

# Expanding Extensional Polymorphism<sup>\*</sup>

Roberto Di Cosmo<sup>1</sup>

Adolfo Piperno<sup>2</sup>

<sup>1</sup> DMI-LIENS - Ecole Normale Supérieure  
45, Rue d'Ulm - 75230 Paris (France)  
roberto@dicosmo.org

<sup>2</sup> 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  $\eta$ -reduction. Our proof technique, based on a simple abstract lemma and a labelled  $\lambda$ -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  $\beta$  equality  $(\lambda x.M)N = M[N/x]$  as fundamental computational mechanism, and with the  $\eta$  (*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  $\beta$  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  $\eta$  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:  $\eta$  becomes then

$$M: A \rightarrow B \longrightarrow \lambda x.Mx.$$

These expansions are suitably restricted to ensure termination<sup>3</sup>, and several *first*

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

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

*order* systems incorporating both the expansive  $\eta$  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  $\eta$ -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  $\lambda$ -calculus fail, most notably  $\eta$ -postponement, that would allow a very simple proof of normalization for the calculus<sup>4</sup>, 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  $\beta$  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.*

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

*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:

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

**Proposition 1.3** Let  $\mathcal{R}_1$  and  $\mathcal{R}_2$  be two reduction systems and  $\mathcal{T}$  a translation from  $\mathcal{R}_1$ -terms to  $\mathcal{R}_2$ -terms.

- (i) If for every reduction  $M_1 \xrightarrow{\mathcal{R}_1} M_2$  there is a non empty sequence  $P_1 \xrightarrow[\oplus]{\mathcal{R}_2} P_2$  such that  $\mathcal{T}(M_i) = P_i$ , for  $i = 1, 2$  (simulation property), then the strong normalization of  $\mathcal{R}_2$  implies that of  $\mathcal{R}_1$ .
- (ii) If in addition the translation is the identity on  $\mathcal{R}_1$  normal forms, and these normal forms are included in the  $\mathcal{R}_2$  normal forms, then the confluence of  $\mathcal{R}_2$  also implies the confluence of the full system.

The translation used in[DCK94b] inserts in all positions where an expansion could take place a special term  $\Delta_A$  (called an *expansor*) depending on the type  $A$  of the expansion redex, and then all that one is left to prove is the simulation property.

A different non-modular approach is taken in [JG92] and [Dou93], where the proofs of strong normalization are based on an extension of the traditional techniques of reducibility and allow to handle also the peculiarity of the expansion rules. But that is not all, since one is left to prove weak confluence separately, which is not an easy task in the presence of expansion rules (see[DCK94b] for details).

Finally, an even different technique is used in [Cub92], where confluence is shown by a careful study of the residuals in the reduction.

As suggested in the introduction, in the presence of second order quantification, the type of a subterm can evolve during evaluation, and this fact allows us to build very simple examples suggesting that the modular approaches[Aka93, DCK94b] cannot be extended to the second order case.

### Expansions and polymorphism are not modular

The following simple example shows that we cannot use the modular techniques developed up to now to separate the complexities introduced by expansion rules and polymorphic typing by singling out the expansions in a separate rewriting system.

**Example 1.4** Let  $M = (\lambda\sigma.\lambda x: (\forall\mu.\mu \rightarrow A).(x[\sigma \rightarrow \sigma])(\lambda y: \sigma.y))[A \rightarrow B]$ . Then, the term  $M$  is a normal form w.r.t expansion rules, but its immediate  $\beta^2$  reduct is not:

$$M' = \lambda x: (\forall\mu.\mu \rightarrow A).(x[(A \rightarrow B) \rightarrow (A \rightarrow B)])(\lambda y: A \rightarrow B.y)$$

In fact,  $M'$  reduces to the term

$$M'' = \lambda x: (\forall\mu.\mu \rightarrow A).(x[(A \rightarrow B) \rightarrow (A \rightarrow B)])(\lambda y: A \rightarrow B.\lambda z: A.yz)$$

Now, there is no way to reduce  $M$  to  $M''$  without expansions, so the hypothesis of lemma 1.2 are not satisfied.

This very same example can be used to show how the use of expansor terms is neither viable<sup>5</sup>.

Notice that Akama's lemma fails also if we put  $\beta^2$  together with  $\eta$  in the reduction relation  $R$ , because  $\beta$  does not preserve  $\beta^2$  normal forms.

As for the reducibility technique, it can be adapted as in[JG92] for the first order calculi with expansion rules, but there is a fundamental difference between the proof

<sup>5</sup> For readers acquainted with the techniques used in[DCK94b], it is easy to see that the term

$$M = (\lambda\sigma.\lambda x: (\forall\mu.\mu \rightarrow A).(x[\sigma \rightarrow \sigma])(\lambda y: \sigma.y))[A \rightarrow B]$$

is the translation of itself, as there is no expansion redex, but its  $\beta^2$  reduct

$$\lambda x: (\forall\mu.\mu \rightarrow A).(x[(A \rightarrow B) \rightarrow (A \rightarrow B)])(\lambda y: A \rightarrow B.y)$$

gets translated to  $\lambda x: (\forall\mu.\mu \rightarrow A).(x[(A \rightarrow B) \rightarrow (A \rightarrow B)])(\lambda y: A \rightarrow B.\Delta_{A \rightarrow B}y)$  and there is no way for  $M$  to reduce to it, so the hypothesis of proposition 1.3 are not satisfied.

for the simply typed and the proof for polymorphic lambda calculus: one does not work with just one reducibility candidate, but with all reducibility candidates at once. This requires to deal with many subtle points in the second order case that do not appear at all for the simply typed calculi: for example, in the second order setting one has to show that the set of all strongly normalizing terms is indeed a reducibility candidate: this is straightforward using Girard's original definition of reducibility, but the modifications of Girard's **(CR3)** property imposed by the extensional rules make even this task extremely difficult. Up to now no such proof is known.

## 1.2 Our approach

Since the previous modular techniques are not viable as used traditionally, and the reducibility properties modified as in [JG92] do not extend nicely to second order, we had to look for something new, and since all the problems come up immediately as soon as we add  $\beta^2$  to the first order expansionary system, we focused on a simple system with  $\beta$ , expansive  $\eta$  and  $\beta^2$  first (weak confluence for this calculus is quite straightforward). Here, we first observed that:

- an infinite reduction path in the typed calculus implies the existence of an infinite reduction path containing infinite  $\beta$  steps in the untyped calculus, each untyped term being the erasure of the corresponding typed one (this is the case because  $\beta^2$  alone, that leaves the erasures unchanged, always terminates, and because  $\beta^2$  and  $\eta$  together are easily seen to be strongly normalizing);
- an untyped term that is typable in the second order lambda calculus cannot have an infinite  $\beta$  reduction sequence;
- $\eta$ -postponement holds if we lift the restrictions on expansions, and we can use it on the erasure of the reduction, obtaining a reduction sequence that contains all the  $\beta$  steps, consecutive and at the beginning.

