On modular properties of higher order extensional lambda calculi

Roberto Di Cosmo Neil Ghani

DMI-LIENS (CNRS URA 1327)
Ecole Normale Supérieure
45, Rue d'Ulm
75230 Paris, France
e-mail:{dicosmo,ghani}@dmi.ens.fr

Abstract. We prove that confluence and strong normalisation are both modular properties for the addition of algebraic term rewriting systems to Girard's F^ω equipped with either β-equality or $\beta\eta$-equality. The key innovation is the use of η-expansions over the more traditional η-contractions.

We then discuss the difficulties encountered in generalising these results to type theories with dependent types. Here confluence remains modular, but results concerning strong normalisation await further basic research into the use of η-expansions in dependent type theory.

1 Introduction

A property P is *modular* for the combination of rewrite systems $\mathcal{T}_1$ and $\mathcal{T}_2$ iff whenever both $\mathcal{T}_1$ and $\mathcal{T}_2$ satisfy P, then so does the combined rewrite system $\mathcal{T}_1 \cup \mathcal{T}_2$. This paper studies the modularity of confluence and strong normalization for combinations of higher order lambda calculi and algebraic term rewriting systems. That is, does the addition of a confluent algebraic TRS to a higher order lambda calculus (with or without rewrite rules for η-conversion) produce a system which is still confluent? Similarly, is the combination of a strongly normalising algebraic TRS and a higher order lambda calculus (again, with or without rewrite rules for η-conversion) still strongly normalising? And can these results be generalised to dependent type theories such as the Calculus of Constructions?
These questions are quite relevant both from a theoretical point of view, where one looks for general results on combination of rewriting systems, and from a practical point of view, when one develops higher order semi-unification algorithms, or establishes the formal properties of algebraic-functional languages.

Tannen [9] showed that strong normalization and confluence are both moldular properties for the combination of algebraic TRS's with the simply typed lambda calculus equipped with β-reduction. Gallier and Tannen [10, 11] extended these results to System F. Although strong normalisation remains modular in these type theories if we work with both β- and η-reductions, confluence is no longer a modular property. For example, if s is a base type with constants $f : s \to s$ and $* : s$ and with a rewrite rule $fx \Rightarrow *$, then $\Rightarrow$ is confluent. However, the combination of $\Rightarrow$ with the contractive η-rewrite rule fails to be confluent:

$$\lambda x.* \Leftarrow \lambda x.fx \Rightarrow f$$

Because of these problems with η-contractions, later research was restricted to adding more expressive TRSs to systems equipped only with β-reduction. In particular, translations into intersection type-assignment systems [3, **?**, 27, 6, 5, 7, 4] were used to prove the modularity of strong normalisation and *completeness*, i.e. the property of strong normalisation and confluence together, with confluence following from strong normalisation by Newman's lemma. As far as the authors are aware, modularity of confluence *alone* was not pursued any further and no attempts were made to study modularity results for calculi equipped with $\beta\eta$-equality.

This paper extends the works of Tannen and Gallier in several ways. Firstly, we shall consider more expressive calculi such as Girard's F^ω and Coquand and Huet's Calculus of Constructions. We show that confluence is modular for the combination of algebraic TRS's with these calculi (without η-conversion). As mentioned earlier, these results are surprisingly missing in the literature.

Our second contribution is to extend these modularity results to calculi equipped with $\beta\eta$-equality. This is done by replacing the problematic interpretation of η-conversion as a contractive rewrite relation with its more recent interpretation as an expansionary rewrite rule. Eta-expansions in the simply typed λ-calculus were first studied in the 70's but only recently they made the object of accurate study in a number of papers [1, 18, 13, 22, 28, 19] (for an up-to-date survey, the interested reader can refer to [**?**]) and much of this paper relies on Ghani's recent application of η-expansions to F^ω [**?**] and the Calculus of Constructions [**?**].

In this paper, we prove that confluence is a modular property when algebraic TRS's are added to F^ω and the Calculus of Constructions equipped with both β-reductions *and* η-expansions . We then show strong normalization is a modular property for the combination of algebraic TRS's with F^ω (with and without expansive η-rewrite rules). However we are unable to extend this result to the Calculus of Constructions as the use of η-expansions here remains an area of active research. In summary, the major modularity results proved by this paper are:

System	β		$\beta\eta$	
F^ω	CR	Corollary 10	CR	Corollary 14
	SN	Corollary 22	SN	Corollary 24
Coc	CR	Corollary 27	CR	Corollary 29

2 Extensional and Non-extensional F^ω

We use the standard notions of substitutions, reduction, normal form, confluence, normalization, etc., from the theory of λ-calculus and rewriting systems [8, 14]. The *free variables* of a term M are denoted $FV(M)$ and we write $M\theta$ for the result of applying a substitution θ to the term M. The *domain* of a substitution θ is denoted $\mathtt{dom}(\theta)$. In addition, if $\mathcal{R}$ is a rewrite relation with unique normal forms, then reduction to $\mathcal{R}$-normal form is denoted $\mathcal{R}\downarrow$ and the unique $\mathcal{R}$-normal form of t is denoted $\mathcal{R}(t)$.

In this section, two versions of F^ω will be defined. *Extensional* F^ω uses $\beta\eta$-equality for type conversion while *non-extensional* F^ω has only β-equality for type conversion. Our presentation of these calculi is based on Gallier's [24] and we only give here the typing rules, which are necessary for a self-contained introduction of expansions, while referring the reader to that paper for the formal definition of the usual β rewriting rules.

