# The Church-Rosser Property for $\beta\eta$-reduction in Typed $\lambda$-Calculi

Herman Geuvers
Faculty of Mathematics and Computer Science
University of Nijmegen
Nijmegen, The Netherlands

**Abstract**

In this paper we investigate the Church-Rosser property (CR) for Pure Type Systems with $\beta\eta$-reduction. For Pure Type Systems with only $\beta$-reduction, CR on well typed terms follows immediately from CR on the so called 'pseudoterms' and subject reduction. For $\beta\eta$-reduction, CR on the set of pseudoterms is just false, as was shown by [Nederpelt 1973]. Here we prove that CR (for $\beta\eta$) on the well-typed terms *of a fixed type* holds, which is the maximum we can expect in view of Nederpelts counterexample. The proof is given for a large class of Pure Type systems that contains e.g. LF (for which CR for $\beta\eta$ was proved by [Salvesen 1989] and [Coquand 1991]), F, F$\omega$ and the Calculus of Constructions. In the proof, one key lemma (a very weak form of CR for $\beta\eta$ on pseudoterms) takes a central position. It is remarkable that in the proof of this key lemma the counterexample to CR for $\beta\eta$ is essentially used.

## 1 Introduction

Simply typed lambda calculus (from now on denoted by $\lambda{\to}$) and polymorphic lambda calculus are usually described by first giving the set of types and then the derivation rules for deriving typing judgements of the form $\Gamma \vdash M{:}\sigma$, where $\sigma$ is a type, $\Gamma$ is a 'context' and $M$ is an element of the set of so called 'pseudoterms'. For $\lambda{\to}$ this amounts to the following.

$$\mathsf{Typ}_{\to} ::= \mathsf{TVar} \mid \mathsf{Typ}_{\to} {\to} \mathsf{Typ}_{\to},$$

where $\mathsf{TVar}$ denotes the set of type variables. A *context* is a sequence of declarations $x{:}\sigma$ with $\sigma \in \mathsf{Typ}_{\to}$ and $x \in \mathsf{var}$, the set of term variables. The derivation rules are.

$$(\mathrm{var})\ \frac{}{\Gamma \vdash x : \sigma}\ \text{if } x{:}\sigma \in \Gamma$$

$$(\lambda)\ \frac{\Gamma, x{:}\sigma \vdash M : \tau}{\Gamma \vdash \lambda x{:}\sigma.M : \sigma{\to}\tau}$$

$$(\mathrm{app})\ \frac{\Gamma \vdash M : \sigma{\to}\tau \quad \Gamma \vdash N : \sigma}{\Gamma \vdash MN : \tau}$$

For meta-theory it is convenient to introduce the set of *pseudoterms of* $\lambda{\to}$,

$$\mathsf{T}_{\to} ::= \mathsf{var} \mid (\lambda \mathsf{var}{:}\mathsf{Typ}_{\to}.\mathsf{T}_{\to}) \mid \mathsf{T}_{\to}\mathsf{T}_{\to},$$

then one can see the derivation rules as singling out the well-typed terms from the set of pseudoterms. Let's write $\mathsf{Term}_{\to}$ for the set of pseudoterms typable with some type in some context. Now, $\beta$-reduction on $\mathsf{T}_{\to}$ is Church-Rosser, so with the *Subject Reduction property*: $\Gamma \vdash M{:}\sigma \ \&\ M \twoheadrightarrow_\beta M'$, then $\Gamma \vdash M'{:}\sigma$, we obtain that $\beta$-reduction is CR on the well-typed terms: $M \in \mathsf{Term}_{\to},\ M \twoheadrightarrow_\beta M' \ \&\ M \twoheadrightarrow_\beta M''$, then $\exists N \in \mathsf{Term}_{\to}.M' \twoheadrightarrow_\beta N \ \&\ M'' \twoheadrightarrow_\beta N$.

If one is interested in the combination of $\beta$- and $\eta$-reduction, the situation is a little bit more complicated, because we have the following counterexample to CR for $\beta\eta$-reduction on $\mathsf{T}_{\to}$. (Originally due to [Nederpelt 1973].) Take

$$M := \lambda x{:}\sigma.(\lambda y{:}\tau.y)x,$$

with $\sigma \neq \tau$. Then $M \longrightarrow_\beta \lambda x{:}\sigma.x$ and $M \longrightarrow_\eta \lambda x{:}\tau.x$ and both are in normal form. This problem can be overcome by noticing that the only problematic overlapping of redexes in $\mathsf{T}_{\to}$ is when we have a subterm $\lambda x{:}\sigma.(\lambda y{:}\tau.M)x$ (with $x \notin \mathrm{FV}(M)$) and by proving that in well-typed terms, this can only happen when $\sigma \equiv \tau$. For the polymorphic lambda calculus the situation is quite similar; we now have polymorphic types and polymorphic terms (with abstraction and application), but for the problematic overlapping of redexes in a term $(\lambda x{:}\sigma.(\lambda y{:}\tau.M)x$ with $x \notin \mathrm{FV}(M))$, we still have that $\sigma \equiv \tau$.

For the higher order lambda calculus ($\mathrm{F}_\omega$), the situation is a bit more difficult, because we have reduction inside types. Now types are also formed inside a context and there is an extra rule, stating that convertible types have the same 'inhabitants'. The syntax is the following. There are two sets of variables, $\mathsf{var}$, denoting the term variables, and $\mathsf{Var}$, denoting the so called *constructor variables* (the set of constructors will include the types.) We also distinguish two kinds of contexts, the *constructor contexts*, denoted by $\Delta$, for declaring constructor variables, and the *term contexts*, denoted by $\Gamma$, for declaring term variables. The term variables are declared to types and the constructor variables to so called *kinds*. 'Kind' represents the collection of kinds and 'Type' represents the kind of types. Kind is defined by $\mathsf{Kind} ::= \mathsf{Type} \mid \mathsf{Kind}{\to}\mathsf{Kind}$. We now give the derivation rules. Let $\Delta$ be a constructor context.

1. $(\mathsf{Var})\ \dfrac{}{\Delta \vdash \alpha : A}\ \text{if } \alpha{:}A \in \Delta$

$$2.\ (\lambda 1)\ \frac{\Delta, \alpha{:}A \vdash M : B}{\Delta \vdash \lambda\alpha{:}A.M : A{\to}B}$$

$$3.\ (\text{app1})\ \frac{\Delta \vdash M : A{\to}B \quad \Delta \vdash N : A}{\Delta \vdash MN : B}$$

$$4.\ (\to)\ \frac{\Delta \vdash A : \mathsf{Type} \quad \Delta \vdash B : \mathsf{Type}}{\Delta \vdash A{\to}B : \mathsf{Type}}$$

$$5.\ (\Pi)\ \frac{\Delta, \alpha{:}A \vdash B : \mathsf{Type}}{\Delta \vdash \Pi\alpha{:}A.B : \mathsf{Type}}$$

$$6.\ (\text{context})\ \frac{\Delta \vdash A : \mathsf{Type} \quad \Delta; \Gamma\ ok}{\Delta; \Gamma, x{:}A\ ok}$$

$$7.\ (\mathsf{var})\ \frac{\Delta; \Gamma\ ok}{\Delta; \Gamma \vdash x : A}\ \text{if } x{:}A \in \Gamma$$

$$8.\ (\lambda 2)\ \frac{\Delta; \Gamma, x{:}A \vdash M : B}{\Delta; \Gamma \vdash \lambda x{:}A.M : A{\to}B}$$

$$9.\ (\lambda 3)\ \frac{\Delta, \alpha{:}A; \Gamma \vdash M : B}{\Delta; \Gamma \vdash \lambda\alpha{:}A.M : \Pi\alpha{:}A.B}\ \text{if } \alpha \notin \mathrm{FV}(\Gamma)$$

$$10.\ (\text{app2})\ \frac{\Delta; \Gamma \vdash M : A{\to}B \quad \Delta; \Gamma \vdash N : A}{\Delta; \Gamma \vdash MN : B}$$

$$11.\ (\text{app3})\ \frac{\Delta; \Gamma \vdash M : \Pi x{:}A.B \quad \Delta; \Gamma \vdash N : A}{\Delta; \Gamma \vdash MN : B[N/x]}$$

$$12.\ (\text{conv}_\beta)\ \frac{\Delta; \Gamma \vdash M : A \quad \Delta \vdash B : \mathsf{Type}}{\Delta; \Gamma \vdash M : B}\ A =_\beta B$$

