Thesis Outline: Tools and Techniques for Formalising Structural Proof Theory*

Peter Chapman

Abstract

Whilst results from Structural Proof Theory can be couched in many formalisms, it is the sequent calculus which is the most amenable of the formalisms to metamathematical treatment. As such, we present some tools and techniques so that formalisations of results in sequent calculi can proceed more smoothly. The proof assistant *Isabelle* is used, along with the package for reasoning about first-order languages, *Nominal Isabelle*. We also present a new framework within which one can analyse a wide range of sequent calculi. General conditions can thus to be made upon the rules of such a calculus to ensure certain properties hold, notably invertibility of the rules of the calculus.

1 Introduction

We give a more detailed overview of the aims and objectives of the research. There is a discussion of why the problem is important, defines some terms and gives a further overview of the rest of the thesis.

1.1 Motivations

A brief introduction to the field, along with a discussion of why it is an important area for study, with particular regard to the need for formalisation.

1.2 Aims and Objectives of the Thesis

A discussion on what we hope to achieve. In particular, providing tools and techniques so that Structural Proof Theory, and specifically sequent calculi, results can be formalised in a human-readable way. Some of this endeavour will

*Where references in chapters are given, they refer to work which has already been written, albeit not in thesis-ready form. Some may be published papers, others unpublished research reports. The References contain works which will be referenced in the thesis.

be demonstrated with case-studies of formalisations, and some will be theoretical results. The theoretical results reduce the burden on the user in future formalisations.

1.3 Structure of the Thesis

A plan of the rest of the document.

2 Structural Proof Theory

This section gives an overview of the relevant pieces of proof theory one might like to formalise. It also describes various "shortcuts" which could increase the level of automation possible in formalising proof theory results. There is also a discussion on inductive measures.

2.1 Sequent Calculi

We discuss the main results of the sequent calculus formalism, with particular emphasis on the main result of Cut Admissibility. Different methods for proving the admissibility of Cut will be explored. The wide variety of inductive measures used for such proofs will be investigated.

2.2 Cut Admissibility

An overview of the various attempts to provide sufficient conditions for Cut admissibility.

3 Formalising Mathematics

This section gives an introduction to various proof assistants, including ones which we have used. The applicability of each to formalising structural proof theory will be discussed, as will any attempted formalisations.

3.1 Proof Assistants

The main interactive theorem provers used, including *Isabelle*, EPIGRAM and *Coq*. *Isabelle* has been used for the main body of the thesis; this decision is justified in this section.

3.2 Formalisations of Structural Proof Theory

A discussion of seminal papers in the field, which use a variety of proof assistants.

4 Formalisations using terms

This is a tentative chapter. It highlights the more traditional way of formalising proofs in sequent calculi by use of a term notation. Whilst it is easier to formalise in this manner, the proofs become much less readable and transparent. The non-trivial result will be a translation of Cut elimination steps to normalisations in natural deduction.

4.1 The Background to the Problem

The motivation of the problem, along with the definitions of the sequent calculi and natural deduction system which are used.

4.2 Formalising the Translation

The translation between the two systems is given, and formalised. Some basic results are proved about the translation.

4.3 Strong Normalisation

A reduction system is given for the term calculus. It is shown to be strongly normalising using a lexicographic path order argument.

4.4 Confluence

The reduction rules contain no critical pairs. From this, one can derive local confluence, and thus confluence, of the rewrite system.

5 Formalising Cut and Craig

We present some case studies, showing the use of *Isabelle* and its extension *Nominal Isabelle* to formalise some non-trivial results. The interesting parts, as well as the deficiencies, of the formalisations are highlighted. This chapter will be based upon [8][1],[5],[4][2]. These two papers contain around 75% of the material required for the chapter.

5.1 How to Formalise a Sequent Calculus

A generic pattern which can be used to formalise different sequent calculi is explained. In particular, we give a framework so that derivations can be easily represented. The framework can be extended so that derivations carry information about depth.

[1]This was a published paper, and had a page limit of 15 pages.
[2]These constitute about 20 pages in total.

5.2 Craig's Interpolation Theorem

An account of how we formalised this result in *Nominal Isabelle*, and why this endeavour was undertaken.