Now, this would clearly suffice to prove strong normalization, if  $\eta$ -postponement could be done without deleting some  $\beta$  or  $\eta$  steps from the original reduction, as then from an infinite typed reduction we could get an infinite  $\beta$  reduction starting from a typable term, that is impossible.

Unfortunately,  $\eta$ -postponement with unrestricted expansions *does* delete some  $\eta$  and  $\beta$  steps, as in the following loops that are the very motivation for imposing restrictions on the  $\eta$  expansions:

$$\begin{array}{lcl} MN \xrightarrow{\eta} (\lambda x.Mx)N \xrightarrow{\beta} MN & \text{becomes} & MN = MN \\ \lambda x.M \xrightarrow{\eta} \lambda z.(\lambda x.M)z \xrightarrow{\beta} \lambda x.M & \text{becomes} & \lambda x.M = \lambda x.M \end{array}$$

We carefully analysed these deletions during the  $\eta$  postponement, showing that the only  $\beta$  steps that get erased are the ones that are created by expansions which violate the restrictions. To study such reductions, we work in a labelled calculus where abstractions introduced by  $\eta$  expansions are marked, with unrestricted  $\eta$  turned into  $M: A \rightarrow B \xrightarrow{\eta} \lambda^*x.Mx$ .

In this labelled calculus, it is easy to identify the  $\beta$  steps that get erased during postponement, and we singled them out as the following  $\beta_*$  rule:

$$\begin{array}{l} (\beta_*) \quad \left\{ \begin{array}{l} (\lambda^*x.M)N \xrightarrow{\beta_*} M[N/x] \\ \lambda^*y.D[(\lambda x.M)Y] \xrightarrow{\beta_*} \lambda y.D[M[Y/x]], \text{ if } y \xrightarrow{\eta} Y \wedge [] \xrightarrow{\eta} D[] \end{array} \right. \\ (\beta) \quad (\lambda x.M)N \xrightarrow{\beta} M[N/x], \text{ if } (\lambda x.M)N \text{ is not in a context} \\ \text{where } (\beta_*) \text{ applies.} \end{array}$$

Thus, when a  $\beta_*$ -redex is also a  $\beta$ -redex, we assume that its contraction constitutes a  $\beta_*$ -step.

Now, this is the key step:  $\beta_*$ , unlike the full  $\beta$ , is well behaved wrt  $\beta^2$ , as it preserves  $\beta^2$  normal forms, and one can apply Akama's lemma to show that  $\beta^2$ ,  $\eta$  and  $\beta_*$  together are confluent and normalizing.

This means that an infinite typed reduction must contain infinite  $\beta$  steps that are not  $\beta_*$  steps, and we are done because these  $\beta$  steps are not deleted by the  $\eta$  postponement that we perform on the erasure of the typed reduction: we can finally build an infinite  $\beta$  reduction leaving from a typable term, a contradiction, as we wanted.

What is particularly satisfactory in this proof technique is the fact that we really show that the only source of danger are the real  $\beta$  reductions, and not the  $\beta_*$  redexes produced by  $\eta$  expansion, which are really harmless. This proof technique can be applied successfully also to various extensions of the simple calculus presented here, as we will detail in the Conclusions.

### 1.3 Structure of the paper

The two main technical points in the paper will then be to present the  $\eta$  postponement in the unrestricted case and to prove the hypothesis of Akama's Lemma for  $\beta_* \cup \eta \cup \beta^2$ , but there is something else.

Indeed, we found that applying Akama's Lemma is hard: one has to show commutation of one reduction relation wrt the reduction to normal form for the other reduction relation, and this is done in Akama's paper in an ad hoc fashion for a specific calculus by a difficult technical analysis of  $\eta$  normal forms. We wanted a more easily applicable technique, that we found, and that we decided, for its generality, to present in a section by its own.

So, we will first expose formally the results on the untyped calculus, in the next section, then we will show how to drastically simplify the proofs involved in applying Akama's Lemma, in the following section, and finally we will apply this simpler technique to the typed reduction  $\beta^2 \cup \eta \cup \beta_*$  in order to obtain the proof of strong normalization for the full reduction. Confluence will follow by Newman's Lemma.

We will then conclude with an overview of the applicability of our technique, and with some ideas for further work.

## 2 The Untyped Case

In this Section we introduce a  $\lambda$ -calculus with markers, which enable us to keep under control variables introduced by applications of the expansive version of the  $\eta$  rule. We characterize a relevant class of terminating reductions in such calculus.

Terms of the *untyped marked  $\lambda$ -calculus* are defined by the following syntax

$$M ::= x \mid MM \mid \lambda x.M \mid \lambda^* x.M, \quad (1)$$

where  $x$  ranges over a denumerable set  $\text{Var}$  of term variables;  $FV(M)$  denotes the set of variables occurring free in  $M$ . We call  $\Lambda^*$  the set of terms resulting from (1), while  $\Lambda$  is the set of unmarked terms.

As usual, terms are considered modulo  $\alpha$ -conversion, i.e. modulo names of bound variables.

One step  $\eta$ -reduction is the least binary relation on  $\Lambda^*$  which passes contexts and satisfies

$$(\eta) \quad M \xrightarrow{\eta} \lambda^* x.Mx, \quad \text{if } x \notin FV(M).$$

$\xrightarrow{\eta}$  denotes the reflexive and transitive closure of the one step  $\eta$ -reduction relation, while  $\xrightarrow[\eta]{n}$  denotes the  $n$  times composition of  $\xrightarrow{\eta}$  with itself.

**Remark 2.1** For simplicity of notation, we will always use  $\eta$  to denote *expansion* rules. In the untyped system, such reduction is applicable in any context; in the typed system, some restrictions will be introduced on the applicability of  $\eta$ -reduction. Nevertheless, the same notation will be adopted for both (either untyped and unrestricted or typed and restricted)  $\eta$ -reductions. Indeed, a special notation will be used in the typed case for unrestricted  $\eta$ -reduction.

One step  $\beta_*$  and  $\beta$ -reductions (notation:  $\xrightarrow{\beta_*}$  and  $\xrightarrow{\beta}$ , respectively) are defined as the least binary relations on  $\Lambda^*$  which pass contexts and respectively satisfy:

$$\begin{aligned} (\beta_*) \quad & \left\{ \begin{array}{l} (\lambda^*x.M)N \xrightarrow{\beta_*} M[N/x] \\ \lambda^*y.D[(\lambda x.M)Y] \xrightarrow{\beta_*} \lambda y.D[M[Y/x]], \text{ if } y \xrightarrow{\eta} Y \wedge [] \xrightarrow{\eta} D[] \end{array} \right. \\ (\beta) \quad & (\lambda x.M)N \xrightarrow{\beta} M[N/x], \text{ if } (\lambda x.M)N \text{ is not in a context} \\ & \text{where } (\beta_*) \text{ applies} \end{aligned}$$

Let  $\rho = \beta \cup \beta_* \cup \eta$ . Finally, let  $\xrightarrow{\beta}$  ( $\xrightarrow{\beta_*}$ ,  $\xrightarrow{\rho}$ ) denote the reflexive and transitive closure of one step the  $\beta$ - ( $\beta_*$ ,  $\rho$ ) reduction relation.