Here constructor variables are denoted by Greek characters and term variables by Roman characters. In the rules (Var), (var) and (weak) it is always assumed that the newly declared variable is fresh, that is, it has not yet been declared in $\Gamma$.

If one wants to look at $\eta$-reduction in the system, it is of course reasonable to replace the rule ($\text{conv}_\beta$) by ($\text{conv}_{\beta\eta}$), which is the conversion rule with $A =_{\beta\eta} B$ as side condition. This equality condition is an equality in the set of pseudoterms $\mathsf{T}$, which can be defined similarly to $\mathsf{T}_{\to}$ for $\lambda{\to}$, with extra clauses for $\Pi$-expressions etcetera. This equality on pseudoterms as a side condition may seem strange, because many of these pseudoterms do not have any semantical significance and it would be more intuitive to replace this side condition by an *equality judgement* of the form $\Gamma \vdash A = B : \mathsf{Type}$ (where this judgement means that $A$ converts to $B$ via a reduction/expansion path in which all expressions are well-typed.) A reason for studying the system with equality as a side condition is that meta-theory about reduction and confluence becomes easier, because one can apply well-known properties of the untyped lambda calculus without having to cope with typing conditions. We see that for studying $\beta\eta$-reduction, untyped lambda calculus is not of immediate help. Nevertheless, the general $\mathrm{CR}_{\beta\eta}$ proof given below will make use of the untyped lambda calculus. We shall come back later to the two different ways of introducing and using the equality in the system.

Now $\mathrm{CR}_{\beta\eta}$ on the pseudoterms $\mathsf{T}$ doesn't hold and if $\lambda x{:}A.(\lambda y{:}B.M)x$ (well-typed with $A, B{:}\mathsf{Type}$) is a $\beta$- and an $\eta$-redex, we only obtain (by meta-reasoning) that $A =_{\beta\eta} B$. It looks like before proving $\mathrm{CR}_{\beta\eta}$ for well-typed terms, we have to prove $\mathrm{CR}_{\beta\eta}$ for types. In the system F$\omega$ this is possible; types can only contain redexes in which the $\lambda$-abstraction arises from the ($\lambda 1$)-rule and application from the (app1)-rule, because terms can't be subexpressions of types. So, one first proves confluence of $\beta\eta$-reduction for types (for $A$ and $B$ types, if $A =_{\beta\eta} B$, then $A \longrightarrow\!\!\!\!\!\rightarrow_{\beta\eta} C$ and $B \longrightarrow\!\!\!\!\!\rightarrow_{\beta\eta} C$ for some type $C$.) Then $\mathrm{CR}_{\beta\eta}$ for terms follows.

For systems with so called *dependent types*, that is types that have terms as subexpressions, like LF or the Calculus of Constructions (CC), the Church-Rosser property is very complicated: if $\lambda x{:}A.(\lambda y{:}B.M)x$ is a well-typed term, we know $A =_{\beta\eta} B$, but now this doesn't bring us any further because the equality may arise from an equality between terms. $\mathrm{CR}_{\beta\eta}$ for LF was proved in [Salvesen 1989] and later by [Coquand 1991], but for more complicated systems with dependent types the question of $\mathrm{CR}_{\beta\eta}$ was still open. Here we shall prove $\mathrm{CR}_{\beta\eta}$ for normalizing, functional Pure Type Systems, which includes CC. (There is no proof of normalization for $\beta\eta$-reduction for CC in the literature. However, the strong normalization proof for $\beta$-reduction for CC in [Geuvers and Nederhof 1991] can easily be adapted to the case for $\beta\eta$-reduction.)

## 2 Pure Type Systems and the extension with $\eta$-reduction

Let's take a look at the definition of Pure Type Systems (PTSs for short) and give some of the meta theory. (See [Geuvers and Nederhof 1991], [Barendregt 199+] for more meta-theory and examples.)

**Definition 2.1** For $\mathcal{S}$ a set, $\mathcal{A} \subset \mathcal{S} \times \mathcal{S}$ and $\mathcal{R} \subset \mathcal{S} \times \mathcal{S} \times \mathcal{S}$, $\lambda(\mathcal{S}, \mathcal{A}, \mathcal{R})$ is the typed lambda calculus with the following deduction rules.

$$1.\ (\text{sort}) \vdash s_1 : s_2 \text{ if } (s_1, s_2) \in \mathcal{A}.$$

$$2.\ (\text{var})\ \frac{\Gamma \vdash A : s}{\Gamma, x{:}A \vdash x : A}$$

$$3.\ (\text{weak})\ \frac{\Gamma \vdash A : s \quad \Gamma \vdash M : C}{\Gamma, x{:}A \vdash M : C}$$

$$4.\ (\Pi)\ \frac{\Gamma \vdash A : s_1 \quad \Gamma, x{:}A \vdash B : s_2}{\Gamma \vdash \Pi x{:}A.B : s_3}\ \text{if } (s_1, s_2, s_3) \in \mathcal{R}$$

$$5.\ (\lambda)\ \frac{\Gamma, x{:}A \vdash M : B \quad \Gamma \vdash \Pi x{:}A.B : s}{\Gamma \vdash \lambda x{:}A.M : \Pi x{:}A.B}$$

$$6.\ (\text{app})\ \frac{\Gamma \vdash M : \Pi x{:}A.B \quad \Gamma \vdash N : A}{\Gamma \vdash MN : B[N/x]}$$

$$7.\ (\text{conv}_\beta)\ \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : s}{\Gamma \vdash M : B}\ A =_\beta B$$

In the rules (var) and (weak) it is always assumed that the newly declared variable is fresh, that is, it has not yet been declared in $\Gamma$. If $s_2 \equiv s_3$ in a triple $(s_1, s_2, s_3) \in \mathcal{R}$, one usually just writes $(s_1, s_2) \in \mathcal{R}$. The equality $A =_\beta B$ is the transitive, reflexive, symmetric closure of one-step-$\beta$-reduction, which is

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

compatible with application, $\lambda$-abstraction and $\Pi$-abstraction, on the set of *pseudoterms* $\mathsf{T}$, defined by

$$\mathsf{T} ::= \mathcal{S} \mid \mathsf{Var} \mid (\Pi \mathsf{Var}{:}\mathsf{T}.\mathsf{T}) \mid (\lambda \mathsf{Var}{:}\mathsf{T}.\mathsf{T}) \mid \mathsf{TT}.$$

The notions of *free variable* and *bound variable* of an expression are the usual ones from untyped lambda calculus ($\lambda$ and $\Pi$ are the binders), writing $\mathrm{FV}(M)$ for the set of free variables of $M$. We work modulo *$\alpha$-conversion*, i.e. pseudoterms that only differ in their bound variables are considered to be equal, which equality will be denoted by $\equiv$. (So in fact we work with $\equiv$-equivalence classes of pseudoterms or representants of such classes. For practical reasons we shall only work with representants of $\equiv$-equivalence classes in which all bound variables are pairwise distinct and different from the free variables. (The *variable convention*)) The $\Gamma$ in the judgement $\Gamma \vdash M : A$ is, as usual, called a *context*. For $\Gamma = x_1{:}A_1, \ldots x_n{:}A_n$, $\mathrm{FV}(\Gamma)$ denotes the set $\mathrm{FV}(A_1, \ldots A_n)$.