5.3 Cut Admissibility

A description of various Cut admissibility proofs is given, along with a discussion on how to best structure such proofs; two formal proofs of Cut Admissibility for the same calculus can differ greatly in their lengths.

6 Classifying Sequent Calculi

This chapter underpins §7. We motivate the various definitions we use, and discuss the range of calculi which can be analysed in this way. The whole chapter is based upon [7][3]. This paper contains at least 90% of the material needed for this chapter.

6.1 Formula and Metaformulae, Rules and Inferences

Precise descriptions of these hitherto almost interchangeable terms are given, along with a raft of examples for illustration.

6.2 Active, Passive, Principal and Prime

Definitions of these terms, with comparison to similar notions in the literature. Again, many examples are used.

6.3 Decompositions: Normal and Implicit Weakening

The definition of a decomposable rule is given, along with the two flavours in which they appear. Most structural rules are not decomposable; a discussion of why this is not as harmful as it may at first appear is included.

7 Permutability and Invertibility

Some work is presented which codifies sequent calculi so that one can guarantee invertibility of the rules. The range of calculi which one can analyse using the presented methods are discussed. There is also some discussion as to the faithfulness of the formalisation. As with §6, this chapter is based upon [7], although the work on permutability has not been written yet. As such, it represents around 50% of the material needed for the chapter. In particular, §7.4.1 still requires some significant work.

[3]This paper was accepted for a conference in New Zealand in January 2009, but owing to funding restrictions, it had to be withdrawn. It is around 30 pages long.

7.1 Definition of the Problem

We describe how the problem arises, and why it is important. An explanation of the relationship between permutability and invertibility is included.

7.2 Permutability of Rules

The conditions which guarantee that one rule permutes over others are given: this will form an important part of the proofs that are to follow in later subsections.

7.3 Invertibility of Rules

The conditions which guarantee that a rule is invertible (with respect to a calculus) are given, for a range of calculi.

7.3.1 Propositional Uniprincipal Multisuccedent

7.3.2 Propositional Uniprincipal Single Succedent

7.3.3 First-Order Uniprincipal Calculi

7.3.4 Propositional Multiprincipal Calculi

7.3.5 Modal and Modal-Like Calculi

A modal-like calculi is one which has multisets of formulae each sharing an outermost connective or function symbol.

7.4 A Formalisation

A formalisation of the preceding subsections is given, along with a discussion of how close the formalised results are to the theoretical results. The methods of this section will be compared with the methods of §5.3 to show how useful they can be.

7.4.1 An Example

We formalise again a proof from §5.3 to show the power of the new method.

8 Conclusions

A discussion of what was achieved, who might benefit from the work and what could be achieved in the future.

8.1 Applicability to the Field

We discuss possible uses of the work presented.

8.2 Future Directions

We detail ways in which the work could be extended or improved.

8.3 Final Summary

We examine what has been achieved relative to the stated aims from §1.2.

References

[1] A. Adams. *Tools and Techniques for Machine-Assisted Meta-Theory.* PhD thesis, University of St. Andrews, 1997.

[2] F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1999.

[3] S. Boulmé. A Proof of Craig's Interpolation Theorem in Coq, 1996. Available at citeseer.ist.psu.edu/480840.html.

[4] P. Chapman. A Formalisation of Contraction Admissibility for G4ip. University of St Andrews Computer Science Research Report, available at www.cs.st-andrews.ac.uk/∼pc, 2008.

[5] P. Chapman. A Formalised Proof of Cut Admissibility. University of St Andrews Computer Science Research Report, available at www.cs.st-andrews.ac.uk/∼pc, 2008.

[6] P. Chapman. Formalising Proofs of Cut Admissibility. University of St Andrews Computer Science Research Report, 2008.

[7] P. Chapman. Syntactic Conditions for Invertibility in Sequent Calculi. University of St Andrews Computer Science Research Report, available at www.cs.st-andrews.ac.uk/∼pc, 2008.

[8] Peter Chapman, James McKinna, and Christian Urban. Mechanising a Proof of Craig's Interpolation Theorem for Intuitionistic Logic in Nominal Isabelle. In *AISC/MKM/Calculemus*, volume 5144 of *Lecture Notes in Computer Science*, pages 38–52. Springer, 2008.