**Definition 2.2** We define  $\Lambda_\eta^*$  to be the subset of  $\Lambda^*$  whose elements are obtained from unmarked terms via  $\eta$ -reduction.

$$\Lambda_\eta^* = \{M \in \Lambda^* \mid \exists M' \in \Lambda. M' \xrightarrow{\eta} M\}.$$

Recall that a context  $C[] (\in \Lambda[], \Lambda^*[], \Lambda_\eta^*[])$  is a term (belonging to  $\Lambda, \Lambda^*, \Lambda_\eta^*$ ) with one hole in it.

**Property 2.3** (i)  $M \in \Lambda_\eta^* \Leftrightarrow (\forall C[] \in \Lambda_\eta^*[]. M \equiv C[\lambda^*x.N] \Rightarrow N \equiv D[N'X]),$   
where  $N' \in \Lambda_\eta^* \wedge FV(N') \not\ni x \xrightarrow{\eta} X \wedge [] \xrightarrow{\eta} D[]$ .

(ii)  $(M \in \Lambda_\eta^* \wedge M \xrightarrow{\rho} N) \Rightarrow N \in \Lambda_\eta^*;$

*Proof.* (i)  $(\Leftarrow)$  is trivial. To prove  $(\Rightarrow)$ , observe that if  $M \in \Lambda_\eta^*$ , then  $\exists M' \in \Lambda$  and  $n \in \mathbb{N}$ , s.t.  $M' \xrightarrow[\eta]{n} M$ . We will reason by induction on  $n$ .

$n = 0$ . Vacuously true.

$n = m + 1$ . Then there is an  $M' \in \Lambda$  s.t.  $M' \xrightarrow[\eta]{m} M'' \xrightarrow{\eta} M$ . We know by induction that the property holds for  $M''$ . If  $M \equiv C[\lambda^*x.N]$ , then, two cases:

(a)  $\lambda^*x.N$  was already in  $M''$ , that is, for some  $C'[] \in \Lambda_\eta^*[], M'' \equiv C'[\lambda^*x.N]$ , and we are done by induction hypothesis.

(b)  $\lambda^*x.N$  is the result of the last step  $M'' \xrightarrow{\eta} M$ , and we have two cases:

(b.1)  $M'' \equiv C[N']$  and  $M \equiv C[\lambda^*x.N'x]$ , so  $N = N'x$  with  $x \notin FV(N')$ , and we are done.

(b.2)  $M'' \equiv C[\lambda^*x.N']$  and  $M \equiv C[\lambda^*x.N]$ , with  $N' \xrightarrow{\eta} N$ . By induction hyp. we know that  $N' \equiv D[N''X]$ , where  $FV(N'') \not\ni x \xrightarrow{\eta} X \wedge [] \xrightarrow{\eta} D[]$ . Now, the last expansion can be either  $N'' \xrightarrow{\eta} N'''$ , and then  $M \equiv C[\lambda^*x.D[N'''X]]$ , or  $X \xrightarrow{\eta} X'$ , and then  $M \equiv C[\lambda^*x.D[N''X']]$ , or  $D[N''X] \xrightarrow{\eta} D'[N''X]$ , and then  $M \equiv C[\lambda^*x.D'[N''X]]$ , with  $D[] \xrightarrow{\eta} D'[]$ . In all cases, the conditions are satisfied and we are done.

(ii) If the  $\rho$  reduction is an  $\eta$ , then the property holds by the very definition of  $\Lambda_\eta^*$ . In the other cases ( $\beta$  and  $\beta_*$ ), the proof is deferred after Lemma 2.7 from which it follows immediately.  $\square$

The following facts can be shown by simple calculations

**Fact 2.4** If  $H \in \Lambda_\eta^*$  and  $x \in \text{Var}$  then  $x \xrightarrow{\eta} X \Rightarrow H \xrightarrow{\eta} X[H/x]$ .

**Fact 2.5** If  $H, J \in \Lambda_\eta^*$  and  $x \in \text{Var}$  then

$$(H \xrightarrow{\eta} H', J \xrightarrow{\eta} J') \Rightarrow H[J/x] \xrightarrow{\eta} H'[J'/x].$$

**Fact 2.6** If  $x \in \text{Var}$  and  $x \xrightarrow{\eta} X \xrightarrow{\eta} X'$  then  $X'[X/x] \xrightarrow{\beta_*} X'$ .

**Lemma 2.7** Let  $P \in \Lambda$  and  $M, N \in \Lambda_\eta^*$ .

- (i) If  $P \xrightarrow{\eta} M \xrightarrow{\beta_*} N$ , then  $P \xrightarrow{\eta} N$ .
- (ii) If  $P \xrightarrow{\eta} M \xrightarrow{\beta} N$ , then there exists  $Q \in \Lambda$  such that  $P \xrightarrow{\beta} Q \xrightarrow{\eta} N$ .
- (iii) Let  $M, N \in \Lambda_\eta^*$  and  $\tau = \beta_* \cup \eta$ . If  $M \xrightarrow{\tau} N$ , then there exists  $Q \in \Lambda_\eta^*$  such that  $M \xrightarrow{\eta} Q \xrightarrow{\beta_*} N$ .

*Proof.* (i) We distinguish the two cases for  $\beta_*$ :

- (a) If  $M \equiv C[(\lambda^*x.S)R]$ , then by Property 2.3 we have  $S \equiv D[QX]$  where  $Q \in \Lambda_\eta^*$  and  $x \xrightarrow{\eta} X$ , and then

$$M \equiv C[(\lambda^*x.D[QX])R] \xrightarrow{\beta_*} C[D[QX[R/x]]] \equiv N.$$

Hence  $P \equiv C'[Q'R']$ , where  $Q' \xrightarrow{\eta} Q$ ,  $R' \xrightarrow{\eta} R$  and  $C'[\ ] \xrightarrow{\eta} C[\ ]$ .

This case is settled using Fact 2.4.

- (b)  $M \equiv C[\lambda^*y.D[(\lambda x.Q)Y]] \xrightarrow{\beta_*} C[\lambda y.D[Q[Y/x]]] \equiv N$ , with  $y \xrightarrow{\eta} Y$ .

Hence  $P \equiv C'[\lambda x.Q']$ , where  $Q' \xrightarrow{\eta} Q$  and  $C'[\ ] \xrightarrow{\eta} C[\ ]$ .

This case is settled using Fact 2.4.

- (ii)  $M \equiv C[(\lambda x.Q)R] \xrightarrow{\beta} C[Q[R/x]] \equiv N$ . Hence

$$P \equiv C'[(\lambda x.Q')R'],$$

where  $Q' \xrightarrow{\eta} Q$ ,  $R' \xrightarrow{\eta} R$  and  $C'[\ ] \xrightarrow{\eta} C[\ ]$ .

