Eta-Expansions III — F^ω

Neil Ghani
LIENS-DMI, Ecole Normale Superieure
45, Rue D'Ulm, 75230 Paris Cedex 05, France
e-mail:ghani@ens.fr

April 25, 1996

Abstract

The use of expansionary η-rewrite rules in various typed λ-calculi has become increasingly common in recent years as their advantages over contractive η-rewrite rules have become apparent. Not only does one obtain simultaneously a decision procedure for $\beta\eta$-equality and a procedure for the calculation of the long $\beta\eta$-normal form of a term, but rewrite relations using expansions retain key properties when combined with first order rewrite systems, generalise more easily to other type constructors and are supported by a categorical theory of reduction.

However, until now η-contractions have remained the only possibility in the more powerful type systems of the λ-cube. In this paper we begin to rectify this situation by extending the techniques previously developed to a higher order polymorphic λ-calculus called F^ω, where reduction no longer occurs only at the level of terms but also at the level of types.

1 Introduction

Extensional equality for terms of the simply typed λ-calculus requires η-conversion, whose interpretation as a rewrite rule has traditionally been as a contraction $\lambda x:T.fx \Rightarrow f$ with the side condition $x \not\in \mathtt{FV}(f)$. When combined with the usual β-reduction, the resulting rewrite relation is strongly normalising and confluent, and thus reduction to normal form provides a decision procedure for the associated equational theory.

However η-contractions behave badly when combined with other rewrite rules and the key property of confluence is often lost. For example, if the calculus is extended by a unit type 1 with associated rewrite rule $t \Rightarrow *$ (providing t has type 1), then the divergence $\lambda x:1.* \Leftarrow \lambda x:1.fx \Rightarrow f$ cannot be completed. Another area where η-contractions cannot be used is in the combination of type theories and algebraic rewrite systems. Properties of such rewrite systems such as confluence and strong normalisation, preserved when combined with β-reduction [2, 3], are typically lost in the presence of a contractive η-rewrite rule. For example, if we regard 1 as a base type with constants $f : 1 \rightarrow 1$ and $* : 1$ and with rewrite rule $fx \Rightarrow *$, then $\Rightarrow$ is confluent while the divergence above shows that the combination of $\Rightarrow$ with the contractive η-rewrite rule is not confluent.

Recently several authors [5, 6, 13] have accepted the old proposal [12, 14, 15] that η-conversion be interpreted as an expansion $f \Rightarrow \lambda x : T.fx$ (where T is the type of x) and the resulting rewrite relation has been shown confluent. In these works infinite reduction sequences such as

$$f \Rightarrow \lambda x:T.fx \Rightarrow \lambda x:T.(\lambda y:T.fy)x \Rightarrow \ldots$$

are avoided by imposing syntactic restrictions to limit the possibilities for expansion; namely λ-abstractions cannot be expanded, nor can terms which are applied. This restricted expansion relation is strongly normalising, confluent and generates the same equational theory as the unrestricted expansionary rewrite relation. Thus $\beta\eta$-equality can be decided by reduction to normal form in this restricted fragment and, in addition, the normal forms of this restricted rewrite relation are exactly Huet's *long $\beta\eta$-normal forms* [12, 15]. Most pleasingly of all, these properties tend to be maintained if one adds other type constructors [10], or algebraic rewrite rules [4].

In addition to these practical arguments, the category-theoretic analysis of reduction [10] provides another argument in favour of interpreting η as an expansion. In this analysis the introduction and elimination rules of a type constructor form a pair of locally adjoint functors whose local unit and counit are respectively an expansionary (not *contractive*) η-rewrite rule and contractive β-rewrite rule. The associated local triangle laws assert the existence of looping reductions — for the exponential the triangle laws are

$$\begin{array}{lllll} \lambda x.t & \Rightarrow & \lambda y.(\lambda x.t)y & \Rightarrow & \lambda y.t[y/x] \equiv \lambda x.t \\ tu & \Rightarrow & (\lambda x.tx)u & \Rightarrow & tu \end{array} \tag{1}$$

Thus even the restrictions on η-expansion required to obtain strong normalisation have a categorical formulation, preventing exactly those expansions occurring in the triangle laws 1.

This categorical approach to rewriting has been extended to the more difficult problem of providing a decision procedure for $\beta\eta$-equality for type constructors such as the coproduct and the tensor of linear logic [9, 10]. The resulting η-rewrite rules are substantially more complex than those for the product and exponential — not only is there a facility for expanding terms of sum type analogous to that for the product and exponential, but also the ability to permute the order in which different subterms of sum type are eliminated. This leads to some difficult term rewriting problems — the interested reader may consult the above references.

This paper extends these initial results in a different direction by generalising the techniques previously developed to a higher-order polymorphic λ-calculus called F^ω. The major difference between this calculi and those previously studied is that β and η-redexes may occur in not just the terms of the calculus, but also in the type constructors. Thus the type of a term, which partially determines whether an η-expansion may be applied to it, is no longer unique and so we are required to analyse the equality on types before that on terms. Thus one property which F^ω shares with simply typed λ-calculus, namely the separation between type constructors and terms of the calculus, will be crucial in our work.

There have been several attempts to give a satisfactory account of the more powerfull systems in the λ-cube with $\beta\eta$-equality. One of the major problems of doing so is that if one attempts to define $\beta\eta$-equality on pre-terms with the equation $t = \lambda x : A.tx$ $(x \not\in \mathtt{FV}(T))$, then the associated equational theory is too coarse, equating semantically distinct entities. For instance one has $\lambda x : A.x =_\beta \lambda x : A.(\lambda y : B.y)x =_\eta \lambda y : B.y$ where A and B are arbitrary types. However, as η-expansions are best formulated in a typed setting, we are naturally led to a calculus based on typing and equality judgements. Although this requires a small amount of extra notation, we avoid the inconsistencies inherent in the untyped approach, aswell as develop the notation for more powerful systems where there is no alternative to the use of these judgement forms. In [11], typed and untyped equality are shown to coincide on well-type terms *of the same type*,

$$\Gamma \vdash t_1 =_{\beta\eta} t_2 : T \text{ iff } \Gamma \vdash t_i : T \wedge t_1 =_{\beta\eta} t_2$$

Since β with η-contraction is strongly normalising and confluent on well typed terms, typed equality is decidable. In [7] η-contractions are used to prove the existence of $\beta\eta$-long normal forms for all systems of the λ-cube.

In summary, there are many reasons for using η-expansions in type theory, e.g. the ease with which they generalise to other type constructors and their good modularity properties. This paper shows that η-expansions are also robust enough to be applied to more expressive type theories where until now the only possibility was to use contractive η-rewrite rules. This paper is thus an important first step towards the smooth combination of expressive type theories with $\beta\eta$-equality and algebraic rewrite systems.

