Expanding Extensional Polymorphism*

Roberto Di Cosmo [1] *Adolfo Piperno* [2]

[1] DMI-LIENS - Ecole Normale Supérieure
45, Rue d'Ulm - 75230 Paris (France)
dicosmo@dmi.ens.fr

[2] Dipartimento di Scienze dell'Informazione - Università di Roma "La Sapienza"
Via Salaria 113 - 00198 Roma (Italy)
piperno@dsi.uniroma1.it

Abstract. We prove the confluence and strong normalization properties for second order lambda calculus equipped with an expansive version of η-reduction. Our proof technique, based on a simple abstract lemma and a labelled λ-calculus, can also be successfully used to simplify the proofs of confluence and normalization for first order calculi, and can be applied to various extensions of the calculus presented here.

1 Introduction

The typed lambda calculus provides a convenient framework for studying functional programming and offers a natural formalism to deal with proofs in intuitionistic logic. It comes traditionally equipped with the β equality $(\lambda x.M)N = M[N/x]$ as fundamental computational mechanism, and with the η *(extensional)* equality $\lambda x.Mx = M$ as a tool for reasoning about programs. This basic calculus can then be extended by adding further types, like products, unit and second order types, each coming with its own computational mechanism and/or its extensional equalities.

To reason about programs and the proofs that they represent, one has to be able to orient each equality into a rewriting rule, and to prove that the resulting rewriting system is indeed confluent and strongly normalizing: these properties guarantee that to each program (or proof) P we can associate an equivalent canonical representative which is unique and can be found in finite time by applying the reduction rules to P in whatever order we choose. The β equality, for example, is always turned into the reduction rule $(\lambda x.M)N \longrightarrow M[N/x]$.

Traditionally, the extensional equalities are turned into *contraction* reduction rules, the most known example being the η rule $\lambda x.Mx \longrightarrow M$, but this approach raises a number of difficult problems when trying to add other rules to the system. For example the extensional first order lambda calculus associated to Cartesian Closed Categories, where one needs a special *unit* type T with an axiom $M{:}T = *{:}T$ (see [CDC91] and especially [DCK94b] for a longer discussion and references) is no longer confluent. Another example is the extensional first order lambda calculus enriched with a confluent algebraic rewriting system, where confluence is also broken [DCK94a].

This inconvenient can be fortunately overcome, as proposed in several recent works[Aka93, Dou93, DCK94b, Cub92, JG92], by turning the extensional equalities into *expansion* rules: η becomes then

$$M{:}A \to B \longrightarrow \lambda x.Mx.$$

These expansions are suitably restricted to ensure termination [3], and several *first*

* This work has been partially supported by grants from HCM "Typed Lambda Calculus" and CNR-CNRS projects

[3] We refer the interested reader to[DCK93, DCK94b] for a more detailed discussion of these restrictions.

order systems incorporating both the expansive η rule and an expansive version of the Surjective Pairing extensional rule for products can be proven confluent and strongly normalizing. In[DCK94b] Delia Kesner and the first author even proved that a system with expansions for Surjective Pairing is confluent in the presence of a fixpoint combinator, while it is known that confluence does not hold with the contractive version of Surjective Pairing[Nes89].

These recent works raise a natural question: is it possible to carry on this approach to extensional equalities via expansion rules to the *second order* typed lambda calculus? The answer is not obvious: for an expansion rule to be applicable on a given subterm, we need to look at the type of that subterm, and when we add second order quantification a *subterm* can *change* its type during evaluation. As we will see, this fact rules out a whole class of modular proof techniques that would easily establish the result, and makes the study of expansion rules more problematic.

In this paper we focus on the *second order* typed lambda calculus and extensionality axioms for the arrow type: this system corresponds to the Intuitionistic Positive Calculus with implication, and quantification over propositions.

For this calculus we provide a reduction system based on expansion rules that is confluent and strongly normalizing, by means of an interpretation into a normalizing fragment of the untyped lambda calculus.

This result gives a natural justification of the notion of η-long normal forms used in higher order unification and resolution: they can be now defined simply as the normal forms w.r.t. our extensional rewriting system.

1.1 Survey

The restrictions imposed on the expansion rules in order to insure termination make several usual properties of the λ-calculus fail, most notably η-postponement, that would allow a very simple proof of normalization for the calculus[4], but several proof techniques have been developed over the past years to show that the expansionary interpretation of the extensional equalities yields a confluent and normalizing system in the first order case. One idea is to try to separate the expansion rules from the rest of the reduction, and then try to show some kind of modularity of the reduction systems. One traditional technique for confluence that comes to mind is the well known

Lemma 1.1 (Hindley-Rosen ([Bar84], §3)) *If R and S are confluent, and commute with each other, then $R \cup S$ is confluent.*

Unfortunately, this technique does not work in the presence of restricted expansion rules, because β can destroy expansion redexes, but in[Aka93] Akama gives a modular proof using the following property, requiring some additional conditions on R and S:

Lemma 1.2 *Let S and R be confluent and strongly normalizing reductions, s.t.*

$$\forall M, N \quad (M \xrightarrow{S} N) \quad \textit{implies} \quad (M^R \xrightarrow[+]{S} N^R),$$

where M^R and N^R are the R-normal forms of M and N, respectively; then $S \cup R$ is also confluent and strongly normalizing.

In[Aka93] R is taken to be the expansionary system alone and S is the usual non extensional reduction relation.

In[DCK94b], confluence and strong normalization of the full expansionary system is reduced to that of the traditional one without expansions using the following:

[4] For a very broad presentation of the properties that fail in presence of restricted expansions, see[DCK94b].