

Expanding the Cube

Gilles Barthe

Institutionen för Datavetenskap, Chalmers Tekniska Högskola, Göteborg, Sweden
Departamento de Informática, Universidade do Minho, Braga, Portugal
gilles@di.uminho.pt

Abstract. We prove strong normalization of β -reduction+ η -expansion for the Calculus of Constructions, thus providing the first strong normalization result for β -reduction+ η -expansion in calculi of dependent types and answering in the affirmative a conjecture by Di Cosmo and Ghani. In addition, we prove strong normalization of β -reduction+ η -expansion+algebraic reduction for the Algebraic Calculus of Constructions, which extends the Calculus of Constructions with first-order term-rewriting systems. The latter result, which requires the term-rewriting system to be non-duplicating, partially answers in the affirmative another conjecture by Di Cosmo and Ghani.

1 Introduction

Extensionality, as embodied in η -conversion

$$\lambda x.A. M x = M \quad \text{if } x \notin \text{FV}(M)$$

is a basic notion in λ -calculus and type theory. Traditionally, η -conversion has been oriented from left to right, thus leading to η -reduction. Recently, several authors have advocated a different computational interpretation, in which η -conversion is oriented from right to left, thus leading to η -expansion. The latter, which originates from proof-theoretical considerations [23], has found an increasing number of applications in computer science. The current body of results on η -expansion and its applications is too large to be presented here in any detail but we refer to [9] for a recent survey of the field, including a summary of the applications of η -expansion, an historical account of the subject and pointers to the literature.

The first part of this paper is concerned with normalization of β -reduction+ η -expansion for the λ -cube [4, 5, 14].¹ While weak normalization of β -reduction+ η -expansion is relatively easy to establish [6, 17], even for systems of dependent types, strong normalization of β -reduction+ η -expansion has remained unaddressed thus far. In fact, it was conjectured by Di Cosmo and Ghani [10, 17] that β -reduction+ η -expansion is strongly normalizing for the legal terms of the Calculus of Constructions. In the first part of the paper, we prove their conjecture by means of a model construction inspired from [25] and based on saturated sets.

The second part of this paper is concerned with normalization of β -reduction+ η -expansion+algebraic reduction for the algebraic λ -cube, see e.g. [3, 8]. In contrast to the λ -cube, neither weak normalization results nor strong normalization results are known for systems of dependent types. Again, it was conjectured by Di Cosmo and Ghani [10] that strong normalization is a modular property of the algebraic λ -cube, i.e. that a system of the algebraic λ -cube is strongly normalizing w.r.t. β -reduction+ η -expansion+algebraic reduction if its underlying term-rewriting system is strongly normalizing. In the second part of this paper, we solve their conjecture partially, under the extra assumption that the underlying term-rewriting system is non-duplicating. The proof is obtained by using ideas from [7] and modifying the model construction of the first part of the paper.

Related work

Much work has been devoted to extensionality in type systems so we shall only focus on systems of dependent types.

¹ Throughout the paper, we shall be concerned with the extensional versions of the λ -cube, in which the conversion rule uses $=_{\beta\eta}$. These versions are presented in [14] but differ from Barendregt's original presentation [4, 5], in which the conversion rule uses $=_{\beta}$. In order to avoid confusion, we refer to the latter presentation as the usual λ -cube, the usual Calculus of Constructions. . .

η -reduction The study of β -reduction+ η -reduction in dependent type theories dates back to the early 70's, with Nederpelt's thesis [21]. Nederpelt showed that β -reduction+ η -reduction is not confluent on the pseudo-terms of a dependently typed language *à la* Church, i.e. with typed λ -abstractions. Later, van Daalen [12] proved confluence of β -reduction+ η -reduction for (the typed terms of) a language of the Automath family. More recently, Geuvers [14] and Salvesen [24] proved confluence of β -reduction+ η -reduction for functional, normalizing Pure Type Systems [4, 5, 14]. Finally, the author [6] recently generalized Geuvers and Salvesen's results by showing that all Pure Type Systems have unique normal forms with respect to β -reduction+ η -reduction. As for strong normalization, Geuvers [14] seems to be the standard proof of strong normalization for the extensional version of the Calculus of Constructions.

$\beta\eta$ -long normal forms The existence of $\beta\eta$ -long normal forms for the systems of the λ -cube was first shown by Dowek, Huet and Werner [13], who defined a non-standard induction principle to this end. The induction principle was later simplified and generalized by the author [6] who showed the uniqueness of $\beta\eta$ -long normal forms for all Pure Type Systems and the existence of $\beta\eta$ -long normal forms for most Pure Type Systems of interest.

