

# Logic For Computer Science

## Foundations of Automatic Theorem Proving

Copyright 2003, Jean H. Gallier

June 2003

Jean Gallier  
University of Pennsylvania  
Department of Computer and Information Science  
200 South 33rd Street  
Philadelphia, Pa 19104  
USA  
e-mail: [jean@saul.cis.upenn.edu](mailto:jean@saul.cis.upenn.edu)

To Anne, my wife,

Mia, Philippe and Sylvie, my children,

and my mother, Simone



# Preface (2003 Edition)

This is a slightly revised version of the 1985 edition of my logic book. Many typos and errors have been corrected and the line drawings have been improved. Most mistakes were minor, except for a subtle error in Theorem 4.3.3. Indeed, the second part of the theorem about the complexity of the proof tree  $T$  obtained from a resolution refutation  $D$  is false: It is not necessarily the case that the number of leaves of  $T$  is less than or equal to the number of resolution steps in  $D$ . As a consequence, Lemma 4.3.4 is also false.

In this revised edition, we simply removed all statements about the complexity of the conversion of resolution refutations into proof trees. Hopefully, this part of the book is now correct, as well as the rest of it!

Some parts of the book have now aged, in particular, the parts about PROLOG. Also, eighteen years later, I would prefer to present some of the material in a different order and in a different manner. In particular, I would separate more clearly the material on the resolution method (Chapter 4) from the more proof-theory oriented material. However, I consider this too big a task and this mildly revised version will have to do (at least, for now!).

Ideally, a number of topics should also be covered, for example, some basics of constructive logic, linear logic, temporal logic and model checking. Again, this is too formidable a task for a single individual and I hope that readers enjoy this new version of my book anyway.

It should be noted that this book is of “pre $\text{\TeX}$  vintage.” This means that  $\text{\LaTeX}$  was not available at the time this book was written, which implies that I had to write macros (stolen and adapted from D. Knuth blue  $\text{\TeX}$  book) to do chapter headings, etc. I also had to adapt the infamous macro

`makeindex` to produce the dreaded index (which was turned into final form using a PASCAL program!). Thus, I indulged in the practice of the change of catcode to turn the symbol `^` into an active character, among other strange things! Nevertheless, I am grateful to Knuth for producing  $\text{\TeX}$ . Without it, this book would not be alive.

In retrospect, I realize how much I was inspired by, and thus, how much I owe, Gerhard Gentzen, Jacques Herbrand, Stephen Kleene, Gaisi Takeuti, Raymond Smullyan, Peter Andrews and last but not least, Jean-Yves Girard. You have my deepest respect for your seminal and inspiring work and I thank all of you for what you taught me.

Philadelphia, June 2003

Jean Gallier

# Preface (1985 Edition)

This book is intended as an introduction to mathematical logic, with an emphasis on proof theory and procedures for constructing formal proofs of formulae algorithmically.

This book is designed primarily for computer scientists, and more generally, for mathematically inclined readers interested in the formalization of proofs, and the foundations of automatic theorem-proving.

The book is self contained, and the level corresponds to senior undergraduates and first year graduate students. However, there is enough material for at least a two semester course, and some Chapters (Chapters 6,7,9,10) contain material which could form the basis of seminars. It would be helpful, but not indispensable, if the reader has had an undergraduate-level course in set theory and/or modern algebra.

Since the main emphasis of the text is on the study of proof systems and algorithmic methods for constructing proofs, it contains some features rarely found in other texts on logic. Four of these features are:

- (1) The use of Gentzen Systems;
- (2) A Justification of the Resolution method via a translation from a Gentzen System;
- (3) A presentation of SLD-resolution and a presentation of the foundations of PROLOG;
- (4) Fast decisions procedures based on congruence closures.

A fruitful way to use this text is to teach PROLOG concurrently with the material in the book, and ask the student to implement in PROLOG some of the procedures given in the text, in order to design a simple theorem-prover.

Even though the main emphasis of the book is on the design of procedures for constructing formal proofs, the treatment of the semantics is perfectly rigorous. The following paradigm has been followed: Having defined the syntax of the language, it is shown that the set of well-formed formulae is a freely generated inductive set. This is an important point, which is often glossed over. Then, the concepts of satisfaction and validity are defined by recursion over the freely generated inductive set (using the “unique homomorphic extension theorem”, which can be rigorously justified). Finally, the proof theory is developed, and procedures for constructing proofs are given. Particular attention is given to the complexity of such procedures.

