
The aim of the book is to fill a gap in the literature which exists between introductory books on mathematical logic and structural proof theory on the one hand, and, on the other hand, monographs on generalized Hilbert’s program, which can be loosely characterized as theory of ordinal notations. The text concentrates on so-called structural proof theory of first-order, but a short excurse into first-order arithmetic and second-order logic is also provided. The contents are listed below.

Chapter 1. Introduction. 1. Basic definitions and notations of first-order predicate minimal, intuitionistic and classical logic. 2. Simple type theories: typed combinatory logic and typed lambda calculus. The correlated “simple” proofs of Church–Rosser property and (strong) normalizability. 3. Gentzen’s natural deductions and sequent calculi, and Hilbert-style modus ponens calculi for propositional logic. The correlated formulae-as-types relationships with typed lambda calculus and combinatorial calculus.

Chapter 2. N-systems and H-systems. 1–2. Natural deduction (N-) systems for predicate logic in Gentzen’s original and in sequent-style form and as typed term calculi. 3. The negative translations show relative equipollence of minimal, intuitionistic and classical logic. 4. Hilbert-style (H-) systems for predicate logic and their equivalence (modulo provability) to the corresponding N-systems.

Chapter 3. Gentzen systems. 1–2. Classical and intuitionistic G-systems (sequent calculi), and the cut rule. 3. The equivalence between G-systems with Cut and N-systems. 4. Structural rules and invertibility. 5. One-sided variants (sequents as strings, sets or multisets of formulas) for classical G-systems are discussed.

Chapter 4. Cut elimination with applications. 1. Dragalin’s variant of Gentzen’s cut elimination proof. 2. Applications: weak subformula/separating property, weak disjunction property, explicit definability, decidability of the intuitionistic propositional logic, weak (prenex) version of Herbrand’s theorem. 3. Interpolation and definability, axiomatic treatment of function symbols and equality. 4. Primitive recursive arithmetic and intuitionistic E-logic. 5. Hudelmaier’s intuitionistic G-system and Dyckhoff’s semi-constructive equivalence proof.

Elsevier Science B.V.

PII S0167-6423(98)00002-1
Chapter 5. Refinements. 1. Hyper(super)exponential upper bounds on deductive depth for predicate cut elimination (the proof is inaccurate due to lacking contraction). Brief exposition of Felscher–Gordeev–Hudelmaier propositional cut elimination via inversion, and the resulting elementary upper bounds on deductive depth. 2. Rule permutation properties.


Chapter 7. Resolution. 1–2. Resolution proofs for Horn clauses and unification theorem. 3. Unrestricted linear resolution for Horn clauses. 4. Propositional cutfree G-deductions as general resolution proofs of clause formulas with new propositional variables. 5. Specifications to intuitionistic G-systems.


Chapter 10. Proof theory of arithmetic. 1. Ordinal notations below epsilon 0. 2. Provability of initial cases of TI (transfinite induction) for ordinal notations below epsilon 0 in PA (Peano Arithmetic) (Gentzen's proof). 3. Normalization for the N-system with the omega rule. 4. Schuette's proof that TI for ordinal notations below epsilon 0 is underivable in PA. 5. Kreisel's counterexamples show that ordinal notations are not ordinals.

Chapter 11. Second-order logic. 1. Intuitionistic N-system. Normal deductions. 2.Typed polymorphic lambda calculus. 3. Girard's strong normalization proof. 4–5. Provably recursive functions of second-order intuitionistic and/or classical PA are representable in polymorphic lambda calculus.

The book is carefully organized, and it provides a compact overview of various trends in "descriptive" proof theory. Chapters dealing with typed term calculi, natural deductions and linear logic are excellent. However, there are numerous misprints and notational confusions, and the list of symbols and notations (pp. 331–334) is incomplete. In this regard, the reader will be well advised to consult http://turing.wins.uva.nl/~anne/ for current corrections. Traditional proof theoretical stuff (sequent calculi proper and cut elimination techniques) and computational complexity aspects are underrepresented, and the choice of G-systems and their names needs more justification. Actually, it would suffice to work with only two "canonical" G-systems: (1) GS3 (p. 72), together with the corresponding one-sided cumulative (additive) cut rule, for classical logic, and (2) G4ip (p. 102) extended by the four quantification rules of G3i (p. 65), together with the additive cut rule (p. 58), for the
intuitionistic logic. Among other familiar G-systems, these calculi (first used by Rasiowa and Hudelmaier, respectively) have best proof search and shortest average proofs, and their cut elimination proofs (via inversion, in propositional case) provide optimal speed-ups. The analogous (classical) one-sided cumulative variant of G3s (p. 229) would also simplify the treatment of the modal logic S4.