[9] C. Chen, D. Zhu, and H. Xi. Implementing Cut Elimination: A Case Study of Simulating Dependent Types in Haskell, 2003. Chen, C., Zhu, D., and Xi, H. Implementing Cut Elimination: A Case Study of Simulating Dependent Types in Haskell, October 2003. Available at: http://www.cs.bu.edu/∼hwxi/academic/papers/CutElim.

[10] Agata Ciabattoni and Kazushige Terui. Modular Cut-Elimination: Finding Proofs or Counterexamples. In *Logic for Programming, Artificial Intelligence, and Reasoning, 13th International Conference, LPAR 2006, Phnom Penh, Cambodia, November 13-17, 2006, Proceedings*, pages 135–149, 2006.

[11] Agata Ciabattoni and Kazushige Terui. Towards a Semantic Characterization of Cut-Elimination. *Studia Logica*, 82(1):95–119, 2006.

[12] Coq Development Team. *The Coq Proof Assistant Reference Manual Version 8.1*, 2006. Available at http://coq.inria.fr/V8.1/refman/index.html.

[13] Jeremy E. Dawson. Isabelle files. Available at http://users.rsise.anu.edu.au/∼jeremy/isabelle/, 2008.

[14] Nachum Dershowitz and Zohar Manna. Proving Termination with Multiset Orderings. *Commun. ACM*, 22(8):465–476, 1979.

[15] A. G. Dragalin. *Mathematical Intuitionism.* Number 67 in Translations of Mathematical Monographs. American Mathematical Society, 1988.

[16] R. Dyckhoff, D. Kesner, and S. Lengrand. Strong cut-elimination systems for Hudelmaier's depth-bounded sequent calculus for implicational logic. In U. Furbach and N. Shankar, editors, *Proceedings of the 3rd International Joint Conference on Automated Reasoning (IJCAR'06)*, volume 4130 of *LNAI*, pages 347–361. Springer-Verlag, August 2006.

[17] Roy Dyckhoff. Contraction-Free Sequent Calculi for Intuitionistic Logic. *J. Symb. Log.*, 57(3):795–807, 1992.

[18] Roy Dyckhoff and Sara Negri. Admissibility of Structural Rules for Contraction-Free Systems of Intuitionistic Logic. *J. Symb. Log.*, 65(4):1499–1518, 2000.

[19] Murdoch J. Gabbay and Andrew M. Pitts. A New Approach to Abstract Syntax Involving Binders. In *14th Annual Symposium on Logic in Computer Science*, pages 214–224, Washington, DC, USA, 1999. IEEE Computer Society Press.

[20] Didier Galmiche and Guy Perrier. On Proof Normalization in Linear Logic. *Theor. Comput. Sci.*, 135(1):67–110, 1994.

[21] J. Hudelmaier. Semantische Sequenzenkalküle, 1998. Habilitationsschrift, Fakultät für Informatik der Eberhard-Karls-Üniversität Tübingen, Germany.

[22] Jörg Hudelmaier. An O(n log n)-Space Decision Procedure for Intuitionistic Propositional Logic. *J. Log. Comput.*, 3(1):63–75, 1993.

[23] Ranjit Jhala, Rupak Majumdar, and Ru-Gang Xu. State of the Union: Type Inference Via Craig Interpolation. In *Tools and Algorithms for the Construction and Analysis of Systems, 13th International Conference, TACAS 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007 Braga, Portugal, March 24 - April 1, 2007, Proceedings*, volume 4424 of *Lecture Notes in Computer Science*, pages 553–567. Springer, 2007.

[24] S. C. Kleene. *Introduction to Metamathematics.* North-Holland Publishing Company, 1952.

[25] Tatjana Lutovac and James Harland. Issues in the Analysis of Proof-Search Strategies in Sequential Presentations of Logics. *Electr. Notes Theor. Comput. Sci.*, 125(2):115–147, 2005.

[26] P. Martin-Löf. *Intuitionistic Type Theory.* Number 1 in Studies in Proof Theory. Bibliopolis, 1984.

[27] C. McBride. *Epigram: Practical Programming with dependent types.*

[28] Conor McBride and James McKinna. The View from the Left. *J. Funct. Program.*, 14(1):69–111, 2004.