In our opinion, the choice of Gentzen systems over other formal systems is pedagogically very advantageous. Gentzen-style rules reflect directly the semantics of the logical connectives, lead naturally to mechanical proof procedures, and have the advantage of having duality “built in”. Furthermore, in our opinion, Gentzen systems are more convenient than tableaux systems or natural deduction systems for proof-theoretical investigations (cut-free proofs in particular). In three years of teaching, I have found that Gentzen-like systems were very much appreciated by students.

Another good reason for using a formal system inspired from Gentzen (a sequent calculus), is that the completeness theorem is obtained in a natural and simple way. Furthermore, this approach even yields a program (the *search* procedure) for constructing a proof tree for a valid formula. In fact, in our presentation of the completeness theorem (inspired by Kleene, Kleene 1967), the search for a proof tree is described by a program written in pseudo-PASCAL. We also show how a proof procedure for first-order logic with equality can be developed incrementally, starting with the propositional case.

The contents of the book are now outlined.

Chapter 1 sets the goals of the book.

Chapter 2 has been included in order to make the book as self contained as possible, and it covers the mathematical preliminaries needed in the text. It is recommended to refer to this Chapter only when needed, as opposed to reading it entirely before proceeding to Chapter 3.

Propositional logic is studied in Chapter 3. This includes the syntax and semantics of propositional logic. Gentzen systems are introduced as a method for attempting to falsify a proposition. The completeness theorem is shown as well as some of its consequences (the conjunctive and disjunctive normal forms). By introducing infinite sequents, the extended completeness theorem and the compactness theorem are obtained. An informal exposition of the complexity classes P, NP, and of the concept of NP-completeness is given at the end of the Chapter.



The resolution method for propositional logic is presented in Chapter 4. This Chapter uses a new approach for proving the completeness of resolution. Completeness is obtained by defining a special Gentzen System whose completeness follows easily from the results of Chapter 3, and giving an algorithm for converting proofs in the special Gentzen Systems into resolution refutations. Some complexity issues are also examined.

Chapter 5 is devoted to first-order logic. The syntax and semantics are presented. This includes the notions of first-order languages, structures, and models. Gentzen systems are extended to deal with quantifiers and equality. The concept of a Hintikka set is also introduced. It is shown that every Hintikka set is satisfiable in a model whose domain is (a quotient of) a term algebra. This result, together with a generalization of the *search* procedure, is used to derive the main theorems of first-order logic: completeness, compactness, model existence, Löwenheim-Skolem theorems. One of the main themes in this Chapter is that the *search* procedure is a “Hintikka set builder”.

Chapter 6 is devoted to Gentzen’s “Cut elimination Theorem” and some of its applications. A simple semantic proof derived from the completeness theorem is given for the Gentzen System LK. An entirely proof-theoretic (constructive) argument is also given for a simpler system  $G1^{nnf}$ . This proof due to Schwichtenberg has the advantage that it also yields a precise upper bound on the length of cut-free proofs obtained from a proof with cut. This result is then extended to a system with equality.

A constructive proof of Craig’s Interpolation Theorem is given, and Beth’s Definability Theorem and Robinson’s Joint Consistency Theorem are also proved. This Chapter contains more advanced material than the previous Chapters.

Chapter 7 is devoted to Gentzen’s “Sharpened Hauptsatz”, Herbrand’s Theorem, and the Skolem-Herbrand-Gödel Theorem. As Chapter 6, this Chapter contains more advanced material. Gentzen’s “Sharpened Hauptsatz” for prenex sequents is proved constructively, using proof transformation techniques. A version of the “Sharpened Hauptsatz” is also proved constructively for sequents consisting of formulae in NNF. To prove this result, a new Gentzen system with quantifier rules applying to certain subformulae is defined. This version of the “Sharpened Hauptsatz” for sequents in NNF appears to be new. Using these results, constructive versions of Herbrand’s Theorem are proved, as well as Andrews’s version of the Skolem-Herbrand-Gödel Theorem (Andrews 1981). The class of primitive recursive functions and the class of recursive functions are also briefly introduced.

In Chapter 8, the resolution method for first-order logic is presented. A recursive unification algorithm inspired from Robinson’s algorithm (Robinson 1965) is presented. Using results from Chapter 4 and the Skolem-Herbrand-Gödel Theorem, the completeness of first-order resolution is shown, using the “lifting technique”. Paramodulation is also briefly discussed.