η -expansion for dependent types For systems of dependent types, the notion of η -expansion was first studied by Ghani [17], who showed that unrestricted η -expansion is not normalizing for the Calculus of Constructions, introduced a restricted notion of η -expansion, and showed that β -reduction+ η -expansion is confluent and weakly normalizing for the Calculus of Constructions. More recently, the author [6] generalized Ghani's results by showing that all Pure Type Systems have unique normal forms with respect to β -reduction+ η -expansion. As for strong normalization, Joachimski [18] has shown that a Pure Type System is strongly normalizing for β -reduction+ η -expansion if it is strongly normalizing for β -reduction. Unfortunately, Joachimski's notion of η -expansion is weaker than the notions of η -expansion that appear in the literature [6, 13, 17], and subsequently its underlying notion of $\beta\eta$ -long normal form does not correspond to the well-established notion of [6, 13, 17]. We consider it a severe drawback.

Finally, Di Cosmo and Ghani [10] have considered η -expansion in the context of the algebraic λ -cube, in particular for the Algebraic Calculus of Constructions, and have shown that confluence is a modular property of the Algebraic Calculus of Constructions.

Organization of the paper

The paper is organized as follows: in Section 2, we review the definition of the λ -cube. In Section 3, we prove strong normalization of β -reduction+ η -expansion for the systems of the λ -cube. In Section 4, we define the algebraic λ -cube with $\beta\eta$ -conversion and prove strong normalization of β -reduction+ η -expansion+algebraic reduction for the systems of the algebraic λ -cube. Finally, we conclude in Section 5.

Notation

We use standard notation and terminology from Abstract Rewriting Systems [19]. In particular, \rightarrow_{ij} denotes the union of two relations \rightarrow_i and \rightarrow_j , \rightarrow_i^+ denotes the transitive closure of \rightarrow_i , \twoheadrightarrow_i denotes the reflexive-transitive closure of \rightarrow_i and $=_i$ denotes the reflexive-symmetric-transitive closure of \rightarrow_i . Finally, the relation \downarrow_i is defined by $a \downarrow_i b$ if there exists c such that $a \twoheadrightarrow_i c$ and $b \twoheadrightarrow_i c$.

An object a is a *i -normal form* if there is no b such that $a \rightarrow_i b$; the set of i -normal forms is denoted by $\text{NF}(i)$. An object a is *i -normalizing* if there is some $b \in \text{NF}(i)$ such that $a \twoheadrightarrow_i b$; the set of i -normalizing objects is denoted by $\text{WN}(i)$. An object a is *i -strongly normalizing* if all reduction sequences starting from a are finite; the set of i -strongly normalizing objects is denoted by $\text{SN}(i)$.

2 η -expansion in the λ -cube

2.1 The λ -cube

Throughout this paper, we let $\mathcal{S} = \{*, \square\}$. Elements of \mathcal{S} are called *sorts*. For technical convenience, we distinguish between object variables and constructor variables. This distinction, which originates from [14], yields a neat formulation of the Classification Lemma (Lemma 7).

Definition 1. A rule set is a set \mathbf{S} such that $\mathbf{S} \subseteq \mathcal{S} \times \mathcal{S}$. Elements of sets are called rules.

Every rule set \mathbf{S} yields a Pure Type System $\lambda\mathbf{S}$ as specified below.

Definition 2 (Pure Type Systems of the λ -cube).

1. The set \mathcal{T} of pseudo-terms is given by the abstract syntax

$$\mathcal{T} = V^* \mid V^\square \mid * \mid \square \mid \mathcal{T}\mathcal{T} \mid \lambda V : \mathcal{T}.\mathcal{T} \mid \Pi V : \mathcal{T}.\mathcal{T}$$

where V^* and V^\square are fixed, pairwise disjoint and countably infinite sets of variables.

2. β -reduction \rightarrow_β is defined as the compatible closure of the contraction

$$(\lambda x.A. M) N \rightarrow_\beta M\{x := N\}$$

where $\{.\} := \{.\}$ is the standard substitution operator.

3. η -reduction \rightarrow_η is defined as the compatible closure of the contraction

$$\lambda x.A. (M x) \rightarrow_\eta M$$

provided $x \notin \text{FV}(M)$ where $\text{FV}(M)$ is the standard set of free variables of M .

4. The set \mathcal{G} of pseudo-contexts is given by the abstract syntax

$$\mathcal{G} = \langle \rangle \mid \mathcal{G}, V : \mathcal{T}$$

The domain of a context Γ is $\text{dom}(\Gamma) = \{x \mid \exists t \in \mathcal{T}. x : t \in \Gamma\}$.