Formally, let $*$ be a distinguished symbol and let $\mathtt{TVar}$ and $\mathtt{Var}$ be disjoint sets of type variables and term variables. These variables are used to define the *kinds*, *types* (also called *type constructors*) and *terms* of F^ω as follows:

$$\begin{array}{rl} (Kinds) & K := *|K \to K \\ (Types) & T := t|T{\to}T|\forall t : K.T|\lambda t : K.T|T\,T \\ (Terms) & M := x|\lambda x : T.M|M\,M|\Lambda t : K.M|M[T] \end{array}$$

where $t \in \mathtt{TVar}$ is a type variable and $x \in \mathtt{Var}$ is a term variable. A term is called an *abstraction* iff it is of the form $\lambda x : T.M$ or $\Lambda t : K.M$. In order to ensure that types inhabit unique kinds, we assign to each type variable t a unique kind and denote the set of type variables having kind K as $\mathtt{TVar}(K)$. This kinding information is used to define the kinding judgements of F^ω in Table 1

In order to give the typing judgements of extensional F^ω we define the usual $\beta\eta$-equality relation on well-kinded types; if two types t and s are $\beta\eta$-equal, we denote this by writing $t =_{\beta\eta} s$. The following lemma is proved in [**?**]

$$\frac{t \in \mathtt{TVar}(K)}{t : K} \qquad \frac{s : K_2 \quad t \in \mathtt{TVar}(K_1)}{(\lambda t : K_1.s) : K_1 \to K_2} \qquad \frac{t : K_1 \to K_2 \quad s : K_1}{ts : K_2}$$

$$\frac{t \in \mathtt{TVar}(K) \quad s : *}{\forall t : K.s : *} \qquad \frac{t : * \quad s : *}{t \to s : *}$$

Table 1. Kinding judgements

Lemma 1. *$\beta\eta$-equality over types can be generated by a confluent, strongly normalizing reduction relation containing β reduction and restricted η-expansions. The unique normal form of a type is its long $\beta\eta$-normal form.*

The typing judgements of extensional F^ω are defined in Table 2, while the typing judgements of non-extensional F^ω use only β-equality for type conversion. In the rest of this paper, we confine our attention to only those types that kind check and those terms that type check. In addition, we shall endevour to increase legibility by dropping all reference to the context in which a typing judgement occurs whenever we feel there is no danger of confusion arising.

$$\frac{x : T \in \mathsf{dom}(\Gamma)}{\Gamma \vdash x : T} \qquad \frac{\Gamma \vdash M : t \quad t =_{\beta\eta} s \quad s : K}{\Gamma \vdash M : s}$$

$$\frac{\Gamma, x : t_1 \vdash M : t_2}{\Gamma \vdash (\lambda x : t_1.M) : t_1 \to t_2} \qquad \frac{\Gamma \vdash M : t_1 \to t_2 \quad \Gamma \vdash N : t_1}{\Gamma \vdash MN : t_2}$$

$$\frac{\Gamma, t_1 : K \vdash M : t_2}{\Gamma \vdash \Lambda t_1 : K.M : \forall t_1 : K.t_2} \qquad \frac{\Gamma \vdash M : \forall t_1 : K.t_2 \quad \Gamma \vdash s : K}{\Gamma \vdash M[s] : t_2[s/t_1]}$$

Table 2. Typing judgements of Extensional F^ω

2.1 Eta-expansions in F^ω

As argued in the introduction, any robust result concerning the modularity of confluence in the presence of η-conversion requires its interpretation as an expansion. Therefore, we finish this section by summarizing the use of η-expansions in F^ω as presented in [**?**].

In the simply typed λ-calculus, one permits an expansion $t \Rightarrow \lambda x : A.tx$ providing that t is neither a λ-abstraction nor applied to another term. This restricted expansion relation is strongly normalising, confluent and its reflexive, symmetric and transitive closure is $\beta\eta$-equality. Thus $\beta\eta$-equality can be decided by reduction to normal form in this restricted fragment.

However, defining η-expansion in F^ω requires further care so as to avoid pitfalls caused by the presence of multiple typings for terms. For instance, if an expansion $M \xrightarrow{\eta} \lambda x : A.Mx$ is permitted providing $M : A \to B$, then η-expansion is not even confluent as there are rewrites

$$\lambda x : A'.Mx \xleftarrow{\eta} M \xrightarrow{\eta} \lambda x : A.Mx$$

where we only know that $A =_{\beta\eta} A'$ in the type-conversion relation. Worse, η-expansion defined this way does not have unique normal forms and hence the usual strategy for computing long normal forms (first contract β redexes and then perform all remaining expansions) would no longer be valid. For these reasons we define a *type normalised* form of η-expansion as follows

$$(\overline{\eta})\ M \xrightarrow{\eta} \lambda x : A.Mx,\ \text{ if } \begin{cases} x\ fresh \\ M : A \to C, \textit{with } A \to C \textit{ in type normal form} \\ M \text{ is not a } \lambda\text{-abstraction} \\ M \text{ is not applied} \end{cases}$$

Note that the existence of type normal forms is assured by lemma 1. In addition, there is no need for a type-normalised form of the higher order η-rewrite rule because if a term inhabits the types $\forall t : K.A$ and $\forall t : K'.A'$, then we must have $K = K'$. Hence our higher order η-expansion is:

$$(\eta)\ \ M \xrightarrow{\eta} (\Lambda t : K.M[t]) \text{ if } \begin{cases} t\ fresh \\ M : (\forall t : K.A) \\ M \text{ is not a polymorphic } \lambda\text{-abstraction} \\ M \text{ is not applied} \end{cases}$$

Definition 2. Let β be the rewrite relation consisting of all β-reductions on types and term. Also, let $\overline{\eta}$ be the rewrite relation consiting of all restricted expansions on types and those expansions given in equations 1 and 1. The relation η is defined by ommiting the restrcition to type normal forms in equation 1. Finally define $\beta\overline{\eta} = \beta \cup \overline{\eta}$ and $\beta\eta = \beta \cup \eta$.

We shall prove the major results such as the modularity of confluence and strong normalisation for $\beta\overline{\eta}$ and then lift these results to the more general $\beta\eta$. As the following lemma shows, either choice does not loose any equalities.

Lemma 3 **$\beta\overline{\eta}$ and $\beta\eta$ vs. F^ω-equality.** *The reflexive, symmetric and transitive closure of* $\xrightarrow{\beta\eta}$ *is the usual $\beta\eta$-equality over terms of F^ω. The same holds for* $\xrightarrow{\beta\overline{\eta}}$.