Chapter 9 is devoted to SLD-resolution and the foundations of PROLOG. Using techniques from Chapter 4, the completeness of SLD-resolution is shown, by translating proofs in a certain Gentzen system into SLD-refutations. This approach appears to be new. Logic programs are defined, and a model-theoretic semantics is given. It is shown that SLD-resolution is a sound and complete computational proof procedure for logic programs. Most of this material can only be found in research papers, and should be useful to readers interested in logic programming.

In Chapter 10 (the last Chapter), a brief presentation of many-sorted first-order logic is given. This presentation should be sufficient preparation for readers interested in the definition of abstract data types, or computing with rewrite rules. Finally, an extension of the congruence closure method of Nelson and Oppen (Nelson and Oppen 1980) to the many-sorted case and its application to fast decision procedures for testing the validity of quantifier-free formulae are presented.

This book grew out of a number of class notes written for a graduate course in logic for computer scientists, taught at the University of Pennsylvania (CIS581). The inspiration for writing the book came from Sheila Greibach (my advisor at UCLA) and Ronald Book (my “supervisor” at UCSB, while I was a “Post Doc”), who convinced me that there is no better way to really know a topic than writing about it.

I wish to thank my colleagues Saul Gorn, Dale Miller and Alex Pelin for reading the manuscript very carefully, and for many helpful comments. I also wish to thank my students William Dowling, Tomas Isakowitz, Harry Kaplan, Larry Krablin, Francois Lang, Karl Schimpf, Jeff Stroemer, Stan Raatz and Todd Rockoff for their help in “debugging” the manuscript. This includes reporting of typos, stylistic improvements, additional exercises, and correction of mistakes.

Philadelphia, July 1985

Jean Gallier

# How To Use This Book As A Text

This book is written at the level appropriate to senior undergraduate and first year graduate students in computer science, or mathematics. The prerequisites are the equivalent of undergraduate-level courses in either set theory, abstract algebra, or discrete structures. All the mathematical background necessary for the text itself is contained in Chapter 2, and in the Appendix. Some of the most difficult exercises may require deeper knowledge of abstract algebra.

Most instructors will find it convenient to use Chapter 2 on a “call by need” basis, depending on the background of the students. However, to the authors’s experience, it is usually desirable to review the material contained in Sections 2.1, 2.2 and 2.3.

To help the instructor make up a course, at the end of this section we give a graph showing the dependence of the Sections and Chapters. This graph only applies to the text itself, but not to the exercises, which may depend on any earlier Sections.

The core of the subject which, in the author’s opinion, should be part of any course on logic for computer science, is composed of Sections 3.1, 3.2, 3.3 (excluding 3.3.5), 3.4, 3.5, 5.1, 5.2, 5.3, 5.4, 5.5.

The Sections which are next in priority (as core Sections) are 3.6, 5.6, 6.1, 6.2, 6.3, 7.1, 7.2, 7.3, 7.5.

More advanced topics suitable for seminars are covered in Sections 6.4, 6.5, 6.6, 6.7, 7.4, 7.6 and in Chapter 10.

Sections marked with a star (\*) give a glimpse of topics only sketched in this book. They can be omitted at first reading.

Some results from Section 2.4 are required in Chapter 5. However, in order to shorten Chapter 2, this material as well the material on many-sorted algebras has been made into an Appendix. Similarly, to be perfectly rigorous, Chapter 8 depends on Section 7.6 (since the Skolem-Herbrand-Gödel Theorem proved in Section 7.6 is used to prove the completeness of resolution). However, if the emphasis of the course is on theorem-proving techniques rather than on foundations, it is possible to proceed directly from Section 5.5 to Chapter 8 after having covered Chapter 4). The instructor may simply quote the Herbrand-Skolem-Gödel Theorem from Section 7.6, without proof.

Hence, depending on the time available and the level of the class, there is flexibility for focusing more on automatic theorem-proving methods, or more on foundations. A one semester course emphasizing theorem-proving techniques may consist of the core, plus Chapter 4, Chapter 8, and possibly part of Chapter 9. A one semester course emphasizing foundations may consist of the core, plus Chapter 6 and Chapter 7.