5. A judgment is a triple $\Gamma \vdash M : A$ where $\Gamma \in \mathcal{G}$ and $M, A \in \mathcal{T}$.

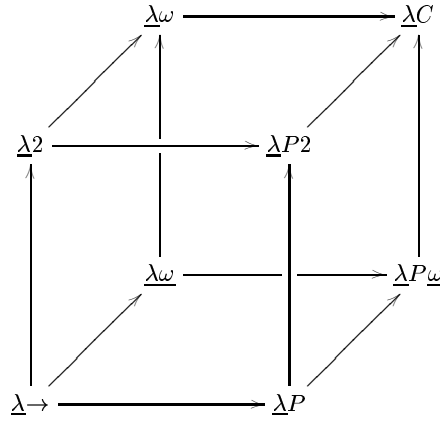
6. The derivability relation \vdash is given by the rules of Figure 1. If $\Gamma \vdash M : A$ is derivable, then Γ , M and A are legal. The set of legal contexts is denoted by \mathcal{H} .

| | | |
|---------------|---|--|
| (axiom) | $\langle \rangle \vdash * : \square$ | |
| (start) | $\frac{\Gamma \vdash A : s}{\Gamma, x : A \vdash x : A}$ | if $x \in V^s \setminus \text{dom}(\Gamma)$ |
| (weakening) | $\frac{\Gamma \vdash A : B \quad \Gamma \vdash C : s}{\Gamma, x : C \vdash A : B}$ | if $x \in V^s \setminus \text{dom}(\Gamma)$ and $A \in V \cup \mathcal{S}$ |
| (product) | $\frac{\Gamma \vdash A : s_1 \quad \Gamma, x : A \vdash B : s_2}{\Gamma \vdash (\Pi x : A. B) : s_2}$ | if $(s_1, s_2) \in \mathcal{R}$ |
| (application) | $\frac{\Gamma \vdash F : (\Pi x : A. B) \quad \Gamma \vdash a : A}{\Gamma \vdash F a : B\{x := a\}}$ | |
| (abstraction) | $\frac{\Gamma, x : A \vdash b : B \quad \Gamma \vdash (\Pi x : A. B) : s}{\Gamma \vdash \lambda x : A. b : \Pi x : A. B}$ | |
| (conversion) | $\frac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s}{\Gamma \vdash A : B'}$ | if $B =_{\beta\eta} B'$ |

Fig. 1. RULES FOR THE λ -CUBE

The eight Pure Type Systems $\lambda\mathbf{S}$ depicted in Figure 2 are collectively known as the λ -cube and relate to well-known systems:

| System | \mathbf{S} | Rule set |
|--|-----------------------|---|
| Simply typed λ -calculus | \rightarrow | $(*, *)$ |
| Polymorphic typed λ -calculus | 2 | $(*, *) \quad (\square, *)$ |
| Logical Frameworks | P | $(*, *) \quad (*, \square)$ |
| | $P2$ | $(*, *) \quad (\square, *) \quad (*, \square)$ |
| | $\underline{\omega}$ | $(*, *) \quad (\square, \square)$ |
| Higher-order typed λ -calculus | ω | $(*, *) \quad (\square, *) \quad (\square, \square)$ |
| | $P\underline{\omega}$ | $(*, *) \quad (*, \square) \quad (\square, \square)$ |
| Calculus of Constructions | $P\omega = C$ | $(*, *) \quad (\square, *) \quad (*, \square) \quad (\square, \square)$ |

Fig. 2. THE λ -CUBE

2.2 η -expansion

Defining η -expansion for systems of dependent types is not straightforward. As pointed out by Ghani [17], full η -expansion $\rightarrow_{\bar{\eta}_f(\Gamma)}$, as defined by the rule

$$M \rightarrow_{\bar{\eta}_f(\Gamma)} \lambda x.A. (M x) \quad \text{if } \Gamma \vdash M : \Pi x. A. B \text{ and } M \text{ not a } \lambda\text{-abstraction}$$

leads to infinite reduction sequences. Indeed, let $\Gamma = X : *, x : X \rightarrow X$ and let $B(y) = (\lambda z : X \rightarrow X. X) y$. Then

$$\begin{aligned} x &\rightarrow_{\bar{\eta}_f(\Gamma)} \lambda z : B(x). (x z) \\ &\rightarrow_{\bar{\eta}_f(\Gamma)} \lambda z : B(\lambda z : B(x). (x z)). (x z) \\ &\rightarrow_{\bar{\eta}_f(\Gamma)} \dots \end{aligned}$$

In order to rule out such infinite reduction sequences, Ghani [17] proposes to impose the extra constraint that A is in $\beta\eta$ -long normal form, where the notion of being $\beta\eta$ -long normal form is defined by induction on the structure of terms.