2 The Calculus F^ω

Although the reader will be assumed to be familiar with F^ω, we shall shall give a brief presentation of the calculus for completeness. Although less concise than the PTS presentation, our formulation makes explicit properties of the calculus, not shared by PTS' in general, which our proofs will rely on. Firstly let $*$ and $\Box$ be special symbols. The *kinds* of F^ω, denoted $\Lambda(\Box)$, are defined by the syntax $K := * \mid K \to K$. Now let $\mathcal{V}(\Box)$ be a set of *type constructor* variables with associated kinds given by a function $\phi : \mathcal{V}(\Box) \to \Lambda(\Box)$ such that each of the inverse images $\mathtt{Var}(K) = \phi^{-1}(K)$ is infinite. The *kinding judgements* of F^ω are of the form $T : K$ and are derived by the inference rules in Table 1. Given a kinding judgement $T : K$, we call

Table 1: Kinding Judgements for F^ω

$$\frac{X \in \mathcal{V}(\Box)}{X : \phi(X)} \qquad \frac{\phi(X) = K \quad T : *}{\Pi X : K.T : *} \qquad \frac{T : * \quad U : *}{T \to U : *}$$

$$\frac{F : K \to K' \quad U : K}{FU : K'} \qquad \frac{\phi(X) = K \quad T : K'}{\lambda X : K.T : K \to K'}$$

T a *type constructor* and also define $\Lambda(K) = \{T \mid \text{there is a kinding judgement } T : K\}$ and $\Lambda_1 = \bigcup_{K \in \Lambda(\Box)} \Lambda(K)$. Type constructors differing only in bound variables of the same kind are regarded as being equivalent. The free variables of a type constructor T, denoted $\mathtt{FV}(T)$, and the result of a substitution θ for free variables of T, denoted $T\theta$, are defined as usual [1].

The inference rules for kinding judgements form a simply typed λ-calculus, whose types are the kinds of F^ω, and which is enriched with constants $\to : * \to * \to *$ (we apologise for the overloading of symbols — if we chose $\Rightarrow$ for exponential types this would clash with the rewrite relations defined later) and $\Pi_K : (K \to *) \to *$. This observation allows us to deduce many results concerning kinding judgements from the literature on the simply typed λ-calculus, and hence we omit proofs for some of the following lemmas.

Lemma 2.1 *Given kinding judgements $T : K$ and $T : K'$, then $K = K'$. Also if there is a kinding judgement $U : \phi(X)$, then there is a kinding judgement $T[U/X] : K$.*

An equational theory, called $\beta\eta$-equality, is defined on type constructors as the least congruence containing the equations the equations $(\lambda X : K.T)U =_{\beta\eta} T[U/X]$ and $T =_{\beta\eta} \lambda X : K.TX$, where the second equation has the side conditions that $T : K \to K'$ and that $X \notin \mathtt{FV}(T)$.

Lemma 2.2 *If $T : K$, $T' : K'$ and $T =_{\beta\eta} T'$, then $K = K'$*

Although confluent, unrestricted use of η-expansion permits infinite reduction sequences, e.g. see equation 1. However, a strongly normalising relation can be obtained by prohibiting the expansions appearing in equation 1. Thus a relation $\Rightarrow_{\mathcal{F}}$ is defined on type constructors which permits arbitrary β-reduction and expansions $T \Rightarrow_{\mathcal{F}} \lambda X : K.TX$ providing $T : K \to K'$, $X \notin \mathtt{FV}(T)$, T is not a λ-abstraction and T is not applied to some other term. This latter context sensitive restriction on η-expansion is enforced by simultaneously defining a further subrelation $\Rightarrow_{\mathcal{I}}$ of $\Rightarrow_{\mathcal{F}}$ which is guaranteed not to include top-level expansions, i.e. rewrites of the form $T \Rightarrow \lambda X : K.TX$. Thus a subterm which is applied to another subterm may be safely $\Rightarrow_{\mathcal{I}}$-rewritten without the risk of violating the restrictions above. A type constructor is *neutral* iff it is not a λ-abstraction. The relations $\Rightarrow_{\mathcal{F}}$ and $\Rightarrow_{\mathcal{I}}$ are formally defined Table 2.

Lemma 2.3 *The reflexive, symmetric, transitive closure of $\Rightarrow_{\mathcal{F}}$ is $\beta\eta$-equality. In addition $\Rightarrow_{\mathcal{F}}$ is strongly normalising and confluent.*

Proof The proof can be found in any of [5, 6, 13]. $\Box$

Table 2: The Restricted Rewrite Relation

$$\frac{T \text{ neutral} \quad T:K{\rightarrow}K'}{T \Rightarrow_{\mathcal{F}} \lambda X:K.TX} \qquad \frac{T \Rightarrow_{\beta} T'}{T \Rightarrow_{\mathcal{I}} T'} \qquad \frac{T \Rightarrow_{\mathcal{I}} T'}{T \Rightarrow_{\mathcal{F}} T'}$$

$$\frac{T \Rightarrow_{\mathcal{F}} T'}{\Pi X:K.T \Rightarrow_{\mathcal{I}} \Pi X:K.T'} \qquad \frac{A \Rightarrow_{\mathcal{F}} A'}{A{\rightarrow}B \Rightarrow_{\mathcal{I}} A'{\rightarrow}B} \qquad \frac{B \Rightarrow_{\mathcal{F}} B'}{A{\rightarrow}B \Rightarrow_{\mathcal{I}} A{\rightarrow}B'}$$

$$\frac{T \Rightarrow_{\mathcal{F}} T'}{\lambda X:K.T \Rightarrow_{\mathcal{I}} \lambda X:K.T'} \qquad \frac{U \Rightarrow_{\mathcal{I}} U'}{UV \Rightarrow_{\mathcal{I}} U'V} \qquad \frac{V \Rightarrow_{\mathcal{F}} V'}{UV \Rightarrow_{\mathcal{I}} UV'}$$

If T is a type constructor, its unique $\Rightarrow_{\mathcal{F}}$-normal form will be denoted $\mathtt{NF}(T)$. One important corollary, known traditionally as *uniqueness of products* is the following:

Lemma 2.4 *If $A{\rightarrow}B =_{\beta\eta} A'{\rightarrow}B'$, then $A =_{\beta\eta} A'$ and $B =_{\beta\eta} B'$. Similarly if $\Pi X:K.A =_{\beta\eta} \Pi X:K'.A'$ then $K = K'$ and $A =_{\beta\eta} A'$.*

Proof By confluence $A{\rightarrow}B$ and $A'{\rightarrow}B'$ have a common $\Rightarrow_{\mathcal{F}}$-reduct. But all $\Rightarrow_{\mathcal{F}}$-reducts of $A{\rightarrow}B$ are induced by $\Rightarrow_{\mathcal{F}}$-reductions of A or B, and hence the common reduct is constructed by forming a common reduct for A and A'. Hence $A =_{\beta\eta} A'$. The rest of the proof is similar. □

3 Term Judgements For F^ω

Let $\mathcal{V}(*)$ be a set of term variables, disjoint from $\mathcal{V}(\Box)$. A *typing context* is a list of pairs $x_1 : T_1, \ldots, x_n : T_n$, where each x_i is a term variable, $T_i \in \Lambda(*)$ and such that the variables are distinct. A typing context $\Gamma = x_1 : T_1, \ldots, x_n : T_n$ is regarded as a function Γ with domain $\mathtt{dom}(\Gamma) = \{x_1, \ldots, x_n\}$ whose action is $\Gamma(x_i) = T_i$. We write $\Gamma \subseteq \Delta$ if $\mathtt{dom}(\Gamma) \subseteq \mathtt{dom}(\Delta)$ and for a variable x declared in both Γ and Δ, $\Gamma(x) = \Delta(x)$. We also write $\Gamma \sim \Delta$ iff $\mathtt{dom}(\Gamma) = \mathtt{dom}(\Delta)$ and for a variable x declared in both contexts, $\Gamma(x) =_{\beta\eta} \Delta(x)$. Finally, the free variables in a context are $\mathtt{FV}(x_1 : T_1, \ldots, x_n : T_n) = \bigcup \mathtt{FV}(T_i)$ and the substitution of type constructors is extended to contexts as expected. Although typing contexts contain only term variable declarations, kinding declarations for all type constructor variables are implicitly assumed.