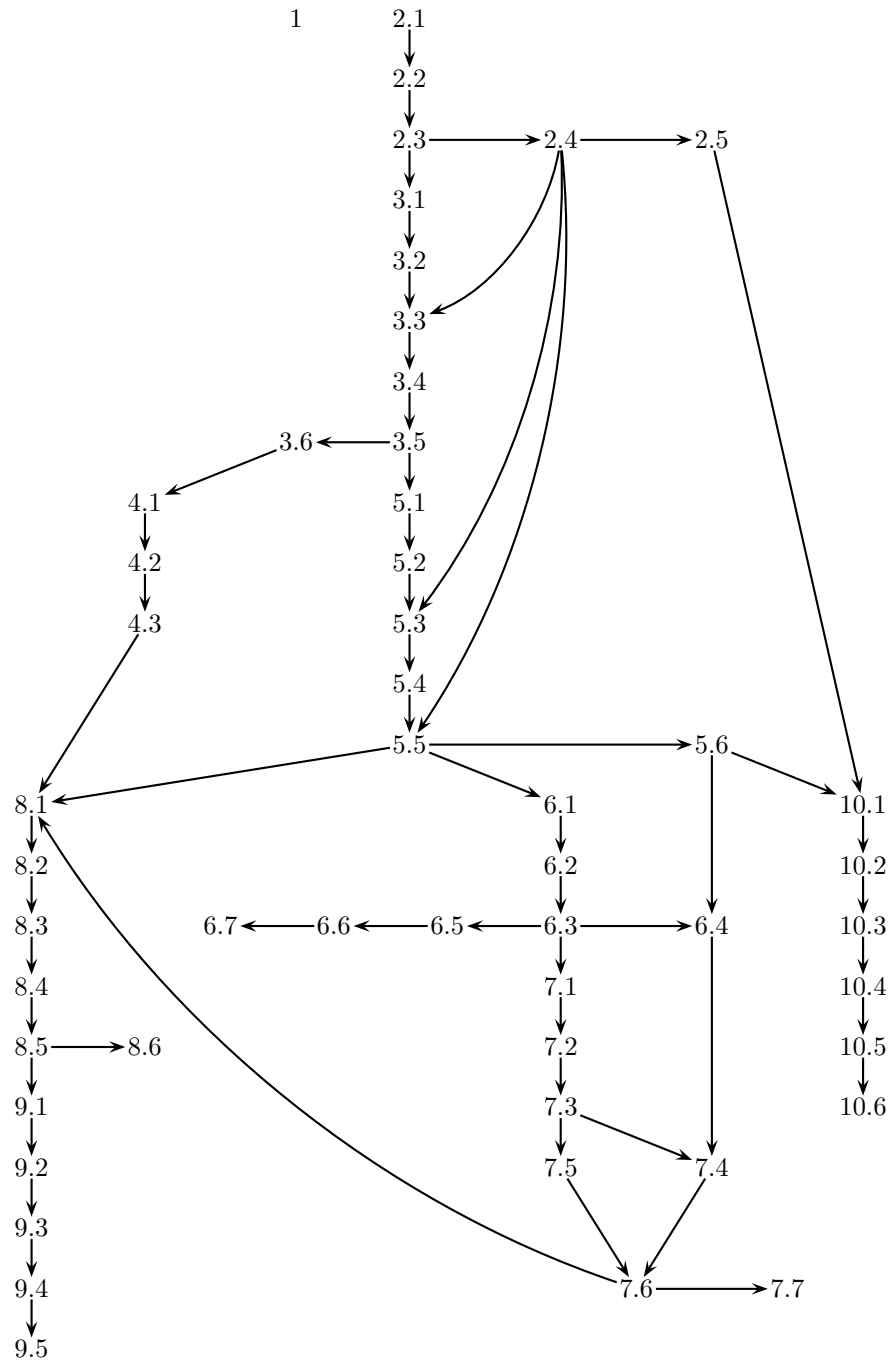
The ideal situation is to teach the course in two semesters, with automatic theorem-proving techniques first. The second semester covers the foundations, and finishes with a more complete coverage of Chapter 9, Chapter 10, and possibly some material on decision procedures, or on rewrite rules.

It is also possible to use Chapter 6 and Chapter 7 as the core of a seminar on analytic versus non-analytic proofs.

Problems are usually found at the end of each Section. The problems range from routine to very difficult. Difficult exercises or exercises requiring knowledge of material not covered in the text are marked with a star (\*). Very difficult exercises are marked with a double star (\*\*). A few programming assignments have been included.

Some historical remarks and suggestions for further reading are included at the end of each Chapter. Finally the end of a proof is indicated by the symbol  $\square$  (box). The word “iff” is used as an abbreviation for “if and only if”.