Definition 3 (Ghani [17]). Let $\Gamma \in \mathcal{H}$. A term $M \in \mathcal{N}$ is a $\beta\eta$ -long normal form in context Γ , written $\xi_\Gamma(M)$, if M is legal in Γ and one of the following conditions holds:

1. $M \in \mathcal{S}$;
2. $M = \Pi x. B. C$, $\xi_\Gamma(B)$ and $\xi_{\Gamma, x : B}(C)$;
3. $M = \lambda x. B. N$, $\xi_\Gamma(B)$ and $\xi_{\Gamma, x : B}(N)$;
4. $M = x P_1 \dots P_n$, $\Gamma \vdash M : A$ for some $A \in \mathcal{C} \cup \mathcal{S}$ and $\xi_\Gamma(P_i)$ for $i = 1, \dots, n$.

The set of $\beta\eta$ -long normal form in context Γ is denoted Ξ_Γ .

The above definition is also implicitly present in [13].

Definition 4 (Ghani [17]). Restricted η -expansion (in context Γ) $\rightarrow_{\bar{\eta}_r(\Gamma)}$ is defined by the rule

$$M \rightarrow_{\bar{\eta}_r(\Gamma)} \lambda x.A. (M x) \quad \text{if } \Gamma \vdash M : \Pi x. A. B \text{ and } M \text{ not a } \lambda\text{-abstraction and } \xi_\Gamma(A)$$

and by the compatibility clauses of Figure 3.

As argued in [6], this definition is somewhat contrived because:

1. one needs to define $\beta\eta$ -long normal forms before defining η -expansion;
2. one can only deduce that x is not in $\rightarrow_{\bar{\eta}(\Gamma, x : A \rightarrow A)}$ -normal form (where A is an arbitrary type) if one knows that $\xi_\Gamma(B)$ for some B such that $A =_{\beta\eta} B$.

Based on the above observations, the author [6] suggested that the definition of η -expansion may be simplified by requiring that $A \in \text{NF}(\beta)$. Note how Ghani's counter-example does not apply to such a notion of η -expansion.

Definition 5 ([6]). η -expansion $\rightarrow_{\bar{\eta}(\Gamma)}$ is defined by the rule

$$M \rightarrow_{\bar{\eta}(\Gamma)} \lambda x:A. (M x) \quad \text{if } \Gamma \vdash M : \Pi x:A. B \text{ and } M \text{ not a } \lambda\text{-abstraction and } A \in \text{NF}(\beta)$$

and by the compatibility clauses of Figure 3.

$$\begin{aligned} M \rightarrow_{\bar{\eta}(\Gamma)} N &\Rightarrow M P \rightarrow_{\bar{\eta}(\Gamma)} N P && \text{if } N \neq \lambda x:B. (M x) \\ M \rightarrow_{\bar{\eta}(\Gamma)} N &\Rightarrow P M \rightarrow_{\bar{\eta}(\Gamma)} P N \\ M \rightarrow_{\bar{\eta}(\Gamma)} N &\Rightarrow \lambda x:M. P \rightarrow_{\bar{\eta}(\Gamma)} \lambda x:N. P \\ M \rightarrow_{\bar{\eta}(\Gamma)} N &\Rightarrow \Pi x:M. P \rightarrow_{\bar{\eta}(\Gamma)} \Pi x:N. P \\ M \rightarrow_{\bar{\eta}(\Gamma, x:A)} N &\Rightarrow \lambda x:P. M \rightarrow_{\bar{\eta}(\Gamma)} \lambda x:P. N \\ M \rightarrow_{\bar{\eta}(\Gamma, x:A)} N &\Rightarrow \Pi x:P. M \rightarrow_{\bar{\eta}(\Gamma)} \Pi x:P. N \end{aligned}$$

Fig. 3. COMPATIBILITY RULES FOR η -EXPANSION

A few words are in order to justify our definition:

1. In the definition of η -expansion it is implicitly assumed that x is fresh;
2. η -expansion is taken to be a compatible relation as it is folklore that η -expanding the first argument of an application would lead to the loop

$$M N \rightarrow_{\bar{\eta}(\Gamma)} (\lambda x:A. M x) N \rightarrow_{\beta} M N$$

Similarly one cannot η -expand λ -abstractions as it would lead to the loop

$$\lambda x:A. M \rightarrow_{\bar{\eta}(\Gamma)} \lambda y:A. (\lambda x:A. M) y \rightarrow_{\beta} \lambda x:A. M$$

It is immediate to see that $\rightarrow_{\bar{\eta}}$ is a generalization of $\rightarrow_{\bar{\eta}_r}$.

Lemma 1. *If $M \rightarrow_{\bar{\eta}_r(\Gamma)} N$ then $M \rightarrow_{\bar{\eta}(\Gamma)} N$.*

3 Strong normalization of β -reduction+ η -expansion for the λ -cube

In this section, we prove that β -reduction+ η -expansion is strongly normalizing on the legal terms of the λ -cube.