The *typing judgements* of F^ω are of the form $\Gamma \vdash t : T$, where Γ is a typing context and T is a type constructor, and are defined by the inference rules in Table 3. If there is a typing

Table 3: Typing Judgements for F^ω

$$\frac{x:T \in \Gamma}{\Gamma \vdash x:T} \qquad \frac{\Gamma \vdash t:T \quad T =_{\beta\eta} T' : K}{\Gamma \vdash t:T'} \qquad \frac{\Gamma \vdash t:T{\rightarrow}T' \quad \Gamma \vdash u:T}{\Gamma \vdash tu:T'}$$

$$\frac{\Gamma, x:T \vdash u:U}{\Gamma \vdash \lambda x:T.u : T{\rightarrow}U} \qquad \frac{\Gamma \vdash t:\Pi X:K.T \quad U:K}{\Gamma \vdash tU:T[U/X]} \qquad \frac{\Gamma \vdash t:T \quad X \notin \mathtt{FV}(\Gamma)}{\Gamma \vdash \lambda X:\phi(X).t : \Pi X:\phi(X).T}$$

judgement $\Gamma \vdash t:T$ we say t is a term of type T under context Γ. Terms which are α-equivalent are treated as equal, the free term variables of a term t are denoted by $\mathtt{FV}(t)$ and substitution of either term variables or type variables is denoted by the same notation as before. We define $\Lambda(T) = \{(\Delta, M) \mid \Delta \vdash M:T\}$ and remark that if $T =_{\beta\eta} T' : K$, then $\Lambda(T) = \Lambda(T')$.

Lemma 3.1 *If there is a typing judgement $\Gamma \vdash t:T$, then there is a kinding judgement $T:*$. If $\Gamma \sim \Gamma'$ or $\Gamma \subseteq \Gamma'$, then there is a typing judgement $\Gamma' \vdash t:T$. Also is $\Delta \subseteq \Gamma$ and $\mathtt{FV}(t) \subseteq \mathtt{dom}(\Delta)$, then there is a typing judgement $\Delta \vdash t:T$.*

Lemma 3.2 *Given judgements $\Gamma, x:A \vdash t:B$ and $\Gamma \vdash u:A$, there is also a typing judgement $\Gamma \vdash t[u/x] : B$. Also if $U:\phi(X)$ and there is a typing judgement $\Gamma \vdash t:T$, then there is a typing judgement $\Gamma[U/X] \vdash t[U/X] : T[U/X]$*

The next lemma we require is a version of the *generation* lemma traditionally proved for PTS' which states that, upto type conversion, terms have unique typing derivations and that all subterms of a term are also terms.

Lemma 3.3 *Let $\Gamma \vdash t:T$ be a typing judgement. Then*

- *If t is a variable x, then there is a declaration $x:T' \in \Gamma$ and $T =_{\beta\eta} T' : *$.*
- *If t is $\lambda X:K.u$, then there is a typing judgement $\Gamma \vdash u:A$ and $T =_{\beta\eta} \Pi X : K.A$.*
- *If t is $\lambda x:A.u$, then there is a typing judgement $\Gamma, x:A \vdash u:B$ and $T =_{\beta\eta} A \rightarrow B$.*
- *If t is rU, where U is a type constructor, then $\Gamma \vdash r : \Pi X:K.A$ and $T =_{\beta\eta} A[U/X]$.*
- *If t is ru, where u is a term, then $\Gamma \vdash r : A \rightarrow B$ and $\Gamma \vdash u:A$ such that $T =_{\beta\eta} B$.*

Lemma 3.4 *Let $\Gamma \vdash t:T_1$ and $\Gamma \vdash t:T_2$ be typing judgements. Then $T_1 =_{\beta\eta} T_2$.*
Proof The proof is by induction on the two typing judgements. If either ends in an instance of type conversion the result follows by induction. If not, consider for example, the case where t is of the form uv. By lemma 3.3 there must be typing judgements $\Gamma \vdash u : A_1 \rightarrow B_1$ and $\Gamma \vdash u : A_2 \rightarrow B_2$. By induction $A_1 \rightarrow B_1 =_{\beta\eta} A_2 \rightarrow B_2$, and by the uniqueness of exponentials proved in lemma 2.4, we deduce $T_1 =_{\beta\eta} B_1 =_{\beta\eta} B_2 =_{\beta\eta} T_2$. All other cases are similar. □

Lemma 3.5 *Given a term t, a context Γ and a type T it is decidable whether there is a typing judgement $\Gamma \vdash t : T$*
Proof The proof is by induction on the structure and uses lemma 3.3 and the decidability of $\beta\eta$-equality on type constructors as proved in lemma 2.3 □

4 Rewriting in F^ω

Since we wish to expand a term if it is of function type, and this inhabitation condition can only be verified in the presence of a context, the rewrite relation on terms must be parameterised by a context, i.e. given as a calculus for deriving judgements $\Gamma \vdash t \Rightarrow_{\mathcal{F}} t'$. As with the rewrite relation on type constructors, the expansion of λ-abstractions or terms which applied to other terms is forbidden by the simultaneous definition of another judgement form $\Gamma \vdash t \Rightarrow_{\mathcal{I}} t'$. The presence of type reductions means that further choices must be made — we have imposed the additional condition that if a term is to be expanded, its type, which must necessarily be $\beta\eta$-equal to an exponential or impredicative type, must be given in normal form. This restriction is needed to ensure that lemma 4.4 holds and at the end of this paper strong normalisation and confluence is also proved for the rewrite relation obtained by dropping this restriction.

The rewriting judgements are defined in Table 4, where $\eta^\sigma(t) = \lambda x:\sigma.(tx)$ for σ either a type or a kind and $x \notin \mathtt{FV}(t)$. The possibility of expanding a term depends on its type, but by lemma 3.5 this is a decidable condition, and hence it is also decidable whether there is a reduction $\Gamma \vdash t \Rightarrow_{\mathcal{R}} t'$. We write $\Gamma \vdash t \Rightarrow_{\mathcal{R}} t' : T$ if there is a rewrite $\Gamma \vdash t \Rightarrow_{\mathcal{R}} t'$ and there are typing judgements $\Gamma \vdash t:T$ and $\Gamma \vdash t':T$. In addition, if T is a type constructor define

$$\mathtt{SN}(T) = \{(\Gamma, M) \mid \Gamma \vdash M:T \text{ and } \not\exists M_1, M_2, ...\Gamma \vdash M \Rightarrow_{\mathcal{F}} M_1 \Rightarrow_{\mathcal{F}} M_2 ...\}$$

and, if $(\Gamma, M) \in \mathtt{SN}(T)$ and Γ may be inferred, we say that M is $\Rightarrow_{\mathcal{F}}$-strongly normalising and speak of the $\Rightarrow_{\mathcal{F}}$-normalisation rank of M. Similarly we speak of the $\Rightarrow_{\mathcal{F}}$-reducts of a term M.