This case is settled using Fact 2.5.

- (iii) We distinguish the two cases for  $\beta_*$  and we observe that:

$$\begin{array}{ccc} C[(\lambda^*x.D[QX])R] & \xrightarrow{\beta_*} & C[D[QX[R/x]]] \\ \downarrow \eta & & \downarrow \eta \\ C'[(\lambda^*x.D'[Q'X'])R'] & \xrightarrow{\beta_*} & C'[D'[Q'X'[R'/x]] \\ \text{where } C'[(D[Q], X, R, \text{ resp.})] & \xrightarrow{\eta} & C'[(D'[Q'], X', R', \text{ resp.})]; \end{array}$$

$$\begin{array}{ccc} C[\lambda^*y.D[(\lambda x.Q)Y]] & \xrightarrow{\beta_*} & C[\lambda y.D[Q[Y/x]]] \\ \downarrow \eta & & \downarrow \eta \\ C'[\lambda^*y.D'[(\lambda x.Q'[Y'_i/x^{(i)}]_{i=1\dots n})Y]] & \xrightarrow[\text{(2.6)}]{\beta_*} & C'[\lambda y.D'[Q'[Y'_i/x^{(i)}]_{i=1,\dots,n}]] \end{array}$$

where  $x^{(1)}, \dots, x^{(n)}$  denote the occurrences of the free variable  $x$  in  $Q$  and

$$C'[(D[Q], Q, \text{ resp.})] \xrightarrow{\eta} C'[(D'[Q'], Q, \text{ resp.})], y \xrightarrow{\eta} Y \xrightarrow{\eta} Y'_i,$$

for  $i = 1, \dots, n$ .

Let now  $M \xrightarrow{\tau} N$ . The lemma follows by an easy induction on the number of  $\beta_*$  steps which are followed by an  $\eta$  step in the reduction from  $M$  to  $N$ .  $\square$

**Definition 2.8** Let  $M_0 \in \Lambda_\eta^*$ . A  $\rho$ -reduction path

$$\Pi: M_0 \xrightarrow{\rho} M_1 \xrightarrow{\rho} M_2 \xrightarrow{\rho} \dots$$

starting from  $M_0$  is called *fair* iff either it is finite or, for any  $i \in \mathbb{N}$ , it satisfies the following conditions

- (i)  $M_i \xrightarrow{\beta} M_{i+1} \Rightarrow \exists k > 0. \neg(M_{i+k} \xrightarrow{\beta} M_{i+k+1})$ ;
- (ii)  $M_i \xrightarrow{\beta_*} M_{i+1} \Rightarrow [\exists k > 0. \neg(M_{i+k} \xrightarrow{\beta_*} M_{i+k+1})] \wedge \neg(M_{i+1} \xrightarrow{\eta} M_{i+2})$ ;
- (iii)  $M_i \xrightarrow{\eta} M_{i+1} \Rightarrow \exists k > 0. \neg(M_{i+k} \xrightarrow{\eta} M_{i+k+1})$ .

**Lemma 2.9 (Main Lemma)** *Let  $M \in \Lambda$  be a  $\beta$ -strongly normalizing term and let  $\Pi$  be a  $\rho$ -reduction path starting from  $M$ .  $\Pi$  is finite iff it is fair.*

*Proof.* Assume the existence of an infinite fair  $\rho$ -reduction path starting from  $M$ . By definition, an infinite fair  $\rho$ -reduction path contains an infinite amount of  $\beta$  steps. Indeed, it does not contain infinite subpaths constituted by all  $\beta$  ( $\beta_*$ ,  $\eta$ , respectively) steps, and also it does not contain any infinite subpath in which  $\beta$  steps do not appear, since by Definition 2.8.(ii) a  $\beta_*$  step is never followed by an  $\eta$  step.

Using Lemma 2.7, we can build an infinite  $\beta$ -reduction starting from  $M$ , which is absurd. Indeed, take a fair reduction sequence starting from a term  $M \in \Lambda$  and containing an infinite number of  $\beta$  steps. Consider now the first  $\beta$  step in the sequence.

By Lemma 2.7, we can assume that all reduction steps from  $M$  to this first  $\beta$  are  $\eta$  steps: if not, these steps must be a sequence of  $\eta$  followed by a sequence of  $\beta_*$ , by definition of fair reduction sequence, and then we can apply Lemma 2.7.(i) and get rid of the  $\beta_*$  sequence, obtaining a reduction sequence that is still fair. Then, from  $\Lambda \ni M \xrightarrow{\eta} M' \xrightarrow{\beta} M'' \xrightarrow{\eta} \dots$  we get, using Lemma 2.7.(iii) a new fair sequence  $\Lambda \ni M \xrightarrow{\beta} M''' \xrightarrow{\eta} M'' \xrightarrow{\eta} \dots$ . Now, it suffices to notice that  $M'''$  is still in  $\Lambda$ , and that the sequence starting from  $M'''$  is again fair and contains an infinite number of  $\beta$  steps, so we can iterate this pumping process and build an infinite  $\beta$  reduction sequence starting from  $M$ .  $\square$

### 3 Simplifying Akama's Lemma

It is now time to turn to Akama's Lemma: applying it directly is hard just like the usual Hindley-Rosen's Lemma 1.1, as one has to handle a multi-step reduction.

But for the Hindley-Rosen's Lemma to be applicable, there is a well known sufficient condition; this just asks us to verify that any divergent diagram  $M' \xleftarrow{S} M \xrightarrow{R} M''$  can be closed using as many  $R$  steps as we want, but no more than one  $S$  step. This sufficient condition is what is always used, for its simplicity (see for example [Bar84]).

Along our investigation, we had to devise a similar sufficient condition for Akama's Lemma, to simplify the otherwise extremely difficult proof of the Lemma's hypothesis. This sufficient condition is so general and nice to prove, that even the results in Akama's original paper can be obtained in a few lines, without the complex syntactic analysis used there.

**Notation 3.1** Let  $\langle \mathcal{A}, \longrightarrow \rangle$  be an Abstract Reduction System. We denote by

- $\equiv \longrightarrow$  the reflexive closure of  $\longrightarrow$  ;
- $\xrightarrow{+}$  the transitive closure of  $\longrightarrow$  ;
- $\xrightarrow{+}$  the reflexive and transitive closure of  $\longrightarrow$  .

