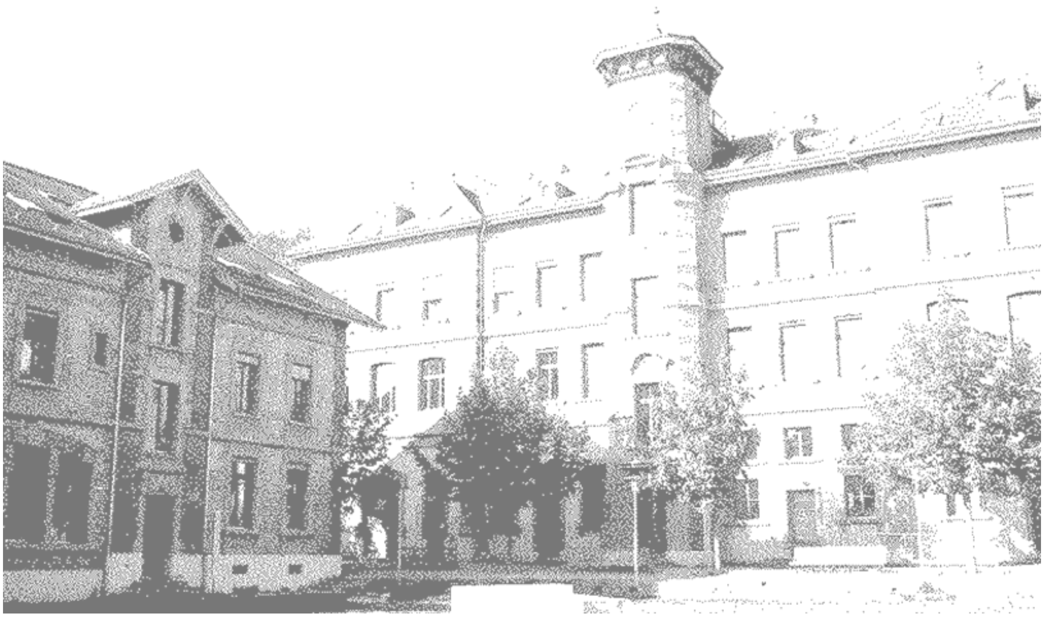
UNIVERSITÄT
BERN

Proof, Computation, Complexity PCC 2010 International Workshop, Proceedings

K. Brünnler and T. Studer (editors)

Technical report IAM-10-001, 18-19 June 2010

Institut für Informatik und angewandte Mathematik, www.iam.unibe.ch



Proof, Computation, Complexity

PCC 2010

International Workshop

Bern, 18 – 19 June 2010

Edited by:

Kai Brünnler
Thomas Studer

Institut für Informatik und angewandte Mathematik
Universität Bern

The aim of PCC is to stimulate research in proof theory, computation, and complexity, focusing on issues which combine logical and computational aspects. Topics may include applications of formal inference systems in computer science, as well as new developments in proof theory motivated by computer science demands. Specific areas of interest are (non-exhaustively listed) foundations for specification and programming languages, logical methods in specification and program development including program extraction from proofs, type theory, new developments in structural proof theory, and implicit computational complexity.

Invited Talks

Provability algebras and scattered topology

Lev Beklemishev

Abstract:

Provability algebra of a reasonable arithmetical theory T is its Lindenbaum boolean algebra enriched by the operators <n> mapping a sentence A to the Gödelian sentence expressing n-consistency of A over T. We consider set-theoretic interpretation of propositions as subsets of a set X, boolean connectives as boolean operations, and operators <n> as operators on the set of all subsets of X. The identities of the provability algebra then impose some restrictions on the choice of possible operators which amount to the following ones:

(1) every operator <n> is a derived set operator w.r.t. some topology T_n, that is, <n>A is the set of limit points of A w.r.t. T_n;

(2) each T_n is scattered, that is, every non-empty subspace A of X has an isolated point;

(3) each T_{n+1} is finer than T_n;

(4) <n>A is open in T_{n+1}, for each subset A of X.

We study the properties of such spaces and their connections with the questions of proof theory and set theory. In particular, the problem of completeness of the system of identities of provability algebras w.r.t. GLP-spaces has unexpected relationships with the axioms of large cardinals and stationary reflection.

Systematic (and algebraic) proof theory for substructural logics

Agata Ciabattoni

Abstract:

I will outline an algebraic and systematic approach to proof theory for substructural logics, which is being developed together with N. Galatos, K. Terui and L. Strassburger.

Unification in nonclassical theories

Rosalie Iemhoff

There are many problems in mathematics that can be cast in terms of unification, meaning that a solution of the problem is a substitution that identifies two terms, either literally, or against a background theory of equivalence. If the theory has a term for "true", then a substitution which applied to a term makes it true, is called a *unifier* of that term. Thus in the context of formulas, a unifier is a substitution under which the formula becomes derivable in the theory. This form of unification plays an important role in automated theorem proving, and it is this kind of unification that is considered in this talk.

In many classical theories, all unifiable formulas have a *most general unifier*, which is a unifier that generates all other unifiers of a formula. Nonclassical theories in general do not have this useful property. But sometimes something weaker holds: every formula has a finite set of maximal unifiers. Many well-known modal and intermediate logics have this property.

The study of unification in nonclassical logics mainly uses semantical techniques. Even though there exist algorithms to find maximal unifiers, proofs of correctness again use semantics. In this talk a purely syntactic treatment of unification is presented, and it is shown how most known results follow easily from this approach. The talk will start with a general introduction to unification.

Contributed Talks

On positive fragments of polymodal provability logic

Evgenij Dashkov

There is not much literature concerning *positive modal logics*, i.e., logics in languages without negation and implication. Dunn [3] has pioneered the field with a study of bimodal minimal normal logic $K_+^{\top\bot}$ in the language with $\Box$, $\Diamond$, $\vee$, $\wedge$, $\top$, $\bot$. However, as far as we know, there have not been any special studies of positive provability logics. We begin such a study and argue that it could be fruitful for proof-theoretic applications.

Lev Beklemishev has suggested an algebraical approach to proof-theoretic analysis based on the notion of *graded provability algebra*, i.e., a theory's Lindenbaum algebra augmented with provability operators. Unlike bare Lindenbaum boolean algebras, which are pairwise isomorphic for all reasonable theories, the structure can capture some proof-theoretical properties of the theory. In [1], the method is applied to Peano arithmetic and it is shown how an ordinal notation system up to ϵ_0 can be canonically recovered from the corresponding algebra. Terms of the graded provability algebra are put in correspondence with formulas of the polymodal provability logic GLP.

We notice that the principal results of [1] and [2] (including ordinal analysis of PA, characterization of PA Π_n-consequences in terms of iterated reflection schemes, and independence of PA for the combinatorial *Worm Principle*) only rely on a special fragment of GLP. Namely, it is sufficient to consider equivalences of formulas built from the truth constant $\top$, propositional variables, conjunction and modalities $\langle n \rangle$ for all $n < \omega$. We call such polymodal formulas *positive*. There arises a natural question how to axiomatize the fragment of GLP in this positive language.

For the positive fragment, we suggest two calculi: sequential Gentzen-style $\mathrm{GLP}_+^{\mathrm{G}}$ and equational $\mathrm{GLP}_+^{\mathrm{e}}$ (see the tables below). We establish the following conservation theorems.

Theorem 1. *Let Γ and Δ be finite sets of positive formulas. Then* $\mathrm{GLP} \vdash \bigwedge \Gamma \to \bigvee \Delta$ *iff* $\mathrm{GLP}_+^{\mathrm{G}} \vdash \Gamma \Rightarrow \Delta$.

Theorem 2. *For any positive formulas ϕ and ψ,* $\mathrm{GLP} \vdash \phi \leftrightarrow \psi$ *iff* $\mathrm{GLP}_+^{\mathrm{e}} \vdash \phi = \psi$.

The restriction of $\mathrm{GLP}_+^{\mathrm{e}}$ calculus to the language with one modality yields an axiomatization for the positive fragments of both K4 and GL. Thus, curiously, we obtain that the positive fragments of K4 and GL are the same.