Lemma 4.1 *Let $\Gamma \vdash t \Rightarrow_{\mathcal{R}} t'$. If $\Gamma \subseteq \Gamma'$ or $\Gamma \sim \Gamma'$, then there is a rewrite $\Gamma' \vdash t \Rightarrow_{\mathcal{R}} t'$. Also, if $\Delta \subseteq \Gamma$ and $\mathtt{FV}(t) \subseteq \mathtt{dom}(\Delta)$, then $\Delta \vdash t \Rightarrow_{\mathcal{R}} t'$*

Table 4: The Restricted Rewrite Relation

$$\frac{t \text{ neutral} \quad \Gamma \vdash t : T \quad \mathtt{NF}(T) = A \to B}{\Gamma \vdash t \Rightarrow_{\mathcal{F}} \eta^{A}(t)} \qquad \frac{}{\Gamma \vdash (\lambda x : A.t)u \Rightarrow_{\mathcal{I}} t[u/x] : T}$$

$$\frac{t \text{ neutral} \quad \Gamma \vdash t : T \quad \mathtt{NF}(T) = \Pi X : K.B}{\Gamma \vdash t \Rightarrow_{\mathcal{F}} \eta^{K}(t)} \qquad \frac{\Gamma \vdash t \Rightarrow_{\mathcal{I}} t'}{\Gamma \vdash t \Rightarrow_{\mathcal{F}} t'}$$

$$\frac{\Gamma \vdash t \Rightarrow_{\mathcal{I}} t'}{\Gamma \vdash tu \Rightarrow_{\mathcal{I}} t'u} \qquad \frac{\Gamma \vdash u \Rightarrow_{\mathcal{F}} u'}{\Gamma \vdash tu \Rightarrow_{\mathcal{I}} tu'}$$

$$\frac{\Gamma, x : T \vdash t \Rightarrow_{\mathcal{F}} t'}{\Gamma \vdash \lambda x : T.t \Rightarrow_{\mathcal{I}} \lambda x : T.t'} \qquad \frac{T \Rightarrow_{\mathcal{F}} T'}{\Gamma \vdash \lambda x : T.t \Rightarrow_{\mathcal{I}} \lambda x : T'.t}$$

$$\frac{\Gamma \vdash t \Rightarrow_{\mathcal{I}} t'}{\Gamma \vdash tU \Rightarrow_{\mathcal{I}} t'U} \qquad \frac{U \Rightarrow_{\mathcal{F}} U' : K}{\Gamma \vdash tU \Rightarrow_{\mathcal{I}} tU'}$$

$$\frac{\Gamma \vdash t \Rightarrow_{\mathcal{F}} t'}{\Gamma \vdash \lambda X : K.t \Rightarrow_{\mathcal{I}} \lambda X : K.t'}$$

Lemma 4.2 *If $\Gamma \vdash t \Rightarrow_{\mathcal{F}} t'$ and $\Gamma \vdash t : U$, then there is also a typing judgement $\Gamma \vdash t' : U$*
Proof The lemma is proved by induction on the structure of the rewrite. □

Lemma 4.3 *There are rewrites $\Gamma \vdash \eta^{A}(\lambda x : A'.t) \Rightarrow_{\mathcal{I}} \lambda x : A.t$ and $\Gamma \vdash \eta^{A}(t)u \Rightarrow_{\mathcal{I}} tu$*

For reasons of space we are unable to give a full definition of $\beta\eta$-equality for the terms of F^{ω} — the reader may consult [11]. However all β-reductions are contained in $\Rightarrow_{\mathcal{F}}$ and, as lemma 4.3 shows, if a term rewrites to another by a prohibited η-expansion, then the terms have a common $\Rightarrow_{\mathcal{F}}^{*}$ reduct. Thus the reflexive, symmetric, transitive closure of $\Rightarrow_{\mathcal{F}}$ is the same as traditional $\beta\eta$-equality on the terms of F^{ω}.

When defining the rewrite relation $\Rightarrow_{\mathcal{F}}$ all triangular expansions where prohibited. However there are "fake" triangular expansion sequences such as

$$(\lambda x : X \to X.x)(\lambda y : X.y) \Rightarrow_{\mathcal{F}} (\lambda x : X \to X.\eta^{X}(x))(\lambda y : X.y) \Rightarrow_{\beta} \eta^{X}(\lambda y : X.y)$$

which seem to η-expand the λ-abstraction. Of course, these fake expansions cannot cause real harm as they all consume β-reductions. To formalise this, call a rewrite relation *substitutive* if whenever $\Gamma, x : A \vdash t \Rightarrow t'$ and $\Gamma \vdash u \Rightarrow u'$ then $\Gamma \vdash t[u/x] \Rightarrow^{*} t'[u'/x]$. The context sensitive restrictions on expansion mean that the rewrite relation $\Rightarrow_{\mathcal{F}}$ is not a congruence and hence not substitutive. The next lemma characterises when substitutivity fails and, in these instances, exhibits alternative reduction sequences.

Lemma 4.4 *Let $\Gamma, x : A \vdash t \Rightarrow_{\mathcal{R}} t'$, $\Gamma \vdash u \Rightarrow_{\mathcal{R}} u'$ and $T \Rightarrow_{\mathcal{R}} T'$ where $\mathcal{R} \in \{\mathcal{I}, \mathcal{F}\}$ and $X \notin \mathtt{FV}(\Gamma)$.*

- *There is either a reduction sequence $\Gamma \vdash t[u/x] \Rightarrow_{\mathcal{R}} t'[u/x]$, or u is of the form $\lambda x : A_0.u_0$ and there is a reduction sequence $\Gamma \vdash t[\eta^{A_0}(u)/x] \Rightarrow_{\mathcal{I}}^{*} t'[u/x]$.*
- *Either $t[u/x]$ and $t[u'/x]$ have a common $\Rightarrow_{\mathcal{I}}^{*}$-reduct, or $t = x$ and u' is obtained from u by an η-expansion.*
- *If $X \notin \mathtt{FV}(\Gamma)$, then either there is a reduction $\Gamma \vdash t[U/X] \Rightarrow_{\mathcal{R}} t'[U/X]$ or U is of the form $\lambda X : K.U_0$ and there is a reduction sequence $\Gamma \vdash t[\eta^{K}(U)/X] \Rightarrow_{\mathcal{I}}^{*} t'[U/X]$.*

- *There is a reduction sequence* $\Gamma \vdash t[U/X] \Rightarrow_{\mathcal{R}} t[U'/X]$.

Proof The proofs are all by induction on t. In the last case there are no alternatives because a type constructor cannot be applied inside a term. $\square$

Note that if the restriction that in an expansion $t \Rightarrow_{\mathcal{F}} \eta^A(t)$, the type constructor A is a normal form were to be dropped, then the first part of lemma 4.4 would be false. For example, given the rewrite $x : T \rightarrow U \vdash x \Rightarrow \eta^T(x)$ and a typing judgement $\vdash \lambda y : T'.u : T \rightarrow U$, there is no rewrite $\vdash \lambda y : T'.u \Rightarrow_{\mathcal{F}} \eta^T(\lambda y : T'.u)$. In addition, for there to be a rewrite sequence

$$\vdash \eta^{T'}(\lambda y : T'.u) \Rightarrow_{\mathcal{I}}^{*} \eta^{T}(\lambda y : T'.u)$$

one needs a reduction sequence $T' \Rightarrow_{\mathcal{F}}^{*} T$ which can only be guaranteed by assuming T is a normal form. We comment further upon the role of substitutivity in our proof of strong normalisation after presenting the reducibility candidates in the next section. In the mean time, one would like to hypothesise that both $\Rightarrow_{\mathcal{I}}$ and $\Rightarrow_{\mathcal{F}}$ are locally confluent. Unfortunately this is not the case, e.g. there are $\Rightarrow_{\mathcal{I}}$-reductions $t[u/x] \Leftarrow (\lambda x : T.t)u \Rightarrow_{\mathcal{I}} (\lambda x : T.\eta(t))u$ which cannot be completed without an $\Rightarrow_{\mathcal{F}}$-reduction and so $\Rightarrow_{\mathcal{I}}$ is not locally confluent. However local confluence of $\Rightarrow_{\mathcal{F}}$ can be proved in conjunction with a slight variant for $\Rightarrow_{\mathcal{I}}$.