We say that $M$ *is of type* $A$ *in* $\Gamma$ if $\Gamma \vdash M : A$ is a conclusion of a deduction tree which uses only the rules stated above. A *well-typed term* is a pseudoterm $A$ for which $\Gamma \vdash A : B$ or $\Gamma \vdash B : A$ for some $\Gamma$ and $B$. $\mathsf{Term}$ denotes the set of well-typed terms in a $\mathsf{PTS}$, and $\mathsf{Term}(\Gamma, A)$ denotes the well-typed terms of type $A$ in $\Gamma$. A $\mathsf{PTS}$ is *normalizing*, if every well-typed term in it reduces to a normal form.

The collection of $\mathsf{PTS}$s contains well-known type systems like simply typed lambda calculus, $\lambda{\to}$, $(\lambda(\{\mathsf{Type}, \mathsf{Kind}\}, \{\mathsf{Type}{:}\mathsf{Kind}\}, \{(\mathsf{Type}, \mathsf{Type})\})$, Girards systems F ($\lambda{\to}$ extended with the rule $(\mathsf{Kind}, \mathsf{Type})$) and F$\omega$ (F extended with $(\mathsf{Kind}, \mathsf{Kind})$) and the Calculus of Constructions (F$\omega$ with the rule $(\mathsf{Type}, \mathsf{Kind})$). (It requires some meta-theory to see that this version of F$\omega$ is the same as the one introduced before.) The inconsistent type system $\lambda*$, where 'Type' is a type is also a $\mathsf{PTS}$ $(\lambda(\{\mathsf{Type}\}, \{\mathsf{Type}{:}\mathsf{Type}\}, \{(\mathsf{Type}, \mathsf{Type})\})$ and the notion of $\mathsf{PTS}$ also captures logical systems like first, second order and higher order predicate logic, as shown by [Berardi 1988]. (See also [Barendregt 199+].)

In this paper we want to study the extension with $\eta$-equality. First we give some motivations why this extension is interesting and why the Church-Rosser property of the combined $\beta\eta$-reduction is important. (This will hardly need any explanation.) Then we want to show what are the problems that arise when trying to prove CR for $\beta\eta$. Section 2 gives an outline of the proof.

## 2.1 Properties of Pure Type Systems

Here we give some of the meta-theory for $\mathsf{PTS}$s, mainly those theorems that will be encountered later in the Church-Rosser proof. For detailed proofs we refer to [Geuvers and Nederhof 1991] or [Barendregt 199+]. In the following, unless noted otherwise, we work in an arbitrary $\mathsf{PTS}$.

**Proposition 2.2 ($\mathbf{CR}_\beta$)** *The reduction relation $\longrightarrow\!\!\!\!\!\rightarrow_\beta$ is Church-Rosser on $\mathsf{T}$. (That is, if $M \longrightarrow\!\!\!\!\!\rightarrow_\beta M_1$ and $M \longrightarrow\!\!\!\!\!\rightarrow_\beta M_2$ then $M_1 \longrightarrow\!\!\!\!\!\rightarrow_\beta N$ and $M_2 \longrightarrow\!\!\!\!\!\rightarrow_\beta N$ for some $N \in \mathsf{T}$.*

(The proof follows precisely the Church-Rosser proof of $\beta$-reduction on the untyped lambda terms in [Barendregt 1984].)

**Corollary 2.3 (Confluence of $\beta$ on $\mathsf{T}$, $\mathbf{CON}_\beta$)** *If $M =_\beta M'$ then $M \longrightarrow\!\!\!\!\!\rightarrow_\beta N$ and $M' \longrightarrow\!\!\!\!\!\rightarrow_\beta N$ for some $N \in \mathsf{T}$.*

We shall write $M \downarrow_\beta M'$ for $\exists N \in \mathsf{T}.M \longrightarrow\!\!\!\!\!\rightarrow_\beta N\ \&\ M' \longrightarrow\!\!\!\!\!\rightarrow_\beta N$.

From the form of the derivation rules we can conclude the following.

**Lemma 2.4 (Stripping)** *1. $\Gamma \vdash \lambda x{:}A.M : C \Rightarrow \Gamma, x{:}A \vdash M : B, \Gamma \vdash \Pi x{:}A.B : s$ for some $B$ and some $s \in \mathcal{S}$ such that $\Pi x{:}A.B =_{\beta\eta} C$.*

*2. $\Gamma \vdash MN : C \Rightarrow \Gamma \vdash M : \Pi x{:}A.B, \Gamma \vdash N : A$ for some $A$ and $B$ such that $B[N/x] =_{\beta\eta} C$.*

**Proposition 2.5 (Subject Reduction for $\beta$, $\mathbf{SR}_\beta$)** *If $\Gamma \vdash M : A$ and $M \longrightarrow\!\!\!\!\!\rightarrow_\beta M'$ then $\Gamma \vdash M' : A$.*

The proof of the proposition is by induction on the derivation, proving the statement simultaneously for a one step reduction in $\Gamma$ or $M$. It uses Stripping (2.4) and the fact that $\Pi x{:}A.B =_\beta \Pi x{:}C.D$ implies $A =_\beta C$ and $B =_\beta D$, which holds by $\mathrm{CON}_\beta$ on $\mathsf{T}$.

We don't have $\eta$ in the $(\text{conv}_\beta)$ rule, but we do have subject reduction for $\eta$.

**Proposition 2.6 (Subject Reduction for $\eta$, $\mathbf{SR}_\eta$)** *If $\Gamma \vdash M : A$ and $M \longrightarrow\!\!\!\!\!\rightarrow_\eta M'$ then $\Gamma \vdash M' : A$.*

The proof of this proposition is, just as for $\mathrm{SR}_\beta$, by induction on the derivation, proving the statement simultaneously for a one step reduction in $\Gamma$ or in $M$. In the proof one needs *strengthening*:

**Proposition 2.7 (Strengthening)** *If $\Gamma_1, x{:}A, \Gamma_2 \vdash M : B$ with $x \notin FV(\Gamma_2, M, B)$, then $\Gamma_1, \Gamma_2 \vdash M : B$.*

The proof is in [van Benthem Jutting 199+]. In [Geuvers and Nederhof 1991] a proof is given for a subcollection of PTSs, the so called *functional* ones: A PTS is *functional* if the relation $\mathcal{A}$ is a function from $\mathcal{S}$ to $\mathcal{S}$ and the relation $\mathcal{R}$ is a function from $\mathcal{S} \times \mathcal{S}$ to $\mathcal{S}$. (That is, if $s : s'$, $s : s'' \in \mathcal{A}$, then $s' \equiv s''$ and if $(s_1, s_2, s_3), (s_1, s_2, s_3') \in \mathcal{R}$ then $s_3 \equiv s_3'$.) Both proofs use $\mathrm{CR}_\beta$ on $\mathsf{T}$.