For positive formulas, we generalize the usual interpretation of polymodal formulas so that a set of arithmetical sentences, rather than a single sentence is assigned to a modal formula. These sets of sentences have to be arithmetically definable, i.e., we consider each with an arithmetical formula defining the set of Gödel numbers for those sentences.

Let us fix a sound arithmetical theory S. Assign a set p^* of arithmetical sentences with its arithmetical definition to each propositional variable p. Then define $\top^* = \varnothing$, $(\phi \wedge \psi)^* = \phi^* \cup \psi^*$, and $(\langle n \rangle \phi)^* = \{\mathrm{RFN}_{\Pi_{n+1}}(S + \phi^*)\}$ alongside with their natural arithmetical definitions. Combining the arithmetical completeness theorem for GLP [4] with our conservation result, we obtain

Corollary 3. *For any positive formulas ϕ and ψ, $\mathrm{GLP}^{\mathrm{e}}_{+} \vdash \phi = \psi$ iff ϕ^* and ψ^* are equivalent over S.*

This interpretation has an advantage for studying provability algebras of theories stronger than PA. For example, the full reflection scheme $\mathrm{RFN}(S)$ is not finitely axiomatizable. Therefore a modal formula representing the scheme cannot be interpreted as one arithmetical formula. However, it can be treated within positive modal logic.

Let us introduce a new modality: $(\langle \omega \rangle \phi)^* = \mathrm{RFN}(S + \phi^*)$. The axioms of $\mathrm{GLP}^{\mathrm{e}}_{+}$ are also meaningful for the extended language. We obtain the following arithmetical completeness theorem.

Theorem 4. *For any positive formulas ϕ and ψ in the language enriched with $\langle \omega \rangle$, $\mathrm{GLP}^{\mathrm{e}}_{+} \vdash \phi = \psi$ iff ϕ^* and ψ^* are equivalent over S.*

We plan to extend this technique to develop provability algebras for systems of predicative analysis.

System $\mathbf{GLP}^{\mathbf{e}}_{+}$

1. The usual rules of equational logic and identities for $\wedge$;
2. $\langle n \rangle(\phi \wedge \psi) \leqslant \langle n \rangle \phi \wedge \langle n \rangle \psi$; *(By $\phi \leqslant \psi$, denote $\phi \wedge \psi = \phi$)*
3. $\langle n \rangle \langle n \rangle \phi \leqslant \langle n \rangle \phi$;
4. $\langle n \rangle \phi \wedge \langle m \rangle \psi = \langle n \rangle(\phi \wedge \langle m \rangle \psi)$, where $m < n$;
5. $\langle n \rangle \phi \leqslant \langle m \rangle \phi$, where $m < n$.

System $\mathbf{GLP}^{\mathbf{G}}_{+}$

(Suppose $\Gamma = \{\gamma_i\}$. Then denote by $\Diamond_{\geqslant n}\Gamma$ any set $\{\langle k_i \rangle \gamma_i\}$, where $k_i \geqslant n$ for all i. Define $\Diamond_{<n}\Gamma$ similarly and put $\langle n \rangle \Gamma = \{\langle n \rangle \gamma_i\}$.)

The axioms are $\phi \Rightarrow \phi$ and $\Rightarrow \top$. The rules are *weakening, cut, left* and *right conjunction introductions*, as well as the following two:

$$(M)\ \frac{\Gamma, \langle m \rangle \phi \ \Rightarrow\ \Delta}{\Gamma, \langle n \rangle \phi \ \Rightarrow\ \Delta} \qquad\qquad \frac{\Sigma, \phi \ \Rightarrow\ \Pi, \Diamond_{\geqslant n}\Theta, \Theta, \langle n \rangle \phi}{\Sigma, \langle n \rangle \phi \ \Rightarrow\ \Pi, \langle n \rangle \Theta}\ (J)$$

where $m < n$, $\Sigma = \Diamond_{<n}\Gamma$ and $\Pi = \Diamond_{<n}\Delta$ for some sets Γ and Δ.

References

[1] L. Beklemishev. *Provability algebras and proof-theoretic ordinals, I.* Annals of Pure and Applied Logic, 128:103–123, 2004.

[2] L. Beklemishev. *The Worm Principle.* In Z. Chatzidakis, P. Koepke, W. Pohlers (eds.), Lecture Notes in Logic 27. Logic Colloquium '02, pp. 75–95, AK Peters, 2006.

[3] J. Dunn. *Positive modal logic.* Studia Logica, 55(2):301–317, 1995.

[4] K. Ignatiev. *On strong provability predicates and the associated modal logics.* The Journal of Symbolic Logic, 58:249–290, 1993.

Cut elimination, substitution and normalisation

Abstract
PCC Bern, June 2010

Roy Dyckhoff
St Andrews University
rd@cs.st-andrews.ac.uk

It is well-known that, in intuitionistic logic, sequent calculus derivations (with or without *Cut*) are recipes for constructing natural deductions, and that, by the Curry-Howard correspondence, with care about variable discharge conventions, one can represent both the former and the latter using terms of a typed lambda calculus. Natural deduction terms may, by various standard reductions, be normalised; there are however many sequent calculi S, reduction systems R for S and reduction strategies for R, including but not limited to those given by Gentzen. We present a complete single-succedent sequent calculus (essentially the Ketonen-Kleene system **G3ip** from [4], including all the usual zero-order connectives, including disjunction and absurdity) and a (we believe) novel reduction system for eliminating cuts, with the virtues that (a) it is strongly normalising (b) it is confluent and (c) allows a surjective homomorphism (as described below) from cut-elimination to normalisation: in other words, a homomorphism from **G3ip** derivations to **NJ** natural deductions with the property that each cut reduction step translates into a sequence of zero or more reductions in the natural deduction setting.

Kreisel asked [1] about the relation between cut-elimination and normalisation. Zucker [6], using Gentzen's cut-elimination steps and an innermost-first strategy, gave a partial answer, but had difficulties with disjunction, including a failure of strong normalisation. Pottinger [3] gave a positive answer covering disjunction; but, as pointed out by Urban [5], the notion of normality for the sequent calculus proof terms does not coincide with cut-freedom, and this renders Pottinger's answer (we believe) defective, despite Pottinger's claim that the difference is "trivial". (Moreover, the closest system in [3] to a conventional sequent calculus is his H_L; but, although it is complete for derivability of formulae, it does not admit *Contraction*; it does not derive, for example, the sequent $p \Rightarrow p \wedge p$. Nor does it admit *Weakening* or derive the sequent $p \wedge p \Rightarrow p \wedge p$. There is a section explaining what one might do if contraction is added as a primitive rule, with no explanation of how the cut-reduction rules might change—one is reminded of Gentzen's difficulties with *Contraction* and his avoidance thereof with his *Mix* rule.) The calculi we present are designed to avoid these difficulties.

The sequent calculus typing rules are rather standard (various freshness constraints are omitted):

$$\frac{}{x : \bot, \Gamma \Rightarrow X(x) : \theta}\ L\bot \qquad \frac{}{x : \phi, \Gamma \Rightarrow x : \phi}\ Ax$$

$$\frac{z : \phi\supset\psi, \Gamma \Rightarrow L : \phi \quad y : \psi, z : \phi\supset\psi, \Gamma \Rightarrow L' : \theta}{z : \phi\supset\psi, \Gamma \Rightarrow A(z, L, y.L') : \theta}\ L\supset \qquad \frac{x : \phi, \Gamma \Rightarrow L : \psi}{\Gamma \Rightarrow \lambda x.L : \phi\supset\psi}\ R\supset$$

$$\frac{x : \phi_i, z : \phi_1 \wedge \phi_2, \Gamma \Rightarrow L : \theta}{z : \phi_1 \wedge \phi_2, \Gamma \Rightarrow E_i(z, x.L) : \theta}\ L\wedge_i \qquad \frac{\Gamma \Rightarrow L : \phi \quad \Gamma \Rightarrow L' : \psi}{\Gamma \Rightarrow (L, L') : \phi \wedge \psi}\ R\wedge$$