Lemma 4.5 *The relation $\Rightarrow_{\mathcal{F}}$ is locally confluent and given any divergence $\Gamma \vdash t \Rightarrow_{\mathcal{I}} t_i$ (where $i = 1, 2$), there is a term s such that $\Gamma \vdash t_1 \Rightarrow_{\mathcal{I}}^{*} s$ or $\Gamma \vdash t_1 \Rightarrow_{\mathcal{F}} s$ and similarly for t_2.*
Proof The proof is by simultaneous induction on the term t, with the tricky cases handled by lemma 4.4 and lemma 4.3. $\square$

4.1 Reducibility and Term Normalisation

A reducibility model for the terms of F^ω is constructed and used to derive strong normalisation of all terms. As with other parts of this paper we follow the proof for the simply typed fragment in [13] but are forced to generalise the argument to one based on type judgements and not just terms. The reader may consult [8] where this generalisation is used to prove β-reduction for F^ω strongly normalising. If $P \subseteq \Lambda(T)$ and $(\Gamma, M) \in P$, we write $\Gamma \vdash M \in P$. Given type constructor $T : K$ define the set of reducibility candidates $\mathtt{RC}(T, K)$ by induction on K

- If $K = *$, then $P \in \mathtt{RC}(T, *)$ iff $P \subseteq \Lambda(T)$ and P satisfies the 5 reducibility predicates CR0-4 in Table 5.

- If $K = K_1 \rightarrow K_2$ then a reducibility candidate $f \in \mathtt{RC}(T, K)$ is a $\Lambda(K_1)$-indexed set of functions $f_U : \mathtt{RC}(U, K_1) \rightarrow \mathtt{RC}(TU, K_2)$.

If $\Delta \vdash M \in P$ and Δ is clear we say M is P-reducible, or simply that M is reducible if P is also clear. Since by lemma 2.1 type constructors have unique kinds, we often omit the kinding information and refer simply to $\mathtt{RC}(T)$. The introduction of the new reducibility predicate CR0 is a direct result of the use of typing judgements to build reducibility candidates. Notice also that CR2 and CR4 ensure that reducibility candidates of kind $*$ are closed under $\Rightarrow_{\mathcal{F}}$-reduction. In addition, if $x : T \in \Gamma$ and $P \in \mathtt{RC}(T)$, then since there are no terms t such that $\Gamma \vdash x \Rightarrow_{\mathcal{I}} t$, we may conclude by CR3 that $\Gamma \vdash x \in P$. It has been suggested to me that these candidates are too strong, in particular that there is no need for CR4. However, all alternative proofs I have seen require something like lemma 4.9 to be proven and these proofs forget that reduction is not substitutive and hence fail to properly re-establish the induction hypothesis. The only solution is to make use of lemma 4.4 and this requires that the η-expansion of a reducible term is reducible — hence the need for CR4.

Lemma 4.6 *If $T =_{\beta\eta} U$, then* $\mathtt{RC}(T) = \mathtt{RC}(U)$
Proof By lemma 3.4 both T and U inhabit a unique kind, say K and the lemma is proved by induction on K. If $K = *$, then consider $P \in \mathtt{RC}(T)$. Firstly if $\Delta \vdash M \in P$, then $\Delta \vdash M : T$ and

Table 5: Reducibility Predicates CR0-4

Let $P \subseteq \Lambda(T)$ for some type constructor T

CR0 If $\Gamma \vdash M \in P$, and $\Gamma \subseteq \Gamma'$, then $\Gamma' \vdash M \in P$

CR1 If $\Gamma \vdash M \in P$, then $\Gamma \vdash M \in \mathtt{SN}(T)$

CR2 If $\Gamma \vdash M \in P$ and $\Gamma \vdash M \Rightarrow_{\mathcal{I}} M'$ then $\Gamma \vdash M' \in P$

CR3 If $\Gamma \vdash M : T$ for a neutral term M and

$$\Gamma \vdash M \Rightarrow_{\mathcal{I}} M' \text{ implies } \Gamma \vdash M' \in P$$

then $\Gamma \vdash M \in P$

CR4 If $\Gamma \vdash M \in P$ and $T =_{\beta\eta} \Pi X : A.B$ or $T =_{\beta\eta} A \rightarrow B$, then $\Gamma \vdash \eta^A(M) \in P$

hence by type conversion $\Delta \vdash M : U$. Thus $P \subseteq \Lambda(U)$. The predicates CR0,1,2 and 3 hold for P by assumption, while CR4 holds because equality on type constructors is transitive.

If on the other hand K is of the form $K_1 \rightarrow K_2$, then an element $f \in \mathtt{RC}(T)$ is a $\Lambda(K_1)$-indexed function $f_V : \mathtt{RC}(V) \rightarrow \mathtt{RC}(TV)$. But since $T =_{\beta\eta} U$, we conclude by induction that $\mathtt{RC}(TV) = \mathtt{RC}(UV)$ and hence $f \in \mathtt{RC}(U)$. □

Reducibility candidates are constructed for all type constructors. This is done initially for the *simple* type constructors $\mathcal{S}(K) \subseteq \Lambda(K)$, which are defined

- If $\phi(X) = K$, then $X \in \mathcal{S}(K)$
- If $\alpha \in \mathcal{S}(K_1 \rightarrow K_2)$ and $T \in \Lambda(K_1)$, then $\alpha T \in \mathcal{S}(K_2)$

Lemma 4.7 *Let $T \in \mathcal{S}(*)$. Then there are no type constructors of the form $A \rightarrow B$ or $\Pi X : K.A$ which are $\beta\eta$-equal to T.*
Proof If there were such a type constructor, say $A \rightarrow B$, then T and $A \rightarrow B$ would have a common reduct. But all reducts of T are simple, and all reducts of $A \rightarrow B$ are exponentials which are not simple type constructors. □

Given a simple type constructor $\alpha \in \mathcal{S}(K)$, we define a canonical reducibility candidate $\mathcal{R}^\alpha \in \mathtt{RC}(\alpha)$ by induction on K. If $K = *$, then $R^\alpha = \mathtt{SN}(\alpha)$, while if $K = K_1 \rightarrow K_2$, then $R^\alpha = \lambda T : \Lambda(K_1).\lambda S : \mathtt{RC}(T).R^{\alpha T}$ (we hope the reader is not confused by this λ-calculus notation).

