STRUCTURAL PROOF THEORY

Structural proof theory is a branch of logic that studies the general structure and properties of logical and mathematical proofs. This book is both a concise introduction to the central results and methods of structural proof theory and a work of research that will be of interest to specialists. The book is designed to be used by students of philosophy, mathematics, and computer science.

The book contains a wealth of new results on proof-theoretical systems, including extensions of such systems from logic to mathematics and on the connection between the two main forms of structural proof theory – natural deduction and sequent calculus. The authors emphasize the computational content of logical results.

A special feature of the volume is a computerized system for developing proofs interactively, downloadable from the web and regularly updated.

Sara Negri is Docent of Logic at the University of Helsinki.

Jan von Plato is Professor of Philosophy at the University of Helsinki and author of the successful *Creating Modern Probability* (Cambridge, 1994).

STRUCTURAL PROOF THEORY

SARA NEGRI

JAN VON PLATO

With an Appendix by Aarne Ranta

CAMBRIDGE
UNIVERSITY PRESS

CAMBRIDGE UNIVERSITY PRESS
Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo

Cambridge University Press
The Edinburgh Building, Cambridge CB2 8RU, UK

Published in the United States of America by Cambridge University Press, New York

www.cambridge.org
Information on this title: www.cambridge.org/9780521793070

First published 2001
This digitally printed version 2008

A catalogue record for this publication is available from the British Library

Library of Congress Cataloguing in Publication data

Negri, Sara, 1967–
Structural proof theory / Sara Negri, Jan von Plato.
p. cm.
Includes bibliographical references.
ISBN 0-521-79307-6
1. Proof theory. I. von Plato, Jan. II. Title.
QA9.54. N44 2001
511.3 – dc21 00-040327

ISBN 978-0-521-79307-0 hardback
ISBN 978-0-521-06842-0 paperback

Contents

Preface

This book grew out of our fascination with the contraction-free sequent calculi. The first part, Chapters 1 to 4, is an introduction to intuitionistic and classical predicate logic as based on such calculi. The second part, Chapters 5 to 8, mainly presents work of our own that exploits the control over proofs made possible by the contraction-free calculi.

The first of the authors got her initial training in structural proof theory in a course given by Prof. Anne Troelstra in Amsterdam in 1992. The second author studied logic in the seventies, when by surprise Dag Prawitz mailed a copy of his book *Natural Deduction*. We thank them both for these intellectual stimuli, brought to fruition by the second author with a considerable delay. Since 1997, collaboration of the first author with Roy Dyckhoff has led us to the forefront of research in sequent calculi.

Dirk van Dalen, Roy Dyckhoff, and Glenn Shafer read all or most of a first version of the text. Other colleagues have commented on the manuscript or papers and talks on which part of the book builds, including Felix Joachimski, Petri Mäenpää, Per Martin-Löf, Ralph Matthes, Grigori Mints, Enrico Moriconi, Dag Prawitz, Anne Troelstra, Sergei Tupailo, René Vestergaard, and students from our courses, Raul Hakli in particular; we thank them all.

Aarne Ranta joined our book project in the spring of 1999, implementing in a short time a proof editor for sequent calculus. He also wrote an appendix describing the proof editor.

Little Daniel was with us from the very first day when the writing began. To him this book is dedicated.

Sara Negri
Jan von Plato

Introduction

STRUCTURAL PROOF THEORY

The idea of mathematical proof is very old, even if precise principles of proof have been laid down during only the past hundred years or so. Proof theory was first based on axiomatic systems with just one or two rules of inference. Such systems can be useful as formal representations of what is provable, but the actual finding of proofs in axiomatic systems is next to impossible. A proof begins with instances of the axioms, but there is no systematic way of finding out what these instances should be. Axiomatic proof theory was initiated by David Hilbert, whose aim was to use it in the study of the consistency, mutual independence, and completeness of axiomatic systems of mathematics.

Structural proof theory studies the general structure and properties of mathematical proofs. It was discovered by Gerhard Gentzen (1909–1945) in the first years of the 1930s and presented in his doctoral thesis *Untersuchungen über das logische Schliessen* in 1933. In his thesis, Gentzen gives the two main formulations of systems of logical rules, **natural deduction** and **sequent calculus**. The first aims at a close correspondence with the way theorems are proved in practice; the latter was the formulation through which Gentzen found his main result, often referred to as Gentzen's "Hauptsatz." It says that proofs can be transformed into a certain "cut-free" form, and from this form general conclusions about proofs can be made, such as the consistency of the system of rules.

