LFCS

Laboratory for Foundations of Computer Science
Department of Computer Science - The University of Edinburgh

Long $\beta\eta$ Normal Forms and Confluence

by

C. Barry Jay

LFCS Report Series
LFCS
Department of Computer Science
University of Edinburgh
The King's Buildings
Edinburgh EH9 3JZ

ECS-LFCS-91-183
October 1991

Copyright © 1991, LFCS

Copyright © 1991, Laboratory for Foundations of Computer Science,
University of Edinburgh. All rights reserved.

Reproduction of all or part of this work
is permitted for educational or research use
on condition that this copyright notice is
included in any copy.

Long $\beta\eta$ Normal Forms and Confluence

C. Barry Jay *

LFCS, Department of Computer Science
University of Edinburgh
The King's Buildings, Mayfield Road
Edinburgh, UK, EH9 3JZ
e-mail: cbj@dcs.ed.ac.uk

October 21, 1991

Abstract

η-conversion and surjective pairing can be treated as expansion rules in the simply-typed λ-calculus with terminal object, surjective pairing and iterator, to obtain a fully confluent and essentially normalising rewriting system. By adding some categorically motivated, but syntactic, restrictions to these expansions, one obtains a confluent reduction system which is strongly normalising to the long $\beta\eta$ normal forms.

1 Introduction

Reduction systems for the simply-typed λ-calculus with terminal type suffer from the tension between the desire for confluence (being Church-Rosser) and and for strong normalisation. If, as is usual, η-conversion is interpreted as a contraction then confluence fails in the presence of non-trivial terms of terminal type (attributed to Obtulowicz in [15]).

Now full confluence is not only desirable in its own right, but is necessary for a new categorical semantics of reduction. The denotations of λ-terms are well understood using the theory of cartesian closed categories, in which types are represented by objects and terms by morphisms [15]. Now the goal [13] is to model reduction of terms by means of a reduction relation, or pre-order, on the morphisms of the category, while retaining the power of the denotational semantics that comes from adjoints and cartesian closure. The formal development of such a semantics

*Research supported by The Royal Society of Edinburgh/BP, and NSERC operating grant OGPIN 016.

requires that the reduction order on the morphisms be fully confluent (though not necessarily strongly normalising). Thus, previous results were inadequate.

Curien and Di Cosmo [3] recovered confluence by applying the Knuth-Bendix algorithm to obtain additional reduction rules, while only preserving weak normalisation. Categorical insights, however, suggest an alternative approach: that η be treated as an expansion rule. That is, on terms $t : A \to B$ we have

$$t \Rightarrow \lambda x.tx$$

Again, confluence holds for all terms, but normalisation is lost, as can be seen from the following infinite reduction sequence

$$t \Rightarrow \lambda x.tx \Rightarrow \lambda y.(\lambda x.tx)y \Rightarrow \ldots \tag{1}$$

Note, however, that this reduction sequence is not actually performing any "work" since after the first step all the reductions can be reversed by β contractions. Another way of viewing this is to consider the following *reduction loops* whose steps are *reversible reductions*

$$\begin{array}{ccccc} \lambda x.t & \Rightarrow & \lambda y.(\lambda x.t)y & \Rightarrow & \lambda y.t[y/x] \equiv \lambda x.t \\ ta & \Rightarrow & (\lambda x.tx)a & \Rightarrow & ta \end{array} \tag{2}$$

These correspond to the triangle laws of an adjunction with β as the counit and η as the unit.

The formal study of adjunctions between ordered categories (and, more generally, 2-categories and bicategories) [5, 14, 10, 11, 2, 12] has always involved the presence of local triangle laws for the adjunctions, in which the local unit (reduction) is an expansion and the local counit (reduction) is a contraction. These structures have inspired models of reduction [19, 18], and program refinement [8], though they aren't always committed to the triangle laws.

In the presence of pairing we can consider a similar expansion law

$$c \Rightarrow \langle \pi c, \pi' c \rangle$$

for terms $c : A \times B$ which is *truely* surjective pairing since every term of product type reduces to a pair. We also obtain a similar pair of loops (the second and third counting as one loop in the product category)

$$\begin{array}{ccccc} \langle a, b \rangle & \Rightarrow & \langle \pi\langle a, b\rangle, \pi'\langle a, b\rangle \rangle & \Rightarrow & \langle a, b \rangle \\ \pi c & \Rightarrow & \pi\langle \pi c, \pi' c\rangle & \Rightarrow & \pi c \\ \pi' c & \Rightarrow & \pi'\langle \pi c, \pi' c\rangle & \Rightarrow & \pi' c \end{array} \tag{3}$$

Thus, we can account for the presence of looping reductions, and perhaps even find them mathematically desirable, but this is not yet sufficient to model computation, which requires termination. The solution is simply to cut the loops by banning those expansions which form part of a triangle law. This can be done

syntactically, and leads to a confluent and strongly normalising reduction system. The proof of normalisation requires a modest extension to that of Girard's for the contraction-based system [4] (inspired by his "Candidates de Reducibilité" techniques). These restrictions on expansions, though not the proof or motivation, were first exploited by Mints [16] for technical purposes, but have been supplanted in his forthcoming collected works.