$$\frac{x : \phi, z : \phi \vee \psi, \Gamma \Rightarrow L : \theta \quad x' : \psi, z : \phi \vee \psi, \Gamma \Rightarrow L' : \theta}{z : \phi \vee \psi, \Gamma \Rightarrow W(z, x.L, x'.L') : \theta}\ L\vee \qquad \frac{\Gamma \Rightarrow L : \phi_i}{\Gamma \Rightarrow in_i(L) : \phi_1 \vee \phi_2}\ R\vee_i$$

$$\frac{\Gamma \Rightarrow L : \phi \quad x : \phi, \Gamma \Rightarrow L' : \theta}{\Gamma \Rightarrow C(L, x.L') : \theta}\ Cut$$

and we give here one, and for lack of space just one, of the 32 cut-reduction rules:

$$\begin{aligned} C(W(w, w_1.L_1, w_2.L_2), x.W(x, x'.L', x''.L'')) \quad \rightsquigarrow \quad & W(w, w_1.C(L_1, x.W(x, x'.C(W(w, w_1.L_1, w_2.L_2), x.L'), \\ & \qquad x''.C(W(w, w_1.L_1, w_2.L_2), x.L''))), \\ & \quad w_2.C(L_2, x.W(x, x'.C(W(w, w_1.L_1, w_2.L_2), x.L'), \\ & \qquad x''.C(W(w, w_1.L_1, w_2.L_2), x.L'')))) \end{aligned}$$

The subject reduction property is routine. Strong normalisation is proved using a lexicographic path ordering. The system of cut-reduction rules is a left-linear orthogonal pattern-rewrite system, without critical pairs; by the results of [2], confluence is immediate.

Thanks are due to Peter Chapman, Jacob Howe, Stéphane Lengrand and Christian Urban for helpful comments and (to the last of these) for a copy of [5] prior to its publication, albeit many years after its presentation in Rio. Chapman's formalisation in *Nominal Isabelle* of the results (along lines started by Urban) was most helpful in identifying some minor errors.

References

[1] Georg Kreisel. *A Survey of Proof Theory II*, Proceedings of Second Scandinavian Logic Symposium, pp 109–170, 1971.

[2] Richard Mayr and Tobias Nipkow, *Higher-Order Rewrite Systems and their Confluence*, Theoretical Computer Science **192**, pp 3–29, 1998.

[3] Garrel Pottinger. *Normalisation as a homomorphic image of cut-elimination*, Annals of Mathematical Logic **12**, pp 323–357, 1977.

[4] Anne Troelstra and Helmut Schwichtenberg. *Basic Proof Theory*, 2nd ed, Cambridge, 2000.

[5] Christian Urban. *Revisiting Zucker's work on the correspondence between cut-elimination and normalisation*, Proceedings of Conference on Natural Deduction 2001, Rio de Janeiro, (eds: Edward Hermann Haeusler, Luiz Carlos Pereira and Valeria de Paiva), Advances in Natural Deduction, Kluwer 2010 (to appear), 21 pp.

[6] Jeff Zucker. *The correspondence between cut-elimination and normalisation*, Annals of Mathematical Logic **7**, pp 1–112, 1974. 2000.

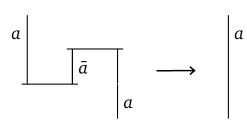
Normalisation by Interaction and Computational Interpretations

Nicolas Guenot

Laboratoire d'Informatique (LIX), École Polytechnique
rue de Saclay, 91128 Palaiseau cedex, France
`nguenot@lix.polytechnique.fr`

We will present a work in progress aiming at the definition of normalisation procedures for proof systems using the deep inference methodology, based only on permutations and interaction of rules instances, and establishing a framework for investigations into the extension of the Curry-Howard correspondence to the calculus of structures [Gug07]. The starting point is a system for purely implicative intuitionistic linear logic, for which we provide a normalisation result, that can be used in a fine-grain analysis of β-reduction in the linear λ-calculus.

We have devised for this system a new technique for proving cut elimination, different from the usual procedures in deep inference, often based on the *splitting lemma*, involving complex transformations of proofs. It is closer to the *substitution* technique [Brü03], which could not be used here because it relies on a non-linear setting, just as the proof based on atomic flows [GG08]. However, the basic idea is the same: as it is done in natural deduction, we eliminate *detours* until no needless cut remains, as can be illustrated using atomic flows:



This procedure is purely internal to the syntax of the calculus of structures, and can be implemented in a modular way, by picking any cut, reducing it to the atomic form, and then eliminate it by interaction with a matching identity. Moreover, it works on open derivations so that cut elimination is a corollary — while other systems in an intuitionistic setting, e.g. [Tiu06], have no such internal proof.

This work started as an attempt to reconcile the deep inference approach with the traditional understanding of computation through cut elimination, based on the idea that there exists a strong link between intuitionistic logic and the λ-calculus, whatever formalism is used. It was accepted that the calculus of structures should allow for an analysis at a lower level, since it generates interesting decompositions on the logical side. As usual in this setting, a correspondence is obtained by typing λ-terms using derivations of the logic. We provide a *nested type system* for a variant of the purely linear λ-calculus, where typing rules are in one-to-one correpondence with inference rules in our proof system.

Our typing system is non-deterministic, so that it requires to choose a strategy, which has an interesting relation to evaluation strategies in the λ-calculus. Indeed, there are more proofs in the calculus of structures than in the sequent calculus, so that each typing derivation can be seen as an execution trace of a λ-term, following a strategy induced by the typing strategy. Then, the normalisation procedure can be applied on typing derivations and thus to λ-terms, yielding an alternative set of rewriting rules, that decomposes the usual β-reduction, using η-expansion and β-reduction on variables:

$$\begin{array}{rcl} x & \longrightarrow & \lambda y.xy \\ (\lambda x.x)y & \longrightarrow & y \end{array}$$

along with a more complex rule, that can be summarised by simplifying it into its shallow variant, moving virtual redexes out to make them explicit:

$$(\lambda x.xM)(\lambda y.N) \quad \longrightarrow \quad (\lambda x.x)((\lambda y.N)M)$$

This analysis of the λ-calculus *through the glasses* of deep inference provides an interesting notion of *nested execution*, and is more intuitive than the only existing algorithmic interpretation of a proof system in the calculus of structures [BM08], which induces a very abstract view of computation. It could also yield interesting insights on evaluation strategies for the λ-calculus.

Finally, we will discuss the extension of this work to larger logics, and classical intuitionistic logic in the first place. The *normalisation by interaction* technique seems quite general, and could apply in many cases if we overcome the problem of handling duplications in logics equipped with a form of contraction.

References

[BM08] K. Brünnler and R. McKinley. An algorithmic interpretation of a deep inference system. In I. Cervesato, H. Veith, and A. Voronkov, editors, *LPAR'08*, volume 5330 of *LNCS*, pages 482–496, 2008.

[Brü03] K. Brünnler. *Deep Inference and Symmetry in Classical Proofs*. PhD thesis, Technische Universität Dresden, September 2003.

[GG08] A. Guglielmi and T. Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1):1–36, 2008.

[Gug07] A. Guglielmi. A system of interaction and structure. *ACM Transactions on Computational Logic*, 8(1):1–64, 2007.

[Tiu06] A. Tiu. A local system for intuitionistic logic. In M. Hermann and A. Voronkov, editors, *LPAR'06*, volume 4246 of *LNCS*, pages 242–256, 2006.