## DEPENDENCY OF SECTIONS



# TABLE OF CONTENTS

**Chapter 1: INTRODUCTION, 1**

**Chapter 2: MATHEMATICAL PRELIMINARIES, 4**

- 2.1 Relations, Functions, Partial Orders, Induction, 4
  - 2.1.1 Relations, 4
  - 2.1.2 Partial Functions, Total Functions, 5
  - 2.1.3 Composition of Relations and Functions, 5
  - 2.1.4 Injections, Surjections, Bijections, 6
  - 2.1.5 Direct Image, Inverse Image, 6
  - 2.1.6 Sequences, 6
  - 2.1.7 Natural Numbers and Countability, 7
  - 2.1.8 Equivalence Relations, 7
  - 2.1.9 Partial and Total Orders, 8
  - 2.1.10 Well-Founded Sets and Complete Induction, 9
  - 2.1.11 Restrictions and Extensions, 12
  - 2.1.12 Strings, 12
- 2.2 Tree Domains and Trees, 13
  - 2.2.1 Tree Domains, 13
  - 2.2.2 Trees, 14
  - 2.2.3 Paths, 15
  - 2.2.4 Subtrees, 15

- 2.2.5 Tree Replacement, 15
- 2.2.6 Ranked Alphabets and  $\Sigma$ -Trees, 16
- 2.3 Inductive Definitions, 17
  - 2.3.1 Inductive Closures, 17
  - 2.3.2 Freely Generated Sets, 20
  - 2.3.3 Functions Defined Recursively over Freely Generated Inductive Sets, 22
- PROBLEMS, 24

### Chapter 3: PROPOSITIONAL LOGIC, 28

- 3.1 Introduction, 28
- 3.2 Syntax of Propositional Logic, 31
  - 3.2.1 The Language of Propositional Logic, 32
  - 3.2.2 Free Generation of PROP, 32
  - PROBLEMS, 36
- 3.3 Semantics of Propositional Logic, 39
  - 3.3.1 The Semantics of Propositions, 39
  - 3.3.2 Satisfiability, Unsatisfiability, Tautologies, 42
  - 3.3.3 Truth Functions and Functionally Complete Sets of Connectives, 45
  - 3.3.4 Logical Equivalence and Boolean Algebras, 48
  - \* 3.3.5 NP-Complete Problems, 50
  - PROBLEMS, 54
- 3.4 Proof Theory of Propositional Logic: The Gentzen System  $G'$ , 60
  - 3.4.1 Basic Idea: Searching for a Counter Example, 60
  - 3.4.2 Sequents and the Gentzen System  $G'$ , 62
  - 3.4.3 Falsifiable and Valid Sequents, 64
  - 3.4.4 Axioms, Deduction Trees, Proof Trees, Counter Example Trees, 65
  - 3.4.5 Soundness of the Gentzen System  $G'$ , 67
  - 3.4.6 The Search Procedure, 68
  - 3.4.7 Completeness of the Gentzen System  $G'$ , 71
  - 3.4.8 Conjunctive and Disjunctive Normal Form, 73
  - 3.4.9 Negation Normal Form, 74
  - PROBLEMS, 76
- 3.5 Proof Theory for Infinite Sequents: Extended Completeness of  $G'$ , 82
  - 3.5.1 Infinite Sequents, 82
  - 3.5.2 The Search Procedure for Infinite Sequents, 83

3.5.3	König's Lemma, 89
3.5.4	Signed Formulae, 89
3.5.5	Hintikka Sets, 90
3.5.6	Extended Completeness of the Gentzen System $G'$ , 92
3.5.7	Compactness, Model Existence, Consistency, 94
3.5.8	Maximal Consistent Sets, 95
	PROBLEMS, 97
3.6	More on Gentzen Systems: The Cut Rule, 109
3.6.1	Using Auxiliary Lemmas in Proofs, 109
3.6.2	The Gentzen System $LK'$ , 110
3.6.3	Logical Equivalence of $G'$ , $LK'$ , and $LK' - \{cut\}$ , 112
3.6.4	Gentzen's Hauptsatz for $LK'$ (Cut elimination theorem for $LK'$ ), 113
3.6.5	Characterization of Consistency in $LK'$ , 114
	PROBLEMS, 115
	Notes and Suggestions for Further Reading, 116
 <b>Chapter 4: RESOLUTION IN PROPOSITIONAL LOGIC, 117</b>	
4.1	Introduction, 117
4.2	A Special Gentzen System, 118
4.2.1	Definition of the System $GCNF'$ , 118
4.2.2	Soundness of the System $GCNF'$ , 121
4.2.3	Completeness of the System $GCNF'$ , 123
	PROBLEMS, 124
4.3	The Resolution Method for Propositional Logic, 126
4.3.1	Resolution DAGs, 126
4.3.2	Definition of the Resolution Method for Propositional Logic, 128
4.3.3	Soundness of the Resolution Method, 131
4.3.4	Converting $GCNF'$ -proofs into Resolution Refutations and Completeness, 131
4.3.5	From Resolution Refutations to $GCNF'$ -proofs, 137
	PROBLEMS, 142
	Notes and Suggestions for Further Reading, 144
 <b>Chapter 5: FIRST-ORDER LOGIC, 146</b>	
5.1	INTRODUCTION, 146