H-systems and their connections with N-systems and G-systems with Cut are underrepresented (the correlated embeddings are all weight-polynomial; the estimate 3.3.4B is too loose), given that these systems played fundamental role in earlier proof theoretical research by the scholars of Hilbert and Bernays, Russell and Whitehead, Lukasiewicz and Tarski. Equational calculi aren’t exposed; these systems are essential for Tarski’s proof theoretical approach to predicate logic via relation algebras. In contrast, N-systems play crucial role in the text. This might be justified by an obvious simplicity of Prawitz’s normalization theory on the one hand, and important links to typed combinational/lambda term calculi, on the other hand. However, more general theory of cut elimination in G-systems justifies Gentzen’s original choice of the sequent calculus approach. Indeed, by Hudelmaier’s remark (cf. 6.8.3, p. 170), there are more, and better, cut elimination options in G-systems than normalization options in the corresponding N-systems. That is, normalization is merely a special case of cut elimination. Therefore, strong cut elimination theorems stating that all sequences of standard cut reductions terminate turn out to be substantially stronger than strong normalization theorems in the corresponding N-systems. This applies to all formal systems used in the book. In fact, strong cut elimination holds for both intuitionistic and classical familiar G-systems corresponding to strongly normalizable N-systems of the same proof theoretical strength. By the reviewer’s result, it still holds for Quine’s New Foundation without Extensionality, possibly extended by consistent well found-ness axioms. Analogous (weaker) results about strong normalization in the corresponding N-systems were obtained by Crabbe. Surely, strong cut elimination proofs require more careful formalizations of basic G-systems and the correlated cut reduction operators, which would certainly enlarge the book’s volume. However, many important results hang on the operational approach to cut elimination (initiated independently by Belnap and Mints); so are e.g. conservation theorems for intuitionistic applicative theories and Friedman’s and Martin–Loef’s constructive set theories [Gordeev 1988] on the one hand, and Gore’s recent cutfree formalization of Tarski’s relation algebra, on the other hand.

Chapter 10 may serve as an introduction to general theory of ordinal notations (along the lines of generalized Hilbert’s programme). Proofs in classical and/or intuitionistic PA are interpreted as infinite primitive recursive normal derivations in N-systems with the infinite-branching omega rule. For obvious reasons, computational contents of these derivations aren’t immediately clear. It is possible to avoid omega rule by adding more sophisticated finite-branching rules for number theoretical induction.

Turning to the bottomline, somewhat surprising is the lack of exhaustive references in the text to the earlier pre-Gentzen proof theoretical work [Herbrand 1928]. In fact,
Herbrand initiated a more general proof theory – with a kind of cut elimination combined with normal form expansion. Contrary to Gentzen’s rules of inference reducing only maximal formulas (occurring in sequences), Herbrand’s rules allow to reduce arbitrary subformulas without changing the environment. According to the reviewer’s research, this advantage leads to a more general “nested” proof theory that continue to hold in finite-variable logic, which fails in Gentzen’s theory.

Lev Gordeev


Elementary algorithms are certainly an important part in a computer-science undergraduate curriculum. Traditionally, elementary algorithms were a component of a first or second year course together with data structures. It seems that nowadays the trend is to separate these two components. A first-year course introduces students to elementary data structures: arrays, strings, stacks, queues, linked lists and a few basic algorithms associated with them. The body of standard techniques which provide solutions to frequently occurring problems are now moved to a second-year course: this introduces searching, sorting, tree and graph algorithms, possibly integrated by some notions of dynamic programming, pattern matching and text compression.

This textbook is explicitly designed for such a second-year course. Elementary programming, recursion and the use of elementary data structures are prerequisite of the book, as well as some basic notions on discrete mathematics and logics (however, some of the needed notions are contained in an appendix). After a first, introductory, chapter on the notion of algorithm and recursive programming techniques, the book provides first the essentials: there are chapters on searching, sorting, growth and maintenance of binary trees and elementary graph algorithms. The last chapters contain some material on additional, more advanced, topics which include text compression, dynamic programming and random numbers. Three appendices on mathematical background, theorems about trees and finite-state automata conclude.

The book is well written and clear. Algorithms are first presented informally by explaining their logic in narrative prose. Afterward a Pascal implementation is provided, usually following the code with further comments. In general, we find this style of presentation very helpful for students since, as the author mentions in the preface, in general one learns by iterative refinement. This style is applied also to mathematical notions which are often defined twice: first informally and then in a more precise way. In some cases, however, the author consciously sacrifices rigor to clarity. This choice is debatable, since, to our opinion, some notions would deserve a more rigorous treatment also in a textbook for non-advanced courses. This is the case, for example, for the complexity analysis of algorithms which in many cases is performed informally, by using the shortcuts outlined in the first chapter for the big-O