**Lemma 3.2** *Let  $\langle \mathcal{A}, \xrightarrow{R}, \xrightarrow{S} \rangle$  be an Abstract Reduction System, where  $R$ -reduction is strongly normalizing. Let the following commutation hold*

$$\forall a, b, c, d \in \mathcal{A} \quad \begin{array}{ccc} a & \xrightarrow{R} & c \\ \downarrow S & & \downarrow S \\ b & \xrightarrow{R} & d \end{array}$$

Then we have



- (i)  $\xrightarrow{R}$  and  $\xrightarrow{S}$  commute.  
(ii) if  $R$  preserves  $S$  normal forms (let  $S\downarrow$  denote reduction to  $S$  normal form), then

$$\forall a, b, c, d \in \mathcal{A} \quad \begin{array}{ccc} a & \xrightarrow{R} & c \\ \downarrow S\downarrow & & \downarrow S\downarrow \\ b & \xrightarrow{R} & d \end{array}$$

- (iii) if  $S$  is also confluent and  $R$  preserves  $S$  normal forms, then

$$\forall a, b, c, d \in \mathcal{A} \quad \begin{array}{ccc} a & \xrightarrow{R} & c \\ \downarrow S\downarrow & & \downarrow S\downarrow \\ b & \xrightarrow{R} & d \end{array}$$

*Proof.* We just prove the first result, as the others are very simple consequences of it. Such result has been independently obtained by Alfons Geser in his PhD Thesis [Ges]. If  $a_1, a_2 \in \mathcal{A}$ , then denote  $deg(a_1)$  the length of the longest  $R$ -reduction path out of  $a_1$  and  $dist(a_1, a_2)$  the length of a  $S$ -reduction sequence from  $a_1$  to  $a_2$ . The proof is by induction on pairs  $(deg(b), dist(a, b))$ , ordered lexicographically. Indeed, if  $deg(b) = 0$  or  $dist(a, b) = 0$ , then the lemma trivially holds. Otherwise, by hypothesis, there exist  $a', a'', a'''$  as in the following diagram.

$$\begin{array}{ccccc} a & \xrightarrow{R} & a' & \xrightarrow{R} & c \\ \downarrow S & & \downarrow S & & \downarrow S \\ a'' & \xrightarrow{R} & a''' & & \\ \downarrow S & & \downarrow S & & \downarrow S \\ b & \xrightarrow{R} & b' & \xrightarrow{R} & d \end{array}$$

We can now apply the inductive hypothesis to the diagram  $D_1$ , since

$$(deg(b), dist(a'', b)) <_{lex} (deg(b), dist(a, b)).$$

Finally, we observe that  $b \xrightarrow{R} b'$ , just composing the diagram in the hypothesis down from  $a$ .

Hence we can apply the inductive hypothesis to the diagram  $D_2$ , since

$$(deg(b'), dist(a', b')) <_{lex} (deg(b), dist(a, b)),$$

and we are done.  $\square$

Lemma 3.2.(iii) tells us that in using Akama's Lemma, before trying to prove directly the commutation between  $R$  and  $S\downarrow$  we should better check the one step commutation between  $R$  and  $S$ , and verify if  $R$  preserves  $S$  normal forms, which can be boring, but usually simple tasks.

### A simple proof of confluence and normalization for $\lambda^1\beta\eta\pi^*$

As a simple application, consider the typed lambda calculus  $\lambda^1\beta\eta\pi^*$  for Cartesian Closed Categories: this consists of  $\beta, \eta, \pi, SP$  and a rule  $Top$  that collapses all terms of a special type  $T$  into a single constant  $*$  (with both  $\eta$  and  $SP$  taken as expansions). If we take  $R = \beta \cup \pi \cup Top$  and  $S = \eta \cup SP$ , it is extremely simple to verify our sufficient condition, and then confluence and normalization for the full system are a consequence of the same properties for the two separate subsystems, that can be shown fairly easily with simple traditional techniques.

## 4 The calculus $\lambda^2\beta\eta$

We briefly recall that in the second order  $\lambda$ -calculus  $\lambda^2\beta\eta$

**Types** are defined by the following grammar:

$$Type ::= At \mid TVar \mid Type \rightarrow Type \mid \forall \sigma. Type,$$

where  $At$  are countably many atomic types and  $TVar$  countably many type variables.

**Terms** ( $M:A$  will stand for  $M$  is a term of type  $A$ ) are such that

- the set of terms contains a countable set  $x, y, \dots$  of term variables for each type
- terms are constructed from variables and constants via the following term formation rules (notice the perfect analogy with the introduction and elimination rules for second order logic in natural deduction style)

$$\frac{\Gamma, x : A \vdash M : B}{\Gamma \vdash \lambda x. M : A \rightarrow B} \quad \frac{\Gamma \vdash M : A \rightarrow B \quad \Gamma \vdash N : A}{\Gamma \vdash (MN) : B}$$

$$\frac{\Gamma \vdash M : A}{\Gamma \vdash \Lambda \sigma. M : \forall \sigma. A} \quad \frac{\Gamma \vdash M : \forall \sigma. A}{\Gamma \vdash M[B] : A[B/\sigma]} \text{ for any type } B.$$

**Equality** is generated by

$$(\beta) \quad (\lambda x. M)N = M[N/x] \quad (\eta) \quad \lambda x. Mx = M \text{ if } x \notin FV(M)$$

$$(\beta^2) \quad (\Lambda \sigma. M)[A] = M[A/\sigma]$$

Now we can introduce marked abstractions as in the previous Section. We have then again a set of pre-terms  $A_*^2$  generated by the grammar

$$M ::= x \mid MM \mid \lambda x. M \mid \lambda^* x. M \mid \Lambda \sigma. M \mid M[A] \quad (2)$$

and from these we define a set of marked second order terms obtained from unmarked terms by means of *unrestricted* expansions.

**Definition 4.1 (Marked terms)**

$$A_{*\eta}^2 = \{M \in A_*^2 \mid \exists M' \in \lambda^2 \beta \eta. M' \eta\text{-expands in an unrestricted way to } M\}.$$

These are the terms of the marked typed calculus  $\lambda^2 \beta \eta_*$ , that has the following associated rewriting system:

$$(\beta_*) \quad \begin{cases} (\lambda^* x. M)N \xrightarrow{\beta_*} M[N/x] \\ \lambda^* y. D[(\lambda x. M)Y] \xrightarrow{\beta_*} \lambda y. D[M[Y/x]], \text{ if } y \xrightarrow{\eta} Y \wedge [] \xrightarrow{\eta} D[] \end{cases}$$

$$(\beta) \quad (\lambda x : A. M)N \xrightarrow{\beta} M[N/x]$$

if  $(\lambda x : A. M)N$  is not in a context where  $(\beta_*)$  applies

$$(\eta) \quad M \xrightarrow{\eta} \lambda^* x : A. Mx \text{ if } \begin{cases} x \notin FV(M) \\ M : A \rightarrow B \\ M \text{ is not applied} \end{cases}$$

$$(\beta^2) \quad (\Lambda \sigma. M)[A] \xrightarrow{\beta^2} M[A/X]$$

Again, we have split  $(\beta)$  into  $(\beta)$  and  $(\beta_*)$ .

The one-step reduction relation between terms is defined as the least relation which includes  $\beta, \beta_*, \beta^2, \eta$  and is closed for all the contexts *except* in the application case:

$$\text{if } M \Longrightarrow M', \text{ then } MN \Longrightarrow M'N \text{ except in the case } M \xrightarrow{\eta} M';$$

but, for the sake of simplicity, we will avoid using an additional symbol  $\Longrightarrow$  to denote it.