5.2	FIRST-ORDER LANGUAGES, 147
5.2.1	Syntax, 147
5.2.2	Free Generation of the Set of Terms, 150
5.2.3	Free Generation of the Set of Formulae, 151
5.2.4	Free and Bound Variables, 153
5.2.5	Substitutions, 155
	PROBLEMS, 156
5.3	SEMANTICS OF FIRST-ORDER LANGUAGES, 158
5.3.1	First-Order Structures, 158
5.3.2	Semantics of Formulae, 159
5.3.3	Satisfaction, Validity, and Model, 162
5.3.4	A More Convenient Semantics, 163
5.3.5	Free Variables and Semantics of Formulae, 169
5.3.6	Subformulae and Rectified Formulae, 171
5.3.7	Valid Formulae Obtained by Substitution in Tautologies, 173
5.3.8	Complete Sets of Connectives, 175
5.3.9	Logical Equivalence and Boolean Algebras, 176
	PROBLEMS, 179
5.4	Proof Theory of First-Order Languages, 187
5.4.1	The Gentzen System G for Languages Without Equality, 187
5.4.2	Deduction Trees for the System G, 188
5.4.3	Soundness of the System G, 189
5.4.4	Signed Formulae and Term Algebras (no Equality Symbol), 192
5.4.5	Reducts, Expansions, 194
5.4.6	Hintikka Sets (Languages Without Equality), 194
5.4.7	Completeness: Special Case of Languages Without Function Symbols and Without Equality, 197
	PROBLEMS, 205
5.5	Completeness for Languages with Function Symbols and no Equality, 207
5.5.1	Organizing the Terms for Languages with Function Symbols and no Equality, 207
5.5.2	The Search Procedure for Languages with Function Symbols and no Equality, 209
5.5.3	Completeness of the System G (Languages Without Equality), 214
5.5.4	Löwenheim-Skolem, Compactness, and Model Existence Theorems for Languages Without Equality, 217
5.5.5	Maximal Consistent Sets, 218

## PROBLEMS, 219

- 5.6 A Gentzen System for First-Order Languages With Equality, 230
  - 5.6.1 Hintikka Sets for Languages with Equality, 230
  - 5.6.2 The Gentzen System  $G_=($  Languages With Equality), 236
  - 5.6.3 Soundness of the System  $G_=$ , 239
  - 5.6.4 Search Procedure for Languages With Equality, 239
  - 5.6.5 Completeness of the System  $G_=$ , 241
  - 5.6.6 Löwenheim-Skolem, Compactness, and Model Existence Theorems for Languages With Equality, 243
  - 5.6.7 Maximal Consistent Sets, 243
  - 5.6.8 Application of the Compactness and Löwenheim-Skolem Theorems: Nonstandard Models of Arithmetic, 244
- PROBLEMS, 246
- Notes and Suggestions for Further Reading, 255

## Chapter 6: GENTZEN'S CUT ELIMINATION THEOREM AND APPLICATIONS, 256

- 6.1 Introduction, 256
- 6.2 Gentzen System LK for Languages Without Equality, 257
  - 6.2.1 Syntax of LK, 257
  - 6.2.2 The Logical Equivalence of the Systems  $G$ ,  $LK$ , and  $LK - \{cut\}$ , 259
- PROBLEMS, 261
- 6.3 The Gentzen System  $LK_e$  with Equality, 262
  - 6.3.1 Syntax of  $LK_e$ , 262
  - 6.3.2 A Permutation Lemma for the System  $G_=$ , 263
  - 6.3.3 Logical equivalence of  $G_=$ ,  $LK_e$ , and  $LK_e$  Without Essential Cuts: Gentzen's Hauptsatz for  $LK_e$  Without Essential Cuts, 266
- PROBLEMS, 268
- 6.4 Gentzen's Hauptsatz for Sequents in NNF, 269
  - 6.4.1 Negation Normal Form, 269
  - 6.4.2 The Gentzen System  $G1^{nnf}$ , 270
  - 6.4.3 Completeness of  $G1^{nnf}$ , 272
  - 6.4.4 The Cut Elimination Theorem for  $G1^{nnf}$ , 273
  - 6.4.5 The System  $G1_{=}^{nnf}$ , 281
  - 6.4.6 The Cut Elimination Theorem for  $G1_{=}^{nnf}$ , 282
- PROBLEMS, 284