Proof. Firstly, all η equalities $M = \lambda x : A.Mx$ that seem to be forbidden by the restrictions of $\xrightarrow{\beta\eta}$ can be obtained by β-reduction of $\lambda x : A.Mx$. Thus the reflexive, symmetric, transitive closure of $\xrightarrow{\beta\eta}$ is $\beta\eta$-equality. For the second part of the lemma, notice that $\xrightarrow{\beta\eta}$ and $\xrightarrow{\beta\overline{\eta}}$ generate the same equational theory: on one side, type-normalized expansions are just a particular case of the general expansion, so $\xrightarrow{Fooexpnf}$ is included in $\xrightarrow{Fooexp}$, and the same holds for the generated equalities. On the other side, we only have to show that if $\lambda x : A.Mx$ is the reduct of a non type-normalised η-expansion of a term M, then we can show the two terms equal in $\xrightarrow{Fooexpnf}$. To do so, first type-reduce A to normal form obtaining $\lambda x : nf(A).Mx$ (this can be done in $\xrightarrow{Fooexpnf}$, and notice that this last term is an $\overline{\eta}$ expansion of M, so $\lambda x : A.Mx$ is equal to M in $\xrightarrow{Fooexpnf}$ too, and we are done.

The major theorems concerning $\beta\eta$ and $\beta\overline{\eta}$ are

Theorem 4. *The rewrite relations $\beta\overline{\eta}$ and $\beta\eta$ are confluent and strongly normalizing to the long $\beta\eta$-normal forms. The long $\beta\eta$-normal form of a term may be calculated by first contracting all β-redexes and then performing any remaining type-normalised η-expansions.*

3 Modularity Results for F^ω

In this section we define algebraic TRSs and show the modularity of confluence and strong normalisation for the unions of algebraic TRSs with F^ω.

3.1 Algebraic TRSs

Algebraic TRSs are defined as follows

Definition 5. A signature Σ consists of disjoint sets $\mathcal{T}$ of *base types* and $\mathcal{F}$ of *function symbols* together with a function which assigns to every function symbol $f \in \mathcal{F}$, a *typing* of the form $f : \alpha_1 \to \ldots \to \alpha_n \to \alpha$, where $\alpha_1, \ldots, \alpha_n, \alpha \in \mathcal{T}$ and $n \geq 0$. We say the *arity* of f is n.

Signatures are used in the definition of algebraic TRSs as follows.

Definition 6. An *algebraic rewrite rule* is an ordered pair (T, U) of algebraic terms such that T is not a variable, and every variable of U also appears in T. An *algebraic term rewriting system* $\mathcal{T}$ is a finite set $\{(T_i, U_i)\}_{i=1}^{n}$ of algebraic rewrite rules.

An algebraic TRS induces a relation on the terms of the underlying signature as follows.

Definition 7. Given an algebraic TRS T, the associated *algebraic rewrite relation* is the least binary relation $\xrightarrow{\mathcal{T}}$ on terms such that if $(T, U) \in \mathcal{T}$, θ is a substitution and C is a context, then $C[T\theta] \xrightarrow{\mathcal{T}} C[U\theta]$

Given an algebraic TRS, its union with calculi such as F^ω is defined as expected. A term of the union of an algebraic TRS and F^ω is *algebraic* if it is either a variable of base type or has the form $f\ T_1 \ldots T_n$, where $f \in \mathcal{F}$ has arity n, and every T_i is an algebraic term. Note that an algebraic term is always of base type.

3.2 Modularity of Confluence

In this section we use the proof strategy of [11], to show the modularity of confluence for the combination of algebraic TRSs with both extensional and non-extensional F^ω. In particular, we prove that reduction to long $\beta\eta$-normal form in F^ω commutes with algebraic reductions — by theorem 4 this may be done in two stages.

Lemma 8. *Reduction to β normal form commutes w.r.t. algebraic reduction, i.e. the following diagram holds:*

$$\begin{tikzcd} {} \arrow[r, "\mathcal{T}"] \arrow[d, "\beta\downarrow"'] & {} \arrow[d, dashed, "\beta\downarrow"] \\ {} \arrow[r, dashed, two heads, "\mathcal{T}"] & {} \end{tikzcd}$$

Proof. See lemma 31 in the appendix for the proof.

Lemma 9 A confluent algebraic reduction stays confluent over mixed terms. *If $\mathcal{T}$ is a confluent algebraic rewriting system (over algebraic terms), then it is confluent over the terms of $F^\omega \cup \mathcal{T}$ (mixed terms).*

Proof. This proof was first done in [11] for System F, but it holds unchanged for F^ω and the Calculus of Construction. This is because, using Gallier-Tannen's terminology, the *only* property of mixed terms involving the non-algebraic terms which is necessary for the proof to go through is the fact that (see [11], page 12) whenever a term $M = A[\phi]$ reduces via a non-trunk algebric reduction to a term M', then $M' = A[\phi']$ for the same A (the trunk is preserved by non-trunk reductions).

These lemmas allow us to derive our first modularity result, namely that of confluence for the addition of algebraic TRSs to non-extensional F^ω. As we have previously remarked, this is a new result as it shows modularity of confluence alone, and not of confluence and strong normalization together as in [7]:

Corollary 10 Modularity of confluence for non-extensional F^ω with algebraic TRSs. *The union of non-extensional F^ω with a confluent algebraic TRS is confluent.*

Proof. By lemma 8, if $t =_{\beta\cup\mathcal{T}} t'$, then $\beta(t) =_{\mathcal{T}} \beta(t')$. As $\mathcal{T}$ is confluent *over mixed terms*, $\beta(t)$ and $\beta(t')$ have a common reduct and hence t and t' have a common reduct.

Proving that confluence is modular for the addition of algebraic TRSs to extensional F^ω requires us to relate algebraic rewriting to expansive normal forms, extending what was done by DiCosmo and Kesner in [19]:

Lemma 11 Algebraic reduction commute with reduction to expansive normal form. *Reduction to $\overline{\eta}$ normal form commutes w.r.t. algebraic reduction, i.e. the following diagram holds:*

$$\begin{tikzcd} \cdot \arrow[r, "\mathcal{T}"] \arrow[d, "\overline{\eta}\downarrow"'] & \cdot \arrow[d, dashed, "\overline{\eta}\downarrow"] \\ \cdot \arrow[r, dashed, "\mathcal{T}"] & \cdot \end{tikzcd}$$

Proof. The proof is by induction on the structure of terms. The fact that the η normal form of a term is unique is necessary for the lemma to hold with arbitrary TRSs and not only left-linear ones.

