A \(\lambda\)-calculus Structure Isomorphic to Gentzen-style Sequent Calculus Structure

Hugo Herbelin *

LITP, University Paris 7, 2 place Jussieu, 78252 Paris Cedex 05, France
INRIA-Rocquencourt, B.P. 105, 78153 Le Chesnay Cedex, France
Hugo.Herbelin@inria.fr

Abstract. We consider a \(\lambda\)-calculus for which applicative terms have no longer the form \(\ldots((u \ u_1) \ u_2)\ldots \ u_n)\) but the form \(u \ [u_1;\ldots; u_n]\), for which \([u_1;\ldots; u_n]\) is a list of terms. While the structure of the usual \(\lambda\)-calculus is isomorphic to the structure of natural deduction, this new structure is isomorphic to the structure of Gentzen-style sequent calculus.

To express the basis of the isomorphism, we consider intuitionistic logic with the implication as sole connective. However we do not consider Gentzen's calculus LJ, but a calculus LJT which leads to restrict the notion of cut-free proofs in LJ. We need also to explicitly consider, in a simply typed version of this \(\lambda\)-calculus, a substitution operator and a list concatenation operator. By this way, each elementary step of cut-elimination exactly matches with a \(\beta\)-reduction, a substitution propagation step or a concatenation computation step.

Though it is possible to extend the isomorphism to classical logic and to other connectives, we do not treat of it in this paper.

1 Introduction

By the Curry-Howard isomorphism between natural deduction and simply-typed \(\lambda\)-calculus, and using Prawitz's standard translation [11] of cut-free LJ into natural deduction, we get an assignment of LJ proofs by \(\lambda\)-terms.

Zucker [14] and Pottinger [10] have studied the relations between normalisation in natural deduction and cut-elimination in LJ. They were considering normalisation without paying special attention to the computational cost of the substitution of a proof in place of an hypothesis. But in sequent calculus, among the different uses of the cut rule, there is one which stands for an explicit operator of substitution and among the elementary rules for cut-elimination, there are rules to compute the propagation of substitution. Therefore, Zucker and Pottinger were led to consider proofs up to the equivalence generated by these substitution propagation computation rules.

Here, we consider a \(\lambda\)-calculus with an explicit operator of substitution and with appropriated substitution propagation rules. This allows to have a more

* This research was partly supported by ESPRIT Basic Research Action "Types for Proofs and Programs" and by Programme de Recherche Coordonnées "Mécanisation du raisonnement".
precise correspondence with the elementary rules for cut-elimination. However, there are two problems. The first one is that several cut-free proofs of LJ are associated to the same normal simply-typed $\lambda$-terms. An answer to this problem is to rather consider a restriction of LJ, called LJT, having the same structure and same strength as LJ but for which there is a one-to-one correspondence with normal simply-typed terms. The second problem is that Gentzen-style sequent calculus and $\lambda$-calculus (or natural deduction) have not the same structure. Consequently, the reduction rules in one and the other calculi do not match. An answer to this second problem is to consider an alternative syntax for $\lambda$-calculus of which, this time, the simply-typed fragment is isomorphic to LJT.

Note that a radically different approach of the computational content of Gentzen's sequent calculus appears in Breazu Tanen et al [1], Gallier [4] and Wadler [13]. Each of them interprets the left introduction rules of sequent calculus as pattern construction rules.

2 A Motivated Approach to LJT and $\overline{\lambda}$-calculus

2.1 The Sequent Calculus LJ

We consider a version of LJ with the implication as sole connective. The formulas are defined by the grammar

$$A ::= X \mid A \to A$$

where $X$ ranges over $\forall_F$, an infinite set of which the elements are called propositional variable names. In the sequel, we reserve the letters $A, B, C, \ldots$ to denote formulas.

Sequents of LJ have the form $\Gamma \vdash A$. To avoid the need of a structural rule we define $\Gamma$ as a set. To avoid confusion between multiple occurrences of the same formula, this set is a set of named formulas. We assume the existence of an infinite set of which the elements are called names. Then, a named formula is just the pair of a formula and a name. Usually, we do not mention the names of formulas (anyway, no ambiguity occurs in the sequents we consider here).

Under the condition that $A$, with its name, does not belong to $\Gamma$, the notation $\Gamma, A$ stands for the set-theoretic union of $\Gamma$ and $\{A\}$.

To avoid the need of a weakening rule, we admit irrelevant formulas in axioms. The rules of LJ are:

$$\begin{align*}
\Gamma, A & \vdash A & \text{Ax} \\
\Gamma, A & \vdash C & \Gamma, A, A \vdash C & \text{Cont} \\
\Gamma, A & \vdash A & \Gamma, B \vdash C & \Gamma, A \vdash B & \text{I}_L \\
\Gamma, A \to B & \vdash C & \Gamma & \vdash A \to B & \Gamma & \vdash B & \text{I}_R \\
& \Gamma & \vdash A & \Gamma, A & \vdash B & \Gamma & \vdash B & \text{Cut}
\end{align*}$$