<sup>6</sup> With the proviso that the type variable  $\sigma$  is not free in the type of any free variable of the term  $M$ .

**Notation 4.2** The transitive and the reflexive transitive closure of  $\longrightarrow$  are noted  $\longrightarrow_+$  and  $\longrightarrow$  respectively. Furthermore, we denote  $\longrightarrow_{\eta_{unr}}$  the one-step unrestricted  $\eta$ -reduction.

The so obtained *typed* calculus still has the following property:

**Property 4.3 (See 2.3)**

- (i)  $M \in \Lambda_{*\eta}^2 \Leftrightarrow (\forall C[] \in \Lambda_{*\eta}^2[]. M \equiv C[\lambda^*x.N] \Rightarrow N \equiv D[N'X]),$   
 where  $N' \in \Lambda_{*\eta}^2 \wedge FV(N') \not\ni x \xrightarrow{\eta} X \wedge [] \xrightarrow{\eta} D[].$
- (ii)  $(M \in \Lambda_{*\eta}^2 \wedge M \longrightarrow N) \Rightarrow N \in \Lambda_{*\eta}^2;$

*Proof.* Property (i) can be shown by induction exactly as in the untyped case. As for property (ii), we just need to focus on and  $\beta^2$  reduction, as for the other ones one can proceed exactly as in 2.3. For this, it suffices to show that if  $M \xrightarrow{\eta_{unr}} M' \xrightarrow{\beta^2} N,$  then there exists an  $M''$  such that  $M \xrightarrow{\beta^2} M'' \xrightarrow{\eta_{unr}} N.$  This is easy, because we are using the unrestricted  $\eta$  expansion. Then, given any term  $M$  in  $\Lambda_{*\eta}^2,$  we have  $M' \xrightarrow{\eta_{unr}} M \xrightarrow{\beta^2} M''$  for some  $M \in \Lambda_{*\eta}^2,$  that can be turned into  $M' \xrightarrow{\beta^2} M''' \xrightarrow{\eta_{unr}} M''$  for some  $M''' \in \Lambda_{*\eta}^2,$  so  $M'' \in \Lambda_{*\eta}^2$  too.  $\square$

## 4.1 Properties of Reduction

Let  $\gamma$  be a notion of reduction; we denote by  $\gamma\downarrow$  an exhaustive  $\gamma$ -reduction path.

**Remark 4.4** If  $Q \xrightarrow{\eta} Q',$  then  $Q[A] \xrightarrow{\eta} Q'[A].$

*Proof.* It is an easy induction on the structure of  $Q.$   $\square$

**Remark 4.5** The reductions  $\beta^2$  and  $\eta$  alone are confluent and strongly normalizing.

*Proof.* Folklore for  $\beta^2,$  see [Kes93, Cub92, DCK94a, Min79] for  $\eta.$   $\square$

**Lemma 4.6 (Commutation of  $\beta^2$  wrt  $\eta$ )**  $\beta^2$  commutes (in one step) with  $\eta.$

*Proof.* We consider all possible critical pairs between  $\eta$  and  $\beta^2:$

$$\begin{array}{ccc}
 (\Lambda\sigma.M)[A] \xrightarrow{\eta} \lambda y : B.((\Lambda\sigma.M)[A])y & (\Lambda\sigma.M)[A] \xrightarrow{\eta} (\Lambda\sigma.\lambda y : B.M)y[A] \\
 \beta^2\downarrow & \downarrow\beta^2 & \beta^2\downarrow & \downarrow\beta^2 \\
 M[A/\sigma] \xrightarrow{\eta} \lambda y : B.(M[A/\sigma])y & & M[A/\sigma] \xrightarrow{\eta} \lambda y : B[A/\sigma].(M[A/\sigma])y
 \end{array}$$

In these diagrams, the erasure of  $(\Lambda\sigma.M)[A]$  is not an abstraction, because we can apply  $\eta;$  but the erasure of  $M[A/\sigma]$  is the same, so we can still apply an  $\eta,$  and close the diagram in one step. Using these diagrams, the one step commutation property for the general case is shown by induction on the structure of contextual reductions.  $\square$

**Lemma 4.7 (Commutation of  $\eta$  with reduction to  $\beta^2$  n.f.)**

$$\begin{array}{ccc}
 M \xrightarrow{\eta} N & & \\
 \beta^2\downarrow & & \downarrow\beta^2 \\
 M' \xrightarrow{\eta} N' & & 
 \end{array}$$

*Proof.* Consider the reduction sequence from  $M$  to the  $\beta^2$  normal form  $M'$  of  $M,$  and the reduction  $M \xrightarrow{\eta} N.$  We can apply repeatedly Lemma 4.6 to close the diagram, obtaining

$$\begin{array}{ccc}
M \xrightarrow{\eta} N & & M \xrightarrow{\eta} N \\
\beta^2 \downarrow & & \beta^2 \downarrow \\
M' \xrightarrow{\eta} N'' & \text{hence} & M' \xrightarrow{\eta} N'' \\
\beta^2 \downarrow & & \beta^2 \downarrow \\
M' \xrightarrow{\eta} N'' & & M' \xrightarrow{\eta} N''
\end{array}$$

since  $\eta$  preserves  $\beta^2$  normal forms. Finally, being  $\beta^2$  normal forms unique,  $N' = N''$  so  $M' \xrightarrow{\eta} N'$  as needed.  $\square$

**Corollary 4.8**  $\beta^2 \cup \eta$  is confluent and strongly normalizing.

*Proof.* Using the previous lemma, and knowing that  $\beta^2$  and  $\eta$  separately are CR and SN, we obtain the result by Akama's Lemma.  $\square$

**Property 4.9** Relationship between:

$$\begin{array}{ccc}
\text{(i) } \beta_* \text{ and } \beta^2: & M \xrightarrow{\beta_*} N & \text{(ii) } \beta_* \text{ and } \eta: \\
\beta^2 \downarrow & \beta^2 \downarrow & \eta \downarrow \\
Q \xrightarrow{\beta_*} R & Q \xrightarrow{\beta_*} R & Q \xrightarrow{\beta_*} R
\end{array}$$

*Proof.* (i) There are no non-trivial critical pairs between  $\beta_*$  and  $\beta^2$  and since  $\beta^2$  is a rewriting rule without restrictions, it is a matter of a simple induction on the derivation of the reductions to prove the commutation. (The fact that we need only one  $\beta_*$  step to close the diagram comes from the fact that  $\beta^2$  cannot duplicate subterms.)