The normal forms of this system have a longer history, being studied by Prawitz [17], as the *extended normal forms*, and later by Huet [9] as the *long $\beta\eta$ normal forms*. Unlike the present system, reduction proceeded there in two stages; first do β-reduction, then η-expansion, subject to restrictions that prevent the infinite reductions of (1). These normal forms appear, for example, in the development of the LF logical framework [6] and in the study of type classes [7].

The defect (from a theoretical viewpoint) of these restrictions on expansion is that reduction now depends on the context, i.e. the result is not a rewriting system. If we insist on a rewriting system and refuse the restrictions then all is not lost as the system is still *essentially normal* in that the amount of "work" (i.e. irreversible reduction) which can be done is bounded. It is expected that efficient implementations will automatically exclude reduction loops without any special effort, in which case the restrictions will benefit neither theory nor practice.

In any event, we now have two confluent systems: an essentially normalising rewriting system and; a strongly normalising reduction system.

2 Modelling the Simply-Typed λ-calculus

Although all systems are defined, some familiarity with λ-calculus is assumed, e.g. one of [1, 15, 4].

Here is the unrestricted reduction system for simply-typed λ-calculus with terminal type, surjective pairing, and iterator. The *types* are given by

$$T := T{\times}T \mid T{\to}T \mid N \mid 1$$

Each type T comes equipped with a collection of variables $x : T$ denoted by the letters x, y, z. Other lower case letters will denote arbitrary terms. The *terms* are given by

$$t := x \mid \lambda x.t \mid t_1 t_2 \mid * \mid \langle t_1, t_2 \rangle \mid \pi t \mid \pi' t \mid 0 \mid S \mid \mathtt{It}(a, h, n)$$

where $0 : N$ and $S : N{\to}N$ representing zero and successor, and $* : 1$. Also, if $a : A$ and $h : A{\to}A$ with $n : N$ then $\mathtt{It}(a, h, n) : A$. For the other terms the usual conventions concerning type-assignment and well-formedness apply.

Now we will define the *expansionary reduction system.* Its *basic reductions* are:

(β)	$(\lambda x.t)a$	$>$	$t[a/x]$
(η')	t	$>$	$\lambda x.tx$
(π)	$\pi\langle a,b\rangle$	$>$	a
(π')	$\pi'\langle a,b\rangle$	$>$	b
(sp)	c	$>$	$\langle \pi c, \pi' c\rangle$
$(*)$	a	$>$	$*$
$(It0)$	$\mathtt{It}(a,h,0)$	$>$	a
(ItS)	$\mathtt{It}(a,h,Sn)$	$>$	$h\mathtt{It}(a,h,n)$

where $t : A{\to}B$ and $c : A{\times}B$ and $a : 1$ is not $*$. The β rule involves implicit α-conversion whenever substitution threatens to capture free variables of a. The *1-step* reductions of a term are obtained by applying a basic reduction to (some or all instances of) one of its sub-terms. The transitive closure of the 1-step reductions is the reduction relation, denoted $\Rightarrow$. Reductions built solely from η' and (sp) are called *expansions.* Conversely, reductions built without any use of expansions are *contractions.* A reduction $t{\Rightarrow}t'$ for which $t'{\Rightarrow}t$ is called *reversible.*

Theorem 2.1 *The expansionary reduction system is confluent.*
Proof To establish confluence it suffices to consider the system whose 1-step reductions are either 1-step expansions or (general) contractions, and show that any critical pair of 1-step reductions can be completed by 1-step reductions. Now the standard arguments show that contractions in isolation are confluent, and also 1-step expansions commute with any reduction. For example, if $t{\Rightarrow}t'$ by a contraction and $t{\Rightarrow}t''$ by a 1-step expansion then there is a term t''' such that $t'{\Rightarrow}t'''$ by a 1-step expansion (or equality) and $t''{\Rightarrow}t'''$ by a contraction. □

Although the looping reductions (2) and (3) show that this system isn't strongly normalising, this defect can be repaired by restricting expansion to avoid these loops, i.e. require that

- (η') is not applied to λ-terms
- (η') is not applied to sub-terms which are evaluated
- (sp) is not applied to pairs
- (sp) is not applied to sub-terms which are projected

This has the effect of removing all the reversible reductions, but note that the result is not a rewriting system, i.e. reduction is not context-free.