As a consequence of the previous lemmas, we have the following

Corollary 12. *Reduction to $\beta\overline{\eta}$ normal form commutes with algebraic reduction, i.e. the following diagram holds:*

$$\begin{tikzcd} \cdot \arrow[r, "\mathcal{T}"] \arrow[d, "\beta\overline{\eta}\downarrow"'] & \cdot \arrow[d, dashed, "\beta\overline{\eta}\downarrow"] \\ \cdot \arrow[r, dashed, two heads, "\mathcal{T}"] & \cdot \end{tikzcd}$$

Proof. Simply by composing the diagrams from the previous lemmas one obtains:

$$\begin{tikzcd} \cdot \arrow[r, "\mathcal{T}"] \arrow[d, "\beta\downarrow"'] & \cdot \arrow[d, dashed, "\beta\downarrow"] \\ \cdot \arrow[r, dashed, two heads, "\mathcal{T}"] \arrow[d, "\overline{\eta}\downarrow"'] & \cdot \arrow[d, dashed, "\overline{\eta}\downarrow"] \\ \cdot \arrow[r, dashed, two heads, "\mathcal{T}"] & \cdot \end{tikzcd}$$

But by theorem 4, performing β normalization first and then $\overline{\eta}$ normalisation computes the $\beta\overline{\eta}$-normal form of a term.

Theorem 13 Extentional F^ω with confluent TRSs. *The union of $\beta\overline{\eta}$ with a confluent algebraic TRS is confluent.*

Proof. As in corollary 10 using corollary 12.

Corollary 14 Modularity of Confluence with general η-expansion. *The union of $\beta\eta$ (where η is not restricted to type normal forms) with a confluent algebraic TRS $\mathcal{T}$ is confluent.*

Proof. If two terms are $T \cup \beta\eta$ equivalent, they are $T \cup \beta\overline{\eta}$ equivalent and hence by theorem 13 there is a $T \cup \beta\overline{\eta}$ completion for these terms. But this is also a $T \cup \beta\eta$ completion.

3.3 Modularity of Strong Normalization

The relations $\beta\overline{\eta}$ and $\beta\eta$ were proved confluent and strongly normalising in [**?**] by a modified reducibility argument, adapted from traditional reducibility proofs [] to cope with the presence of expansionary η-rewrite rules. Reducibility arguments are designed to cope with the higher order features at the level of kinds and type constructors, while the effect of adding algebraic TRSs is only felt at the level of base types. Thus these reducibility arguments generalise to prove the modularity of strong normalisation for the combination of algebraic TRSs with extensional F^{ω}.

The key concept in modular term rewriting is the idea of *layer structure*, i.e. the ability to decompose a term constructed from symbols in the union of two disjoint signatures into a term constructed from symbols in only one signature and strictly smaller subterms whose head symbol comes from the other signature. We follow [] in using the following defintions relating to layer structure.

Definition 15 Trunk Judgements. A typing judgement $\Gamma \vdash M : s$ is called *trunk* iff M is of the form $fM_1, \ldots, M_k$ where f is a constant of arity k. A typing judgement $\Gamma \vdash M : s$ is called *non-trunk* iff it is not a trunk judgement, i.e. M is of the form $HT_1, \ldots, T_n$ where H is an abstraction or a variable, and the T_i's are either terms or types.

Note that all judgements $\Gamma \vdash M : s$ are either trunk or non-trunk because of the restriction that M is of base-sort.

Definition 16 Turnk Decompositions. An *algebraic trunk decomposition* of a typing judgement $\Gamma \vdash M : s$ consists of an algebraic typing judgement $\Delta \vdash A : s$ and a term-valued substitution ϕ such that $M = A\phi$, $\mathtt{dom}(\phi) = FV(A)$ and

- Each free variable in A occurs only once
- For each $x \in FV(A)$, the typing judgement $\Gamma \vdash \phi(x) : s$ is non-trunk.

Induction shows that all typing judgements $\Gamma \vdash M : s$ have algebriac trunk decompositions, and moreover such a decomposition $M = A\phi$ is unique upto the renaming of the free variables of A. We refer to A as a trunk of the term M. Algebraic trunk decompositions are used to analyse the reducts of a term.

Definition 17 Trunk Reductions. A reduction $M = A[\phi] \xrightarrow{\mathcal{T}} N$ is a *trunk* reduction iff the redex contracted is not a subterm of one of the $\phi(x)$'s. A reduction $M = A[\phi] \xrightarrow{\mathcal{T}} N$ is a *non-trunk* reduction step otherwise.

We would like a trunk reduction $A[\phi] \xrightarrow{\mathcal{T}} N$ to induce a reduction of the trunk A, and also that in a non-trunk reduction $M \xrightarrow{\mathcal{T}} N$, the trunk of M is the same as the trunk of N. Unfortunatley neither of these properties actually hold. The first problem is caused by non-left linear rewrite rules, eg if $f : s \to s \to s$ is a constant with associated rewrite rule $fxx \xrightarrow{\mathcal{T}} x$ and a is a non-trunk term of type s, then although there is a reduction $faa \xrightarrow{\mathcal{T}} a$, there is not one between the respective trunks, i.e. there is no reduction $fxy \xrightarrow{\mathcal{T}} x$. The second problem is caused by the ability of β-reduction to collapse the *layer structure* in a term, eg consider constants $g : s \to s$ and $a : s$. Then there is a reduction $g((\lambda x : s.x)(a)) \xrightarrow{\beta} ga$, and while the trunk of the first term is gy, the trunk of the reduct is ga.

We solve the first problem by introducing a special term variable $\jmath^s$ for each sort and then defining a special substitution $\jmath$ which maps every term variable of type s to $\jmath^s$. There is also simple patch for the second problem as outlined below.