Analysis Many proofs of strong normalization for the Calculus of Constructions (see e.g. [1, 15, 20, 25]) are achieved through the definition of a suitable model construction, in which types are interpreted as specific sets of strongly normalizing λ -terms, called *saturated sets*. Our proof proceeds in a similar way and is largely inspired from [25]. There are some minor differences however:

1. the notion of saturated set is modified so as to account for the peculiarities of β -reduction+ η -expansion, typically of not being substitutive;
2. in the proof of the soundness theorem, the induction step for the abstraction rule is modified so as to take into account the characterization of strongly normalizing terms w.r.t. β -reduction+ η -expansion.

The proof is not fully satisfactory, as one would hope to have a modular proof of strong normalization of β -reduction+ η -expansion for the λ -cube. In other words, one would like to reduce (in a weak system of arithmetic) strong normalization of β -reduction+ η -expansion for the λ -cube to strong normalization of β -reduction for the usual λ -cube (without having to prove the latter). One possibility, which needs to be further explored, is to extend the technique of simulating-expansions-without-expansions [11, 26] to the Calculus of Constructions. However, this is a challenging problem which remains unresolved so far.

Prerequisites The proof is self-contained, except that we make use of the following results.

Theorem 1 (Geuvers [14]). *For the systems of the λ -cube:*

1. $\rightarrow_{\beta\eta}$ is confluent on legal terms: $(\Gamma \vdash M : A \wedge \Gamma \vdash N : A \wedge M =_{\beta\eta} N) \Rightarrow M \downarrow_{\beta\eta} N$
2. $\rightarrow_{\beta\eta}$ enjoys subject reduction: $(\Gamma \vdash M : A \wedge M \rightarrow_{\beta\eta} N) \Rightarrow \Gamma \vdash N : A$

For the point of view of the proof, the most important consequence of confluence and subject reduction is that the systems of the λ -cube are *sound w.r.t. convertibility* [16], i.e. that every two convertible types or kinds are convertible through well-typed terms. Soundness w.r.t. convertibility is required to prove that the interpretation of types $\langle\langle \cdot \rangle\rangle$ preserves convertibility.

Theorem 2 ([6]). *For the systems of the λ -cube:*

1. $\rightarrow_{\bar{\eta}(\Gamma)}$ is strongly normalizing on Γ -legal terms;
2. $\rightarrow_{\bar{\eta}(\Gamma)}$ preserves β -normal forms.

It follows:

Lemma 2. *If $\Gamma \vdash M : A$ and $M \in \text{NF}(\beta)$ then $M \in \text{SN}(\beta\bar{\eta})$.*

Proof. By the above theorem, every reduction sequence starting from $M \in \text{NF}(\beta)$ is a $\bar{\eta}$ -reduction sequence and hence terminates.

Finally, we need the following characterization of strongly normalizing terms.

Lemma 3. *Assume $(\lambda x:A.M) N P_1 \dots P_n \in \text{SN}(\beta\bar{\eta}(\Gamma))$. If $A, N \in \text{SN}(\beta\bar{\eta}(\Gamma))$ and for every $N' \in \mathcal{T}$ such that $N \rightarrow_{\beta\bar{\eta}(\Gamma)} N'$, $M\{x := N'\} P_1 \dots P_n \in \text{SN}(\beta\bar{\eta}(\Gamma))$ then $(\lambda x:A.M) N P_1 \dots P_n \in \text{SN}(\beta\bar{\eta}(\Gamma))$.*

Proof. By an analysis of the possible reduction sequences starting from $(\lambda x:A.M) N P_1 \dots P_n$. Necessarily an infinite sequence will be of the form

$$\begin{aligned} (\lambda x:A.M) N P_1 \dots P_n &\rightarrow_{\beta\bar{\eta}} \lambda z_1 : B_1. \dots \lambda z_m : B_m. (\lambda x:A'.M') N' P'_1 \dots P'_n z'_1 \dots z'_m \\ &\rightarrow_{\beta\bar{\eta}} \lambda z_1 : B_1. \dots \lambda z_m : B_m. (M'\{x := N'\}) P'_1 \dots P'_n z'_1 \dots z'_m \\ &\rightarrow_{\beta\bar{\eta}} \dots \end{aligned}$$

with $B_1, \dots, B_m, z'_1, \dots, z'_m \in \text{NF}(\beta) \subseteq \text{SN}(\beta\bar{\eta}(\Gamma))$ and $M', N', P'_1, \dots, P'_n \in \text{SN}(\beta\bar{\eta}(\Gamma))$. By assumption $(M'\{x := N'\}) P_1 \dots P_n \in \text{SN}(\beta\bar{\eta}(\Gamma))$. Now

$$(M'\{x := N'\}) P_1 \dots P_n \rightarrow_{\beta\bar{\eta}} \lambda z_1 : B_1. \dots \lambda z_n : B_n. (M'\{x := N'\}) P'_1 \dots P'_n z'_1 \dots z'_n$$

so we conclude that the reduction sequence cannot be infinite, a contradiction.

