Computing with intuitionistic sequent proof terms: progress report

José Espírito Santo and Maria João Frade and Luís Pinto

University of Minho, Portugal

It is well-known that the formulation as sequent calculi of logics and type theories is to be preferred, if one is interested in a formalism suitable for proof search [13, 10]. But in type theories one needs not only to search for proofs and proof terms, but also to compute with them. Now it is not too controversial to say that not everything is understood regarding computation with proof terms in the sequent calculus format, as progress in the matter is still seen recently, e.g. either in the computational interpretation of cut-elimination on focused proofs [15, 1], or in the understanding of variables in sequent proof terms [4]. Hence "structural" matters about sequent proof terms still hinder the formulation of type theories as sequent calculi.

In the past decade two of the authors proposed and studied the system $\lambda\mathbf{Jm}$ as a vehicle for studying reduction procedures in the sequent calculus [6, 7]. The approach was modular, with the system designed to be as simple as possible, so that only the intricacies of the reduction procedures remained: the logic was the simplest one (intuitionistic implications as sole connective); the cut=redex paradigm [8, 2] was not followed, so that variables in proof terms could be treated as ordinary term variables [4]; substitution was treated as a meta-operation, with the corresponding cut-rule treated as an admissible typing rule, from which the call-by-name character of cut-elimination followed [2].

But, in order not to fall in mere natural deduction with generalized elimination [16, 9], actual use of the formulas in the l.h.s. of sequents was permitted in the inference process of $\lambda\mathbf{Jm}$. This entails, at the level of proof terms, the existence of a primitive mechanism of vectorization of arguments, familiar from the $\overline{\lambda}$-calculus [8]; and, at the level of reduction, not only that cut-elimination corresponds to the "multiary"[14] version of the $\beta\pi$-reduction found in λ-calculus with generalized applications [9], but also the existence of a second reduction process. The latter may be seen as the η-reduction for the $\tilde{\mu}$-operator [2], but is named μ-reduction, as in [14], where it was introduced.

Finally, a third process of reduction, most typical of sequent calculus, has been permanently considered in the study of $\lambda\mathbf{Jm}$: permutative conversion [3, 14]. The three reduction procedures were studied in isolation for their properties and computational interpretation, but also in their possible combinations. At some point, the authors attempted (in [5]) not only some systematization of the multiplicity of subsystems and kinds of normal forms that such combinations give rise to, but also some harmony out of the syntactic noise and explosion. Recent progress allows us to say now that we could have done better, and that is what we intend to report in this talk.

Following [12, 3], let us call *normal* a cut-free term that is irreducible for permutative conversion (the justification of the terminology is that normal terms are in bijection with normal natural deductions). Normal terms have a characterization that is applicable to terms in general (not necessarily cut-free). We call *natural* the terms which enjoy such characterization, so that a sequent proof term is normal iff it is natural and cut-free. Natural terms are closed for cut-elimination and μ-reduction. We can now give a simple and transparent computational interpretation of this subsystem: non-values consist of a term "applied" to a kind of generalized argument, consisting of a list of lists of ordinary arguments, with cut-elimination allowing the call of a function with the first argument of the first list (β rule), or appending two such lists of lists (π rule), and with μ being an operation of flattening. So, the generality of generalized applications is here reduced to a second vectorization mechanism. In addition, we characterize proof search for normal proofs, identifying the relaxation of the LJT focusing discipline [8, 11] that it follows.

We also embrace the view that permutative conversion is a process of *conversion to natural form*, named γ for short, and amend the definition of γ found in [5] after having realized that the substitution process involved is a refinement of ordinary term substitution. Conversion to natural form is then studied systematically together with cut-elimination and μ-reduction to know when a procedure commutes and/or preserves another. Based on this analysis, we can conclude that a proof in $\lambda\mathbf{Jm}$ determines not one, but eight possibly distinct cut-free proofs. We also see how to combine γ-reduction and μ-reduction in order to define *focalization* - a process of reduction to the LJT-form - and observe the commutation of cut-elimination with focalization. Finally, since conversion to natural form commutes with cut-elimination, we see that the two immediate senses for the concept of *normalization* in $\lambda\mathbf{Jm}$, either conversion of cut-free terms to normal form, or cut-elimination in the natural subsystem, are coherent and have a common generalization to the entire set of proof terms.

References

[1] T. Brock-Nannestad, N. Guenot, and D. Gustafsson. Computation in focused intuitionistic logic. In *Proceedings of PPDP'15*, pages 43–54. ACM, 2015.

[2] P.-L. Curien and H. Herbelin. The duality of computation. In *Proceedings of International Conference on Functional Programming 2000*. IEEE, 2000.

[3] R. Dyckhoff and L. Pinto. Permutability of proofs in intuitionistic sequent calculi. *Theoretical Computer Science*, 212:141–155, 1999.

[4] J. Espírito Santo. Curry-Howard for sequent calculus at last! In *Proceedings TLCA'15*, volume 38 of *LIPIcs*, pages 165–179. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2015.

[5] J. Espírito Santo, M. J. Frade, and L. Pinto. Structural proof theory as rewriting. In *Proceedings of RTA 2006*, volume 4098 of *LNCS*, pages 197–211. Springer-Verlag, 2006.

[6] J. Espírito Santo and L. Pinto. Confluence and strong normalisation of the generalised multiary λ-calculus. In *Proceedings of TYPES'04 2003*, volume 3085 of *LNCS*. Springer-Verlag, 2004.

[7] J. Espírito Santo and L. Pinto. A calculus of multiary sequent terms. *ACM Trans. Comput. Log.*, 12(3):22, 2011.

[8] H. Herbelin. A λ-calculus structure isomorphic to a Gentzen-style sequent calculus structure. In *Proceedings of CSL'94*, volume 933 of *LNCS*, pages 61–75. Springer-Verlag, 1995.

[9] F. Joachimski and R. Matthes. Standardization and confluence for a lambda calculus with generalized applications. In *Proc. of RTA'00*, volume 1833 of *LNCS*, pages 141–155. Springer, 2000.

[10] S. Lengrand, R. Dyckhoff, and J. McKinna. A focused sequent calculus framework for proof search in pure type systems. *Logical Methods in Computer Science*, 7(1), 2011.

[11] C. Liang and D. Miller. Focusing and polarization in linear, intuitionistic, and classical logic. *Theoretical Computer Science*, 410:4747–4768, 2009.

[12] G. Mints. Normal forms for sequent derivations. In P. Odifreddi, editor, *Kreiseliana*, pages 469–492. A. K. Peters, Wellesley, Massachusetts, 1996.

[13] D. Pym. A note on the proof theory of the lambda-π-calculus. *Studia Logica*, 54(2):199–230, 1995.

[14] H. Schwichtenberg. Termination of permutative conversions in intuitionistic Gentzen calculi. *Theoretical Computer Science*, 212, 1999.

[15] R. J. Simmons. Structural focalization. *ACM Trans. Comput. Log.*, 15(3):21:1–21:33, 2014.

[16] J. von Plato. Natural deduction with general elimination rules. *Annals of Mathematical Logic*, 40(7):541–567, 2001.