Contents

Foreword vii

Preface ix

Chapter 1 The Menu, The Map, and the Magic 1
1.1 The Menu for the Grand Feast . . . 3
1.2 The Book's Audience . . . 5
1.3 The Map . . . 7
1.4 Reasoning in Review . . . 8
1.5 Reasoning by Computer versus Reasoning by a Person . . . 10
1.6 Obstacles to the Effective Automation of Reasoning . . . 13
1.6.1 Language . . . 14
1.6.2 Inference Rules . . . 15
1.6.3 Assignment Completion . . . 16
1.6.4 Strategy . . . 17
1.6.5 Redundancy . . . 17
1.6.6 Specific versus General Information . . . 18
1.6.7 Conclusion Retention . . . 18
1.6.8 Conclusion Generation . . . 19
1.6.9 Inadequate Focus . . . 20
1.6.10 Conclusion Repetition . . . 21
1.6.11 Redundancy-Control Transformations . . . 22
1.6.12 Size of Deduction Step . . . 22
1.6.13 Metarules for Program Use . . . 23
1.6.14 Indexing . . . 23
1.7 Paradigms for Reasoning and for Research . . . 23
1.8 The Future of Automated Reasoning . . . 26

Chapter 2 Learning Logic by Example 29
2.1 and, or, not, if-then (implies) . . . 29

2.2 A Language for Automated Reasoning Programs 35
2.2.1 Predicates and Constants 36
2.2.2 Variables 37
2.2.3 Functions 39
2.3 Combinations of or with and, Complex if-then, and DeMorgan's Laws 44
2.4 Assumptions and Axioms, Types of Reasoning, and Proof 45
2.4.1 Assumptions and Axioms 46
2.4.2 Types of Reasoning, Inference Rules 47
2.4.3 Proof 50
2.5 Summary 56

Chapter 3 Automated Reasoning in Full **63**
3.1 Logic 64
3.1.1 and 64
3.1.2 or 64
3.1.3 not 65
3.1.4 if-then and implies 65
3.1.5 is-equivalent-to 65
3.1.6 Relationships and Laws in Logic 65
3.2 A Language Understood by an Automated Reasoning Program 67
3.2.1 Variables 68
3.3 Submitting a Problem to a Reasoning Program 70
3.3.1 Assumptions and Axioms 70
3.3.2 Special Facts and the Special Hypothesis 70
3.3.3 Denial of the Goal or Theorem 71
3.4 Inference Rules 72
3.4.1 Unification 72
3.4.2 Binary Resolution 73
3.4.3 UR-Resolution 74
3.4.4 Hyperresolution 75
3.4.5 Paramodulation 76
3.4.6 Other Inference Rules 78
3.5 The Empty Clause 79
3.6 Proof by Contradiction 79
3.7 Demodulation 80
3.8 Subsumption 82
3.9 Strategy 84
3.9.1 The Set of Support Strategy 84
3.9.2 Weighting 85
3.9.3 Unit Preference Strategy 85
3.9.4 Other Strategies 86

3.10 An Automated Reasoning Program in Action 87
3.11 OTTER and Earlier Automated Theorem-Proving Programs 89
3.11.1 The Basic Argonne Paradigm 93
3.11.2 A Sequential Theorem-Proving Algorithm 94
3.11.3 Implementing the Algorithm on Shared-Memory Multiprocessors 95
3.11.4 Implementing the Algorithm on Distributed-Memory Machines 98
3.12 Answers to Exercises 100

Chapter 4 Logic Circuit Design **107**
4.1 Introduction to Logic Circuit Design 107
4.1.1 Components of Logic Circuits 108
4.1.1.1 AND, OR, and NOT gates 108
4.1.1.2 Other Basic Components 109
4.1.2 Specifications 110
4.1.3 Problems for the Circuit Designer 111
4.2 Circuit Design Using Demodulation 112
4.2.1 Designing from Functional Specifications 112
4.2.2 Designing from Descriptive Sentences 116
4.2.3 Designing from Tables 117
4.2.4 Advantages and Disadvantages of the Demodulation Approach . 119
4.3 Circuit Design Using Hyperresolution 119
4.3.1 The OUTPUT Predicate 120
4.3.2 Construction Rules 120
4.3.3 Examples 121
4.3.4 Achievable Signals and the OUTPUT Predicate 124
4.4 Solution to the Two-Inverter Puzzle 124
4.4.1 Some New Complexities 125
4.4.2 The First Formulation 126
4.4.3 Reversions 131
4.4.4 Solving the Puzzle Faster 134
4.5 Multivalued Logic Design Using Negative Hyperresolution 137
4.5.1 T-gates 138
4.5.2 Representation of Multivalued Logic Functions 138
4.5.3 Construction Rules 139
4.5.4 Negative Hyperresolution 139
4.5.5 An Example 140
4.6 Answers to Exercises 142

