A common thread throughout is the application of logic to computers and computation.

Part 1 on Proof Theory introduces a deductive system (resolution logic) that comes from an area of research known as automated deduction.

Part 2 on Computability Theory explores the limits of computation using an abstract model of computers called register machines.

Part 3 on Philosophical Logic presents a certain non-classical logic (relevance logic) and a semantics for it that is useful for automated reasoning systems that must deal with the possibility of inconsistent information.

The book can serve a variety of needs. For the first time there is now available a text for an instructor who would like to offer a course that teaches the role of logic in several disciplines. The book could also be used as a supplementary text for a logic course that emphasizes the more traditional topics of logic but also wishes to include a few special topics. The book is also designed to be a valuable resource for researchers and academics who want an accessible yet substantial introduction to the three topics.

The three areas from which the special topics are drawn — proof theory, computability theory, and philosophical logic — exhibit the different roles that logic plays in three different disciplines: computer science, mathematics, and philosophy. The three parts of the book were written by a computer scientist, a mathematician, and a philosopher, respectively, and each part was reviewed by the other two authors for accessibility to students in their fields. The three parts of the book are roughly of equal length. The second part, on computability theory, is largely independent of the first, but the third part, on philosophical logic, is best presented after the first two parts.

Although it is helpful to have had a previous course in logic, we present the topics in such a way that this is not necessary. However, some mathematical background is useful, especially if no logic background is offered. We had a number of freshmen and sophomores take this course with success, but they had a strong analytic preparation. In particular, prior exposure to proofs by induction is important. (We do offer a summary review of the induction methods employed in an appendix.)

The three topics covered are both timely and important. Although the use of automated theorem proving in artificial intelligence (AI) is often associated with the early decades of AI, it is also of great value in some current AI research programs. Watson, IBM's question-answering computer, made famous in 2011 by an impressive performance on the quiz show *Jeopardy!*, employed resolution logic (presented in **Part 1**) through the resolution-based programming language Prolog.

Part 2, on computability theory, presents one of the great success stories of mathematical logic. We now have a methodology for proving that certain problems cannot be solved by an algorithm. The ideas required to reach this goal can be traced back to Gödel and Turing and moreover played an important role in the development of the modern-day computer; it is for this reason that Gödel's and Turing's names are on *Time* magazine's list of the twenty most influential scientists and thinkers of the twentieth century.

Classical logic was motivated by considerations in mathematics. The important role that logic plays in other disciplines has given rise to logics that extend or differ from classical logic; examples include modal logic, intuitionistic logic, fuzzy logic, and relevance logic. **Part 3** explores the ideas of extensions and alternatives to classical logic, with an in-depth treatment of one of these, relevance logic.

The book begins with proof theory for both propositional logic and first-order logic. In each case, there is a quick review of the semantics of that logic. This has the advantage of serving as background for the subsequent parts on computability theory and non-classical logics.

A computer-oriented deductive logic based on the resolution inference rule is employed. At the propositional level all proofs are given, including both soundness and completeness proofs. In resolution logic the completeness theorem proof has an intuitive graphical form that makes the proof easier to comprehend. At the first-order level proofs are deferred to a set of problems to be undertaken by the mathematically oriented students. They cover most of the major results, including the steps to the completeness theorem. Plausibility arguments are used instead. This pedagogical strategy works well without losing the important content because first-order proof theory based on resolution employs lifting proofs almost verbatim from the propositional counterpart proof. The lifting process is discussed in detail. There is an extensive treatment of restrictions of resolution logic based on linear resolution that serves as the basis of Prolog, a computer programming language based on deduction. No programming experience is required.