(ii) We use our knowledge of the structure of a marked abstraction to distinguish two cases:

$$\begin{array}{l}
\text{(a) } M \equiv C[(\lambda^*x.D[PX])T], \text{ where } x \notin FV(P), x \xrightarrow{\eta} X, [] \xrightarrow{\eta} D[[]]. \\
\text{We have } M \xrightarrow{\beta_*} C[D[PX[T/x]]] \equiv N, \text{ i.e.} \\
Q \xleftarrow{\eta} C[(\lambda^*x.D[PX])T] \equiv M \xrightarrow{\beta_*} N \equiv C[D[PX[T/x]]].
\end{array}$$

Now, four cases are possible:

1.  $Q \equiv C'[(\lambda^*x.D[PX])T]$ , with  $C[[]] \xrightarrow{\eta} C'[[]]$ . Then we have two cases:
  - $C'[[]] \equiv C[\lambda^*y.[[]]y]$  and  $D[PX[T/x]]$  is an abstraction. This can happen only if  $D[[]] \equiv \lambda^*z.D'[[]]z$ , but then
$$\begin{array}{l}
Q \xrightarrow{\beta_*} C[\lambda^*y.(\lambda^*z.(D'[PX[T/x]])z)y] \\
\xrightarrow{\beta_*} C[\lambda^*y.D'[PX[T/x]]y] \\
\equiv C[D[PX[T/x]]];
\end{array}$$
  - $Q \xrightarrow{\beta_*} C'[D[PX[T/x]]] \equiv R \xleftarrow{\eta} C[D[PX[T/x]]]$ , otherwise.
2.  $Q \equiv C[(\lambda^*x.D[P'X])T]$ , with  $P \xrightarrow{\eta} P'$ .  
The expansion in  $P'$  cannot be at the root ( $P'$  is applied) and it can be performed after the  $\beta_*$ , closing the diagram with  $R \equiv C[D[P'X[T/x]]]$ .
3.  $Q \equiv C[(\lambda^*x.D[PX'])T]$ , with  $X \xrightarrow{\eta} X'$ .  
Two cases are possible here: if  $N \xrightarrow{\eta} C[D[PX'[T/x]]]$ , then we are done. Otherwise,  $x \equiv X \xrightarrow{\eta} \lambda^*t.xt \equiv X'$  and  $T$  has an initial abstraction. Hence we have the thesis observing that

$$\begin{array}{l}
Q \equiv C[(\lambda^*x.D[P(\lambda^*t.xt)])(\lambda w.T')] \xrightarrow{\beta_*} \\
C[D[P(\lambda^*t.(\lambda w.T')t)]] \xrightarrow{\beta_*} \\
C[D[P(\lambda t.T'[t/w])] ] \equiv N.
\end{array}$$

The case where the  $\lambda$  binding the variable  $w$  is a marked one is similar.

4.  $Q \equiv C[(\lambda^*x.D[PX])T']$ , with  $T \xrightarrow{\eta} T'$ .

Here again, if  $\eta$  is not allowed after the  $\beta_*$  reduction, it is the case that we can perform another  $\beta_*$  step to close the diagram.

(b)  $M \equiv C[\lambda^*y.D[(\lambda x.P)Y]]$ , where  $y \notin FV(P)$ ,  $y \xrightarrow{\eta} Y$ ,  $[] \xrightarrow{\eta} D[]$ .

We have :

$$Q \xleftarrow{\eta} C[\lambda^*y.D[(\lambda x.P)Y]] \equiv M \xrightarrow{\beta_*} N \equiv C[\lambda y.D[P[Y/x]]].$$

Now, four cases are possible:

1.  $Q \equiv C'[\lambda^*y.D[(\lambda x.P)Y]]$ , with  $C[] \xrightarrow{\eta} C'[]$ .

Here  $Q \xrightarrow{\beta_*} R$  and  $N \xrightarrow{\eta} R$ , where  $R \equiv C'[\lambda y.D[P[Y/x]]]$ , and this case is settled.

2.  $Q \equiv C[\lambda^*y.D'[(\lambda x.P)Y]]$ , with  $D[] \xrightarrow{\eta} D'[]$ .

Two cases are possible here: if  $N \xrightarrow{\eta} C[\lambda y.D'[P[Y/x]]]$ , the thesis follows exactly as in case 1. Otherwise, we are in the case that

$$\neg(N \xrightarrow{\eta} C[\lambda y.D'[P[Y/x]]]).$$

This may only happen when  $D \equiv []$ ,  $D' \equiv \lambda^*t.[ ]t$  and  $P$  has an external abstraction. Hence we have the thesis observing that

$$\begin{aligned} Q &\equiv C[\lambda^*y.\lambda^*t.(\lambda x.(\lambda w.P'))Yt] \\ &\xrightarrow{\beta_*} C[\lambda y.\lambda^*t.(\lambda w.P')[Y/x]t] \\ &\xrightarrow{\beta_*} C[\lambda y.\lambda t.P'[Y/x, t/w]] \equiv N. \end{aligned}$$

The case where the  $\lambda$  binding the variable  $w$  is a marked one is similar.

3.  $Q \equiv C[\lambda^*y.D[(\lambda x.P')Y]]$ , with  $P \xrightarrow{\eta} P'$ .

Similar to case 2, with some small adjustments.

4.  $Q \equiv C[\lambda^*y.D[(\lambda x.P)Y']]$ , with  $Y \xrightarrow{\eta} Y'$ .

Two cases are possible here: if  $N \xrightarrow{\eta} C[\lambda y.D[P[Y'/x]]]$ , the thesis follows exactly as in case 1. Otherwise, we are in the case that

$$\neg(N \xrightarrow{\eta} C[\lambda y.D[P[Y'/x]]]).$$

This may happen when  $y \equiv Y$  and some occurrences of  $x$  are  $P$  is in functional position in applications. Let us then distinguish such occurrences, denoting them by  $\bar{x}$ ; moreover, let us assume that  $P[Y'/x, Y/\bar{x}]$  denotes the term obtained from  $P$  substituting  $Y'$  for occurrences of  $x$  which are not in functional position in  $P$ , and  $Y$  for those in functional position. Hence we have the thesis observing that

$$Q \xrightarrow{\beta_*} C[\lambda y.D[P[Y'/x]]] \xrightarrow{\beta_*} C[\lambda y.D[P[Y'/x, Y/\bar{x}]]] \xleftarrow{\eta} N.$$

□

**Property 4.10**  $\beta_*$  preserves  $\beta^2$  and  $\eta$ -normal forms.

*Proof.* We show that if a redex is not in  $\beta^2$  ( $\eta$ )-normal form, then the redex is not in  $\beta^2$  ( $\eta$ )-normal form either.

It is not possible to create  $\eta$  expansion redexes by  $\beta$ -reduction in general, since this reduction preserves the type of all subterms: imagine indeed we have a reduction  $C[(\lambda x: A.M)N] \xrightarrow{\beta} C[M[N/x]]$ , where the second term has an  $\eta$ -redex. If the redex is inside  $N$  or  $M$  or  $C[]$ , then it already exists in the first term. If it is  $M$  or  $C[]$ , then again it is already in the first term. If it is one of the new occurrences of  $N$ , then notice that these occurrences have the same type as  $N$  in the first term, so  $N$  in the first term is a redex too.