Chapter 5 Logic Circuit Validation **147**
5.1 What Is Validation? 147
5.2 Simple Example 148

5.3 Important Features of the Example 150
5.3.1 Validation as Language Translation 150
5.3.2 Canonicalization 151
5.3.3 Simplification 152
5.3.4 Commutativity-Type Demodulators and Lexical Ordering 152
5.4 A More Complex Example 153
5.5 Validating an Adder 157
5.5.1 The One-Bit Full Adder 157
5.5.2 Combining Validated Subcircuits 161
5.6 Answers to Exercises 163

Chapter 6 Research in Mathematics 165
6.1 A Simple Example 166
6.2 A Simple Example Revisited 171
6.2.1 The Set of Support Strategy and the Simple Example 174
6.2.2 Weighting Applied to the Example 178
6.2.3 Demodulation, Canonicalization, and Simplification 180
6.2.4 Subsumption and Redundancy 182
6.3 Answering Open Questions 187
6.4 Model and Counterexample Generation 190
6.5 Representable Concepts: Easy and Difficult 197
6.6 Review 199
6.7 Using OTTER to Find Shorter Proofs 200
6.8 Answers to Exercises 205

Chapter 7 Research in Formal Logic 211
7.1 Equivalential Calculus 212
7.2 A Simple Example 213
7.3 Imposing Knowledge and Intuition 215
7.4 Answering Open Questions in Equivalential Calculus 217
7.4.1 Finding Useful Notation 219
7.4.2 Suggesting Conjectures 220
7.5 Propositional Calculus 221
7.5.1 Proof by Analogy and Other Techniques for Solving Hard Problems 222
7.5.2 Seeking Shorter Proofs 224
7.5.3 Finding New Axiom Systems 228
7.6 Combinatory Logic 230
7.6.1 Constructing Objects 231
7.6.2 Successes in Combinatory Logic 233
7.7 Using OTTER to Seek Shorter Proofs in Two-Valued Sentential Calculus 234
7.8 Using OTTER to Construct Combinations 242
7.9 Answers to Exercises 248

Chapter 8 The Formal Treatment of Automated Reasoning 255
8.1 First-Order Predicate Calculus 255
8.1.1 The Language of FOPC 256
8.1.2 The Semantics of FOPC 260
8.1.3 The Calculus of FOPC and the Concept of Proof 262
8.2 The Clause Language 265
8.3 Inference Rules 270
8.3.1 Binary Resolution 274
8.3.2 Refutation Completeness 278
8.4 Strategy 285
8.5 Other Inference Rules 291
8.6 Subsumption 298
8.7 Demodulation 299
8.8 Answers to Exercises 301

Chapter 9 Wos's Biased Guide for the Effective Use of OTTER 309
9.1 Overview to Option Selection 312
9.2 Option Choosing 313
9.2.1 Representation 313
9.2.2 Strategy 315
9.2.2.1 Set of Support Strategy 315
9.2.2.2 Weighting 317
9.2.2.3 Level Saturation 320
9.2.2.4 Ratio Strategy 321
9.2.2.5 Tail Strategy and Recursive Tail Strategy 322
9.2.2.6 Resonance Strategy and Resonance-Restriction Strategy 324
9.2.2.7 Subtautology 326
9.2.3 Hot List Strategy and Dynamic Hot List Strategy 326
9.2.4 Inference Rules 327
9.2.4.1 UR-Resolution 327
9.2.4.2 Hyperresolution 328
9.2.4.3 Paramodulation 329
9.2.4.4 Hyperparamodulation 330
9.2.4.5 Binary Resolution 330
9.2.5 Subsumption, for Purging Redundant Information 330
9.2.6 Demodulation, for Simplification and Canonicalization 332
9.2.7 List Usage 334
9.3 An Emphasis on Experimentation 338
9.4 Four Useful Input Files 339
9.4.1 A Puzzle to Solve 339
9.4.2 A Theorem to Prove 342
9.4.3 A Shorter Proof to Find 343

