Lambda terms for natural deduction, sequent calculus and cut elimination

Henk Barendregt* and Silvia Ghilezan†

Abstract

It is well-known that there is a good correspondence between natural deduction derivations and typed lambda terms. The relation between lambda terms and derivations in sequent calculus, between normal lambda terms and cut-free derivations in sequent calculus and finally between normalization of terms and cut elimination of derivations has been observed by several authors (Prawitz [1965], Zucker [1974] and Pottinger [1977]). This relation is less perfect because several cut-free sequent derivations correspond to one normal lambda term.

The correspondence between sequent calculus derivations and natural deduction derivations is, however, not a one-to-one map. Different cut-free sequent derivations correspond to one normal natural derivation. This causes some syntactic technicalities. The correspondence is best explained by two extensionally equivalent type assignment systems for untyped lambda terms, one corresponding to natural deduction (λN) and the other to sequent calculus (λL). These two systems constitute different grammars for generating the same type assignment relation (for untyped) lambda terms.

The main technical tool is the type assignment system λL corresponding exactly to sequent calculus (for minimal propositional logic). The type assignment system λL is a subsystem of a system studied in Barbanera et al. [1995]. The terms involved in λL are also in Mints [1996]. The difference between the present approach and the one by Mints is that in that paper derivations in intuitionistic logic are first order citizens, whereas in λL the provable formulas and the lambda terms are.

The system λL has a cut-free fragment λL^{cf} which generates the same type assignment relation as λN restricted to normal terms and cut elimination corresponds exactly to normalization. The mismatch between the logical systems that was observed above, is due to the fact that λN is a syntax directed system, whereas both λL and λL^{cf} are not.

In λN typeable terms are built up as usual (following the grammar of lambda terms). In λL^{cf} only normal terms are typeable. They are built up from variables by transitions like

$$P \mapsto \lambda x.P$$

and

$$P \mapsto P[x := yQ].$$

This is an ambiguous way of building terms, in the sense that one term can be built up in several ways. For example, one can assign to the term $\lambda x.yz$ the type $C \to B$ (in the context $z : A$, $y : A \to B$) via two different cut-free derivations.

These, correspond, respectively, to the following two formations of terms

$$\begin{array}{ccccc} u & \mapsto & yz & \mapsto & \lambda x.yz, \\ u & \mapsto & \lambda x.u & \mapsto & \lambda x.yz. \end{array}$$

*Department of Computer Science, Catholic University, Toernooiveld 1, 6525 ED Nijmegen, The Netherlands; e-mail: henk@cs.kun.nl.

†Faculty of Engineering, University of Novi Sad, Trg Dositeja Obradovića 6, 21000 Novi Sad, Yugoslavia; e-mail: gsilvia@uns.ns.ac.yu.

Therefore there are more sequent calculus derivations giving rise to the same lambda term. This is the cause of the mismatch between sequent calculus and natural deduction as described in Zucker [1974], Pottinger [1977] and Mints [1996]. See also Dyckhoff and Pinto [1997], Schwichtenberg [1997] and Troelstra [1998].

In Herbelin [1995] the mismatch between sequent calculus derivations and lambda terms is repaired by translating these into terms with explicit substitution:

$$\lambda x.(u < u := yz >),$$
$$(\lambda x.) < u := yz > .$$

In our paper lambda terms are considered as first class citizens also for sequent calculus. This gives an insight into the mentioned mismatch by understanding it as an intensional aspect how the sequent calculus generates these terms.

It is interesting to note, how in the full system λL the rule (cut) generates terms not in β-normal form. The extra transition now is

$$P \mapsto P[x := F].$$

This will introduce a redex, if x occurs actively (in a context xQ) and F is an abstraction ($F \equiv \lambda x.R$), the other applications of the rule (cut) being superfluous. Also, it can be observed that if one wants to manipulate derivations in order to obtain a cut-free proof, then the term involved gets reduced. By the strong normalization theorem for $\lambda N(= \lambda \rightarrow)$ it follows that eventually a cut-free proof will be reached. The cut elimination theorem becomes a simple consequence of the fact that typed lambda terms posses a normal form.

We have not studied in detail whether cut elimination can be done along the lines of this paper for the full system of intuitionistic predicate logic, but there seems to be no problem. More interesting is the question, whether there are similar results for classical and linear logic.

References

[1] Barbanera, F., M. Dezani-Ciancaglini and U. de'Liguoro [1995]
Intersection and union types: syntax and semantics, *Information and Computation* **119**, pp. 202–230.

[2] Barendregt, H.P. [1992]
Lambda calculi with types, *in*: S. Abramsky, D.M. Gabbai and T.S.E. Maibaum (eds.), *Handbook of Logic in Computer Science*, Vol. 2, Oxford University Press, Oxford, pp. 117–309.

[3] Dyckhoff, R. and L. Pinto [1997]
Permutability of proofs in intuitionistic sequent calculi, *Theoretical Computer Science*. To appear.

[4] Herbelin, H. [1995]
A lambda calculus structure isomorphic to Gentzen-style sequent calculus structure, *Computer Science Logic (CSL'94)*, Lecture Notes in Computer Science **933**, Springer Verlag, Berlin, pp. 61–75.

[5] Mints, G. [1996]
Normal forms for sequent derivations, *in*: P. Odifreddi (eds.), *Kreiseliana. About and Around Georg Kreisel*, A.K. Peters, Wellesley, Massachusetts, pp.469–492.

[6] Pottinger, G. [1977]
Normalization as a homomorphic image of cut-elimination, *Annals of Mathematical Logic* **12**, pp. 323–357.

[7] Prawitz, D. [1965]
Natural deduction. A proof-theoretical study, Almqvist and Wiksell, Stockholm.

[8] Schwichtenberg, H. [1997]
Termination of permutative conversion inintuitionistic Gentzen calculi, *Theoretical Computer Science*. To appear.

[9] Troelstra, A.S. [1998]
Marginalia on sequent calculi, *Studia Logica*. To appear.

[10] Zucker, J. [1974]
Cut-elimination and normalization, *Annals of Mathematical Logic* **7**, pp 1–112.