**Proposition 2.8 (Uniqueness of types, UT)** *In a functional* PTS*: If* $\Gamma \vdash M : A$, $\Gamma \vdash M : A'$, *then* $A =_\beta A'$.

The proof is by induction on the derivation. It uses $\mathrm{CON}_\beta$ in the sense that one needs $\Pi x{:}A.B =_\beta \Pi x{:}C.D \Rightarrow B =_\beta D$ to hold. (The proposition is false for non-functional PTSs.)

## 2.2 The extension with $\eta$

Extending the PTSs with $\eta$ means just replacing the conversion rule by

$$(\mathrm{conv}_{\beta\eta})\ \frac{\Gamma \vdash M : A \quad \Gamma \vdash B : s}{\Gamma \vdash M : B}\ A =_{\beta\eta} B$$

To distinguish the two, we write $\mathsf{PTS}_\beta$ for the ones with $(\mathrm{conv}_\beta)$ as conversion rule and $\mathsf{PTS}_{\beta\eta}$ for the ones with $(\mathrm{conv}_{\beta\eta})$.

There are many reasons for studying this extension. For one thing, the stronger $\beta\eta$-equality is quite natural, especially if we think about a PTS as representing a logic. The need for $\beta\eta$-equality also becomes apparent when we look at type systems that are used as frameworks, like LF. The completeness of the interpretation of a formal system in a signature relies on defining a mapping back, from terms in the signature to terms of the formal system, which is defined on (representants of) $\beta\eta$-equivalence classes. (See [Harper et al. 1987].) A nice consequence of $\mathrm{CR}_{\beta\eta}$ is that the unification algorithms for the Calculus of Constructions, as discussed in [Pfenning 1991] are complete. A semantical reason for strengthening the equality is that in almost all models of type systems the $\eta$ rule is valid. Further $\mathrm{CR}_{\beta\eta}$ is essential if we relate the syntax of PTSs to the semantics: The systems that are really interpreted in e.g. [Streicher 1989] and [Jacobs 1991] are not the ones as formulated above (with equality on the pseudoterms as a side condition), but with an equality judgement inside the context. If $\zeta$ is a PTS, write $\zeta_=$ for the associated 'semantical' version of the system (with equality judgement), replacing $\vdash$ by $\vdash_=$. The conversion rule of $\zeta_=$ is

$$(\mathrm{conv}'_{\beta\eta})\ \frac{\Gamma \vdash_= M : A \quad \Gamma \vdash_= A = B : s}{\Gamma \vdash_= M : B}$$

where the judgement $\Gamma \vdash_= A = B : s$ is generated by

$$(\beta)\ \frac{\Gamma \vdash_= \lambda x{:}A.M : \Pi x{:}C.D \quad \Gamma \vdash_= N : C}{\Gamma \vdash_= (\lambda x{:}A.M)N = M[N/x] : D[N/x]}$$

$$(\eta)\ \frac{\Gamma, x{:}A \vdash_= Mx : B \quad \Gamma \vdash_= \Pi x{:}A.B : s}{\Gamma \vdash_= \lambda x{:}A.Mx = M : \Pi x{:}A.B}\ \text{if}\ x \notin \mathrm{FV}(M)$$

taking the transitive, reflexive, symmetric closure, made compatible with application, $\lambda$- and $\Pi$-abstraction. In this version of the system the conversion rule can only be applied to two equal types if they are equal via a path through the well-typed terms. In the original PTS version one only claims that the types are equal as pseudoterms. If we have only $\beta$-conversion (i.e. in $\zeta_=$ we don't have $(\eta)$), the two versions are equivalent: If $M, M' \in \mathsf{Term}$ with $M =_\beta M'$, then there is a path between them through $\mathsf{Term}$ (by $\mathrm{CR}_\beta$ on $\mathsf{T}$ and $\mathrm{SR}_\beta$). This is expressed by the following theorem for any $\mathsf{PTS}_\beta$.

**Theorem 2.9**

$$\left.\begin{array}{c}\Gamma \vdash M : A \\ \Gamma \vdash M' : A \\ M =_\beta M'\end{array}\right\} \Longleftrightarrow \Gamma \vdash_= M = M' : A.$$

In fact, this theorem also states that the system with equality on the pseudoterms is a 'sound' system: there would really be something wrong if $\Gamma \vdash M, M' : A$ and $M =_\beta M'$ as pseudoterms without there being a path from $M$ to $M'$ through $\mathsf{Term}(\Gamma, A)$. If we extend both versions of PTSs with $\eta$, we need some form of Church-Rosser to prove the equivalence of the two versions. It suffices to have that if $M$ and $M'$ are of the same type in some context $\Gamma$ and $M =_{\beta\eta} M'$, then $M \downarrow_{\beta\eta} M'$. This is precisely what will be proved, for normalizing functional PTSs. As consequences we find that $\zeta$ and $\zeta_=$ are equivalent (for $\zeta$ functional and normalizing) and that $\mathrm{CR}_{\beta\eta}$ and $\mathrm{CON}_{\beta\eta}$ hold for $\zeta_=$. (It is not clear how a proof of $\mathrm{CR}_{\beta\eta}$ for $\zeta_=$ could be essentially simpler then via the proof for $\zeta$.)

## 2.3 Problems with proving $\mathrm{CR}_{\beta\eta}$

As already remarked, the main problem with proving $\mathrm{CR}_{\beta\eta}$ is that on the pseudoterms $\mathsf{T}$ it is just false. The counterexample $(\lambda x{:}A.(\lambda y{:}B.y)x$ with $A \neq_{\beta\eta} B)$ implies that also $\mathrm{CON}_{\beta\eta}$ on $\mathsf{T}$ (see 2.3) is not true, even if we restrict ourselves to well-typed terms. For the meta-theory for PTSs with $\eta$ this has some serious consequences. UT is not immediate anymore for functional PTSs, with $=_\beta$ replaced by $=_{\beta\eta}$. (One needs that $\Pi x{:}A.B =_{\beta\eta} \Pi x{:}C.D \Rightarrow B =_{\beta\eta} D$.) For $\mathrm{SR}_\beta$ and $\mathrm{SR}_\eta$ the situation is also problematic: The first requires $\Pi x{:}A.B =_{\beta\eta} \Pi x{:}C.D \Rightarrow A =_{\beta\eta} C \ \&\ B =_{\beta\eta} D$. The second requires strengthening, the proof of which uses Church-Rosser. Our solution to this will be to prove a very weak form of Church-Rosser for $\beta\eta$-reduction on $\mathsf{T}$ (using the counterexample to $\mathrm{CR}_{\beta\eta}$(!)), which will turn out to be sufficient to get UT and $\mathrm{SR}_\beta$. Also this key lemma will be sufficient to prove strengthening (and so $\mathrm{SR}_\eta$) for functional, normalizing PTSs. Using all this, we shall prove (for functional, normalizing PTSs)

$$\Gamma \vdash M, M' : A,\ M =_{\beta\eta} M' \Rightarrow \mathsf{nf}(M) \equiv \mathsf{nf}(M')$$

Where we write $\mathsf{nf}(M)$ for the normal form obtained by some normalization procedure. We have as an immediate corollary $\mathrm{CON}_{\beta\eta}$ for $\mathsf{Term}(\Gamma, A)$.