Breaking Paths in Atomic Flows for Classical Logic

Alessio Guglielmi
INRIA Nancy – Grand Est, LORIA and
University of Bath
A.Guglielmi AT bath.ac.uk

Tom Gundersen
INRIA Saclay – Île-de-France and
École Polytechnique, LIX
teg AT jklm.no

Lutz Straßburger
INRIA Saclay – Île-de-France and
École Polytechnique, LIX
lutz AT lix.polytechnique.fr

Abstract

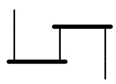
This work belongs to a wider effort aimed at eliminating syntactic bureaucracy from proof systems. We present a novel cut elimination procedure for classical propositional logic. It is based on the recently introduced atomic flows: *they are purely graphical devices that abstract away from much of the typical bureaucracy of proofs. We make crucial use of the* path breaker*, an atomic-flow construction that avoids some nasty termination problems, and that can be used in any proof system with sufficient symmetry.*

The investigation of the cut elimination property of logical systems is a central topic in current proof theory, and, as pointed out by Girard [Gir87b], the *lack of modularity* is one of its main technical limitations. More precisely, the argument for showing cut elimination is usually based on heavy syntactic arguments and a tedious case analysis depending on the shape of the inference rules. A slight change in design makes the whole proof break down, and if one wishes to add some rules, one usually has to redo the whole cut elimination proof from scratch.

This work suggests that the source of this "lack of modularity" might not be in the nature of the cut elimination property, but in the method that is used for proving it. We present here a cut elimination procedure for classical propositional logic that is independent from the shape of the logical rules. It is not based on the permutation of inference rules but on the manipulation of *atomic flows* [GG08].

Atomic flows capture the structural skeleton of a proof and ignore the logical content. Due to their "graphical nature", atomic flows can be seen as relatives of Girard's *proof nets* [Gir87a, DR89] and Buss' *logical flow graphs* [Bus91]. Proof nets have originally been proposed only for linear logic, but there have been various proposals for proof nets for classical logic with different purpose and design, e.g., by Laurent [Lau99], by Robinson [Rob03] and by Lamarche and Straßburger [LS05]. Logical flow graphs have only been defined for classical logic, but their definition for linear logic would be literally the same. In fact, for the multiplicative fragment of linear logic (MLL) the two notions essentially coincide. This, however, is no longer the case for classical logic, which can be obtained from MLL by adding the rules for contraction and weakening. Atomic flows can be seen as a development that takes the best out of both worlds. Like proof nets they simplify proof normalization because they avoid unnecessary bureaucracy due to trivial rule permutations, and like logical flow graphs they precisely capture the information flow inside the proof. In this respect, they are very similar to the variant of proof nets discussed in [Str09]. Since atomic flows contain for each atom occurrence every contraction and weakening that is applied to it, they can be used for controlling the size of proofs, and thus can also play a role in proof complexity (see [BGGP10]).

Atomic flows are also very similar to *string diagrams* for representing morphisms in monoidal categories (see [Sel09] for a survey). However, in (classical) logic one usually finds two dual monoidal structures and not just one. Thus, atomic flows are, in spirit, more closely related to *coherence graphs* in monoidal closed categories [KM71]. Nonetheless, it should be stressed that atomic flows do not form a monoidal closed category. The following two flows are *not* the same

 and (1)

although, during the normalization process, we wish to reduce the atomic flow on the left (a cut connected to an identity) to the atomic flow on the right (a single edge). In linear logic one can simply "pull the edges" and directly reduce the left atomic flow in (1) to the right one, whereas in classical logic this step might involve duplication of large parts of the proof.

The main insight coming from atomic flows is that this duplication and the whole normalization process is independent from the logical content of the proof and independent from the design of the logical rules in use, as is discussed in [GG08].

This work is structured in two parts:

- We define local transformations on atomic flows that are similar to the reduction steps in linear logic proof nets or in interaction nets [Laf90], and that have the goal to normalize the proof. However, due to the presence of contractions, the atomic flows can contain cycles that prevent these local reductions from terminating. To solve this problem, we define a global transformation on the atomic flows, called the *path breaker*, that treats the proof as a black box; it simply duplicates the whole proof and combines the copies. Note that this is conceptually different from the cut elimination in standard proof nets [Gir87a, Lau99, Rob03], where cut reduction steps are mixed local/global: A single step involves a local cut reduction and some duplication of a part of a proof (a box or an empire). In our case the procedure consists of two phases, a purely global one followed by a purely local one. However it remains an important research objective to investigate the computational meaning of these reductions.

- We show how formal proofs in a deductive system are mapped to atomic flows, and how the operations on atomic flows that we defined before can be lifted to the deductive system, and thus can be used to provide a cut elimination procedure. This can be done because the symmetry of the deductive system we use allows to reduce the cut to its atomic form, in the same way as it is done for the identity rule in traditional systems.

References

[BGGP10] Paola Bruscoli, Alessio Guglielmi, Tom Gundersen, and Michel Parigot. A quasipolynomial cut-elimination procedure in deep inference via atomic flows and threshold formulae. In *LPAR-16*, 2010.

[Bus91] Samuel R. Buss. The undecidability of k-provability. *Annals of Pure and Applied Logic*, 53:72–102, 1991.

[DR89] Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Annals of Mathematical Logic*, 28:181–203, 1989.

[GG08] Alessio Guglielmi and Tom Gundersen. Normalisation control in deep inference via atomic flows. *Logical Methods in Computer Science*, 4(1:9):1–36, 2008.

[Gir87a] Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Gir87b] Jean-Yves Girard. *Proof Theory and Logical Complexity, Volume I*, volume 1 of *Studies in Proof Theory*. Bibliopolis, edizioni di filosofia e scienze, 1987.

[KM71] Gregory Maxwell Kelly and Saunders Mac Lane. Coherence in closed categories. *Jour. of Pure and Applied Algebra*, 1:97–140, 1971.

[Laf90] Yves Lafont. Interaction nets. In *POPL*, pages 95–108, 1990.

[Lau99] Olivier Laurent. Polarized proof-nets: proof-nets for LC (extended abstract). In Jean-Yves Girard, editor, *Typed Lambda Calculi and Applications (TLCA 1999)*, volume 1581 of *LNCS*, pages 213–227. Springer, 1999.

[LS05] François Lamarche and Lutz Straßburger. Naming proofs in classical propositional logic. In Paweł Urzyczyn, editor, *Typed Lambda Calculi and Applications, TLCA 2005*, volume 3461 of *LNCS*, pages 246–261. Springer, 2005.

[Rob03] Edmund P. Robinson. Proof nets for classical logic. *Journal of Logic and Computation*, 13:777–797, 2003.

[Sel09] Peter Selinger. A survey of graphical languages for monoidal categories. In Bob Coecke, editor, *New Structures for Physics*, Lecture Notes in Physics. Springer, 2009. to appear.

[Str09] Lutz Straßburger. From deep inference to proof nets via cut elimination. *Jour. of Logic and Comp.*, 2009. to appear.

Proof Nets for Free Sum-Product Categories

Willem Heijltjes

LFCS
University of Edinburgh

Extended abstract

The category $\Sigma\Pi(\mathcal{C})$ is the free completion with finite products and coproducts of a base category $\mathcal{C}$.[1] Such free sum-product completions are the discrete, finite version of the *bicompletions* described by Joyal in [4], the free completion with all limits and colimits.

Analogous to the lattice completion of a partial order, Joyal shows that the free bicompletion of a category is completely characterised by four properties, one of which is *softness*. A particular consequence of softness is that any map from a limit to a colimit must factor through one of the limit's projections or one of the colimit's injections. Since a map into a limit can be deconstructed as a collection of maps, one into each component (and similarly for a map out of a colimit), Joyal's characterisation is akin to a cut-elimination result: the maps in a free bicompletion have representations in *normal form*.