Lemma 18 Analysis of Reductions. *Let $A\phi$ be a trunk decomposition for M*

- *If $M \xrightarrow{\mathcal{T}} N$ is not a trunk reduction, then there is an algebraic trunk decomposition $N = A\phi'$ such that for some $x \in FV(A)$, $\phi(x) \xrightarrow{\mathcal{T}} \phi'(x)$, while for all other $y \in FV(A)$, $\phi(y) = \phi'(y)$.*

– *If $M \xrightarrow{\mathcal{T}} N$ is a trunk reduction, then there is an algebraic trunk decomposition $N = A'\phi'$ such that $A\jmath \xrightarrow{\mathcal{T}} A'\jmath$ and for every $y \in FV(A')$, there exists an $x \in FV(A)$ such that $\phi'(y) = \phi(x)$.*

– *If $M \xrightarrow{\beta\overline{\eta}} N$, then there is an algebraic trunk decomposition $N = A'\phi'$ and for every $y \in FV(A')$, there exists an $x \in FV(A)$ such that either $\phi(x) \xrightarrow{\beta\overline{\eta}} N_x$ and $\phi'(y)$ is a subterm of N_x or $\phi(x) = \phi'(y)$*

Proof. The lemma is proved by induction on the term M.

Lemma 19 $\mathcal{T}$ is strongly normalising on all terms. *If $\mathcal{T}$ is a strongly normalising algebraic TRS, then its extension to F^ω is also strongly normalising.*

Proof. The lemma is proved by induction on the structure of terms with the only interesting case being a trunk term $M = A\phi$. By lemma 18, any infinite reduction sequence of M induces either an infinite reduction sequence of a $\phi(x)$, or an infinite reduction sequence of $A\jmath$. The first possiblity is impossible by the induction hypothesis, while the second possibility is also impossible as $\mathcal{T}$ is strongly normalising on algebraic terms and $A\jmath$ is algebraic.

We now prove the main result of this section, namely that the union of a strongly normalising algebraic TRS and $\beta\overline{\eta}$-reduction in F^ω is strongly normalising. The proof follows the modified reducibility argument of [] and thus we only sketch the general reducibility argument and concentrate instead on the particular novelties which arise via the addition of algebraic TRSs. One defines a notion of *reducibility candidate* and *reducibility parameter* exactly as in [] and proves that if T is a type and θ is a reducibility parameter, then $T\theta$ is a reducibility candidate. The only new case is when T is a sort s and here the reducibility candidate $s\theta$ is defined to be the strongly normalising terms of type s. The following pair of lemmas are the key to completeing the proof.

Lemma 20 Induction Hypothesis For Reducibility. *If the terms $t_1, \ldots, t_n$ are strongly normalising, then so is $ft_1 \ldots t_n$.*

Proof. That there are no infinite $\beta\overline{\eta}$ reduction sequences is proved in []. By corollary 12, a rewrite $ft_1 \ldots t_n = M \xrightarrow{\mathcal{T}} N$ induces a sequence of rewrites $M_0 \xrightarrow{\mathcal{T}^*} N_0$ where M_0 and N_0 are the long $\beta\eta$-normal forms of M and N. Close inspection of the proof shows that if the initial rewrite is of the trunk, then this induced rewrite sequence is of length at least one. Hence there can be no infinite reduction sequences containing an infinite number of trunk rewrites. By lemma 18, all other infinite reduction sequences of $ft_1 \ldots t_n$ induce infinite reduction sequences of one of the terms t_i which is prohibited by assumption.

We now use lemma 20 to devise conditions under which constants applied to terms are reducible.

Lemma 21. *If t_i is a strongly normalising term of sort s_i for $i = 1, \cdots, m$, and f has type $s_1 \to \ldots \to s_n$ where $m < n$, then $ft_1 \ldots t_m$ is reducible.*

Proof. The proof is by induction on the type of the term $ft_1 \ldots t_m$. If this type is a sort, then we must show that $ft_1 \ldots t_m$ is strongly normalising under the assumption that each of the t_i are strongly normalising. But this is precisely lemma 20. If however the type of $ft_1 \ldots t_m$ is of the form $s \to T$, then we must show that if t is a reducible term of type s, then $ft_1 \ldots t_m t$ is reducible. Since the reducible terms of type s are exactly the strongly normalising ones, this follows from the induction hypothesis.

Lemma 22 Modularity of Strong Normalisation. *If $\mathcal{T}$ is a strongly normalising algebraic TRS, then $\beta\overline{\eta} \cup \mathcal{T}$ is also strongly normalising. Hence $\beta \cup \mathcal{T}$ is also strongly normalising.*

Proof. Having set up the machinery of reducibility candidates as in [], the proof concludes by showing that if t is an arbitrary term, θ is a reduciblity parameter, the free term variables of t are among $x_j : T_j$ and u_j are members of the reducibility candidate $T_j\theta$, then $t[|\theta|][u_j/x_j]$ is a member of the reducibility candidate $T\theta$ (note $|\theta|$ is the type-valued substitution which underlies the reduciblity parameter θ).

The only new case is when t is of the form $ft_1 \dots t_n$ and one must show $(ft_1 \dots t_n)[|\theta|][u_j/x_j]$ is reducible when each of the terms $t_i[|\theta|][u_j/x_j]$ is reducible. But this follows from lemma 21. Strong normalisation of $\beta\overline{\eta}\cup\mathcal{T}$ follows by taking the identity substitution and identity reducibility parameter, while strong normalisation of $\beta \cup \mathcal{T}$ follows as this is a subrelation of $\beta\overline{\eta} \cup \mathcal{T}$.

There is a simple trick to extend strong normalisation of $\beta\overline{\eta} \cup \mathcal{T}$ to $\beta\eta \cup \mathcal{T}$. If t is a term, let $\mathtt{TNF}(t)$ be the *type normal form* of t, ie the term that is obtained by normalising all the types occuring as subterms and in λ-abstractions in t. A reduction $t \xrightarrow{\beta\eta} t'$ is called *type induced* iff the redex contracted occurs inside a subterm of t which is actually a type.

Lemma 23. *If there is a rewrite $t \xrightarrow{\beta\eta\cup\mathcal{T}} t'$, then there is a rewrite $\mathtt{TNF}(t) \xrightarrow{\beta\overline{\eta}\cup\mathcal{T}}{}^{*} \mathtt{TNF}(t')$. In addition, if the original rewrite is not type induced then the final rewrite sequence is not of zero length.*

Proof. The lemma is proved exactly as in []

Corollary 24. *If $\mathcal{T}$ is a strongly normalising algebraic TRS, then $\beta\eta\cup\mathcal{T}$ is also strongly normalising.*

Proof. There are no infinite sequences of type induced reductions because reduction on types is strongly normalising. In addition, if $t \xrightarrow{\beta\eta} t'$ is type induced, then $\mathtt{TNF}(t) = \mathtt{TNF}(t')$. Thus any infinite $\beta\eta \cup \mathcal{T}$ reduction sequence is mapped by type normalisation to an infinite $\beta\overline{\eta} \cup \mathcal{T}$ reduction sequence.