Lemma 3 is used crucially in proving the soundness of the model construction, more specifically in the case of the (abstraction) rule.

3.1 Environments

The notion of environment conveniently captures the notion of infinite context.

Definition 6 (Pottinger [22]). *An environment \mathcal{E} is an infinite sequence of variable declarations $\mathcal{E} = x_1 : E_1, x_2 : E_2, \dots$ such that for every $i > 0$*

- $\mathcal{E}^i = x_1 : E_1, \dots, x_i : E_i$ is a legal context;
- if $\mathcal{E}^i \vdash A : s$ and $s \in S$ then there exists infinitely many k s.t. $E_k = A$.

We write $\mathcal{E} \vdash M : A$ whenever there exists $i \geq 0$ s.t. $\mathcal{E}^i \vdash M : A$.

Note that the definition of environment implicitly embeds our variable convention, i.e. for every $(x : A) \in \mathcal{E}$, $\mathcal{E} \vdash A : s$ iff $x \in V^s$.

Lemma 4 (Pottinger [22]). *Every context can be extended to an environment.*

Environments are convenient for the purpose of strong normalization proofs, because (up to convertibility) an expression has at most one type in an environment and because η -expansion is stable under thinning, i.e. for $i \leq j$, we have $M \rightarrow_{\bar{\eta}(\mathcal{E}^i)} N \Rightarrow M \rightarrow_{\bar{\eta}(\mathcal{E}^j)} N$. In the sequel, we therefore omit \mathcal{E}^i and write $M \rightarrow_{\bar{\eta}} N$ when there is no risk of confusion.

3.2 Saturated sets

Throughout this section, we let \mathcal{E} be a fixed environment. We say A is an \mathcal{E} -type (or simply a type) if $\mathcal{E} \vdash A : s$ with $s \in \mathcal{S}$. Moreover, we let $\text{SN}(A)$ denote the set of $\beta\bar{\eta}$ -strongly normalizing terms of type A in \mathcal{E} . Finally we let $V^A = \{x \in V \mid \mathcal{E} \vdash x : A\}$.

Definition 7 (Base terms). *The set $\text{Base}(A)$ of A -base terms is defined inductively as follows:*

1. $V^A \subseteq \text{Base}(A)$,
2. if $a \in \text{Base}(\Pi x.B.A)$ and $b \in \text{SN}(B)$, then $ab \in \text{Base}(A\{x := b\})$.

Key reduction, as defined below, is a specific strategy reducing terms to weak-head normal form.

Definition 8 (Key reduction). *Key-reduction \rightarrow_k is the smallest reduction relation satisfying*

$$(\lambda x.B. a) b c_1 \dots c_n \rightarrow_k a\{x := b\} c_1 \dots c_n$$

Note that we do not take the compatible closure of the above rule.

Note that, for every $b \in \mathcal{T}$ there exists at most one $b' \in \mathcal{T}$ such that $b \rightarrow_k b'$.

Definition 9 (Saturated set). *Let A be a type. A set $X \subseteq \text{SN}(A)$ is A -saturated if $\text{Base}(A) \subseteq X$ and X is closed under the rules:*

$$\frac{b \in X \quad b \rightarrow_{\beta\bar{\eta}} b'}{b' \in X} \qquad \frac{b' \in X \quad b \in \text{SN}(A) \quad b \rightarrow_k b'}{b \in X}$$

The collection of A -saturated sets is denoted by $\mathbf{SAT}(A)$.

The extra clause in the definition of saturated sets is due to Lemma 3.² In order for the definition of saturated sets to make sense, we need to prove base terms are strongly normalizing; otherwise there would not be any saturated set.

Lemma 5. *For every type A , $\text{Base}(A) \subseteq \text{SN}(A)$.*

Proof. Similar to that of Lemma 3.

We conclude this subsection by stating some fundamental closure properties of saturated sets.

Lemma 6.

1. $\text{SN}(A) \in \mathbf{SAT}(A)$.
2. If $X_i \in \mathbf{SAT}(A)$ for every $i \in I$ and $I \neq \emptyset$ then $\bigcap_{i \in I} X_i \in \mathbf{SAT}(A)$.
3. If $X \in \mathbf{SAT}(A)$ and for every $x \in X$, $Y_x \in \mathbf{SAT}(B)$ then $\prod x \in X. Y_x \in \mathbf{SAT}(\Pi x.A. B)$, where

$$\prod x \in X. Y_x = \{M \in \text{SN}(\Pi x.A. B) \mid \forall N \in X. M N \in Y_N\}$$

3.3 The Classification Lemma

It is convenient to base the model construction on a classification lemma, which stratifies the different classes of pseudo-terms. The formulation used below is a variant of Geuvers' Classification Lemma, see e.g. [14].