For sum-product categories the connection with cut-elimination has been made explicit by Cockett and Seely in [2]. In that paper they show that a sequent calculus similar to that in Figure 1 has cut-elimination.

$$\frac{}{A \xrightarrow{a\in\mathcal{C}} B} \qquad \frac{X \xrightarrow{f_0} Y_0 \qquad X \xrightarrow{f_1} Y_1}{X \xrightarrow{\langle f_0, f_1\rangle} Y_0 \times Y_1} \qquad \frac{X_i \xrightarrow{f} Y}{X_0 \times X_1 \xrightarrow{f\circ\pi_i} Y}$$

$$\frac{}{\mathbf{0} \xrightarrow{?} X} \qquad \frac{X \xrightarrow{f} Y \qquad Y \xrightarrow{g} Z}{X \xrightarrow{g\circ f} Z}\ \text{cut}$$

$$\frac{}{X \xrightarrow{!} \mathbf{1}} \qquad \frac{X_0 \xrightarrow{f_0} Y \qquad X_1 \xrightarrow{f_1} Y}{X_0 + X_1 \xrightarrow{[f_0, f_1]} Y} \qquad \frac{X \xrightarrow{f} Y_i}{X \xrightarrow{\iota_i \circ f} Y_0 + Y_1}$$

Figure 1. A sequent calculus for $\Sigma\Pi(\mathcal{C})$-maps

Although the calculus in Figure 1 provides normal forms for term representations of maps, these are not generally *canonical*. Rather, a categorical map is represented by a class of terms, related to one another by a range of permutations induced by the categorical laws. But since the set of normal terms for maps in a given hom-set is finite—and checking whether two terms are permutations of each other is simple—term equality is at least decidable. A recent paper [1] by Cockett and Santocanale shows that the problem of term equality in $\Sigma\Pi$-categories is also tractable.

One reason why this result is far from obvious is the unpredictable behaviour of the units, the initial object $\mathbf{0}$ and the terminal object $\mathbf{1}$. This is not surprising: characterising the units is notoriously difficult also in linear logic, of which sum-product logic is a fragment—that of sequents $X \vdash Y$ where X and Y are strictly additive formulae.

[1] For an introduction to products and coproducts see [5]

The project that I will talk about, which is ongoing research, uses an elegant notion of *proof nets* to approach the problem of term equality in sum-product categories. These nets can be seen as proof nets for the sequent calculus in Figure 1. Specifically, in the case without units these proof nets are a fragment of the MALL-nets of Hughes and Van Glabbeek [3], and provide canonical representations of categorical maps.

For the full sum-product logic these nets are not canonical. However, switching to proof nets provides a clearer picture of the difficulties surrounding the units, by factoring out the permutations that do not involve them. Although the remaining equational theory is by no means trivial, current investigations suggest that it can be decided by a remarkably simple algorithm. This suggestion is backed by extensive testing on an implementation of the algorithm, but unfortunately a conclusive proof is still forthcoming.

The proof nets in this project are interesting, firstly, because they clarify many of the difficulties and peculiarities encountered in the logic of sum-product categories. Secondly, they provide an interesting take on the problems posed by the additive units of linear logic, and may well inspire new ways of dealing with them.

Acknowledgements

Many thanks go out to Alex Simpson for guidance and support. Thanks also to Jeff Egger for helpful comments.

References

[1] J.R.B. Cockett and L. Santocanale: On the word problem for ΣΠ-categories, and the properties of two-way communication. ArXiv: 0904.1529v1 (2009)

[2] J.R.B. Cockett and R.A.G. Seely: Finite sum–product logic. Theory and Applications of Categories, Vol. 8 (5), pp. 63–99 (2001)

[3] D.J.D. Hughes and R.J. van Glabbeek: Proof nets for unit-free multiplicative-additive linear logic. Transactions on Computational Logic, Vol. 6 (4), pp. 784–842 (2005)

[4] A. Joyal: Free bicomplete categories. C.R. Math. Rep. Acad. Sci. Canada, Vol. 17 (5), pp. 219–224 (1995)

[5] S. Mac Lane: Categories for the working mathematician. Second edition. Graduate Texts in Mathematics, Vol. 5, Springer-Verlag, New York (1998)

Subrecursive Degrees and Statements Independent of PA

Lars Kristiansen[1]
(joint work with Jan-Christoph Schlage-Puchta[2] and Andreas Weiermann[2])

[1] Department of Mathematics, University of Oslo
[2] Department of Mathematics, Gent University

An honest function is a monotone increasing function with a Kalmar elementary graph. We study the structure of honest elementary degrees, that is, the degree structure induced on the honest functions by the reducibility relation "being Kalmar elementary in".

Theorem 1 (The Growth Theorem). *An honest function f is elementary in an honest function g iff there exists a fixed number k such that $f(x) \leq g^k(x)$.*

Theorem 1 makes it possible to abandon classical computability-theoretic constructions (involving enumerations, diagonalisation, etc.) and investigate the structure of elementary honest degrees by asymptotic analysis and methods of number theoretic nature. The structure turns out to be a distributive lattice with strong density properties. We have a jump operator and canonical degrees $\mathbf{0} < \mathbf{0}' < \mathbf{0}'' < \ldots$. Low$_n$, high$_n$ and intermediate degrees exist with respect to these canonical degrees. For more on elementary honest degrees, see [1] and [2].

Our methods and proof techniques can be generalised to work for weaker subrecursive reducibility relations, including the relation $\leq_{PA}$ where $f \leq_{PA} g$ iff f is provably total in the first order theory $PA + Tot(g)$. The degree structure induced on the honest ordinal recursive functions by $\leq_{PA}$ is expected to be very similar to the structure of elementary honest degrees. This entails a number of interesting independence results for PA, e.g. the next theorem.

Theorem 2. *There exist two computable functions f_0 and f_1 not provable total in PA such that any function provable total in both $PA + Tot(f_0)$ and $PA + Tot(f_1)$ also will be provable total in PA.*

References

1. Kristiansen, L.: A jump operator on honest subrecursive degrees. Archive for Mathematical Logic **37** (1998) 105–125
2. Kristiansen, L.: Subrecursive degrees and fragments of Peano Arithmetic. Archive for Mathematical Logic **40** (2001) 365–397

A Note on the Abnormality of Realizations of S4LP

Roman Kuznets*

Institut für Informatik und angewandte Mathematik
Universität Bern
kuznets@iam.unibe.ch

Justification logics [Art08] are essentially refined analogs of modal epistemic logics. Whereas a modal epistemic logic uses the formula $\Box F$ to indicate that F is known to be true, a justification logic uses $t{:}F$ instead, where t is a term that describes a 'justification' or proof of F. The structure of justification terms t depends on which modal logic needs to be represented in this explicit format. This structure notwithstanding, the formal correspondence between a modal logic ML and its justification counterpart JL is given by a *realization theorem.* It has two directions: first, each theorem of ML can be turned into a theorem of JL by *realizing* all occurrences of the modality $\Box$ with appropriate justification terms; second and vice versa, if all terms in a theorem of JL are replaced with $\Box$, a process called the *forgetful projection*, then the resulting modal formula is provable in ML.

Such correspondences have been established for many normal modal logics between K and S5 (see [Art08]). The first such result was established by Artemov [Art95, Art01] between the modal logic S4 and the so-called Logic of Proofs LP. The idea behind the realization process is that $\Box F$ is interpreted as "there exists a proof of F." Under this interpretation, each modal formula becomes a first-order statement with quantifiers over proofs. The realization theorem for a particular justification logic then states that the logic's operations on terms are rich enough to represent all Skolem functions necessary for Skolemizing valid modal statements. It is, therefore, natural to add a restriction that different negative occurrences of $\Box$, which are interpreted as universal quantifiers over proofs, be realized by distinct justification variables since the Skolemization process replaces such quantifiers by distinct Skolem variables. This additional property of realization is called *normality*, and all justification logics that enjoy a realization theorem do enjoy it in the strong sense that every modal theorem can be realized normally.

In a series of papers culminating in [AN05], Artemov and Nogina developed a logic S4LP that combines modal representation of knowledge as in S4 with justification terms of LP. The connection between implicit modal knowability and explicit evidence terms in this logic is given by the *connection principle* $t : F \to \Box F$ that essentially states that "whatever is known for a reason t must be known."

In their very first paper on the subject, Artemov and Nogina posed the following question about the realization theorem for S4LP: *Whether* [S4LP] *enjoys the realization property: given a derivation D in* [S4LP] [...] *one could find a realization r of all occurrences of $\Box$ in D* [...] *such that the resulting formula F^r is derivable in* [LP]*?*[1] (see [AN04, Problem 2][2]).