The second part of the book introduces the student to computability theory, an area of mathematical logic that should be of interest to a broad audience due to its influence on the development of the computer. There are two major goals: clarify the intuitive notion of an algorithm; and develop a methodology for proving that certain problems cannot be solved by an algorithm. Four famous problems whose solution requires an algorithm are emphasized: Hilbert's Decision Problem, Hilbert's Tenth Problem, the Halting Problem, and Thue's Word Problem. A wide range of explicit algorithms are described, after which attention is restricted to the set of natural numbers. In this setting three informal concepts are defined (each in terms of an algorithm): computable function, decidable relation, and semi-decidable relation. The first three problems mentioned above are semi-decidable (in a more general sense), but are they decidable? Two models of computation are described in considerable detail, each with the motivation of giving a precise counterpart to the three informal concepts. The first model is a machine model, namely the register machine and RM-computable functions. Turing's diagonal argument that the Halting Problem is unsolvable is given, together with an outline of his application of that result, namely that Hilbert's Decision Problem is unsolvable. The Post-Markov result that the Word Problem is unsolvable is also proved. The second model of computation is a mathematical model, the recursive functions. Precise counterparts of the three informal concepts are defined: recursive functions, recursive relations, and recursively enumerable relations. There is a detailed proof that the two models give the same class of functions. The relationship between the informal concepts and their formal counterparts, together with the important role of the Church-Turing Thesis, is emphasized.

The third part of the book consists of topics from philosophical logic, with an emphasis on the propositional calculus of a particular non-classical logic known as relevance logic. We follow Anderson and Belnap's own presentation of it here. The topic is presented by first considering some well-known theorems of classical propositional logic that clash with intuitions about the use of "if ... then ...," which have been known as "paradoxes of implication." The student is invited to reflect on the features of classical logic that give rise to them. This is approached by presenting the rules for a natural deduction system for classical logic and examining which features of these rules permit derivation of the non-intuitive theorems or (so-called) paradoxes. This motivates considering alternative rules for deriving theorems, which is an occasion for a discussion of the analysis of the conditional (if ... then ...) and its relation to deduction and derivation. The non-classical logic known as relevance logic is presented as one such alternative. Both a natural deduction style proof system and a four-valued semantics (told true, told false, told both, told neither) for this logic are given. This is important, as some philosophers present relevance logic as a paraconsistent logic. The pedagogical approach we take here shows that is by no means mandatory: by the use of the engaging example of its application in a question-answering computer, we present a practical application in which this non-classical logic accords well with intuitions about what one would want in a logic to deal with situations in which we are faced with conflicting information. This example broadens the student's ideas of the uses and capabilities of logic. The inferential semantics is presented using a mathematical structure called a lattice. A brief introduction to mathematical lattices is provided. Then, drawing on the points in the classic paper "How a Computer Should Think," it is shown that, in certain contexts in which automated deduction is employed, relevance logic is to be preferred over classical logic. Some connections with the two earlier parts of the course on computer deduction and computability theory are made. Part 3 closes with some remarks on the impact of relevance logic in various disciplines.

Acknowledgments

I am grateful to the many people who inspired and guided my education, of which this book is one consequence. Angelo Margaris and Robert Stoll, Oberlin College, showed me the elegance of mathematics. Paul Gilmore, at IBM, introduced me to the beauty of mathematical logic. Marvin Minsky, MIT, one of the founders of the field of artificial intelligence (AI), kindled my interest in AI which led, ultimately, to the study of logic. It is Martin Davis, at the Courant Institute of NYU and Yeshiva University, to whom I owe the greatest thanks for the knowledge of logic I possess. Books, particularly Davis' *Decidability and Unsolvability*, Kleene's *Introduction to Metamathematics*, Enderton's *A Mathematical Introduction to Logic*, and Shoenfield's *Mathematical Logic* replaced my lack of many formal courses in logic. Earlier work for Martin, and my interest in AI, led to my lifelong interest in automated theorem proving. Peter Andrews, at Carnegie Mellon University, also contributed significantly to my logic education. To the many others who go unnamed but contributed to my education, in the field of automated theorem proving in particular, I extend my great thanks. I dedicate this book to my father, to whom I owe my interest in science and mathematics, to my wife for her patience and support throughout my entire career, and to sons Rob and Doug who so enrich our lives.