Chapter 10 An Author's Appraisal of His Papers **347**
10.1 On Commutative Prime Power Subgroups of the Norm 349
10.2 The Unit Preference Strategy in Theorem Proving 351
10.3 Efficiency and Completeness of the Set of Support Strategy in Theorem Proving 355
10.4 Automatic Generation of Proofs in Mathematics 364
10.5 The Concept of Demodulation in Theorem Proving 366
10.6 Paramodulation and Set of Support 371
10.7 Axiom Systems in Automatic Theorem Proving 375
10.8 Paramodulation and Theorem-Proving in First-Order Theories with Equality 377
10.9 Maximal Models and Refutation Completeness 386
10.10 A Theorem-Proving Language for Experimentation 388
10.11 Unit Refutations and Horn Sets 388
10.12 Problems and Experiments for and with Automated Theorem-Proving Programs 389
10.13 Complexity and Related Enhancements for Automated Theorem-Proving Programs 391
10.14 Unnatural Attack on the Structure Problem for the Free Jordan Ring on 3 Letters 395
10.15 Automated Generation of Models and Counterexamples 396
10.16 Hyperparamodulation: A Refinement of Paramodulation 399
10.17 Semigroups, Antiautomorphisms, and Involutions 402
10.18 An Automated Reasoning System 404
10.19 Solving Open Questions with an Automated Theorem-Proving Program 405
10.20 Demodulation and Related Tricks 406
10.21 Automated Theorem-Proving 1965–1970 408
10.22 Shortest Single Axioms in Equivalential Calculus 408
10.23 Automated Reasoning: Real Uses and Potential Uses 411
10.24 Equivalential Calculus and Infinite Domains 412
10.25 Open Questions Solved with the Assistance of AURA 413
10.26 The Linked Inference Principle, II: The User's Viewpoint 414
10.27 Achievements in Automated Reasoning 417
10.28 Automated Reasoning Programs: How They Work 417
10.29 Automated Reasoning 418
10.30 What Is Automated Reasoning? 419
10.31 Automating Reasoning 419
10.32 Job-Shop Scheduling Using Automated Reasoning 420
10.33 Negative Paramodulation 421
10.34 Set Theory in First-Order Logic: Clauses for Gödel's Axioms 423
10.35 Resolution, Binary 425
10.36 Finding Sages in Combinatory Logic 426

10.37 Challenge Problems Focusing on Equality and Combinatory Logic . . . 428
10.38 Automated Theorem Proving and Logic Programming: A Natural Symbiosis . . . 431
10.39 Subsumption, A Sometimes Undervalued Procedure . . . 432
10.40 Applications of Automated Reasoning . . . 433
10.41 Meeting the Challenge of Fifty Years of Logic . . . 434
10.42 The Absence and the Presence of Fixed Point Combinators . . . 437
10.43 Automated Reasoning Contributes to Mathematics and Logic . . . 438
10.44 Automated Reasoning and Bledsoe's Dream for the Field . . . 439
10.45 A Logical Basis for the Automation of Reasoning: Case Studies . . . 440
10.46 The Linked Inference Principle, I: The Formal Treatment . . . 441
10.47 The Application of Automated Reasoning to Questions in Mathematics and Logic . . . 443
10.48 Automated Reasoning and Enumerative Search . . . 443
10.49 Benchmark Problems in Which Equality Plays the Major Role . . . 444
10.50 Automated Deduction with Condensed Detachment . . . 445
10.51 Single Axioms for Exponent Groups . . . 447
10.52 The Impossibility of the Automation of Logical Reasoning . . . 449
10.53 The Kernel Strategy and Its Use for the Study of Combinatory Logic . 450
10.54 Automated Reasoning Answers Open Questions . . . 453
10.55 The Field of Automated Reasoning . . . 456
10.56 The Resonance Strategy . . . 456
10.57 Searching for Circles of Pure Proofs . . . 461
10.58 The Power of Combining Resonance with Heat . . . 463
10.59 The Hot List Strategy . . . 466
10.60 OTTER and the Moufang Identity Problem . . . 471
10.61 Automating the Search for Elegant Proofs . . . 472
10.62 Otter: The CADE-13 Competition Incarnations . . . 478
10.63 Programs That Offer Fast, Flawless, Logical Reasoning . . . 480
10.64 Hints and Answers . . . 480
10.64.1 Hints . . . 481
10.64.2 Answers . . . 482

Chapter 11 Open Questions, Hard Problems, Intriguing Challenges 493
11.1 Distinctions, Limits, and Overview . . . 495
11.2 Questions and Problems from Combinatory Logic . . . 497
11.3 Questions and Problems from Mathematics . . . 505
11.3.1 Group Theory . . . 506
11.3.2 Ring Theory . . . 507
11.3.3 Robbins Algebra . . . 508

11.3.4 More Algebra 508
11.4 Questions and Problems from Logic 510
11.4.1 Equivalential Calculus 510
11.4.2 Propositional Logic 514
11.4.3 Many-Valued Sentential Calculus 517

Chapter 12 Epilogue and After-Dinner Liqueur **521**

Appendix A Featuring Input Files, Proofs, and Output File Fragments **523**

References 561

Index 571