## 3 The proof of $\mathrm{CR}_{\beta\eta}$ for functional, normalizing pure type systems

In the proof of Church-Rosser we shall relate the $\beta\eta$-reduction on typed terms to the reductions on untyped lambda terms. Properties of reduction on the untyped terms will be used to obtain results about reduction in $\mathsf{T}$. We therefore define an *erasure* mapping from $\mathsf{T}$ to $\Lambda$ and give some properties for it. Then we give some properties of $\beta\eta$-reduction and equality on $\mathsf{T}$, which will enable us to prove $\mathrm{SR}_\beta$ and $\mathrm{SR}_\eta$.

The counterexample of [Nederpelt 1973] shows that, if one tries to prove $\mathrm{CR}_{\beta\eta}$, there is a problem in the types of the $\lambda$-abstracted variables. We shall call these types *domains*: A subterm $A$ of $M$ is a *domain* if it occurs as $\lambda x{:}A$ in $M$. (So we are not concerned with $\Pi$-abstractions.) The erasure map removes all domains.

**Definition 3.1** *The map $|\,|: \mathsf{T} \to \Lambda^\Pi$ is defined with induction on the structure of pseudoterms as follows.*

$$\begin{array}{rcl} |x| & := & x, \\ |s| & := & s, \\ |\lambda x{:}A.M| & := & \lambda x.|M|, \\ |\Pi x{:}A.B| & := & \Pi x{:}|A|.|B|, \\ |MN| & := & |M||N|. \end{array}$$

*Here, $\Lambda^\Pi$ is $\Lambda$ extended with the extra variable binder $\Pi$ and constants $s$ for each $s \in \mathcal{S}$. If one views $\Pi x{:}|A|.|B|$ just as $G|A|(\lambda x.|B|)$, with $G$ some fixed constant, it's easy to see that all the facts (like $CR_{\beta\eta}$) about $\beta\eta$-reduction in $\Lambda$ hold for $\Lambda^\Pi$.*

It is easy to see that if for $M, M' \in \mathsf{T}$, $|M| \equiv |M'|$, then $M$ and $M'$ have the same 'structure' apart from the domains that may be very different. If $|M| \equiv |M'|$ and the respective domains in $M$ and $M'$ are all $\beta\eta$-equal, we say that $M$ and $M'$ are *domain-equal*, notation $M \equiv_d M'$. We have the following proposition, relating reduction in $\mathsf{T}$ to reduction in $\Lambda^\Pi$.

**Proposition 3.2** *Let $M$ and $M'$ be in $\mathsf{T}$.*

$$M \longrightarrow_\beta M' \Rightarrow |M| \longrightarrow_\beta |M'| \vee |M| \equiv |M'|,$$

*and similar for $\longrightarrow_\eta$ and so for $=_{\beta\eta}$.*

$$|M| \twoheadrightarrow_\beta |M'| \Rightarrow \exists N[M \twoheadrightarrow_\beta N \;\&\; |N| \equiv |M'|].$$

*(This doesn't hold for $\twoheadrightarrow_\eta$.)*

**Proof** The first is trivial: If the redex is erased by $|\,|$, then $|M| \equiv |M'|$ and otherwise the same redex can still be done in $\Lambda^\Pi$, so $|M| \longrightarrow |M'|$. The second is almost trivial; we are done if we give the proof for a one-step $\beta$-reduction. As $|\,|$ only erases domains, a $\beta$-redex in $|M|$ is also a $\beta$-redex in $M$, and by evaluating it we find $N \in \mathsf{T}$ with $M \longrightarrow_\beta N$ and $N \equiv M'$. This is not valid for $\eta$, as is shown by the counterexample $M \equiv \lambda x{:}\sigma.y(\lambda z{:}Px.z)x$. (This term can even be well-typed in e.g. the Calculus of Constructions: Take $P \equiv \lambda x{:}\sigma.\tau,\ y{:}(\tau{\to}\tau){\to}\sigma{\to}\sigma$. In Lemma 3.14 we shall see that nevertheless, if $M$ is well-typed in the Calculus of Constructions and $|M|$ is in $\beta\eta$-nf, then $M$ is in $\beta\eta$-nf.) $\square$

The following is an immediate corollary of the counterexample to $\mathrm{CR}_{\beta\eta}$ on $\mathsf{T}$.

**Lemma 3.3 (Domain Lemma)** *If $C[\lambda x{:}A.M]$ and $B$ are in $\mathsf{T}$ (i.e. $C$ is a pseudoterm with subterm $\lambda x{:}A.M$), then*

$$C[\lambda x{:}A.M] =_{\beta\eta} C[\lambda x{:}B.M]$$

**Proof**

$$C[\lambda y{:}B.(\lambda x{:}A.M)y]$$

$$C[\lambda x{:}A.M] \qquad\qquad C[\lambda x{:}B.M]$$

where $y$ is any variable not occurring free in $A$ or $M$. $\square$

**Lemma 3.4 (Key Lemma)** *Let $c$ be a variable or a sort.*

1. *$cP_1 \ldots P_n =_{\beta\eta} Q \Rightarrow Q \longrightarrow\!\!\!\!\!\twoheadrightarrow_\beta \lambda\vec{y}{:}\vec{A}.cQ_1 \ldots Q_n\vec{y}$, with $Q_i =_{\beta\eta} P_i\,(1 \le i \le n)$.*
2. *$\Pi x{:}P_1.P_2 =_{\beta\eta} Q \Rightarrow Q \longrightarrow\!\!\!\!\!\twoheadrightarrow_\beta \lambda\vec{y}{:}\vec{A}.(\Pi x{:}Q_1.Q_2)\vec{y}$, with $P_i =_{\beta\eta} Q_i\,(i = 1, 2)$.*

**Proof** We only proof the first case, since the second is totally similar. Some notation: For $D \in \mathsf{T}$ and $M \in \mathsf{T}$, $M^D \in \mathsf{T}$ is the pseudoterm obtained by replacing all domains in $M$ by $D$. For $D \in \mathsf{T}$ and $t \in \Lambda^\Pi$, $t^{+D} \in \mathsf{T}$ is the pseudoterm obtained by adding $D$ as domain to every $\lambda$ abstraction in $t$. (So e.g. $\lambda x.x$ is replaced by $\lambda x{:}D.x$.) For reasons of readibility we adapt here the convention to use capitals for pseudoterms and small characters for elements of $\Lambda^\Pi$.
Let $cP_1 \ldots P_n$ and $Q$ be as in the first case of the lemma. By $\mathrm{CR}_{\beta\eta}$ on $\Lambda^\Pi$ we find $t_1, \ldots, t_n \in \Lambda^\Pi$ with $c|P_1| \ldots |P_n| \twoheadrightarrow_{\beta\eta} ct_1 \ldots t_n$ and $|Q| \twoheadrightarrow_{\beta\eta} ct_1 \ldots t_n$. Using postponement of $\eta$-reduction, we find that $|Q| \twoheadrightarrow_\beta \lambda\vec{y}.cq_1 \ldots q_n\vec{y} \twoheadrightarrow_\eta ct_1 \ldots t_n$. (Doing as many $\beta$-reductions as possible, i.e. we $\beta$-reduce all the $\eta$-redexes that are also $\beta$-redexes.) By 3.2 we find a term $\lambda\vec{y}{:}\vec{A}.cQ_1 \ldots Q_n\vec{y}$ with $Q \twoheadrightarrow_\beta \lambda\vec{y}{:}\vec{A}.cQ_1 \ldots Q_n\vec{y}$ and $|\lambda\vec{y}{:}\vec{A}.cQ_1 \ldots Q_n\vec{y}| \equiv \lambda\vec{y}.cq_1 \ldots q_n\vec{y}$. The situation is as follows.