The years when Gentzen began his researches were marked by one great but puzzling discovery, Gödel's incompleteness theorem for arithmetic in 1931: Known principles of proof are not sufficient for deriving all of arithmetic; moreover, no single system of axioms and rules can be sufficient. Gentzen's studies of the proof theory of arithmetic led to **ordinal proof theory**, the general task of which is to study the deductive strength of formal systems containing infinitistic principles of proof. This is a part of proof theory we shall not discuss.

Of the two forms of structural proof theory that Gentzen gave in his doctoral thesis, natural deduction has remained remarkably stable in its treatment of rules

of proof. Sequent calculus, instead, has been developed in various directions. One line leads from Gentzen through Ketonen, Kleene, Dragalin, and Troelstra to what are known as **contraction-free** systems of sequent calculus. Each of these logicians added some essential discovery, until a gem emerged. What it is can be only intimated at this stage: There is a way of organizing the principles of proof so that one can start from the theorem to be proved, then analyze it into simpler parts in a guided way. The gem is this "guided way"; namely, if one lays down what the last rule of inference was, the premisses of that last step are uniquely determined. Next, one goes on analyzing these premisses, and so on. Gentzen's basic discovery is reformulated as stating that a proof can be so organized that the premisses of each step of inference are always simpler than its conclusion. (To be more accurate, it can also happen that the premisses are not more complicated than the conclusion.)

Given a purported theorem, the question is whether it is provable or unprovable. In the first case, the task is to find a proof. In the second case, the task is to show that no proof can exist. How can we, then, prove unprovability? The possibility of such proofs depends crucially on having the right kind of calculus, and these proofs can take various forms: In the simplest cases we go through all the rules and find that none of them has a conclusion of the form of the claimed theorem. For certain classes of theorems, we can show that it makes no difference in what order we analyze the theorem to be proved. Each step of analysis leads to simpler premisses and therefore the process stops. From the way it stops we can decide if the conclusion really is a theorem or not. In other cases, it can happen that the premisses are at least as complicated as the conclusion, and we could go on indefinitely trying to find a proof. Some ingenious discovery is usually needed to prove unprovability, say, some analyses stop without giving a proof, and we are able to show that all of the remaining alternatives never stop and thus never give a proof.

One line of division in proof theory concerns the methods used in the study of the structure of proofs. In his original proof-theoretical program, Hilbert aimed at an "absolutely reliable" proof of consistency for formalized mathematics. The methods he thought acceptable had to be finitary, but the goal was shown to be unattainable already for arithmetic by Gödel's results. Later, parts of proof theory remained **reductive**, using different constructive principles, whereas other parts have studied proofs by unrestricted means. Most of our methods can be classified as reductive, but the reasons for restricted methods do not depend on arguments such as reliability. It is rather that we want results about systems of formal proof to have a computational significance. Thus it would not be sufficient to show by unrestricted means the mere existence of proofs with some desirable property. Instead, a constructive method for actually transforming any given proof so that it has the property is sought. From this point of view, our treatment of

structural proof theory belongs, with a few exceptions, to what can be described as **computational proof theory**.

Since 1970, a branch of proof theory known as **constructive type theory** has been developed. A theorem typically states that a certain claim holds under given assumptions. The basic idea of type theory is that proofs are **functions** that convert any proofs of the assumptions of a theorem into a proof of its claim. A connection to computer science is established: In the latter, formal languages have been developed for constructing functions (programs) that act in a specified way on their input, but there has been no formal language for expressing what this specified way, the **program specification**, is. Logical languages, in turn, are suitable for expressing such specifications, but they have totally lacked a formalism for constructing functions that effect the task expressed by the specification. Constructive type theory unites specification language and programming language in a unified formalism in which the task of verifying the correctness of a program is the **same** as the logical task of controlling the correctness of a formal proof. We do not cover constructive type theory in detail, as another book would be needed for that, but some of the basic ideas and their connection to natural deduction and normalization procedures are explained in Appendix B.

At present, there are many projects in the territory between logic, mathematics, and computer science that aim at fully formalizing mathematical proofs. These projects use computer implementations of **proof editors** for the interactive development of formal proofs, and it cannot be said what all the things are that could come out of such projects. It has been observed that even the most detailed informal proofs easily contain gaps and cannot be routinely completed into formal proofs. More importantly, one finds imprecision in the conceptual foundation. The most optimistic researchers find that formalized proofs will become the standard in mathematics some day, but experience has shown formalization beyond the obvious results to be time-consuming. At present, proof editors are still far from being practical tools for the mathematician. If they gain importance in mathematics, it will be due to a change in emphasis through the development of computer science and through the interest in the computational content of mathematical theories. On the other hand, proof editors have been used for program verification even with industrial applications for some years by now. Such applications are bound to increase through the critical importance of program correctness.