However, there are reasons to doubt whether this formulation is the right one. Suppose, we want to realize a theorem $t{:}\Box F \to s{:}\Box G$. The formulation above suggests that the realization must be of the form $t{:}t'{:}F^r \to s{:}s'{:}G^r$ for some terms t' and s' (or $t{:}x{:}F^r \to s{:}s'{:}G^r$ for some justification variable x and some term s' if the normality condition is imposed). This, however, changes the meaning of terms t and s: according to the connection principle, the statements justified by them become stronger; as a result, the assumption is weakened while the conclusion is simultaneously strengthened. In this note, we formalize this objection by proving that the realization theorem for S4LP does not hold if the requirement of normality is imposed.

* Supported by Swiss National Science Foundation grant 200021–117699.

[1] The original problem is more precise in that it is formulated for particular *constant specifications.* However, the phenomenon we are going to describe is completely independent of constant specifications, hence, we omit them from both the formulation of the problem and from the following discussion. In fact, our result holds for an arbitrary constant specification.

[2] The logic was called LPS4 there.

Theorem 1. *Theorem* $z\!:\!\neg\Box(P \vee Q) \to z\!:\!\neg\Box(P \vee Q) \wedge \neg\Box P$ *of* S4LP*, where* z *is a justification variable,* P *and* Q *are propositional letters, does not have a normal realization in* LP*.*

This example of a theorem without a normal realization is inspired by work of Ghari [Gha09]. The proof of the theorem is based on the semantics developed by Mkrtychev [Mkr97] for LP.

Definition 2. *An* M-model $\mathcal{M}$ *is a pair* $(\mathcal{E}, V)$*, where* $V\colon \mathrm{Prop} \to \{\mathit{True}, \mathit{False}\}$ *is a* valuation function *and* $\mathcal{E}\colon \mathrm{Tm} \to 2^{\mathrm{Fm}}$ *is an* evidence function *that satisfies several closure conditions that can be found in [Mkr97] and are omitted here for space considerations. Truth for propositional letters and for Boolean connectives is defined in the standard way;* $\mathcal{M} \Vdash t : F$ *iff* $\mathcal{M} \Vdash F$ *and* $F \in \mathcal{E}(t)$*.*

Theorem 3 ([Mkr97]). *A justification formula is a theorem of* LP *iff it is valid in all M-models.*

Instead of giving the full definition of closure conditions, we will use the following lemma that easily follows from them.

Lemma 4. *For any requirements* $F_i \in \mathcal{E}(t_i)$*,* $i = 1, \ldots, n$*, on the evidence function, there exists a unique minimal function that satisfies the requirements. Moreover, for this minimal function* $\mathcal{E}(x) = \{F_i \mid x = t_i\}$ *for any justification variable* x*.*

Proof (of Theorem 1.). The normality condition requires both negative occurrences of $\Box$ in the given theorem of S4LP to be realized by distinct justification variables, say x and y, whereas the only positive $\Box$ can be realized by an arbitrary term t:

$$z\!:\!\neg t\!:\!(P \vee Q) \to z\!:\!\neg x\!:\!(P \vee Q) \wedge \neg y\!:\!P \ . \tag{1}$$

It is easy to refute (1) if $t \neq x$. Indeed, let $V(P) = V(Q) = \mathit{False}$ and let $\mathcal{E}$ be the minimal evidence function such that $\neg t\!:\!(P \vee Q) \in \mathcal{E}(z)$. Then, $\mathcal{M} \nVdash t\!:\!(P \vee Q)$ simply because both P and Q are false. Therefore, the antecedent of the implication holds. However, for $t \neq x$, the first conjunct in the consequent is false since, by Lemma 4, $\neg x\!:\!(P \vee Q) \notin \mathcal{E}(z)$.

Thus, $t = x$, and the only normal realization possible is $z\!:\!\neg x\!:\!(P \vee Q) \to z\!:\!\neg x\!:\!(P \vee Q) \wedge \neg y\!:\!P$. Here is a model that refutes it. Let $V(P) = \mathit{True}$ and $\mathcal{E}$ be the minimal evidence function such that $P \in \mathcal{E}(y)$ and $\neg x\!:\!(P \vee Q) \in \mathcal{E}(z)$. Then, by Lemma 4, $P \vee Q \notin \mathcal{E}(x)$, hence, $\mathcal{M} \nVdash x\!:\!(P \vee Q)$, which is sufficient to make the antecedent true. On the other hand, the second conjunct of the consequent is clearly false. □

Whether the realization theorem holds for S4LP without the normality condition remains an open question.

References

[AN04] Sergei [N.] Artemov and Elena Nogina. Logic of knowledge with justifications from the provability perspective. Technical Report TR–2004011, CUNY Ph.D. Program in Computer Science, August 2004.

[AN05] Sergei [N.] Artemov and Elena Nogina. Introducing justification into epistemic logic. *Journal of Logic and Computation*, 15(6):1059–1073, December 2005.

[Art95] Sergei N. Artemov. Operational modal logic. Technical Report MSI 95–29, Cornell University, December 1995.

[Art01] Sergei N. Artemov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, 7(1):1–36, March 2001.

[Art08] Sergei [N.] Artemov. The logic of justification. *The Review of Symbolic Logic*, 1(4):477–513, December 2008.

[Gha09] Meghdad Ghari. Realization property for S4LP and S4LPN. Manuscript, 2009.

[Mkr97] Alexey Mkrtychev. Models for the logic of proofs. In Sergei Adian and Anil Nerode, editors, *Logical Foundations of Computer Science, 4th International Symposium, LFCS'97, Yaroslavl, Russia, July 6–12, 1997, Proceedings*, volume 1234 of *Lecture Notes in Computer Science*, pages 266–275. Springer, 1997.

A framework for expressing sequent proofs

Richard McKinley

IAM, Universität Bern

Abstract. The Curry-Howard isomorphism is most fully developed as a connection between typed *functional* programs and proofs in *natural deduction*. What computational interpretations exist for *sequent systems* are, in most cases, for restricted sequent systems with a distinguished conclusion/premise (for example, a *stoup*). The programs corresponding to proofs in these systems are functional programs together with some additional features, evaluated in some abstract machine. The work I will discuss is part of a project to give computational interpretations as *processes* to genuinely multiple-succedent sequent proofs. We present a system of untyped higher-order sequent-style derivations (the *Computational Sequent Calculus*), in which some familiar and not-so-familiar sequent systems can be expressed as type systems; among them Herbelin's $\bar{\lambda}$ calculus, Linear Logic, and a multiple-conclusioned calculus for classical logic.

Monotone schemes within FPSPACE

Isabel Oitavem

(joint work with Amir Ben-Amram and Bruno Loff)

Abstract

An algorithm working in polynomial space is allowed to reuse space, and if we look at known algorithms for PSPACE-complete problems, they always seem to rely heavily on this possibility. Our intuition then indicates that this is a crucial point concerning the problem PTIME versus PSPACE. A rigorous formulation of this intuition is the known fact that a "write-once" Turing machine, given polynomial space, decides exactly PTIME. A write-once machine is one which is not allowed to erase (or rewrite) cells which have been previously written on.

Since the write-once restriction can, in some sense, be seen as a monotonicity constraint on the contents of the storage, this suggests investigating how sensitive characterisations of PSPACE are to "monotonicity constraints".

One explores, in particular, the interplay between recursion and iteration. We take as starting point a recursion-theoretic characterisation of FPSPACE, and we show that substituting predicative primitive iteration for predicative primitive recursion also leads to FPSPACE

We show that imposing a monotonicity constraint on the above recursion and iteration operators leads, in the case of primitive iteration, to FPTIME, and, in the case of primitive recursion, to the polynomial hierarchy FPH. We form a hierarchy based on the nesting-level of the restricted primitive recursion operator, and this provides a new implicit characterisation of all levels of the polynomial hierarchy.

Inversion of Logical Rules by Definitional Reflection

Thomas Piecha
Universität Tübingen
Wilhelm-Schickard-Institut
piecha@informatik.uni-tuebingen.de

Wagner de Campos Sanz
Universidade Federal de Goiás
Faculdade de Filosofia
sanz@fchf.ufg.br

The inversion principle expresses a relationship between left and right introduction rules for logical constants. Hallnäs & Schroeder-Heister [2] have been presenting the principle of definitional reflection as a means of capturing the idea embodied in the inversion principle. Using the principle of definitional reflection, we show for minimal propositional logic that the left introduction rules are admissible when the right introduction rules are given as the definition of logical constants, and vice versa (cf. de Campos Sanz & Piecha [1]).