4 Modularity for Algebraic TRS and the Calculus of Constructions

We have proven a series of modularity results concerning the addition of algebraic TRSs to F^ω. The next logical step is to apply the same ideas to a much more powerful and complex system: the Calculus of Constructions [12]. Due to lack of space, we cannot introduce it here in detail, but we recall that the most important feature is that the distinction between types and terms is blurred and types can contain terms embedded within them. This type dependency introduces infinite reduction sequences which are not present in non-dependent type theories. For example, if we define expansions by

$$\frac{\Gamma \vdash t : \Pi x : A.B}{\Gamma \vdash t \Rightarrow \lambda x : A.tx} \tag{1}$$

and define the term $B(x) = (\lambda z : X \to X.X)(x)$, then there is a typing judgement $X : *, x : X \to X \vdash x : \Pi z : B(x).X$ and hence an infinite reduction sequence

$$X : *, x : X \to X \vdash x \Rightarrow \lambda z : B(x).xz \Rightarrow \lambda z : B(\lambda z : B(x).xz).xz \Rightarrow \dots \tag{2}$$

Notice that this example does not uses higher order types and so is present in simpler dependent type theories such as LF. The existence of infinite reduction sequences such as in equation 2 forces us to restrict our attention to a type normalised form of η-expansion which we again denote by $\overline{\eta}$. In F^ω the existence of type normal forms is easy to prove as reduction at the level of types is defined independently to reduction at the level of terms. However in a dependent type theory such as the Calculus of Constructions the existence of long $\beta\eta$-normal forms is much harder to prove. One can either use the standard theory of η-contractions as in [23] or prove their existence while simultaneously developing the theory of expansions as in [].

Let $\beta\overline{\eta}$ be the rewrite relation containing all β-reductions and type normalised expansions, providing that the term being expanded is not an abstraction nor applied to another term. In addition, let $\beta\eta$ be defined as in $\beta\overline{\eta}$ but without the type normal form restriction. The following results concerning $\beta\overline{\eta}$ and $\beta\eta$ are proved in [**?**]

Theorem 25. *The relations $\beta\overline{\eta}$ and $\beta\eta$ are confluent and weakly normalising to the long $\beta\eta$-normal forms.*

We conjecture that $\beta\overline{\eta}$ is strongly normalising but a proof awaits further research.

4.1 Modularity of Confluence

As we have described above, the theory of strong normalization for η-expansions in Coc is not settled. Nevertheless, we can use confluence and weak normalization of $\beta\overline{\eta}$ to good avail and get the modularity of confluence for the union of algebraic TRSs with CoC.

Lemma 26. *Algebraic reduction commutes with β-normalization in CoC.*

Proof. As in [11]. Again, see lemma 31.

Corollary 27 Modularity of confluence for non-extensional Coc. *If $\mathcal{T}$ is a confluent algebraic TRS, then $\beta \cup \mathcal{T}$ is also confluent*

Proof. As in corollary 10 and using lemma 26

Proving that confluence is modular for the union of algebraic TRSs with extensional Calculus of Constructions requires another commutation lemma.

Lemma 28. *Algebraic reduction commutes with $\overline{\eta}$-normalisation.*

Proof. Similar to lemma 11.

Corollary 29 Modularity of Confluence. *If $\mathcal{T}$ is a confluent algebraic TRS, then $\beta\overline{\eta}\cup\mathcal{T}$ and $\beta\eta\cup\mathcal{T}$ are also confluent.*

Proof. $\beta\overline{\eta}\cup\mathcal{T}$ is proven confluent by a similar argument to theorem 13 using the commutation lemmas 26 and 28. The confluence of $\beta\eta \cup \mathcal{T}$ is proved as in corollary 14.

5 Conclusions

We have proved a variety of modularity results for the combination of algebraic TRSs with higher order typed λ-calculi. In generalising the previous results in the literature, our key innovation is the use of η-expansions instead of the more problematic η-contractions.

There are several directions in which we wish to persue this research. Most importantly we want a modularity result for strong normalisation for the addition of algebraic TRSs to the Calculus of Constructions. As we remarked in the paper, this research awaits further basic research into the use of η-expansions in the Calculus of Constructions. In particular we conjecture that $\beta\overline{\eta}$ is strongly normalising and we further conjecture that the combination of a SN algebraic TRS with $\beta\overline{\eta}$ remains SN.