For  $\beta^2$ , we use the fact that the substitutions done by  $\beta_*$  always involve terms that are not of quantified type, and hence cannot create  $\beta^2$  redexes.

**Lemma 4.11 (Commutation of  $\beta^2$  and  $\eta$  n.f. wrt  $\beta_*$ )** *If  $M \xrightarrow{\beta_*} N$ , then at least one step of  $\beta_*$  can be performed on the  $\beta^2 \cup \eta$ -n.f. of  $M$  to reach the  $\beta^2 \cup \eta$ -n.f. of  $N$ .*

*Proof.* Just notice that Properties 4.9 and 4.10 fulfill the hypothesis of Lemma 3.2.(iii).  $\square$

**Corollary 4.12** *The reduction  $\beta^2 \cup \eta \cup \beta_*$  is strongly normalizing.*

*Proof.* By the previous lemma, and the separate strong normalization of  $\beta^2 \cup \eta$  reduction and  $\beta_*$  reduction. Notice that, since  $\beta_*$  is *not* confluent, one cannot apply here directly Akama's Lemma. Indeed, one can prove that  $\beta^2 \cup \eta \cup \beta_*$  is confluent also, but it is not necessary for the general result.  $\square$

**Theorem 4.13** *The reduction  $\beta^2 \cup \eta \cup \beta \cup \beta_*$  is confluent and strongly normalizing.*

*Proof.* Assume the existence of an infinite reduction in the typed  $\lambda$ -calculus:

$$II: M_0 \longrightarrow M_1 \longrightarrow \dots$$

We associate to  $II$  a sequence

$$II': M'_0 \xrightarrow{=} M'_1 \xrightarrow{=} \dots$$

in the untyped  $\lambda$ -calculus, such that, for all  $i$ ,  $M'_i = \text{erasure}(M_i)$ . We observe that  $II'$  is still infinite, since, by Corollary 4.12,  $II$  must contain an infinite amount of  $\beta$  steps, and

$$\forall M, N \in \Lambda_{*\eta}^2, (M \xrightarrow{\beta} N) \Rightarrow (\text{erasure}(M) \xrightarrow{\beta} \text{erasure}(N)).$$

By Lemma 2.7.(iii),  $II'$  can be transformed into a fair sequence  $II''$ . Now, we know that  $M'_0$  is strongly  $\beta$ -normalizing, since it is the erasure of a typed term. Hence  $II''$  contradicts Lemma 2.9, and this proves the strong normalization property.

Finally, the system is weakly confluent (for independent reasons, the diagrams in the previous Lemmas show almost all relevant cases), so confluence follows by Newman's Lemma.  $\square$

**Corollary 4.14 (Strong normalization and confluence for  $\lambda^2\beta\eta$ )**

*The reduction  $\beta^2 \cup \eta \cup \beta$  is confluent and strongly normalizing.*

*Proof.* A simple consequence of the previous result, because of the direct correspondence between reduction sequences in the marked and in the unmarked calculi.  $\square$

## 5 Conclusion

In this paper, not only we presented the very first proof that the expansive approach to extensional equalities, most notably  $\eta$ , can be successfully carried on to the second order typed  $\lambda$ -calculus, but we did it by means of extremely elementary methods, that do not involve reducibility candidates, complex translations or difficult syntactic analysis of expansionary normal forms.

This elementarity can be clearly seen by considering the first order case: in the absence of  $\beta^2$ , there is no need to single out a  $\beta_*$  reduction as in the second order case, and using the Lemma in Section 3 one can get a proof much simpler than all the known proofs mentioned in the Introduction.

The key of the success is twofold: on one side, the marking that tracks the  $\beta$ -redexes created because of expansions, and on the other side, the simple Lemma 3.2, whose hypothesis are easy to verify (this last can have, in these authors' opinion, wide applicability in the theory of abstract term rewriting systems).

It is now important to turn towards several extensions of this result: is it possible to handle in the same way extensionality for quantified types ( $\eta^2$ )? What about combinations with algebraic rewriting systems? What about the *Top* type? All these questions are currently under active investigation.

## Acknowledgements

We would like to thank Delia Kesner, for many discussions and her fundamental help with all matters concerning expansion rules.

## References

- [Aka93] Yohji Akama. On Mints' reductions for ccc-calculus. In *Typed Lambda Calculus and Applications*, number 664 in LNCS, pages 1–12. Springer Verlag, 1993.
- [Bar84] Henk Barendregt. *The Lambda Calculus; Its syntax and Semantics (revised edition)*. North Holland, 1984.
- [CDC91] Pierre-Louis Curien and Roberto Di Cosmo. A confluent reduction system for the  $\lambda$ -calculus with surjective pairing and terminal object. In Leach, Monien, and Artalejo, editors, *Intern. Conf. on Automata, Languages and Programming (ICALP)*, volume 510 of *Lecture Notes in Computer Science*, pages 291–302. Springer-Verlag, 1991.
- [Cub92] Djordje Cubric. On free ccc. Distributed on the `types` mailing list, 1992.
- [DCK93] Roberto Di Cosmo and Delia Kesner. Simulating expansions without expansions. Technical Report LIENS-93-11/INRIA 1911, LIENS-DMI and INRIA, 1993.
- [DCK94a] Roberto Di Cosmo and Delia Kesner. Modular properties of first order algebraic rewriting systems, recursion and extensional lambda calculi. In *Intern. Conf. on Automata, Languages and Programming (ICALP)*, Lecture Notes in Computer Science. Springer-Verlag, 1994.
- [DCK94b] Roberto Di Cosmo and Delia Kesner. Simulating expansions without expansions. *Mathematical Structures in Computer Science*, 1994. A preliminary version is available as Technical Report LIENS-93-11/INRIA 1911.
- [Dou93] Daniel J. Dougherty. Some lambda calculi with categorical sums and products. In *Proc. of the Fifth International Conference on Rewriting Techniques and Applications (RTA)*, 1993.
- [Ges] Alfons Geser. *Relative termination*. PhD thesis, Dissertation, Fakultät für Mathematik und Informatik, Universität Passau, Germany, 1990. Also available as: Report 91-03, Ulmer Informatik-Berichte, Universität Ulm, 1991.
- [JG92] Colin Barry Jay and Neil Ghani. The virtues of eta-expansion. Technical Report ECS-LFCS-92-243, LFCS, 1992. University of Edinburgh.
- [Kes93] Delia Kesner. *La définition de fonctions par cas à l'aide de motifs dans des langages applicatifs*. Thèse de doctorat, Université de Paris XI, Orsay, december 1993. To appear.
- [Min79] Gregory Mints. Teorija kategorii i teorija dokazatelstv.I. *Aktualnye problemy logiki i metodologii nauky*, pages 252–278, 1979.
- [Nes89] Dan Nesmith. An application of Klop's counterexample to a higher-order rewrite system. Draft Paper, 1989.