Following Schroeder-Heister [3], the inversion principle is based on the idea that if we have certain defining rules $\alpha \Leftarrow \beta_1^1, \ldots, \beta_{n_1}^1, \ldots, \alpha \Leftarrow \beta_1^k, \ldots, \beta_{n_k}^k$ for some atom α, then a rule $\gamma \Leftarrow \alpha$ with premiss α and conclusion γ is justified if γ is consequence of each *defining condition* Γ_i of α, where $\Gamma_i = \beta_1^i, \ldots, \beta_{n_i}^i$; that is, if γ is derivable from each Γ_i, then γ is derivable from α.

This principle can be stated by means of a sequent calculus inference (cf. Hallnäs & Schroeder-Heister [2]) and is then called the *principle of definitional reflection* $(\mathcal{D}\vdash)$:

$$(\mathcal{D}\vdash)\ \frac{\Delta_1, \Gamma_1 \vdash \gamma \quad \ldots \quad \Delta_k, \Gamma_k \vdash \gamma}{\Delta_1, \ldots, \Delta_k, \alpha \vdash \gamma} \qquad \text{(definitional reflection)}$$

where for the set $\mathcal{D}(\alpha)$ of the defining conditions of α and for substitutions σ of vaiables by terms the condition $\mathcal{D}(\alpha^\sigma) \subseteq (\mathcal{D}(\alpha))^\sigma$ has to be obeyed. It has the form of a left introduction rule for atoms α defined by definitional clauses with bodies $\Gamma_1, \ldots, \Gamma_k$, and it is thus a way of stating the inversion principle for definitions. It complements the *principle of definitional closure* $(\vdash\mathcal{D})$ for the corresponding right introduction of defined atoms.

When both principles are added as inference principles for atoms to a given logical system $\mathscr{L}$, we obtain an extended system $\mathscr{L}(\mathcal{D})$, which is a definitional logic based on definition $\mathcal{D}$. The definitional clauses are then the basis for sequent style right and left introduction inferences. A natural candidate for an underlying system $\mathscr{L}$ consists of the structural inferences *identity* (Id), *thinning* (Thin) and *cut* (Cut):

$$(\text{Id})\ \frac{}{A \vdash A} \qquad (\text{Thin})\ \frac{\Delta \vdash A}{B, \Delta \vdash A} \qquad (\text{Cut})\ \frac{\Delta \vdash C \quad C, \Sigma \vdash A}{\Delta, \Sigma \vdash A}$$

Concerning a given rule R and a given definition $\mathcal{D}$, rule R is *admissible* in $\mathcal{D}$, if for every α the implication "if $\Vdash_{\mathcal{D}+R} \alpha$, then $\Vdash_{\mathcal{D}} \alpha$" holds. The principles of definitional reflection and definitional closure can be interpreted as principles for admissibility (cf. Schroeder-Heister [3]) if sequents $\beta_1, \ldots, \beta_n \vdash \alpha$ are interpreted as stating the admissibility of rules $\alpha \Leftarrow \beta_1, \ldots, \beta_n$ relative to a given definition $\mathcal{D}$. For the principle of definitional reflection $(\mathcal{D}\vdash)$ consider the rule $\gamma \Leftarrow \alpha, \Delta_1, \ldots, \Delta_k$ which corresponds to the conclusion of definitional reflection. Then α was derived by a rule $\alpha \Leftarrow \beta_1^i, \ldots, \beta_{n_i}^i$, for some i, in the last step and $\beta_1^i, \ldots, \beta_{n_i}^i$ were derived in previous steps (likewise for $\Delta_1, \ldots, \Delta_k$). Thus, if the rules $\gamma \Leftarrow \beta_1^i, \ldots, \beta_{n_i}^i, \Delta_i$ (corresponding to the premisses of definitional reflection) are admissible, then the rule corresponding to the conclusion of definitional reflection is admissible as well since all consequences γ following from $\beta_1^i, \ldots, \beta_{n_i}^i, \Delta_i$ should be consequences of α.

Sequent calculus rules can be understood as definitions for logical constants. For the right and left introduction rules we use the following representation for object language sequents $\mathfrak{s}$, called *o-sequents*: $\overset{\Omega}{\underset{A}{\triangledown}}$ ("A follows from Ω"). The o-sequents are to be distinguished from the sequents in the framework $\mathscr{L}(\mathcal{D})$, which are called *f-sequents* and are expressed with the turnstile '$\vdash$'. Finite sets of o-sequents are denoted by $\mathfrak{S}$. Our aim is to represent sequent style minimal propositional logic. This is why o-sequents have exactly one formula at the bottom; it corresponds to the succedent of sequents. What is written on top (corresponding to the antecedent of sequents) is either a (possibly empty) finite multiset of formulas or a comma-separated list of such sets, the comma representing multiset union. The sequent symbol '$\triangledown$' represents the relation of deductive consequence. Hence, the logical constants will be defined in the context of deductive consequence.

The properties of the usual deductive consequence relation are captured in sequent calculus by the inferences *identity*, *thinning* and *cut*. These inferences for o-sequents are combined in the axiom (Ax) (which incorporates also (Id) and (Thin) for f-sequents) of the following framework $\mathscr{F}(\mathcal{D})$:

$$(\mathrm{Ax})\ \frac{}{\mathfrak{S},\ \begin{array}{c}\Psi_1\\ \triangledown\\ A_1\end{array},\ \begin{array}{c}\Psi_2, A_1\\ \triangledown\\ A_2\end{array},\ \dots,\ \begin{array}{c}\Psi_n, A_{n-1}\\ \triangledown\\ A_n\end{array}\ \vdash\ \begin{array}{c}\Omega, \Psi_1, \Psi_2, \dots, \Psi_n\\ \triangledown\\ A_n\end{array}} \qquad (\mathrm{Cut_{Id}})\ \frac{\mathfrak{S},\ \begin{array}{c}A\\ \triangledown\\ A\end{array} \vdash \mathfrak{s}}{\mathfrak{S} \vdash \mathfrak{s}}$$

together with the principles of definitional reflection ($\mathcal{D}\vdash$) and definitional closure ($\vdash\mathcal{D}$). Instead of (Cut) the inference rule ($\mathrm{Cut_{Id}}$) is used, which allows only for cuts on o-sequents of the form $\begin{array}{c}A\\ \triangledown\\ A\end{array}$.

For the right introduction rules, that is, rules for the introduction of a logical constant in the bottom of an o-sequent, we can derive f-sequents of the form $\mathfrak{S} \vdash \mathfrak{s}$ representing the left introduction rules, that is, rules for the introduction of a logical constant in the top of an o-sequent, inside the framework $\mathscr{F}(\mathcal{D})$ by using the corresponding definitional reflections (cf. de Campos Sanz & Piecha [1]).