Acknowledgements

References

1. Y. Akama. On Mints' reductions for ccc-Calculus. In *Typed Lambda Calculus and Applications*, number 664 in LNCS, pages 1–12. Springer Verlag, 1993.
2. F. Barbanera. Combining term-rewriting and type-assignment systems. In *Third Italian Conference on Theoretical Computer Science*, Mantova, 1989. World Scientific Publishing Company.
3. F. Barbanera. Combining term rewriting and type assignment systems. *Int. Journal of Found. of Comp. Science*, 1:165–184, 1990.
4. F. Barbanera and M. Fernandez. Intersection type assignment systems with higher-order algebraic rewriting. *Theoretical Computer Science*. To appear.
5. F. Barbanera and M. Fernandez. Modularity of termination and confluence in combinations of rewrite systems with λ_ω. In A.Lingas, R.Karlsson, and S.Carlsson, editors, *Intern. Conf. on Automata, Languages and Programming (ICALP)*, number 700 in Lecture Notes in Computer Science, Lund, 1993.
6. F. Barbanera and M. Fernandez. Modularity of termination and confluence in combinations of rewrite systems with the typed lambda-calculus of order omega. Technical report, Universit Paris Sud, 1994.
7. F. Barbanera, M. Fernandez, and H. Geuvers. Modularity of strong normalization and confluence in the algebraic-λ-cube. In *Proceedings of the Symposium on Logic in Computer Science (LICS)*, Paris, 1994. IEEE Computer Society Press.
8. H. Barendregt. *The Lambda Calculus; Its syntax and Semantics (revised edition)*. North Holland, 1984.
9. V. Breazu-Tannen. Combining algebra and higher order types. In IEEE, editor, *Proceedings of the Symposium on Logic in Computer Science (LICS)*, pages 82–90, July 1988.
10. V. Breazu-Tannen and J. Gallier. Polymorphic rewriting preserves algebraic strong normalization. *Theoretical Computer Science*, 83:3–28, 1991.
11. V. Breazu-Tannen and J. Gallier. Polymorphic rewiting preserves algebraic confluence. *Information and Computation*, 114:1–29, 1994.
12. T. Coquand and G. Huet. Constructions: a higher-order proof system for mechanizing mathematics. *EUROCAL85 in LNCS 203*, 1985.
13. D. Cubric. On free CCC. Distributed on the `types` mailing list, 1992.
14. N. Dershowitz and J.-P. Jouannaud. Rewrite systems. In J. Van Leeuwen, editor, *Handbook of theoretical computer science*, volume Vol. B : Formal Models and Semantics, chapter 6, pages 243–320. The MIT Press, 1990.
15. R. Di Cosmo. On the power of simple diagrams. In *Rewriting Techniques and Applications*, number 1103 in Lecture Notes in Computer Science, pages 200–214, July 1996.
16. R. Di Cosmo and D. Kesner. A confluent reduction for the extensional typed λ-calculus with pairs, sums, recursion and terminal object. In A. Lingas, editor, *Intern. Conf. on Automata, Languages and Programming (ICALP)*, volume 700 of *Lecture Notes in Computer Science*, pages 645–656. Springer-Verlag, July 1993.
17. R. Di Cosmo and D. Kesner. Combining first order algebraic rewriting systems, recursion and extensional lambda calculi. In S. Abiteboul and E. Shamir, editors, *Intern. Conf. on Automata, Languages and Programming (ICALP)*, volume 820 of *Lecture Notes in Computer Science*, pages 462–472. Springer-Verlag, July 1994.
18. R. Di Cosmo and D. Kesner. Simulating expansions without expansions. *Mathematical Structures in Computer Science*, 4:1–48, 1994. A preliminary version is available as Technical Report LIENS-93-11/INRIA 1911.
19. R. Di Cosmo and D. Kesner. Combining algebraic rewriting, extensional lambda calculi and fixpoints. *Theoretical Computer Science*, 1995. To appear.
20. R. Di Cosmo and D. Kesner. Rewriting with polymorphic extensional λ-calculus. In *CSL'95*, volume 1092 of *Lecture Notes in Computer Science*, pages 215–232. Springer-Verlag, 1996. Extended abstract presented in Paderborn, Sept. 1995.
21. R. Di Cosmo and A. Piperno. Expanding extensional polymorphism. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Typed Lambda Calculus and Applications*, volume 902 of *Lecture Notes in Computer Science*, pages 139–153, Apr. 1995.
22. D. J. Dougherty. Some lambda calculi with categorical sums and products. In *Proc. of the Fifth International Conference on Rewriting Techniques and Applications (RTA)*, 1993.
23. G. Dowek, G. Huet, and B. Werner. On the definition of the eta-long normal form in the type systems of the cube. In *Informal Proceedings of the Workshop "Types"*, Nijmegen, 1993.
24. J. Gallier. *On Girard's "Candidats de Reductibilité"*, pages 123–203. Logic and Computer Science. Academic Press, 1990. Odifreddi, editor.
25. N. Ghani. $\beta\eta$-equality for coproducts. In M. Dezani-Ciancaglini and G. Plotkin, editors, *Typed Lambda Calculus and Applications*, volume 902 of *Lecture Notes in Computer Science*, Apr. 1995.
26. N. Ghani. Extensionality and polymorphism. University of Edimburgh, Submitted, 1995.
27. B. Howard and J. Mitchell. Operational and axiomatic semantics of pcf. In *Proceedings of the LISP and Functional Programming Conference*, pages 298–306. ACM, 1990.
28. C. B. Jay and N. Ghani. The Virtues of Eta-expansion. Technical Report ECS-LFCS-92-243, LFCS, 1992. University of Edimburgh, preliminary version of [**?**].
29. G. Mints. Teorija categorii i teoria dokazatelstv.I. *Aktualnye problemy logiki i metodologii nauky*, pages 252–278, 1979.

30. V. van Oostrom. Developing developments. Submitted to Theoretical Computer Science should appear in volume 145, 1994.

A Commutation of algebraic reduction with reduction to β or Coc normal form.

In this section we simply reformulate lemma 4.1 of [11] in the framework of non extensional F^ω and Coc. It is to be noticed that there is really nothing new in the proof, as the clever argument used in that lemma is tight enough to only involve the first order fragment of the caculi, so that extensions to other calculi is straightforward.

In the following, let $A \xrightarrow{r} B$ be an algebraic rewrite rule, with s being the sort of the algebraic term A (and B) and $\overrightarrow{x} = x_1 : s_1, \ldots, x_n : s_n = FV(A) \cup FV(B)$ with the s_i's being the sorts of the variables used in the algebraic rule. Let also z be a chosen variable of type $s_1 \to \ldots \to s_n \to s$. We also suppose a given typing and kinding context that we omit for readability.

We say that a term has the *z-algebraic* property if all occurrences of the variable z in it are fully applied, i.e. at the head of a subterm $zP_1 \ldots P_n$ that possesses the type s with all the P_i's possessing the type s_i. This property is clearly inherited by subterms.

The central property which is needed is the following (where by $\beta - n.f.$ we mean reduction to n.f. only w.r.t. the first order rule β while F^ω (resp. Coc)-n.f. is w.r.t the full non extensional reduction system, which we will also call *full normal form*):

Proposition 30. *If Z is an F^ω (resp. Coc) normal form having the* z-algebraic *property, then*

$$X \equiv \beta - n.f.(Z[\lambda \overrightarrow{x} : \overrightarrow{s}.A/z]) \quad and \quad Y \equiv \beta - n.f.(Z[\lambda \overrightarrow{x} : \overrightarrow{s}.B/z])$$

are F^ω (resp. Coc) normal forms and moreover $X \xrightarrow{r}\!\!\!\!\rightarrow Y$.