Lemma 4.8 *If $\alpha \in \mathcal{S}(K)$ is an simple type constructor, then $R^\alpha \in \mathtt{RC}(\alpha)$*
Proof The lemma is proved by induction on K. If $K = *$ and $\alpha \in \mathcal{S}(*)$, we must prove that $\mathtt{SN}(\alpha) \in \mathtt{RC}(\alpha)$. Well typedness follows by definition while CR0 holds as by lemma 4.1 any infinite reduction sequence in an extended context induces one in the original. CR1 is a tautology, while CR2 holds as $\Rightarrow_{\mathcal{I}} \subseteq \Rightarrow_{\mathcal{F}}$. By lemma 4.7, any term of type α has only $\Rightarrow_{\mathcal{I}}$-reducts and if these are all strongly normalising, then so must be the term. Thus CR3 holds. By lemma 4.7, CR4 holds vacuously. If however K is of the form $K_1 \rightarrow K_2$, then assume $T : K_1$. By induction we may infer $R^{\alpha T} \in \mathtt{RC}(\alpha T)$ and so the definition of R^α is of the correct form. □

4.1.1 Exponential Types

If $R \in \mathtt{RC}(T, *)$ and $S \in \mathtt{RC}(U, *)$ then define the set

$$R \rightarrow S = \{(\Delta, M) \in \Lambda(T \rightarrow U) \mid \forall \Delta' \supseteq \Delta.\ \text{if } \Delta' \vdash N \in R \text{ then } \Delta' \vdash MN \in S\}$$

We follow [13] in showing that $R \rightarrow S$ is a reducibility candidate, but as usual have to make certain alterations as the model is built from typing judgements.

Lemma 4.9 *Let $R \in \mathtt{RC}(T,*)$ and $S \in \mathtt{RC}(U,*)$ be reducibility candidates and $\Delta \vdash \lambda x:T'.u : T \rightarrow U$ be a typing judgement. If forall judgements $\Delta' \vdash a \in R$, where $\Delta \subseteq \Delta'$, $\Delta' \vdash u[a/x] \in S$, then $\Delta \vdash \lambda x:T'.u \in R \rightarrow S$*

Proof By lemma 3.3, $T' =_{\beta\eta} T$. In addition, T' is $\Rightarrow_{\mathcal{F}}$-strongly normalising and we may assume that $x \not\in \mathtt{FV}(\Delta)$. Thus by assumption $\Delta, x:T \vdash u[x/x] \in S$ and so $\Delta, x:T \vdash u \in \mathtt{SN}(U)$.

We prove that for every $\Delta' \vdash a:T$, where $\Delta \subseteq \Delta'$, $\Delta' \vdash (\lambda x:T'.u)a \in S$ by induction on the sum of the normalisation ranks of T', u and a. By CR3 for S, it suffices to prove reducible those one step $\Rightarrow_{\mathcal{I}}$-reducts of $(\lambda x:T'.u)a$. Those reducts induced by reductions of a and T' are reducible by induction, while given any reduction $\Delta \vdash u \Rightarrow_{\mathcal{F}} u'$ we must re-establish the induction hypothesis. Given any judgement $\Delta' \vdash a:T$ (where $\Delta \subseteq \Delta'$) we must prove that $\Delta' \vdash u'[a/x] \in S$. Now if there is a reduction $\Delta' \vdash u[a/x] \Rightarrow_{\mathcal{F}} u'[a/x]$ then we are done by either CR2 or CR4 for S. In the only other case, by lemma 4.4, a is of the form $\lambda x:A_0.a_0$ and there is a reduction $\Delta' \vdash u[\eta^{A_0}(a)/x] \Rightarrow_{\mathcal{I}}^{*} u'[a/x]$. Now by CR4, $\Delta' \vdash \eta^{A_0}(a) \in R$, and so by assumption $\Delta' \vdash u[\eta^{A_0}(a)/x] \in R$. Thus the induction hypothesis has been re-established and the lemma is proved. $\square$

Note the essential use of substitutivity and CR4 in the proof of lemma 4.9.

Lemma 4.10 *If $R \in \mathtt{RC}(T,*)$ and $S \in \mathtt{RC}(U,*)$, then $R \rightarrow S \in \mathtt{RC}(T \rightarrow U,*)$.*

Proof By definition $R \rightarrow S \subseteq \Lambda(T \rightarrow U,*)$. We shall establish the 5 reducibility predicates.

CR0 Let $\Delta \subseteq \Delta'$ and $\Delta \vdash M \in R \rightarrow S$. If $\Delta' \subseteq \Delta''$ and $\Delta'' \vdash N \in R$, then $\Delta \subseteq \Delta'$ and so $\Delta'' \vdash MN \in S$. Thus $\Delta' \vdash M \in R \rightarrow S$

CR1 If $\Delta \vdash M \in R \rightarrow S$, then for some $x \notin \mathtt{dom}(\Delta)$, $\Delta, x:T \vdash Mx \in S$. thus $\Delta, x:T \vdash Mx \in \mathtt{SN}(U)$ and hence $\Delta \vdash M \in \mathtt{SN}(T \rightarrow U)$.

CR2 Let $\Delta \vdash M \in R \rightarrow S$ and $\Delta \vdash M \Rightarrow_{\mathcal{I}} M'$. If $\Delta \subseteq \Delta'$ and $\Delta' \vdash N \in R$, then by lemma 4.1 there is a rewrite $\Delta' \vdash M \Rightarrow_{\mathcal{I}} M'$ and hence one $\Delta' \vdash MN \Rightarrow_{\mathcal{I}} M'N$. Thus $\Delta' \vdash M'N \in S$ and hence $\Delta \vdash M' \in R \rightarrow S$.

CR3 Assume $\Delta \vdash M:T \rightarrow U$ is neutral and satisfies the premises of CR3. We show by induction on the normalisation rank of N, that if $\Delta \subseteq \Delta'$ and $\Delta' \vdash N \in R$, then $\Delta' \vdash MN \in S$. The one-step $\Rightarrow_{\mathcal{I}}$-reducts of MN are either induced by a rewrite $\Delta' \vdash M \Rightarrow_{\mathcal{I}} M'$ or by one $\Delta' \vdash N \Rightarrow_{\mathcal{F}} N'$. In the former case, by lemma 4.1 there is a rewrite $\Delta \vdash M \Rightarrow_{\mathcal{I}} M'$, and since by assumption $\Delta \vdash M' \in R \rightarrow S$, we may conclude $\Delta' \vdash M'N \in S$. The latter case is dealt with by the induction hypothesis.

CR4 Assume $\Delta \vdash M \in R \rightarrow S$ and that there is a judgement $\Delta \vdash M : A \rightarrow B$. Then $A =_{\beta\eta} T$, and by lemma 4.9, in order to show $\Delta \vdash \eta^{A}(M) \in R \rightarrow S$, we need only show that if $\Delta \subseteq \Delta'$ and $\Delta' \vdash N \in R$, then $\Delta' \vdash Mx[N/x] \in S$. But since $x \not\in \mathtt{FV}(M)$, this follows precisely from $\Delta \vdash M \in R \rightarrow S$.

$\square$

4.2 Reducibility parameters

A *reducibility parameter* is function θ which maps $\mathcal{V}(\Box)$ to a reducibility candidate $\theta(X) \in \mathtt{RC}(T, \phi(X))$. Every reducibility parameter has an underlying substitution which is denoted $|\theta|$. Given a type constructor $T \in \Lambda(K)$ and a reducibility parameter θ, we shall construct in Table 6 a reducibility candidate $T\theta \in \mathtt{RC}(T|\theta|)$.