Definitional clauses can be given also for the left introduction rules. Then for each left introduction rule the admissibility of the corresponding right introduction rule can be shown within $\mathscr{F}(\mathcal{D})$. As an example we show for the given definitional clause $\mathcal{D}^{\rightarrow}$ of left implication introduction the admissibility of the right implication introduction rule by deriving the corresponding f-sequent in $\mathscr{F}(\mathcal{D}^{\rightarrow})$. The definitional clause $\mathcal{D}^{\rightarrow}$ for the left implication introduction rule and the corresponding definitional reflection ($\mathcal{D}^{\rightarrow}\vdash$) are as follows:

$$\mathcal{D}^{\rightarrow}\left\{\begin{array}{c}\Omega, A\rightarrow B\\ \triangledown\\ C\end{array} \Leftarrow \begin{array}{c}\Omega\\ \triangledown\\ A\end{array},\ \begin{array}{c}\Omega, B\\ \triangledown\\ C\end{array}\right. \qquad (\mathcal{D}^{\rightarrow}\vdash)\ \frac{\mathfrak{S},\ \begin{array}{c}\Omega\\ \triangledown\\ A\end{array},\ \begin{array}{c}\Omega, B\\ \triangledown\\ C\end{array} \vdash \mathfrak{s}}{\mathfrak{S},\ \begin{array}{c}\Omega, A\rightarrow B\\ \triangledown\\ C\end{array} \vdash \mathfrak{s}}$$

The derivation showing the admissibility of the right implication introduction rule is then:

$$(\mathrm{Cut_{Id}})\ \frac{(\mathcal{D}^{\rightarrow}\vdash)\ \dfrac{(\mathrm{Ax})\ \dfrac{}{\begin{array}{c}\Theta, A\\ \triangledown\\ B\end{array},\ \begin{array}{c}\\ \triangledown\\ A\end{array},\ \begin{array}{c}B\\ \triangledown\\ A\rightarrow B\end{array} \vdash \begin{array}{c}\Theta\\ \triangledown\\ A\rightarrow B\end{array}}}{\begin{array}{c}\Theta, A\\ \triangledown\\ B\end{array},\ \begin{array}{c}A\rightarrow B\\ \triangledown\\ A\rightarrow B\end{array} \vdash \begin{array}{c}\Theta\\ \triangledown\\ A\rightarrow B\end{array}}}{\begin{array}{c}\Theta, A\\ \triangledown\\ B\end{array} \vdash \begin{array}{c}\Theta\\ \triangledown\\ A\rightarrow B\end{array}}$$

The logical constants of minimal propositional logic can be defined by right introduction rules as well as by left introduction rules. If the right introduction rules are given as definitions, then the left introduction rules are consequences of them in the sense of being admissible relative to the given definitions, and if the left introduction rules are given as definitions, then the right introduction rules are consequences of them in the same sense of being admissible.

References

[1] de Campos Sanz, W. & Piecha, T. (2009). Inversion by Definitional Reflection and the Admissibility of Logical Rules. *The Review of Symbolic Logic*, **2**, 550–569.

[2] Hallnäs, L. & Schroeder-Heister, P. (1990/91). A Proof-Theoretic Approach to Logic Programming. I. Clauses as Rules. *Journal of Logic and Computation*, **1**, 261–283 (1990); II. Programs as Definitions. *Ibid.*, 635–660 (1991).

[3] Schroeder-Heister, P. (2007). Generalized Definitional Reflection and the Inversion Principle. *Logica Universalis*, **1**, 355–376.

Strong normalization and confluence for reflexive combinatory logic

Daniyar S. Shamkanov
daniyar.shamkanov@gmail.com

The story of reflexive combinatory logic RCL [2, 3] started with the invention of proof carrying formulas and the logic of proofs LP [1]. Proof carrying formulas seemingly bring into a system of types the possibility to operate simultaneously with objects of different abstraction level: functions, high level programs, low level codes, etc. In addition, the internalization property of LP and Hurry-Howard isomorphism provoked the idea to design a typed system capable to represent its own derivations by its own typed terms. RCL was introduced to meet these expectations.

Reflexive combinatory logic RCL extends typed combinatory logic CL by a new type constuctor $t : F$ with the intended interpretation 't has type F'. Any type of the form $t : F$ has a canonical element $!t$ which represents the high level description of t or the term t supplied with additional metadata(for example, this can be data about types for all subterms of t). Futhermore, RCL contains new combinators:

$$\mathbf{d}^{u:F\to F}, \quad \mathbf{o}^{u:(F\to G)\to(v:F\to uv:G)}, \quad \mathbf{c}^{u:F\to !u:u:F}$$

The combinator $\mathbf{d}^{u:F\to F}$ maps the high level description of an object u into u itself. The combinator $\mathbf{o}^{u:(F\to G)\to(v:F\to uv:G)}$ implements application on high level descriptions(or terms with metadata). $\mathbf{c}^{u:F\to !u:u:F}$ maps the description into the higher description.

Reflexive combinatory logic was designed as a system capable to iterate the type assignment but its operational aspects remained to be clarified. Consider the following reduction rules:

$$\mathbf{k}uv \mapsto u, \quad \mathbf{s}uvw \mapsto (uw)(vw), \quad \mathbf{d}!u \mapsto u, \quad \mathbf{c}!u \mapsto !!u, \quad \mathbf{o}(!u)(!v) \mapsto !(uv)$$

The unrestricted application of the reduction rules may transform a well formed expression into the illegal one, thereby we will use the simultaneous contraction. The application of a reduction $a \mapsto b$ to an expression e will be denoted by $e[a \mapsto b]$ and means simultaneous replacement of all occurrences of a in e by b. This definition secures well formedness preservation under the reductions for expressions without the combinator $\mathbf{o}^F$. To provide the preservation for all expressions, V. N. Krupski proposed to extend RCL to RCL^+ with the following conditions:

- if $\mathbf{o}^F$ is a combinator and $a \mapsto b$ is a reduction, then $\mathbf{o}^{F[a \mapsto b]}$ is also a combinator,

- if $a \mapsto b$ and $c \mapsto d$ are reductions and a is not graphically equal to c, then $a[c \mapsto d] \mapsto b[c \mapsto d]$ is also a reduction.

Theorem 0.1. *RCL^+ has strong normalization and confluence properties.*

References

[1] S. Artemov. Explicit provability and constructive semantics. *Bulletin of Symbolic Logic*, **7**:1 (2001), 1–36.

[2] S. Artemov. Kolmogorov and Gödel's approach to intuitionistic logic: current developments. *Russian Mathematical Surveys*, **59**:2 (2004), 203–229.

[3] N. Krupski. Typing in Reflective Combinatory Logic. *Annals of Pure and Applied Logic*, **141**: 1-2 (2006), 243–256.

On the p-equivalence of extension and substitution in deep inference

Lutz Straßburger

INRIA Saclay - Île-de-France — Équipe-projet Parsifal

École Polytechnique — LIX — Rue de Saclay — 91128 Palaiseau Cedex — France
`lutz at lix dot polytechnique dot fr`

Abstract. Since the work of Cook, Reckhow, Krajícek, and Pudlák, it is known that Frege systems with extension and Frege systems with substitution p-simulate each other. Bruscoli and Guglielmi studied the concepts of extension and substitution also in framework of deep inference. However, for showing the p-equivalence between the two, they rely on the result by Krajícek and Pudlák. Furthermore, substitution in deep inference is weaker than substitution in Frege systems. In this talk I will give an alternative way of presenting extension and substitution in a deep inference setting, and then use deep inference to give a new proof of the Krajícek-Pudlák result on the p-equivalence of substitution and extension for Frege systems.

Program

Friday, June 18th

9:00	Registration
9:30	Lev Beklemishev: Provability algebras and scattered topology
10:30	Coffee
11:00	Evgenij Dashkov: On positive fragments of polymodal provability logic
11:30	Lars Kristiansen: Subrecursive Degrees and Statements Independent of PA
12:00	Isabel Oitavem: Monotone schemes within FPSPACE
12:30	Lunch
14:30	Agata Ciabattoni: Systematic (and algebraic) proof theory for substructural logics
15:30	Coffee
16:00	Roy Dyckhoff: Cut elimination, substitution and normalisation
16:30	Richard McKinley: A framework for expressing sequent proofs
17:00	Lutz Straßburger: On the p-equivalence of extension and substitution in deep inference
19:00	Dinner at Altes Tramdepot

Saturday, June 19th

10:00	Rosalie Iemhoff: Unification in nonclassical theories
11:00	Coffee
11:30	Roman Kuznets: A Note on the Abnormality of Realizations of S4LP
12:00	Daniyar Shamkanov: Strong normalization and confluence for reflexive combinatory logic
12:30	Lunch
14:30	Nicolas Guenot: Normalisation by Interaction and Computational Interpretations
15:00	Tom Gundersen: Breaking Paths in Atomic Flows for Classical Logic
15:30	Coffee
16:00	Thomas Piecha: Inversion of Logical Rules by Definitional Reflection
16:30	Willem Heijltjes: Proof Nets for Free Sum-Product Categories