Proof. This is by induction on the size of Z. Since Z is a normal form, it must be of the shape $\lambda v_1 \ldots v_k.hT_1 \ldots T_m$ with v_i being either a term variable $x_i : S_i$ with S_i a normal form, or a type variable $t_i : K$.
We have now two cases:

$h \not\equiv z$ then $X \equiv \beta - n.f.(Z[\lambda \overrightarrow{x} : \overrightarrow{s}.A/z]) = \lambda \overrightarrow{v}.hT_1^A \ldots T_m^A$ and $Y \equiv \beta - n.f.(Z[\lambda \overrightarrow{x} : \overrightarrow{s}.B/z]) = \lambda \overrightarrow{v}.hT_1^B \ldots T_m^B$ with $T_i^A = \beta{-}n.f.(T_i[\lambda \overrightarrow{x} : \overrightarrow{s}.A/z])$ and $T_i^B = \beta{-}n.f.(T_i[\lambda \overrightarrow{x} : \overrightarrow{s}.B/z])$. But T_i is still a full normal form, of size strictly smaller than Z (as at least h is removed), and it still possesses the *z-algebraic* property as it is a subterm of Z. So, by induction hypothesis, T_i^A is a full normal form and $T_i^A \xrightarrow{r}\!\!\!\!\rightarrow T_i^B$, hence X is a full normal form and $X \xrightarrow{r}\!\!\!\!\rightarrow Y$.

$h \equiv z$ In this case, $k = m$ and we have that

$$Z[\lambda \overrightarrow{x} : \overrightarrow{s}.A/z]) = \lambda \overrightarrow{v}.(\lambda \overrightarrow{x} : \overrightarrow{s}.A)T_1 \ldots T_m \xrightarrow{\beta}\!\!\!\!\rightarrow \lambda \overrightarrow{v}.A[T_1/x_1 \ldots T_m/x_n]$$

and

$$Z[\lambda \overrightarrow{x} : \overrightarrow{s}.B/z]) = \lambda \overrightarrow{v}.(\lambda \overrightarrow{x} : \overrightarrow{s}.B)T_1 \ldots T_m \xrightarrow{\beta}\!\!\!\!\rightarrow \lambda \overrightarrow{v}.B[T_1/x_1 \ldots T_m/x_n].$$

Then, since no β-reduction can take place at the junction points of the T_i with A, as they have as type a base sort, $X \equiv \beta - n.f.(Z[\lambda \overrightarrow{x} : \overrightarrow{s}.A/z]) = \lambda \overrightarrow{v}.A[T_1^A/x_1 \ldots T_m^A/x_n]$ and $Y \equiv \beta - n.f.(Z[\lambda \overrightarrow{x} : \overrightarrow{s}.B/z]) = \lambda \overrightarrow{v}.B[T_1^B/x_1 \ldots T_m^B/x_n]$. As above, the T_i^A (resp. T_i^B) are smaller normal forms than X (resp. Y), so by induction hypothesis we have that the T_i^A and T_i^B are full normal forms and that $T_i^A \xrightarrow{r}\!\!\!\!\rightarrow T_i^B$. Then, both X and Y are full normal forms and moreover $X \equiv \lambda \overrightarrow{v}.A[T_1^A/x_1 \ldots T_m^A/x_n] \xrightarrow{r}\!\!\!\!\rightarrow \lambda \overrightarrow{v}.A[T_1^B/x_1 \ldots T_m^B/x_n] \xrightarrow{r} \lambda \overrightarrow{v}.B[T_1^B/x_1 \ldots T_m^B/x_n]$.
We are done.

Using this crucial result it is then quite easy to show the equivalent of Lemma 4.1 of [11]:

Lemma 31 Algebraic reduction vs. full normal forms. *Let $A \xrightarrow{r} B$ be an algebraic rewrite rule. If $M \xrightarrow{r} N$, then $fnf(M) \xrightarrow{r}\!\!\!\!\rightarrow fnf(N)$, where $fnf(M)$ is the full non-extensional normal form w.r.t. F^ω or Coc.*

Proof. If $M \xrightarrow{r} N$, then $M \equiv C[A\phi]$ and $N \equiv C[B\phi]$ with ϕ a substitution $[P_1/x_1, \ldots, P_n/x_n]$. Then, for a suitable variable z of type $s_1 \rightarrow \ldots \rightarrow s_n \rightarrow s$, we can write terms

$$M' \equiv C[zP_1 \ldots P_n][\lambda \overrightarrow{x} : \overrightarrow{s}.A/z] \quad \textit{and} \quad N' \equiv C[zP_1 \ldots P_n][\lambda \overrightarrow{x} : \overrightarrow{s}.B/z]$$

s.t. $M' \xrightarrow{\beta}\!\!\!\!\rightarrow M$ and $N' \xrightarrow{\beta}\!\!\!\!\rightarrow N$. Now, $C[zP_1 \ldots P_n]$ has the *z-algebraic* property, and since this property is preserved by the non-extensional F^ω and Coc reductions, also $fnf(C[zP_1 \ldots P_n])$ has it. Now, we can apply the previous theorem to such a full normal form and obtain that $M'' = \beta - n.f.(fnf(C[zP_1 \ldots P_n])[\lambda \overrightarrow{x} : \overrightarrow{s}.A/z])$ and $N'' = \beta - n.f.(fnf(C[zP_1 \ldots P_n])[\lambda \overrightarrow{x} : \overrightarrow{s}.B/z])$ are full normal forms and that $M'' \xrightarrow{r}\!\!\!\!\rightarrow N''$. Since $M' \longrightarrow\!\!\!\!\rightarrow M''$ (resp. $N' \longrightarrow\!\!\!\!\rightarrow N''$) and $M' \xrightarrow{\beta}\!\!\!\!\rightarrow M$ (resp. $N' \xrightarrow{\beta}\!\!\!\!\rightarrow N$), we have, due to confluence of F^ω and Coc, that $M'' = fnf(M)$ and $N'' = fnf(N)$, and we are done.