$$cP_1 \ldots P_n \qquad\qquad Q$$

$$c|P_1| \ldots |P_n| \qquad |Q|$$

$$\lambda\vec{y}{:}\vec{A}.cQ_1 \ldots Q_n\vec{y}$$

$$ct_1 \ldots t_n \qquad \lambda\vec{y}.cq_1 \ldots q_n\vec{y}$$

Take for $D$ some closed pseudoterm (or fresh variable), then $\lambda\vec{y}{:}\vec{D}.cQ_1^D \ldots Q_n^D\vec{y} \longrightarrow\!\!\!\!\!\rightarrow_\eta ct_1^{+D} \ldots t_n^{+D}$ and $cP_1^D \ldots P_n^D \longrightarrow\!\!\!\!\!\rightarrow_{\beta\eta} ct_1^{+D} \ldots t_n^{+D}$. Using Lemma 3.3 we obtain (for $1 \leq i \leq n$),

$$P_i \quad =_{\beta\eta} \quad P_i^D \qquad\qquad Q_i^D \quad =_{\beta\eta} \quad Q_i$$

$$t_i^{+D}$$

so $Q_i =_{\beta\eta} P_i$ $(1 \leq i \leq n)$ and we are done. □

Using the Key Lemma we get the following important properties.

**Proposition 3.5 (UT for functional PTSs)** *In a functional* PTS *we have* $\Gamma \vdash M : A \& \Gamma \vdash M : A' \Rightarrow A =_{\beta\eta} A'$.

**Theorem 3.6 ($\mathbf{SR}_\beta$)** *For $\zeta$ a* $\mathsf{PTS}_{\beta\eta}$, $\zeta \models SR_\beta$.

The proofs of the proposition and the theorem are straightforward by redoing the proofs for $\mathsf{PTS}_\beta$ in [Geuvers and Nederhof 1991], using the Key Lemma. As corollaries of $\mathrm{SR}_\beta$ and the Key Lemma we find: If $\Gamma \vdash Q, \Pi x{:}P_1.P_2 : s$ $(s \in \mathcal{S})$ with $Q =_{\beta\eta} \Pi x{:}P_1.P_2$, then $Q \longrightarrow\!\!\!\!\!\rightarrow_\beta \Pi x{:}Q_1.Q_2$ with $P_i =_{\beta\eta} Q_i$ and similar for $xP_1 \ldots P_n$. Further, if $\Gamma \vdash M : Q$ with $Q =_{\beta\eta} s$ $(s \in \mathcal{S})$, then $Q \longrightarrow\!\!\!\!\!\rightarrow_\beta s$. (These are proved using the fact that a type is not a $\lambda$-abstraction.)

We shall now turn to the proof of $\mathrm{SR}_\eta$ for functional, normalizing systems. Again the Key Lemma is used and, as said, we also need some form of strengthening to prove $\mathrm{SR}_\eta$. First we state another lemma, which is required for strengthening.

**Lemma 3.7** *In a functional normalizing* $\mathsf{PTS}_{\beta\eta}$*:*

$$\left.\begin{array}{r}\Gamma \vdash D : s \\ \Gamma \vdash D' : s' \\ D =_{\beta\eta} D'\end{array}\right\} \Rightarrow s \equiv s'.$$

There are essentially two ways to prove this lemma: It can be proved directly (it requires some sublemmas), but also by first adding strengthening as a rule to the system, that is we have an extra derivation rule

$$(\text{streng})\ \frac{\Gamma_1, x{:}A, \Gamma_2 \vdash M : B}{\Gamma_1, \Gamma_2 \vdash M : B} \text{ if } x \notin \mathrm{FV}(\Gamma_2, M, B).$$

In the second case one has to redo the proof for $\mathrm{SR}_\beta$ (straightforward) and prove $\mathrm{SR}_\eta$ (using the rule (streng)). Then the proof of $\mathrm{CR}_{\beta\eta}$ and $\mathrm{CON}_{\beta\eta}$ can be done in the same way as will be done here (in 3.12 - 3.16.) From $\mathrm{CON}_{\beta\eta}$ one then proves that Lemma 3.7 holds in the system without the (streng) rule and so $\mathrm{CR}_{\beta\eta}$ and $\mathrm{CON}_{\beta\eta}$ hold in the system without (streng). The direct proof involves some sublemmas. We give them here without a detailed proof.

**Sublemma 3.8** $\Gamma, x{:}D \vdash M : C$, $\Gamma \vdash D' : s$ $(s \in \mathcal{S})$, $D =_{\beta\eta} D'$ *then* $\Gamma, x{:}D' \vdash M : C$

**Sublemma 3.9** $\Gamma \vdash B : s$, $\Gamma \vdash B' : s'$, $B =_{\beta\eta} B'$, *$B$ in $\beta$*nf *then* $s \equiv s'$.

The proof of the first is by changing everywhere in the derivation tree of $\Gamma, x{:}D \vdash M : C$ the declaration $x : D$ into $x : D'$: We introduce $x$ as $x : D'$ (with the rule (var) or (weak)) and if $x$ is introduced with the (var) rule, we immediately apply $(\mathrm{conv}_{\beta\eta})$ to conclude $\Gamma, x{:}D' \vdash x : D$. The proof of the second is by induction on $B$, using the first sublemma and the Key Lemma. Now Lemma 3.7 easily follows.

Using 3.7 we can prove a weak form of strengthening for functional normalizing $\mathsf{PTS}_{\beta\eta}$, which is sufficient to prove $\mathrm{SR}_\eta$ and which also implies strengthening itself. The proof uses also the Key Lemma and it's corollaries.

**Lemma 3.10** *In a functional, normalizing* $\mathsf{PTS}_{\beta\eta}$*:*

$$\left.\begin{array}{r}\Gamma_1, x{:}A, \Gamma_2 \vdash M : B \\ x \notin FV(\Gamma_2, M)\end{array}\right\} \Rightarrow \begin{array}{l}\Gamma_1, \Gamma_2 \vdash M : B' \\ \textit{for some } B' =_{\beta\eta} B.\end{array}$$

**Proposition 3.11 ($\mathbf{SR}_\eta$)** *For $\zeta$ a functional, normalizing* $\mathsf{PTS}_{\beta\eta}$, $\zeta \models SR_\eta$.