[29] James McKinna and Robert Pollack. Pure Type Systems Formalized. In M. Bezem and J. F. Groote, editors, *Proceedings 1st Int. Conf. on Typed Lambda Calculi and Applications, TLCA'93, Utrecht*, volume 664 of *LNCS*, pages 289–305. Springer-Verlag, 1993.

[30] Kenneth L. McMillan. Applications of Craig Interpolants in Model Checking. In *Tools and Algorithms for the Construction and Analysis of Systems, 11th International Conference, TACAS 2005, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2005, Edinburgh, UK, April 4-8, 2005, Proceedings*, volume 3440 of *Lecture Notes in Computer Science*, pages 1–12. Springer, 2005.

[31] Dale Miller and Alwen Tiu. A Proof Theory for Generic Judgments. *ACM Trans. Comput. Log.*, 6(4):749–783, 2005.

[32] Sara Negri. Proof Analysis in Modal Logic. *Journal of Philosophical Logic*, 34:507–544, 2005.

[33] Sara Negri and Jan von Plato. *Structural Proof Theory.* Cambridge University Press, Cambridge, 2001.

[34] T. Nipkow, L. Paulson, and M. Wenzel. *A Proof Assistant for Higher-Order Logic.* Number 2283 in Lecture Notes in Computer Science. Springer-Verlag, 2005.

[35] Tobias Nipkow. Structured Proofs in Isar/HOL. In *TYPES*, pages 259–278, 2002.

[36] M. de Rijke P. Blackburn and Y. Venema. *Modal Logic.* Number 53 in Cambridge Tracts in Computer Science. Cambridge University Press, 2000.

[37] L. Paulson. Isabelle's Logics. Available at http://www.cl.cam.ac.uk/research/hvg/Isabelle, 2005.

[38] F. Pfenning and C. Schuermann. *Twelf User's Guide*, 2005.

[39] Frank Pfenning. Structural Cut Elimination: I. Intuitionistic and Classical Logic. *Inf. Comput.*, 157(1-2):84–141, 2000.

[40] A. M. Pitts. Nominal Logic, a First-Order Theory of Names and Binding. *Inf. Comput.*, 186(2):165–193, 2003.

[41] João Rasga. Sufficient conditions for cut elimination with complexity analysis. *Ann. Pure Appl. Logic*, 149(1-3):81–99, 2007.

[42] Greg Restall. *An Introduction to Substructural Logics*. Routledge, London, 1999.

[43] T. Ridge. Craig's Interpolation Theorem formalised and mechanised in Isabelle/HOL. Arxiv preprint cs.LO/0607058, 2006 - arxiv.org.

[44] R. Smullyan. *First-order logic*. Springer-Verlag, 1968.

[45] M. Sörensen and P. Urzyczyn. *Lectures on the Curry-Howard Isomorphism*. Number 149 in Studies in Logic and the Foundations of Mathematics. Elsevier, 2006.

[46] G. Takeuti. *Proof Theory*. Number 81 in Studies in Logic and the Foundations of Mathematics. North-Holland Publishing Company, 1975.

[47] C. Tasson and C. Urban. Nominal techniques in Isabelle/HOL. In *Proceedings of the 20th International Conference on Automated Deduction (CADE 2005)*, volume 3632 of *LNCS*, pages 38–53. Springer-Verlag, 2005.

[48] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Number 43 in Cambridge Tracts in Computer Science. Cambridge University Press, second edition, 2000.

[49] C. Urban and G. M. Bierman. Strong Normalisation of Cut-Elimination in Classical Logic. In *Typed Lambda Calculus and Applications*, pages 365–380, 1999.

[50] N. Vorob'ev. A New Algorithm for Derivability in the Constructive Propositional Calculus. *American Mathematical Society Translations*, 94:37–71, 1970.

[51] M. Wenzel. *Isabelle/Isar Reference Manual*, 2002.

[52] Anna Zamansky and Arnon Avron. Canonical Gentzen-Type Calculi with (n, k)-ary Quantifiers. In *Automated Reasoning, Third International Joint Conference, IJCAR 2006, Seattle, WA, USA, August 17-20, 2006, Proceedings*, pages 251–265, 2006.