Lemma 4.11 *Let $\Delta \vdash \lambda X:K.M : \Pi X:K.T$ be a typing judgement and θ a reducibility parameter. If for every $U:K$ and $\alpha \in \mathtt{RC}(U)$, $T\theta[X \leftarrow \alpha]$ is a reducibility candidate (of kind $*$) and $\Delta \vdash M[U/X] \in T\theta[X \leftarrow \alpha]$, then $\Delta \vdash \lambda X:K.M \in (\Pi X:K.T)\theta$.*

Proof By lemma 4.8, there is a reducibility candidate $R^{X} \in \mathtt{RC}(X,K)$ and by assumption $\Delta \vdash t[X/X] \in T\theta[X \leftarrow R^{X}]$. Thus t is $\Rightarrow_{\mathcal{F}}$-strongly normalising. For every $U:K$ and $\alpha : \mathtt{RC}(U)$,

Table 6: Reducibility parameters for Type Constructors

$$X\theta = \theta(X)$$

$$(A\to B)\theta = A\theta\to B\theta$$

$$(\Pi X:K.T)\theta = \{\Delta\vdash M\in\Lambda(\Pi X:K.T)|\theta| \mid \forall U\in\Lambda(K).\forall\alpha\in\mathtt{RC}(U).\Delta\vdash MU\in T\theta[X\leftarrow\alpha]\}$$

$$(\lambda X:K.T)\theta = \lambda U:K.\lambda\alpha:\mathtt{RC}(U).T\theta[X\leftarrow\alpha]$$

$$(FU)\theta = (F\theta)(U|\theta|)(U\theta)$$

we must prove $\Delta\vdash(\lambda X:K.t)U\in T\theta[X\leftarrow\alpha]$. This is done by induction on the sum of the $\Rightarrow_{\mathcal{F}}$-normalisation ranks of t and U in a fashion similar to the proof of lemma 4.9. □

Lemma 4.12 *If T is a type constructor and θ a reducibility parameter, then $T\theta\in\mathtt{RC}(T|\theta|,K)$.*
Proof The proof is by induction on the structure of T. If T is a variable or application the lemma is trivial, while if T is a λ-abstraction the result follows from lemma 4.6. If T is an exponential then use lemma 4.10 and so we are left to consider the case of an impredicative quantification, say $\Pi X:K.T'$ and this is proved in a similar fashion to lemma 4.10. □

Theorem 4.13 *Let $\Delta\vdash t:T$. If θ is a reducibility parameter and for each $x_i\in\mathtt{dom}(\Gamma)$, $u_i\in\Gamma(x_i)\theta$, then $\Delta|\theta|\vdash t|\theta|[u_i/x_i]\in T\theta$. Thus all terms are $\Rightarrow_{\mathcal{F}}$-strongly normalising and hence $\beta\eta$-equality is decidable.*
Proof The proof of the first part of the lemma is by induction over the structure of t and follows the standard procedure. The second part is proved by instantiating the first part with $\theta(X)=\mathcal{R}^X$ and $u_i=x_i$ so as to obtain t is reducible and hence by CR1, $\Rightarrow_{\mathcal{F}}$ is strongly normalising. Since $\Rightarrow_{\mathcal{F}}$ is locally confluent, we may thus deduce that $\Rightarrow_{\mathcal{F}}$ is also confluent and so reduction to normal form provides a decision procedure for $\beta\eta$-eqaulity. □

5 A More Liberal Expansion

In this paper, η-expansions were permitted providing, amongst other conditions, that the type of the term to be expanded is expressed in normal form. We conclude this paper by showing this restriction, introduced only for technical reasons, may be lifted. So let $\Rightarrow_{\mathcal{F}'}$ and $\Rightarrow_{\mathcal{I}'}$ be the rewrite relations defined by using the following alternate forms of η-expansion.

$$\frac{t\ \text{neutral}\quad \Gamma\vdash t:A\to B}{\Gamma\vdash t\Rightarrow_{\mathcal{F}}\eta^A(t)}\qquad \frac{t\ \text{neutral}\quad \Gamma\vdash t:\Pi X:K.T}{\Gamma\vdash t\Rightarrow_{\mathcal{F}}\eta^K(t)}$$

Strong normalisation of $\Rightarrow_{\mathcal{F}'}$ is proved by showing that an infinite $\Rightarrow_{\mathcal{F}'}$-reduction sequence $t_1\Rightarrow_{\mathcal{F}'}t_2\Rightarrow_{\mathcal{F}'}\cdots$ induces an infinite reduction sequence $\mathtt{TNF}(t_1)\Rightarrow^*_{\mathcal{F}}\mathtt{TNF}(t_2)\Rightarrow^*_{\mathcal{F}}\cdots$, where if t is a term, then $\mathtt{TNF}(t)$ is its type normal form. In order to define this notion, if T is a type constructor, its $\Rightarrow_{\mathcal{F}}$-normal form is denoted $\overline{T}$. The *type normal form* of a term is:

$\mathtt{TNF}(x)$	$=$	x	
$\mathtt{TNF}(tU)$	$=$	$\mathtt{TNF}(t)\overline{U}$	if U is a type constructor
$\mathtt{TNF}(tu)$	$=$	$\mathtt{TNF}(t)\mathtt{TNF}(u)$	if u is a term
$\mathtt{TNF}(\lambda x:A.t)$	$=$	$\lambda x:\overline{A}.\mathtt{TNF}(t)$	if A is a type constructor
$\mathtt{TNF}(\lambda x:K.t)$	$=$	$\lambda x:K.\mathtt{TNF}(t)$	if K is a kind

Lemma 5.1 *There is a reduction sequence $\Gamma\vdash t\Rightarrow^*_{\mathcal{I}}\mathtt{TNF}(t)$ and the following are true.*

- *If T and U are type constructors and $U : \phi(X)$, then* $\overline{T}[\overline{U}/X] \Rightarrow^*_{\mathcal{F}} \overline{T[U/X]}$.
- *If* $\Gamma \vdash t : T$ *is a term judgement and* $U : \phi(X)$ *a type constructor, then* $\Gamma[U/X] \vdash \mathtt{TNF}(t)[\overline{U}/X] \Rightarrow^*_{\mathcal{I}} \mathtt{TNF}(t[U/X])$
- *If* $\Gamma, x : A \vdash t : T$ *and* $\Gamma \vdash u : A$ *are term judgements, then* $\mathtt{TNF}(t)[\mathtt{TNF}(u)/x] = \mathtt{TNF}(t[u/x])$

Proof By lemma ??, $\overline{T}[\overline{U}/X]$ and $\overline{T[U/X]}$ are $\beta\eta$-equal type constructors, and as the latter is a normal form, the former must reduce to it. The other parts of the lemma are proved by induction on t. $\square$

Lemma 5.2 *If* $\Gamma \vdash t \Rightarrow_{\mathcal{R}'} t'$, *then* $\Gamma \vdash \mathtt{TNF}(t) \Rightarrow^*_{\mathcal{R}} \mathtt{TNF}(t')$ *for* $\mathcal{R} \in \{\mathcal{I}, \mathcal{F}\}$
Proof The lemma is proved by induction on the rewrite. Consider first a basic expansion $\Gamma \vdash t \Rightarrow_{\mathcal{F}'} \eta^A(t)$ with premises $\Gamma \vdash t : A \rightarrow B$ and t is neutral. Then by lemma 5.1, subject reduction and type conversion, there is also a typing judgement $\Gamma \vdash \mathtt{TNF}(t) : \overline{A} \rightarrow \overline{B}$ and hence a rewrite $\Gamma \vdash \mathtt{TNF}(t) \Rightarrow_{\mathcal{F}} \lambda x : \overline{A}.\mathtt{TNF}(t)x$ which meets the requirement of the lemma. The same form of argument deals with an expansion $\Gamma \vdash t \Rightarrow \lambda X : K.tX$, except that a few more reductions are required to normalise the type variable X.

