HANDBOOK OF LOGIC IN ARTIFICIAL INTELLIGENCE AND LOGIC PROGRAMMING

Volume 2
Deduction Methodologies

Edited by
DOV M. GABBAY
and
C. J. HOGGER
Imperial College of Science, Technology and Medicine
London
and
J. A. ROBINSON
Syracuse University, New York

Volume Co-ordinator
J. SIEKMANN
DFKI/University of Saarland, Saarbrücken
Technische Hochschule Darmstadt
FACHBEREICH INFORMATIK
BIBLIOTHEK
Inventar-Nr.: 0.96-001.27
Sachgebiete: ..............................................
Standort: ...................................................

CLARENDON PRESS • OXFORD
1994
Contents

List of contributors xv

Logical basis for the automation of reasoning:
Case studies 1
Larry Wos and Robert Veroff

1 Introduction 1
2 The clause language paradigm 4
   2.1 Components of the clause language paradigm 4
   2.2 Interplay of the components 6
3 A fragment of the history of automated reasoning from 1960 to 1990 8
4 Answering open questions 11
   4.1 Equivalential calculus 12
   4.2 Combinatory logic 14
   4.3 Sentential calculus 20
5 Relation to logic programming and to person-oriented reasoning 21
   5.1 Logic programming 21
   5.2 Person-oriented reasoning 24
6 Challenge problems for testing, comparing, and evaluating 25
7 Current state of the art: Basic research problems to solve 30
8 Programs, books, and a problem database 32
9 Summary and the future 35

Unification theory 41
Franz Baader and Jörg Siekmann

1 Outline 41
2 What is E-unification? 42
3 Unification in the empty theory 47
   3.1 An informal description of Robinson’s algorithm 48
   3.2 Motivation for using Ø-unification 49
   3.3 Efficient algorithms for Ø-unification 51
4 Unification in non-empty theories 56
   4.1 Motivation for using E-unification 58
   4.2 Single equations versus systems of equations 59
   4.3 A closer look at the signature 61
   4.4 Restricted versus unrestricted instantiation ordering 62
Contents

4.5 Notions and notation revisited 63
5 Unification results for special theories 65
6 General results 71
   6.1 General $E$-unification 72
   6.2 Semantic approaches to unification 80
   6.3 The combination problem 85
7 Related areas of research 91
   7.1 Higher-order unification 91
   7.2 Unification in sort theories 96
   7.3 Constraint solving 99
8 Applications of unification 101

Mathematical induction 127
Christoph Walther

1 Introduction 128
2 Admissible specifications and theories 132
   2.1 Data structures and algorithms 132
   2.2 Theorem proving 136
   2.3 Induction 138
   2.4 Expanded theories 141
   2.5 Incomplete specifications 142
   2.6 Summary 145
3 Computing induction axioms 147
   3.1 Defining induction axioms 147
   3.2 The induction heuristic 154
   3.3 Modification of induction axioms by domain
       generalization 155
   3.4 Comparing induction axioms 161
   3.5 Modification of induction axioms by range
       generalization 168
   3.6 Modification of induction axioms by separation 172
   3.7 Summary 175
4 Proving well-foundedness 177
   4.1 Termination of algorithms 177
   4.2 Proving termination by induction lemmata 180
   4.3 Proving termination with argument bounded
       algorithms 185
   4.4 Computing well-founded domain and
       range generalizations 192
   4.5 Summary 196
5 Proving induction formulas 197
   5.1 The basic theorem prover 197
   5.2 Symbolic evaluation 200
Higher order logic

Daniel Leivant

1 Introduction 230
2 The expressive power of second order Logic 231
   2.1 The language of second order logic 231
   2.2 Expressing size 232
   2.3 Defining data types 234
   2.4 Describing processes 236
   2.5 Expressing convergence using second order validity 237
   2.6 Truth definitions: the analytical hierarchy 238
   2.7 Inductive definitions 241
3 Canonical semantics of higher order logic 243
   3.1 Tarskian semantics of second order logic 243
   3.2 Function and relation formulations 243
   3.3 Normal forms 244
   3.4 Finite order logic 245
   3.5 Functional types 246
   3.6 Formulas as higher order functions 247
   3.7 Truth definitions revisited 248
4 Proof theory 250
   4.1 Basic formalisms 250
   4.2 Additional set existence principles 252
   4.3 Constructive finite order logics 254
   4.4 Normalization and the subformula property 255
5 Ontology 257
   5.1 The gulf between first order and second order logic 257
   5.2 Lindström’s and Quine’s tests 259
   5.3 Slipping from first to second order logic 261
5.4 Higher order logic as mathematics: Henkin's semantics
5.5 Henkin completeness for full finite order logic
5.6 Finite order logic as a second order theory

6 Restricted higher order logic
6.1 Restricted expressiveness 1: Monadic second order logic
6.2 Restricted expressiveness 2: Fixpoint logics
6.3 Restricted semantics: Weak second order logic
6.4 Predicative logic: Restricted comprehension

7 Mathematical practice
7.1 Second order axioms vs. first order schemas
7.2 Higher order aspects of set theory: from higher order to first order and back
7.3 Analysis and reductive proof theory
7.4 Speed-up

8 Higher order logic in relation to computing and programming
8.1 Higher order data and types
8.2 The computational nature of higher order natural deduction
8.3 Higher order logic in the meta-theory of formal systems
8.4 Higher order logic and computational complexity

Meta-languages, reflection principles, and self-reference
Donald Perlis and V. S. Subrahmanian

1 Introduction
1.1 Aims
1.2 Aboutness and self-reference
1.3 Truth and paradoxes
1.4 Ideal, pseudo-ideal and situated reasoning

2 Formal definitions and difficulties
2.1 Meta-languages and Tarski hierarchies
2.2 Reflection principles
2.3 Fixed points

3 Applications
3.1 Ideal and pseudo-ideal approaches (to knowledge)
3.2 Situated temporal reasoning and self-reference
3.3 Logic programming
3.4 Naive vs. amalgamated LP

4 Future developments
## Contents

4.1 Agent referential semantics 353

5 Conclusions 355

### Classical vs non-classical logics (the universality of classical logic)

D. M. Gabbay

1 Introduction—the debate 359

2 What is a logical system?—the challenge 366

2.1 Logical systems as consequence relations 368

2.2 Logical systems as algorithmic proof systems 371

2.3 Logical systems as algorithmic structured consequence relations 372

2.4 Logical systems as labelled deductive systems 374

2.5 Aggregated systems 382

2.6 Practical reasoning systems 384

3 How to construct a logic for an application: the case studies 385

3.1 Case study 1: temporal logic in two-sorted classical logic 388

3.2 Case study 2: priority logic and PROLOG 395

4 Algebraic LDS: a unifying solution 403

5 Reductions to classical logic: the options 427

6 Translations into classical logic: technical case study 436

7 Linked predicate languages 443

7 Linked predicate languages: classical logic as a target for translation 443

8 The meta-language HFP: computational classical logic 451

9 Semi-algebraic semantics for propositional logics 465

10 An automated universal translator into classical logic 478

10.1 The translation steps 479

10.2 The SCAN algorithm 483

11 Conclusion: the current state of the debate 495

Index 501