**Proof** The proof is by proving simultaneously the case for a one step reduction in the context or in the subject. The only interesting case is when $\Gamma \vdash \lambda x{:}A.Mx : C$ with $x \notin \mathrm{FV}(M)$ and we have to prove $\Gamma \vdash M : C$. By the Stripping Lemma (2.4), we find a term $B$ with $\Pi x{:}A.B =_{\beta\eta} C$ and $\Gamma, x{:}A \vdash Mx : B$. Again with Stripping Lemma we find a term $\Pi y{:}D.E$ with $\Gamma, x{:}A \vdash M : \Pi x{:}D.E$, $E[x/y] =_{\beta\eta} B$ and $D =_{\beta\eta} A$. By Lemma 3.10 we find that $\Gamma, x{:}A \vdash M : F$ with $F =_{\beta\eta} \Pi y{:}D.E =_{\beta\eta} \Pi x{:}A.B =_{\beta\eta} C$. Applying $(\mathrm{conv}_{\beta\eta})$ we conclude $\Gamma, x{:}A \vdash M : C$. □

The next lemmas will establish the proof of $\mathrm{CON}_{\beta\eta}$ on $\mathsf{Term}(\Gamma, A)$ for functional normalizing Pure Type Systems. So suppose we work in such a $\mathsf{PTS}_{\beta\eta}$.

**Lemma 3.12**

$$\left.\begin{array}{r}\Gamma \vdash M{:}A \\ \Gamma \vdash M'{:}A' \\ A =_{\beta\eta} A' \\ |M| \equiv |M'| \\ M, M' \textit{ in } \beta\text{-}nf\end{array}\right\} \Rightarrow M \equiv_d M'$$

*(For the definition of $\equiv_d$, see the remarks after Definition 3.1.)*

**Proof** $M$ and $M'$ have the same structure (apart from the domains), say $M = \lambda x_1{:}A_1 \ldots \lambda x_n{:}A_n.N$ and $M' = \lambda x_1{:}A_1' \ldots \lambda x_n{:}A_n'.N'$, with $N$ and $N'$ not abstractions. From the type of $M$ and $M'$ we conclude that $A_i =_{\beta\eta} A_i'$. Now compare from left to right all domains in $N$ and $N'$. Say $B$ occurs as $zR_1 \ldots R_q(\lambda y_1{:}E_1 \ldots \lambda y_p{:}E_p.\lambda x{:}B.P)$ in $N$ and $B'$ occurs as $zR_1' \ldots R_q'(\lambda y_1{:}E_1' \ldots \lambda y_p{:}E_p'.\lambda x{:}B'.P')$ in $N'$ and for all domains to the left of $B$ (respectively $B'$) we are already done. (This implies that

$R_i =_{\beta\eta} R'_i$ for all $i$ and $E_i =_{\beta\eta} E'_i$ for all $i$.) We look at the types of $z, zR_1, \ldots, zR_1 \ldots R_q$ in the derivation tree and compare them with the types of $z, zR'_1, \ldots, zR'_1 \ldots R'_q$ in the other derivation tree. Notice that they are pairwise $\beta\eta$-equal. This implies that the types of $\lambda y_1{:}E_1 \ldots \lambda y_p{:}E_p.\lambda x{:}B.P$ and $\lambda y_1{:}E'_1 \ldots \lambda y_p{:}E'_p.\lambda x{:}B'P'$ are $\beta\eta$-equal, so $B =_{\beta\eta} B'$. □

**Lemma 3.13**

$$\left.\begin{array}{r} \Gamma \vdash A{:}s \\ A \textit{ in } \beta\eta\textit{-nf} \\ A =_{\beta\eta} C \\ x \notin FV(C, \Gamma) \end{array}\right\} \Rightarrow x \notin FV(A).$$

**Proof** The proof is by induction on the structure of $A$. For $A$ a variable or a sort it's trivial. For $A \equiv \Pi x{:}A_1.A_2$, we are done by induction hypothesis. Suppose now $A$ is an application term and $x \in \mathrm{FV}(A)$. Then $x$ is only free in domains of $A$. (Note that $|C| =_{\beta\eta} |A| \longrightarrow\!\!\!\!\!\rightarrow_\eta \mathsf{nf}(|A|)$, and in untyped lambda calculus $\eta$-reductions do not remove any free variables, so $x \notin \mathrm{FV}(|A|)$.) Say $B$ is the leftmost domain of $A$ in which $x$ occurs free, say in the subterm $zR_1 \ldots R_q(\lambda y_1{:}E_1 \ldots \lambda y_p{:}E_p.\lambda y{:}B.P)$. Then there is a type for $z$ in which $x$ is not free ($z$ is declared in the context or left to $B$), so there is a type for $zR_1 \ldots R_q$ in which $x$ is not free, so $B =_{\beta\eta} E$, for some $E$ in which $x$ is not free. Now by induction hypothesis, $x \notin FV(B)$. □

**Lemma 3.14** *For $M \in \mathsf{Term}$, if $M$ in $\beta\eta$-nf, then $|M|$ in $\beta\eta$-nf.*

**Proof** Suppose $|M|$ is not in $\beta$-nf. Then there is an $\eta$-redex in $|M|$, which is not an $\eta$-redex in $M$, say $\lambda x.|N|x$ is the first such. Then $x \in \mathrm{FV}(N)$, but $x \notin \mathrm{FV}(|N|)$, so $x$ is only free in domains of $N$. Say $B$ is the leftmost domain in which $x$ is free, and say $B$ occurs in the subterm $zR_1 \ldots R_q(\lambda y_1{:}E_1 \ldots \lambda y_p{:}E_p.\lambda y{:}B.P)$. The type of $z$ does not have $x$ as a free variable in it. (If $z$ is abstracted in the context or left from the abstraction over $x$, then not by variable convention and if $z$ is abstracted right from $x$, then not by the assumption that $B$ is the leftmost domain containing $x$.) With an argument similar to that in the proof of 3.13, there is a type $B'$ with $B =_{\beta\eta} B'$, and $x \notin \mathrm{FV}(B')$. By 3.13: $x \notin \mathrm{FV}(B)$. □

**Lemma 3.15 ($\mathrm{CON}_{\beta\eta}$ for types)** *Let $s, s' \in \mathcal{S}$.*

$$\left.\begin{array}{r} \Gamma \vdash A{:}s \\ \Gamma \vdash B{:}s' \\ A =_{\beta\eta} B \\ A, B \textit{ in } \beta\eta\textit{-nf} \end{array}\right\} \Rightarrow A \equiv B.$$

**Proof** By induction on the structure of $A$, using the Key Lemma, 3.12 and 3.14.
If $A \equiv \Pi x{:}A_1.A_2$, then $B =_{\beta\eta} \Pi x{:}B_1.B_2$ with $A_1 =_{\beta\eta} B_1$ and $A_2 =_{\beta\eta} B_2$ (by Key Lemma). By induction hypothesis $A_1 \equiv B_1$ and $A_2 \equiv B_2$.
If $A \equiv xP_1 \ldots P_n$, then $B =_{\beta\eta} xQ_1 \ldots Q_n$ with $P_i =_{\beta\eta} Q_i$ (by Key Lemma.) Now, the types of $xP_1 \ldots P_i$ and $xQ_1 \ldots Q_i$ in the derivations of $\Gamma \vdash A{:}s$ resp $\Gamma \vdash B{:}s'$ are pairwise $\beta\eta$-equal. Further, all $P_i$ and $Q_i$ are in $\beta\eta$-nf, so, by 3.14, all $|P_i|$ and $|Q_i|$ are, so $|P_i| \equiv |Q_i|$ for all $i$. We can apply 3.12 to conclude that $P_i \equiv_d Q_i$ for all $i$ and so $A \equiv_d B$ (all respective domains in $A$ and $B$ are $\beta\eta$-equal.) By induction hypothesis (comparing the respective domains in $A$ and $B$ from left to right) we conclude that $A \equiv B$. □