Gentzen's structural proof theory has achieved perfection in pure logic, the predicate calculus. Intuitionistic natural deduction and classical sequent calculus are by now mastered, but the extension of this mastery beyond pure logic has been limited. A new approach that we exploit is to formulate mathematical theories as systems of **nonlogical rules** added to a suitable sequent calculus. As examples of proof analyses, theories of order, lattice theory, and plane affine geometry are treated. These examples indicate a way to an emerging field of study that could

be called **proof theory in mathematics**. It is interesting to note that a large part of abstract mathematical reasoning seems to be finitary, thus not requiring any strong transfinite methods in proof analysis.

In the rest of this Introduction we comment on use of this book in teaching proof theory and what is new in it.

USE OF THIS BOOK IN TEACHING

Chapters 1–4 are based on courses in proof theory we have given at the University of Helsinki. The main objective of these courses was to give to the students a concise introduction to contraction-free intuitionistic and classical sequent calculi. The first author has also given a more specialized course on natural deduction, based on Chapter 1, the first two sections of Chapter 5, Chapter 8, and Appendices A and B.

The presentation is self-contained and the book should be readable without any previous knowledge of logic. Some familiarity with the topic, as in Van Dalen's *Logic and Structure*, will make the task less demanding.

Chapter 1 starts with general observations about logical languages and rules of inference. In a first version, we had defined logical languages through categorial grammars, but this was judged too difficult by most colleagues who read the text. With some reluctance, the categorial grammar approach was moved to Appendix A. Some traces of the definition of logical languages through an abstract syntax remained in the first section of Chapter 1, though.

The introduction rules of natural deduction are explained through the computational semantics of intuitionistic logic. A generalization of the inversion principle, to the effect that "whatever follows from the direct grounds for deriving a proposition, must follow from that proposition," determines the corresponding elimination rules. By the inversion principle, three rules, those of conjunction elimination, implication elimination, and universal elimination, obtain a form more general than the standard natural deduction rules. Using these general elimination rules, we are able to introduce sequent calculus rules as formalizations of the derivability relation in natural deduction. Contraction-free intuitionistic and classical sequent calculi are treated in detail in Chapters 2 and 3. These chapters work as a concise introduction to the central methods and general results of structural proof theory. The basic parts of structural proof theory use combinatorial reasoning and elementary induction on formula length, height of derivation, and so on, therefore perhaps giving an impression of easiness on the newcomer. The main difficulty, witnessed by the long development of structural proof theory, is to find the right rules. The first part of our text shows in what order structural proof theory is built up once those rules have been found. The second part of the

book, Chapters 5–8, gives ample further illustration of the methodology. There is usually a large number of details, and a delicate order is required for putting things together, and mistakes happen. For such reasons, our first cut elimination theorem, in Chapter 2, considers, to our knowledge, absolutely all cases, even at the expense of perhaps being a bit pedantic.

In Chapter 3, following a suggestion of Gentzen, multisuccedent sequents are presented as a natural generalization of single succedent sequents into sequents with several (classical) open cases in the succedent. We find it important for students to avoid the denotational reading of sequents in favor of one in terms of formal proofs.

Chapter 4 contains a systematic treatment of quantifier rules in sequent calculi, again introduced through natural deduction and the general inversion principle.

Connected to the book is an **interactive proof editor** for developing formal derivations in sequent calculi. The system has been implemented by Aarne Ranta in the functional programming language Haskell. A description of the system, with instructions on how to access and use it, is given in Appendix C written by Ranta.

The proof editor serves several purposes: First, it makes the development of formal derivations in sequent calculi less tedious, thereby helping the student. It also checks the formal correctness of derivations. The user can give axiomatic systems to the editor that converts them into systems of nonlogical rules of inference by which the logical sequent calculi are extended. Formal derivations are quite feasible to develop in those extensions we have so far studied. Even though the extensions need not permit a terminating proof search, the user will soon notice how neatly the search space can become limited, often into one or two applicable rules only, or no rules at all, which establishes underivability.

The proof editor produces provably correct LaTeX code, with the advantage that the rewriting of parts of sequents is not needed. The editor is in its early stages; more is hoped to be included in later releases, including translation algorithms between various calculi, cut elimination algorithms, a natural language interface, and so on.