Theorem 2.2 (Mints) *The restricted reduction system is confluent and strongly normalising with normal forms given by the long $\beta\eta$ normal forms.*
Proof The proof of confluence for the expansionary system can be repeated, with the following modification to its final example. The expansion $t'{\Rightarrow}t'''$ may

not be legal in the restricted system, in which case there is a contraction $t'''\Rightarrow t'$ and we can replace t''' by t'.

The proof of strong normalisation is an adaptation of Girard's proof for η-contractions, based on his "Candidates de Reducibilité" techniques [4]. Each type T has a set RED_T of *reducible terms* of that type, defined by induction on its type structure. Let $t : T$ be a term.

- If T is an atomic type then t is reducible if it is strongly normalisable.
- If $T = U \times V$ then t is reducible if πt and $\pi' t$ are.
- If $T = U \rightarrow V$ then t is reducible if $tu : V$ is reducible for all reducible $u : U$.

The *neutral* terms are those which are neither λ-abstractions nor pairs. Girard's proof proceeds by simultaneously establishing the following three hypotheses.

(CR1) If $t \in \text{RED}_T$ then t is strongly normalisable.

(CR2) If $t \in \text{RED}_T$ and $t \Rightarrow t'$ then $t' \in \text{RED}_T$.

(CR3) If t is neutral and every 1-step reduct of t is in RED_T then $t \in \text{RED}_T$.

This proof would apply unchanged to the present system, except that it assumes that variables are reducible, a consequence of their normality. Here, however, expansions can be applied to variables, and so we must introduce yet one more hypothesis to the induction, which implies the reducibility of variables.

(CR5) If $x : T$ is a variable then all its tests are reducible.

where the *tests* for a term t form the smallest set $\text{TEST}(t)$ of terms containing t such that if $t' : T'$ is in $\text{TEST}(t)$ then

- if $T' = U \times V$ then $\pi t, \pi' t \in \text{TEST}(t)$
- if $T' = U \rightarrow V$ and $u : U$ is reducible then $t'u \in \text{TEST}(t)$

The simultaneous proof of CR1-3 and CR5 proceeds by induction on the type structure. The proof steps for the first three hypotheses are unchanged. The proof of CR5 is by induction on the structure of the type of the test term t.

If t is of base type (1 or N) we must show it is normalisable. The only redexes that it has are those of the (finite number of) reducible terms u_i introduced in its construction. By CR1 these are all strongly normalisable, whence t is.

If t is of compound type then its reducibility follows directly from the definition and the induction hypothesis. □

Consider again the (unrestricted) expansionary system. We will see that there is a sense in which it is normalising, namely that one cannot perform unlimited numbers of irreversible steps in a reduction sequence.

Definition 2.3 *A term t in a reduction system R is an* essential normal form *if any reduction of it is reversible. It is* essentially normalising *if there is a number k, called an* essential bound *for t, such that each reduction sequence from t has at most k irreversible steps. The reduction system is* essentially normalising *if every term is so.*

Theorem 2.4 *The expansionary system is essentially normalising. Its essential normal forms are all equivalent to Huet's long $\beta\eta$ normal forms [9].* □

A direct proof of this result would require us to establish which reductions are reversible. That this is somewhat tedious is demonstrated by the following reduction loop for $c : A\times(B\times C)$

$$\begin{aligned} \pi c &\Rightarrow \pi\langle \pi c, \pi' c\rangle \\ &\Rightarrow \pi\langle \pi c, \langle \pi\pi' c, \pi'\pi' c\rangle\rangle \\ &\Rightarrow \pi c \end{aligned}$$

whose second step is not built from the reversible steps we have already met.