Now consider a β-reduction $\Gamma \vdash (\lambda x : T.t)u \Rightarrow_{\mathcal{I}'} t[u/x]$. Then the type normal form of the redex is $(\lambda x : \overline{T}.\mathtt{TNF}(t))\mathtt{TNF}(u)$ which $\Rightarrow_{\mathcal{I}}$-reduces to $\mathtt{TNF}(t)[\mathtt{TNF}(u)/x]$ which by lemma 5.1 is $\mathtt{TNF}(t[u/x])$. A β-reduction $\Gamma \vdash (\lambda X : K.t)U \Rightarrow_{\mathcal{I}'} t[U/X]$ is dealt with similarly, again using lemma 5.1. All other cases are dealt with by the induction hypothesis. $\square$

Of course this does not quite prove that $\Rightarrow_{\mathcal{F}'}$ is strongly normalising as some of the induced $\Rightarrow^*_{\mathcal{F}}$-rewrites may be of zero length. A $\Rightarrow_{\mathcal{F}'}$-rewrite is said to be *type-induced* iff it is formed by a reduction inside a subterm which is a type. Such rewrites are also $\Rightarrow_{\mathcal{F}}$-rewrites as $\Rightarrow_{\mathcal{F}'}$ and $\Rightarrow_{\mathcal{F}}$ are both built using the same reductions on type constructors as given in Table 2.

Lemma 5.3 *If a reduction* $\Gamma \vdash t \Rightarrow_{\mathcal{F}'} t'$ *is type induced, then* $\mathtt{TNF}(t) = \mathtt{TNF}(t')$. *If the reduction is not type-induced, then the reduction sequence* $\Gamma \vdash \mathtt{TNF}(t) \Rightarrow^*_{\mathcal{F}} \mathtt{TNF}(t')$ *is of length at least 1.*
Proof The first part of the lemma is proved by induction on the rewrite, while the second follows by inspection of the proof of lemma 5.2. $\square$

Lemma 5.4 *The relation* $\Rightarrow_{\mathcal{F}'}$ *is strongly normalising.*
Proof First note that there are no infinite type-induced $\Rightarrow_{\mathcal{F}'}$-reduction sequences, because reduction on type constructors is strongly normalising. Thus an infinite $\Rightarrow_{\mathcal{F}'}$-reduction sequence must be of the form $s_1 \Rightarrow^+_{\mathcal{F}'} u_1 \Rightarrow^+_{\mathcal{F}'} s_2 \Rightarrow^+_{\mathcal{F}'} u_2 \cdots$ where non of the reductions in the sequence $s_i \Rightarrow^+_{\mathcal{F}'} u_i$ are type-induced, while all of the reductions in the sequence $u_i \Rightarrow^+_{\mathcal{F}'} s_{i+1}$ are type-induced and either (i) this sequence doesn't terminate; or (ii) there is a j such that there is an infinite sequence of non-type induced reductions starting from s_j. In either case, lemma 5.3 is applied to derive an infinite $\Rightarrow_{\mathcal{F}}$-reduction sequence $\mathtt{TNF}(s_1) \Rightarrow^+_{\mathcal{F}} \mathtt{TNF}(s_2) \cdots \mathtt{TNF}(s_j) \Rightarrow^+_{\mathcal{F}} \cdots$. $\square$

Lemma 5.5 *The relation* $\Rightarrow_{\mathcal{F}'}$ *is confluent.*
Proof If two terms are $\Rightarrow_{\mathcal{F}'}$-equivalent, then because $\Rightarrow_{\mathcal{F}'}$ and $\Rightarrow_{\mathcal{F}}$ generate the same equational theory, these terms are also $\Rightarrow_{\mathcal{F}}$-equivalent. Thus these terms have a $\Rightarrow^*_{\mathcal{F}}$-common reduct, but as $\Rightarrow_{\mathcal{F}}$ is a subrelation of $\Rightarrow_{\mathcal{F}'}$ this term is also a $\Rightarrow^*_{\mathcal{F}'}$-common reduct of the two original terms. Thus $\Rightarrow_{\mathcal{F}'}$ is confluent. $\square$

6 Conclusions and Further Work

This paper has shown that mathematical framework developed in earlier papers is sufficiently robust to handle expansionary η-rewrite rules in F^ω. This is the first step towards generalising [4, 3] by showing that the combination of F^ωequipped with $\beta\eta$-equality and algebraic rewrite systems is a smooth one. In particular, research currently focusses on proving that if the algebraic rewrite system is confluent and/or strongly normalising, then so is its union with $\Rightarrow_{\mathcal{F}}$. The other main extension is to apply η-expansions to the Calculus of Constructions, where the added feature of type dependency greatly complicates matters, e.g. one can no longer analyse $\beta\eta$-equality on types before $\beta\eta$-equality on terms. Also, note that if expansions are permitted without requiring the type of the term to be expanded to be in normal form, then if $B(x) = (\lambda z : X \rightarrow X.X)(x)$, there is a typing judgement $X : *, x : X \rightarrow X \vdash x : \Pi z : B(x).X$ and hence an infinite reduction sequence

$$X : *, x : X \rightarrow X \vdash x \Rightarrow \lambda z : B(x).xz \Rightarrow \lambda z : B(\lambda z : B(x).xz).xz \Rightarrow \ldots \qquad (2)$$

Hence the restriction that the type of a term to be expanded must be in normal form, introduced in this paper purely for technical reasons, will take on a more fundamental role in the Calculus of Constructions.

References

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

[2] Val Breazu-Tannen. Combining algebra and higher-order types. In *LICS*, pages 82–90, 1988.

[3] Val Breazu-Tannen and Jean Gallier. Polymorphic rewriting preserves algebraic confluence. In *Information and Computation*, 1994.

[4] R. Di Cosmo and D. Kesner. Combining first order algebraic rewrite systems, recursion and extensional λ-calculi. In *ICALP*, volume 820 of *Lecture Notes in Computer Science*, pages 462–472. Springer Verlag, 1994.

[5] R. Di Cosmo and D. Kesner. Simulating expansions without expansions. *Mathematical Structures in Computer Science*, 4:1–48, 1994.

[6] D. Dougherty. Some λ-calculi with categorical sums and products. In *Rewriting Techniques and Applications*, volume 690 of *Lecture Notes in Computer Science*, pages 137–151. Springer Verlag, 1993.

[7] G. Dowek. On the definition of η-long normal forms in the type systems of the cube. *Informal Proceedings of the 1993 Workshop on Types for Proofs and Programs*, 1993.

[8] J. Gallier. On girard's candidats de reductibilite. *Logic and Computer Science*, pages 123–203, 1990.

[9] N. Ghani. $\beta\eta$-equality for coproducts. In *Typed λ-calculus and Applications*, number 902 in Lecture Notes in Computer Science, pages 171–185. Springer Verlag, 1995.

[10] N. Ghani. *Adjoint Rewriting*. PhD thesis, University of Edinburgh, Department of Computer Science, submitted.

[11] H. Guevers. The Church-Rosser property for $\beta\eta$-reduction in typed λ-calculi. In *LICS*, pages 453–460. IEEE, 1992.

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

[13] C. B. Jay and N. Ghani. The virtues of eta-expansion. *Journal of Functional Programming*, to appear.

[14] G. E. Mints. Teorija categorii i teoria dokazatelstv.i. *Aktualnye problemy logiki i metodologii nauky*, pages 252–278, 1979.

[15] D. Prawitz. Ideas and results in proof theory. In J.E. Fenstad, editor, *Proc. 2nd Scandinavian Logic Symposium*, pages 235–307. North Holland, 1971.