**Theorem 3.16 ($\mathrm{CON}_{\beta\eta}$)**

$$\left.\begin{array}{r} \Gamma \vdash M{:}A \\ \Gamma \vdash M'{:}A \\ M =_{\beta\eta} M' \end{array}\right\} \Rightarrow M \downarrow_{\beta\eta} M'.$$

**Proof** Define $N := \mathsf{nf}(M), N' := \mathsf{nf}(M')$. We prove $N \equiv N'$ and we are done.
By $\mathrm{SR}_\beta$ and $\mathrm{SR}_\eta$ we find $\Gamma \vdash N{:}A$ and $\Gamma \vdash N'{:}A$. By 3.14, $|N|$ and $|N'|$ are in normal form, so $|N| \equiv |N'|$. By 3.12, $N \equiv_d N'$: All respective domains in $N$ and $N'$ are $\beta\eta$-equal. By $\mathrm{CON}_{\beta\eta}$ for types (3.15), all respective domains in $N$ and $N'$ are syntactically equal ($\equiv$), so $N \equiv N'$. □

## 4 Discussion

We have proved $\mathrm{CON}_{\beta\eta}$ for terms in a fixed context of a fixed type, but only for functional normalizing $\mathsf{PTS}_{\beta\eta}$. This immediately implies $\mathrm{CR}_{\beta\eta}$ on $\mathsf{Term}$. We have also seen that confluence for well-typed terms of different types doesn't hold. (And the same for well-typed terms in different contexts; take $\Gamma$ and $\Gamma'$ such that $\Gamma \vdash x(\lambda y : A.y) : \mathsf{Type}$ and $\Gamma' \vdash x(\lambda y : B.y) : \mathsf{Type}$.)

We think that, using the work of [van Benthem Jutting 199+], who gives an analysis of typing in $\mathsf{PTS}$s, these results can be extended to arbitrary normalizing type systems. The most interesting extension, however, seems to be the one to non-normalizing type systems, like $\lambda*$. First because the proof given here relies very heavily on the normalization, which makes the $\mathrm{CR}_{\beta\eta}$ theorem a higher order property, where we believe it is essentially combinatoric. Second, because from $\mathrm{CON}_{\beta\eta}$ on $\mathsf{Term}(\Gamma, A)$ in $\lambda*$ (with $(\mathrm{conv}_{\beta\eta})$) we hope to get $\mathrm{CON}_{\beta\eta}$ on $\mathsf{Term}(\Gamma, A)$ for an arbitrary $\mathsf{PTS}_{\beta\eta}$, by imitating the reduction steps in $\lambda*$ in the other $\mathsf{PTS}_{\beta\eta}$, using the terminality of $\lambda*$ in the category $\mathsf{PTS}_{\beta\eta}$. This would also require $\mathrm{SR}_\eta$ for an arbitrary $\mathsf{PTS}_{\beta\eta}$. (Here we only have a proof of $\mathrm{SR}_\eta$ for normalizing systems.)

Because of the restriction to normalizing systems, we need to prove normalization of $\beta\eta$-reduction without using the Church-Rosser property. This may look problematic but in practice it isn't. For example for the Calculus of Constructions, the strong normalization proof in [Geuvers and Nederhof 1991] for the system with $(\mathrm{conv}_\beta)$ can be adapted to a proof of strong normalization for the system with $(\mathrm{conv}_{\beta\eta})$. In that

paper a reduction preserving map from Term(CC) to Term(F$\omega$) is defined. This mapping can easily be extended to CC with (conv$_{\beta\eta}$) such that also $\eta$-reductions are preserved and all the required properties of the mapping stil hold (using the Key Lemma.) We conjecture here the general theorem that, if a PTS$_\beta$ is (strongly) normalizing, then the PTS$_{\beta\eta}$ is.

If we look at the Church-Rosser property from a point of view as to how to compute the common reduct, we see that the situation is really a bit more complicated then for untyped lambda calculus. In untyped lambda calculus, if $M \longrightarrow\!\!\!\!\!\rightarrow M_1$ and $M \longrightarrow\!\!\!\!\!\rightarrow M_2$, a common reduct of $M_1$ and $M_2$ can be found using complete developments. (See [Barendregt 1984].) Here one has to do something more, namely reducing the domains: Consider e.g. $M := \lambda x{:}A.(\lambda y{:}B.y)x$, $M_1 := \lambda x{:}A.x$ and $M_2 := \lambda y{:}B.y$. There are no residuals of the $\beta$-redex in $M_2$, nor are there any residuals of the $\eta$-redex in $M_1$, so we have a complete development of the set containing both redexes, but $M_1 \not\equiv M_2$. (They would have been in the untyped case.) We still have to unify $A$ and $B$.


## Acknowledgements

We would like to thank Anne Salvesen for pointing out some mistakes in an earlier version of this paper.


## References


[Barendregt 1984] H.P. Barendregt, *The lambda calculus: its syntax and semantics*, revised edition. Studies in Logic and the Foundations of Mathematics, North Holland.

[Barendregt 199+] H.P. Barendregt, Typed lambda calculi. In Abramski et al. (eds.), *Handbook of Logic in Computer Science*, Oxford Univ. Press, to appear.

[van Benthem Jutting 199+] L.S. van Benthem Jutting, Typing in Pure Type Systems. To appear in *Information and Computation*.

[Berardi 1988] S. Berardi, Towards a mathematical analysis of the Coquand-Huet calculus of constructions and the other systems in Barendregts cube. Dept. Computer Science, Carnegie-Mellon University and Dipartimento Matematica, Universita di Torino, Italy.

[Coquand 1991] Th. Coquand, An algorithm for testing conversion in type theory. In Huet and Plotkin (eds.), *Logical Frameworks*, Cambridge Univ. Press.

[van Daalen 1980] D.T. van Daalen, The language theory of AUTOMATH. Ph.D. thesis, Eindhoven Technological University, The Netherlands.

[Geuvers and Nederhof 1991] J.H. Geuvers and M.J. Nederhof, A modular proof of strong normalization for the calculus of constructions. *Journal of Functional Programming*, vol 1 (2), pp 155-189.

[Harper et al. 1987] R. Harper, F. Honsell and G. Plotkin, A framework for defining logics. *Proceedings Second Symposium on Logic in Computer Science*, (Ithaca, N.Y.), IEEE, Washington DC, pp 194-204.

[Jacobs 1991] B.P.F. Jacobs, Categorical type theory. Ph.D. thesis, University of Nijmegen, The Netherlands.

[Nederpelt 1973] R.P. Nederpelt, Strong normalization in a typed lambda calculus with lambda structured types. Ph.D. thesis, Eindhoven Technological University, The Netherlands.

[Pfenning 1991] F. Pfenning, Unification and anti-unification in the Calculus of Constructions. *Proceedings Sixth Symposium on Logic in Computer Science*, (Amsterdam, the Netherlands), IEEE, Washington DC, pp 74-85.

[Salvesen 1989] A. Salvesen, The Church-Rosser Theorem for LF with $\eta$ reduction. Notes of a talk presented at the BRA-Logical Frameworks meeting, Antibes 1990.

[Streicher 1989] Th. Streicher, Correctness and completeness of a categorical semantics of the Calculus of Constructions. Ph. D. thesis, University of Passau, Germany.