Instead we will work with the *triangular reductions* which are generated by the ten 1-step reductions of (2) and (3) and closed under applications to sub-terms, reflexivity and transitivity. All other basic reductions generate the *square reductions.*

Lemma 2.5 *Let $t\Rightarrow t'$ by a sequence of (1-step) reductions in the expansionary system. Then there is another such sequence employing the same number of square steps, which is given by a contraction $t\Rightarrow t''$ followed by an expansion $t''\Rightarrow t'$.*

Proof Consider an η-expansion followed by a β-contraction. If they don't interact then they commute without affecting their triangularity. If they do interact then we have a reduction loop whose triangular reductions can be deleted. □

Proof (Theorem 2.4) Consider a reduction sequence $t\Rightarrow t'$. It suffices to show that any bound k for reduction on t in the restricted system is an essential bound in the expansionary system. Without loss of generality we may assume that t' is not reducible by contractions (e.g. a long $\beta\eta$ normal form). By the previous lemma we may also assume this of t, whence the reduction sequence consists of expansions. If any of these were triangular then t' would be reducible by a contraction (since expansions cannot reduce the number of contractible sub-terms), contradicting our assumption. Thus the reduction sequence consists of square expansions, whose length is thus bounded by the previous theorem. □

3 Acknowledgements

I would like to thank Pierre-Louis Curien, Neil Ghani, Edmund Kazmierczak, Gordon Plotkin, and the members of the Applications of Categories Club for many stimulating conversations. I would also like to thank Sergei Soloviev for providing a translation of some of the key passages in [16].

References

[1] H.P. Barendregt, *The Lambda Calculus Its Syntax and Semantics (Revised Edition)* Studies in Logic and the Foundations of Mathematics 103 (North Holland, 1984).

[2] A. Carboni, G.M. Kelly and R. Wood, *A 2-categorical approach to change of base and geometric morphisms I*, Sydney Category Seminar Reports 90-1 (1990).

[3] P-L. Curien and R. Di Cosmo, A confluent reduction for the λ-calculus with surjective pairing and terminal object, in: *Proceedings ICALP '91, Madrid* LNCS 510 (Springer, 1991) 291–302 (also appearing as Tech. Rep. LIENS–91–11).

[4] J-Y. Girard, P. Taylor and Y. Lafont, *Proofs and Types*, Cambridge Tracts in Theoretical Computer Science (Cambridge University Press, 1989).

[5] J.W. Gray, *Formal category theory: adjointness for 2-categories*, Lecture Notes in Mathematics 391 (Springer-Verlag, 1974).

[6] R. Harper, F. Honsell and G. Plotkin, *A framework for defining logics*, Technical Report LFCS-91-162 (LFCS, 1991).

[7] B.P. Hilken and D.E. Rydeheard, *Towards a categorical semantics of type classes*, Fundamenta Informatica, to appear.

[8] C.A.R. Hoare, He Jifeng & C.E. Martin, *Pre-adjunctions in Order Enriched Categories*, Oxford University Computing Laboratory (1989).

[9] G. Huet, *Résolution d'équations dans des langages d'ordre* $1, 2, \ldots, \omega$. Thèse d'Etat, Université de Paris VII, 1976.

[10] C.B. Jay, *Local adjunctions*, J. Pure and Appl. Alg. 53 (1988) 227–238.

[11] C.B. Jay, *Extendinging properties to categories of partial maps*, LFCS Tech. Rep. 90-107.

[12] C.B. Jay, Partial functions, ordered categories, limits and cartesian closure, in: G. Birtwistle (ed), *IV Higher Order Workshop, Banff, 1990*, (Springer, 1991).

[13] C.B. Jay, Modelling reduction in confluent categories, in: *Proceedings of the Durham Symposium on Applications of Categories in Computer Science*, submitted.

[14] S. Kasangian, G.M. Kelly and F. Rossi, *Cofibrations and the realization of non deterministic automata*, Cahiers de Topologie et Géometrie Différentielle 24 (1983) 23–46.

[15] J. Lambek and P. Scott, *Introduction to higher order categorical logic*, Cambridge Studies in Advanced Mathematics 7 (Cambridge Univ. Press, 1986).

[16] G.E. Mints, Teorija categorii i teoria dokazatelstv.I., in: *Aktualnye problemy logiki i metodologii nauky , Kiev, 1979* 252-278.

[17] D. Prawitz, Ideas and results in proof theory, in: J.E. Fenstad (ed) *Proc. 2nd Scandinavion Logic Symp.* (North-Holland, 1971) 235–307.

[18] D.E. Rydeheard & J.G. Stell, *Foundations of equational deduction: A categorical treatment of equational proofs and unification algorithms*, in: Pitt et al, (eds), *Category Theory and Computer Science*, Lecture Notes in Computer Science 283 (Springer, 1987) 114 – 139.

[19] R.A.G. Seely, *Modelling computations: a 2-categorical framework*, in: *Proceedings of the Second Annual Symposium on Logic in Computer Science* (1987).