Exercises, mostly to Chapters 1–4 and 8, can be found in the book's home page (see p. 243). We welcome suggestions for further exercises. Basic exercises are just derivations of formulas in the various calculi. Other exercises consist in filling in details of proofs of theorems. Another type of task, for those conversant with the Haskell language, is to formalize theorems about sequent calculi. Since these theorems are, almost without exception, proved constructively in the book, their formalizations give proof-theoretical algorithms for the transformation of proofs. An example is the proof of Glivenko's theorem in Section 5.4.

Through the use of contraction-free sequent calculi, it is possible for students to find proofs of results that were published as research results only a few decades

ago, say, Harrop's theorem in Section 2.5. This should give some idea of what a powerful tool is being put into their hands.

Finally, a description of what is new in this book (for the expert, mostly).

WHAT IS NEW IN THIS BOOK?

Chapter 1 contains a generalization of the inversion principle of Gentzen and Prawitz, one that leads to elimination rules that are more general than the usual ones. Contrary to earlier inversion principles that only justify the elimination rules, our principle actually determines what the elimination rules corresponding to given introduction rules should be. The elimination rules are all of the form of disjunction elimination, with an arbitrary consequence.

Starting from natural deduction with general elimination rules, the rules of sequent calculus are presented in Section 1.3 as straightforward formalizations of the derivability relation of natural deduction.

Section 3.3 gives a proof of completeness of the contraction-free invertible sequent calculus for classical propositional logic known as **G3c**-calculus in the literature. The proof is an elaboration of Ketonen's original 1944 proof. It uses a novel notion of validity defined as a negative concept, the inexistence of a refuting valuation.

Chapter 4 contains proofs of height-preserving α-conversion and the substitution lemma that we have not found done elsewhere in such detail. Section 4.4 gives a proof of completeness of classical predicate calculus worked out for the definition of validity as a negative notion.

Chapter 5 studies various sequent calculi, most of which are new to the literature. Section 5.1 gives a sequent calculus with independent contexts in all two-premiss rules and explicit rules of weakening and contraction. The calculus is motivated by the independent treatment of assumptions in natural deduction. A classical multisuccedent version is also given. Proofs of cut elimination for these calculi are given that do not use Gentzen's mix rule, or rule of multicut. In Section 5.2, the calculi of Section 5.1 are modified so that weakening and contraction are treated implicitly as in natural deduction. Section 5.4 gives a single succedent calculus for classical propositional logic based on a formulation of the law of excluded middle as a sequent calculus rule. A proof of Glivenko's theorem is given through an explicit proof transformation. It is shown that in the derivation of a sequent $\Gamma \Rightarrow C$, the rule can be restricted to atoms of C, from which a full subformula property follows.

Chapter 6 studies extensions of logical sequent calculi by nonlogical rules corresponding to axioms. Contrary to widespread belief, it is possible to add axioms to sequent calculus as rules of a suitable form while maintaining the eliminability of cut. These extensions have no structural rules, which gives a

strong control over the structure of possible derivations. As a first application, a formulation of predicate calculus with equality as a cut-free system of rules is given. It is proved through an explicit proof transformation that predicate logic with equality is conservative over predicate logic. It is essential for the proof that no cuts, even on atoms, be permitted. In Section 6.6, examples of structural proof analysis in mathematics are given. Topics covered include intuitionistic and classical theories of order, lattice theory, and affine geometry. The last one goes beyond the expressive means of first-order logic, but the methods of structural proof analysis of the previous chapters still apply. As an example of such analysis of derivations without structural rules, a proof of independence of Euclid's fifth postulate in plane affine geometry is given.

In Chapter 8 the structure of derivations in natural deduction with general elimination rules is studied. A uniform definition of normality of derivations is given: A derivation is normal when all major premisses of elimination rules are assumptions. This structure follows from the applicability of permutation conversions to all cases in which the major premiss of an elimination rule is concluded by another elimination rule. Translations are given that establish isomorphism of normal derivations and cut-free derivations in the sequent calculus with independent contexts of Chapter 5. It is shown what the role of the structural rules of sequent calculus is in terms of natural deduction: Weakening and contraction in sequent calculus correspond to the vacuous and multiple discharge of assumptions in natural deduction. Cuts in which the cut formula is principal in the right premiss correspond to such steps of elimination in which the major premiss has been derived. (No other cuts can be expressed in terms of natural deduction.) A translation from non-normal derivations to derivations with cuts is given, from which follows a normalization procedure consisting of said translation, followed by cut elimination and translation back to natural deduction.

In the Conclusion, a uniform logical calculus is given that encompasses both sequent calculus and natural deduction.