- 6.5 Craig's Interpolation Theorem, 287
  - 6.5.1 Interpolants, 287
  - 6.5.2 Craig's Interpolation Theorem Without Equality, 288
  - 6.5.3 Craig's Interpolation Theorem With Equality, 292
 PROBLEMS, 294
- 6.6 Beth's Definability Theorem, 295
  - 6.6.1 Implicit and Explicit Definability, 295
  - 6.6.2 Explicit Definability Implies Implicit Definability, 296
  - 6.6.3 Beth's Definability Theorem, Without Equality, 297
  - 6.6.4 Beth's Definability Theorem, With Equality, 299
 PROBLEMS, 299
- 6.7 Robinson's Joint Consistency Theorem, 300
  - PROBLEMS, 301
  - Notes and Suggestions for Further Reading, 302

**Chapter 7: GENTZEN'S SHARPENED HAUPTSATZ;  
HERBRAND'S THEOREM, 303**

- 7.1 Introduction, 303
- 7.2 Prenex Normal Form, 305
  - PROBLEMS, 309
- 7.3 Gentzen's Sharpened Hauptsatz for Prenex Formulae, 310
  - 7.3.1 Pure Variable Proofs, 310
  - 7.3.2 The Permutability Lemma, 314
  - 7.3.3 Gentzen's Sharpened Hauptsatz, 320
 PROBLEMS, 324
- 7.4 The Sharpened Hauptsatz for Sequents in NNF, 325
  - 7.4.1 The System  $G2^{nnf}$ , 325
  - 7.4.2 Soundness of the System  $G2^{nnf}$ , 328
  - 7.4.3 A Gentzen-like Sharpened Hauptsatz for  $G2^{nnf}$ , 330
  - 7.4.4 The Gentzen System  $G2_{=}^{nnf}$ , 336
  - 7.4.5 A Gentzen-like Sharpened Hauptsatz for  $G2_{=}^{nnf}$ , 337
 PROBLEMS, 337
- 7.5 Herbrand's Theorem for Prenex Formulae, 338
  - 7.5.1 Preliminaries, 338
  - 7.5.2 Skolem Function and Constant Symbols, 339
  - 7.5.3 Substitutions, 342
  - 7.5.4 Herbrand's Theorem for Prenex Formulae, 344

	PROBLEMS, 353
7.6	Skolem-Herbrand-Gödel's Theorem for Formulae in NNF, 355
7.6.1	Skolem-Herbrand-Gödel's Theorem in Unsatisfiability Form, 355
7.6.2	Skolem Normal Form, 357
7.6.3	Compound Instances, 359
7.6.4	Half of a Herbrand-like Theorem for Sentences in NNF, 360
7.6.5	Skolem-Herbrand-Gödel's Theorem (Sentences in NNF), 361
7.6.6	Comparison of Herbrand and Skolem-Herbrand-Gödel Theorems, 365
	PROBLEMS, 367
* 7.7	The Primitive Recursive Functions, 369
7.7.1	The Concept of Computability, 369
7.7.2	Definition of the Primitive Recursive Functions, 371
7.7.3	The Partial Recursive Functions, 372
7.7.4	Some Primitive Recursive Functions, 373
	PROBLEMS, 374
	Notes and Suggestions for Further Reading, 375
 <b>Chapter 8: RESOLUTION IN FIRST-ORDER LOGIC, 376</b>	
8.1	Introduction, 376
8.2	Formulae in Clause Form, 378
8.3	Ground Resolution, 379
8.4	Unification and the Unification Algorithm, 381
8.4.1	Unifiers and Most General Unifiers, 381
8.4.2	The Unification Algorithm, 383
	PROBLEMS, 394
8.5	The Resolution Method for First-order Logic, 395
8.5.1	Definition of the Method, 395
8.5.2	Soundness of the Resolution Method, 398
8.5.3	Completeness of the Resolution Method, 400
	PROBLEMS, 405
8.6	A Glimpse at Paramodulation, 407
	Notes and Suggestions for Further Reading, 409