Donald W. Loveland

I wish to express my deep appreciation to my Ph.D. thesis advisor at Duke, the topologist J. H. Roberts. Roberts was a charismatic professor whose use of the Moore method inspired me to pursue an academic research career in set-theoretic topology. Although Roberts is my mathematical father, J. R. Shoenfield is surely my mathematical uncle. During the 1980s I greatly enjoyed attending his crystal-clear graduate level lectures at Duke on computability theory, set theory, and model theory. I also wish to express my thanks to my many other teachers and professors of mathematics and logic throughout the years. These are: High school: E. Whitley; Davidson College: R. Bernard, B. Jackson, J. Kimbrough, W. McGavock; Duke University: L. Carlitz, T. Gallie, S. Warner, N. Wilson. I am also grateful for the opportunity to spend my two years of active military duty (1963–1965) writing machine language programs for the Bendix G-15 computer at the Weapons Analysis Branch of the Army Artillery at Fort Sill, OK. Finally, I dedicate my portion of this book to my wife Margaret, son Richie, daughter Katie, and my parents whom I miss very much.

Richard E. Hodel

I owe thanks to many for reaching a point in my life at which I could participate in this interdisciplinary book project. Of the many professors from whom I took courses on mathematics and logic, I especially thank those whose lectures revealed something of the grand ideas in the field of mathematical logic: the mathematician Peter B. Andrews and the philosopher Gerald Massey (both of whom, I later learned, studied with Alonzo Church at Princeton) and the late Florencio Asenjo, whose lectures on set theory I especially enjoyed. A seminar in philosophy of mathematics run by Kenneth Manders and Wilfried Sieg showed me how the work of mathematicians from earlier centuries (Descartes, Dedekind, Cantor, Hilbert) could be appreciated both in historical context and from the perspective of twentieth century logic and model theory; I owe a lot to both of them for discussions outside that seminar, too. I feel fortunate to have been able to study with such mathematically informed philosophers in tandem with studying mathematical topics that I then only suspected of utility in philosophy: graph theory, combinatorics, abstract algebra, and foundations of geometry. My interest in logic probably first began, though, with computer science, as awe of the power of formalization while listening to R. W. Conway's lectures on structured programming in PL/1 and PL/C at Cornell University.

In philosophical logic, my debt is almost wholly to Nuel D. Belnap, Jr., for many wonderful seminar meetings, and for his elegant, highly instructive papers. It is Belnap's work in *Entailment: The logic of relevance and necessity*, of course, that I present here, but with the larger idea of conveying the role of logician as actively working to capture and formalize what we recognize as good reasoning, revising and inventing along the way as needed. It is for that vision of philosophical logic, especially, that I owe much to Nuel.

Both Nuel Belnap and his successor, Anil D. Gupta, reviewed earlier versions of the manuscript of Part 3 and provided useful suggestions; David Zornek did the same specifically for the section on four-valued logic. Jeff Horty provided encouragement, and Bill McDowell provided a rare and valuable test of the accessibility and clarity of the content of Part 3 frorn the standpoint of a student who had not previously studied relevance logic nor heard our lectures, by reviewing the manuscript and developing problem solutions. Rob Lewis offered his services to convert Part 3 of the manuscript to LaTeX under a demanding time constraint, and Peter Spirtes dropped everything to proofread the resulting formal proofs while publisher deadlines loomed.

S. G. Sterrett

The inspiration for this textbook project owes much to the students in the course on which it is based; their situations varied widely, from students such as Jacqueline Ou, who took the course as a freshman, excelled, and went on to a career as a scientist, to Justin Bledin, who took the course as a senior , excelled, and then decided to switch fields, making logic, methodology and philosophy of science his career. This book project would not have been attempted without examples of such student interest in, and mastery of, the course material, and we thank the many students we have had the pleasure of knowing who similarly inspired us but are not specifically named here.

We thank three anonymous reviewers for their thorough reviews, which contained constructive comments and excellent suggestions that greatly improved all three parts of the book. We thank the Princeton University Press staff for their devotion to producing a premier product, and to their understanding and cooperation when we made special requests. In particular, we thank Vickie Kearn, Executive Editor for mathematics, for her strong belief in the value of our proposed book, and her subsequent support and guidance through the publication process.

Donald Loveland
Richard Hodel
S. G. Sterrett
August 5, 2013

Contents