Definition 10 (Pseudo-objects, pseudo-constructors and pseudo-kinds). *The classes \mathcal{O} , \mathcal{C} and \mathcal{K} of pseudo-objects, pseudo-constructors and pseudo-kinds are given by the abstract syntaxes*

$$\begin{aligned} \mathcal{O} &= V^* \mid \lambda V^*. \mathcal{C}. \mathcal{O} \mid \lambda V^\square. \mathcal{K}. \mathcal{O} \mid \mathcal{O} \mathcal{O} \mid \mathcal{O} \mathcal{C} \\ \mathcal{C} &= V^\square \mid \Pi V. \mathcal{C}. \mathcal{C} \mid \Pi V. \mathcal{K}. \mathcal{C} \mid \lambda V^*. \mathcal{C}. \mathcal{C} \mid \lambda V^\square. \mathcal{K}. \mathcal{C} \mid \mathcal{C} \mathcal{C} \mid \mathcal{C} \mathcal{O} \\ \mathcal{K} &= * \mid \Pi V. \mathcal{C}. \mathcal{K} \mid \Pi V. \mathcal{K}. \mathcal{K} \end{aligned}$$

We write $M \sim N$ if $M \in \mathcal{D}$ and $N \in \mathcal{D}$ for some $\mathcal{D} \in \{\mathcal{O}, \mathcal{C}, \mathcal{K}\}$.

We have:

Lemma 7 (Classification Lemma). *If $\mathcal{E} \vdash M : A$ then exactly one of the three conditions hold: (1) $M \in \mathcal{O}$ and $A \in \mathcal{C}$ (2) $M \in \mathcal{C}$ and $A \in \mathcal{K}$ (3) $M \in \mathcal{K}$ and $A = \square$.*

² Compare with the similar result for β -reduction: if $A, N, M\{x := N\} P_1 \dots P_n \in \text{SN}(\beta)$ then $(\lambda x : A. M) N P_1 \dots P_n \in \text{SN}(\beta)$. Here we need to consider all reducts of N and hence the first clause.

3.4 The model construction

The first step of the model construction is to define an interpretation which assigns to every pseudo-term the set of possible interpretations for its inhabitants. Below we let $\mathbf{1}$ denote an arbitrary one element set and write $B : K$ instead of $B \in \{M \in \mathcal{T} \mid \mathcal{E} \vdash M : K\}$.

Definition 11 (Possible values). *The map $a : \mathcal{T} \rightarrow \mathbf{Set}$ and the relation $\dot{=}_A \subseteq a(A) \times a(A)$ (where $A \in \mathcal{T}$) are defined by*

$$a(A) = \begin{cases} \mathbf{SAT}(A) & \text{if } \mathcal{E} \vdash A : * \text{ or } \mathcal{E} \vdash A : \square \\ \{(f_B)_{B:K_1} \mid \forall B : K_1. f_B \in a(B) \rightarrow a(A \ B) \\ \wedge (\forall B, B' : K_1. B =_{\beta\eta} B' \Rightarrow f_B =_A f_{B'})\} & \text{if } \mathcal{E} \vdash A : \Pi\alpha: K_1. K_2 \text{ and } K_2 \in \mathcal{K} \\ \{\mathbf{1}\} & \text{otherwise} \end{cases}$$

and

$$x \dot{=}_A y = \begin{cases} x = y & \text{if } \mathcal{E} \vdash A : * \text{ or } \mathcal{E} \vdash A : \square \\ \forall B : K_1. \forall z \in a(B). x_B z \dot{=}_{K_2\{x:=B\}} y_B z & \text{if } \mathcal{E} \vdash A : \Pi x: K_1. K_2 \text{ and } K_2 \in \mathcal{K} \\ \text{true} & \text{otherwise} \end{cases}$$

In the sequel, we let $\mathbf{A} = \bigcup_{M \in \mathcal{T}} a(M)$.

The interpretation preserves convertibility.

Lemma 8. *If A and B are legal terms in \mathcal{E} and $A =_{\beta\eta} B$ then $a(A) = a(B)$.*

Proof. By induction on the type of A and B .

The next interpretation maps types and kinds to saturated sets.

Definition 12.