**Chapter 9: SLD-RESOLUTION AND LOGIC PROGRAMMING  
(PROLOG), 410**

- 9.1 Introduction, 410
- 9.2  $GCNF'$ -Proofs in SLD-form, 411
  - 9.2.1 The Case of Definite Clauses, 411
  - 9.2.2  $GCNF'$ -Proofs in SLD-Form, 413
  - 9.2.3 Completeness of Proofs in SLD-Form, 413  
PROBLEMS, 421
- 9.3 SLD-Resolution in Propositional Logic, 422
  - 9.3.1 SLD-Derivations and SLD-Refutations, 422
  - 9.3.2 Completeness of SLD-Resolution for Horn Clauses, 425  
PROBLEMS, 427
- 9.4 SLD-Resolution in First-Order Logic, 427
  - 9.4.1 Definition of SLD-Refutations, 428
  - 9.4.2 Completeness of SLD-Resolution for Horn Clauses, 431  
PROBLEMS, 432
- 9.5 SLD-Resolution, Logic Programming (PROLOG), 433
  - 9.5.1 Refutations as Computations, 433
  - 9.5.2 Model-Theoretic Semantics of Logic Programs, 434
  - 9.5.3 Correctness of SLD-Resolution as a Computation Procedure, 439
  - 9.5.4 Completeness of SLD-Resolution as a Computational Procedure, 444
  - 9.5.5 Limitations of PROLOG, 445  
PROBLEMS, 445  
Notes and Suggestions for Further Reading, 447

**Chapter 10: MANY-SORTED FIRST-ORDER LOGIC, 448**

- 10.1 Introduction, 448
- 10.2 Syntax, 448
  - 10.2.1 Many-Sorted First-Order Languages, 449
  - 10.2.2 Free Generation of Terms and Formulae, 452
  - 10.2.3 Free and Bound Variables, Substitutions, 452  
PROBLEMS, 452
- 10.3 Semantics of Many-Sorted First-Order Languages, 453
  - 10.3.1 Many-Sorted first-Order Structures, 453

10.3.2	Semantics of Formulae, 453
10.3.3	An Alternate Semantics, 455
10.3.4	Semantics and Free Variables, 456
10.3.5	Subformulae and Rectified Formulae, 456
	PROBLEMS, 456
10.4	Proof Theory of Many-Sorted Languages, 456
10.4.1	Gentzen System $G$ for Many-Sorted Languages Without Equality, 456
10.4.2	Deduction Trees for the System $G$ , 458
10.4.3	Soundness of the System $G$ , 458
10.4.4	Completeness of $G$ , 458
10.5	Many-Sorted First-Order Logic With Equality, 458
10.5.1	Gentzen System $G_{=}$ for Languages with Equality, 458
10.5.2	Soundness of the System $G_{=}$ , 459
10.5.3	Completeness of the System $G_{=}$ , 459
10.5.4	Reduction of Many-Sorted Logic to One-Sorted Logic, 459
	PROBLEMS, 460
10.6	Decision Procedures Based on Congruence Closure, 460
10.6.1	Decision Procedure for Quantifier-free Formulae Without Predicate Symbols, 460
10.6.2	Congruence Closure on a Graph, 461
10.6.3	The Graph Associated With a Conjunction, 462
10.6.4	Existence of the Congruence Closure, 467
10.6.5	Decision Procedure for Quantifier-free Formulae, 467
10.6.6	Computing the Congruence Closure, 471
	PROBLEMS, 474
	Notes and Suggestions for Further Reading, 476

## APPENDIX, 477

2.4	Algebras, 477
2.4.1	Definition of an Algebra, 477
2.4.2	Homomorphisms, 478
2.4.3	Subalgebras, 479
2.4.4	Least Subalgebra Generated by a Subset, 479
2.4.5	Subalgebras Freely Generated by a Set $X$ , 480
2.4.6	Congruences, 481
2.4.7	Quotient Algebras, 482
2.5	Many-Sorted Algebras, 483
2.5.1	$S$ -Ranked Alphabets, 483

2.5.2	Definition of a Many-Sorted Algebra, 483
2.5.3	Homomorphisms, 484
2.5.4	Subalgebras, 484
2.5.5	Least Subalgebras, 485
2.5.6	Freely Generated Subalgebras, 485
2.5.7	Congruences, 486
2.5.8	Quotient Algebras, 487
2.5.9	Many-Sorted Trees, 487
	PROBLEMS, 488

**REFERENCES, 490****INDEX OF SYMBOLS, 495****INDEX OF DEFINITIONS, 498****SUBJECT INDEX, 502**