1. A valuation is a pair (ρ, ζ) with $\rho : V \rightarrow \mathcal{T}$ and $\zeta : V \rightarrow \mathbf{A}$.
2. The extension $\langle \cdot \rangle_\rho : \mathcal{T} \rightarrow \mathbf{Set}$ of ρ is defined as the unique capture-avoiding substitution extending ρ .
3. The extension $\langle \langle \cdot \rangle \rangle_{(\rho, \zeta)} : \mathcal{T} \rightarrow \mathbf{A}$ of ζ is defined as follows:

$$\begin{aligned} \langle \langle x \rangle \rangle_{(\rho, \zeta)} &= \zeta(x) && \text{if } x \in V \\ \langle \langle s \rangle \rangle_{(\rho, \zeta)} &= \mathbf{SN}(s) && \text{if } s \in \mathcal{S} \\ \langle \langle \Pi x : A. B \rangle \rangle_{(\rho, \zeta)} &= \prod P \in \langle \langle A \rangle \rangle_{(\rho, \zeta)}. \mathbf{int}_{(\rho, \zeta)}(B, P) \\ \langle \langle M \ N \rangle \rangle_{(\rho, \zeta)} &= (\langle \langle M \rangle \rangle_{(\rho, \zeta)}) \langle \langle N \rangle \rangle_{(\rho, \zeta)} && \text{if } \langle \langle MN \rangle \rangle_\rho \in \mathcal{C} \\ \langle \langle \lambda x : A. M \rangle \rangle_{(\rho, \zeta)} &= (\mathbf{fun}_{(\rho, \zeta)}(M, P))_{P: \langle \langle A \rangle \rangle_\rho} && \text{if } \langle \langle \lambda x : A. M \rangle \rangle_\rho \in \mathcal{C} \\ \langle \langle M \rangle \rangle_{(\rho, \zeta)} &= \mathbf{1} && \text{otherwise} \end{aligned}$$

where

$$\begin{aligned} \mathbf{int}_{(\rho, \zeta)}(B, P) &= \bigcap_{c \in a(P)} \langle \langle B \rangle \rangle_{(\rho(x:=P), \zeta(x:=c))} \\ \mathbf{fun}_{(\rho, \zeta)}(M, P) &= \lambda c \in a(P). \langle \langle M \rangle \rangle_{(\rho(x:=P), \zeta(x:=c))} \end{aligned}$$

and λ denotes the set-theoretic abstraction.

In order to prove strong normalization one first needs to show that, under suitable conditions, valuations preserve typability and that the $\langle \langle \cdot \rangle \rangle$ -interpretation of a term is an element of the possible values of its $\langle \cdot \rangle$ -interpretation.

Definition 13.

1. A valuation (ρ, ζ) satisfies $M : A$, written $(\rho, \zeta) \models M : A$ if
 - (a) $\mathcal{E} \vdash \langle \langle M \rangle \rangle_\rho : \langle \langle A \rangle \rangle_\rho$
 - (b) $\langle \langle M \rangle \rangle_{(\rho, \zeta)} \in a(\langle \langle M \rangle \rangle_\rho)$.

2. A valuation (ρ, ζ) satisfies Γ , written $(\rho, \zeta) \models \Gamma$, where Γ is a pseudo-context, if $(\rho, \zeta) \models x : A$ for every $(x : A) \in \Gamma$.
3. A judgment $\Gamma \vdash M : A$ is valid, written $\Gamma \models M : A$, if $(\rho, \zeta) \models M : A$ for every valuation (ρ, ζ) s.t. $(\rho, \zeta) \models \Gamma$.

We need to start with a few technical lemmas. First of all, we show that the interpretation is well-behaved w.r.t. substitution.

Lemma 9. *Assume that $\Gamma_1, x : B, \Gamma_2 \vdash M : A$ and $\Gamma_1 \vdash N : B$. If $(\rho, \zeta) \models \Gamma_1, x : B, \Gamma_2$ then $\llbracket M \{x := N\} \rrbracket_{(\rho, \zeta)} = \llbracket M \rrbracket_{(\rho(x := \llbracket N \rrbracket_\rho), \zeta(x := \llbracket N \rrbracket_{(\rho, \zeta)}))}$.*

Proof. By induction on the structure of M .

We also need to show that valuations can always be extended in such a way that satisfaction is preserved.

Lemma 10. *Let $\Gamma, x : A$ be a legal context and let (ρ, ζ) be a valuation s.t. $(\rho, \zeta) \models \Gamma, A : s$.*

1. $(\rho(x := N), \zeta(x := c)) \models \Gamma, x : A$ for every $N \in \mathcal{T}$ s.t. $\mathcal{E} \vdash N : \llbracket A \rrbracket_\rho$ and $c \in a(N)$.
2. There exists $z \in V$ s.t. $(\rho(x := z), \zeta(x := c)) \models \Gamma, x : A$ for every $c \in a(z)$.

Proof. By definition of \models .

The interpretation is preserved under conversion.

Lemma 11.

1. Assume $\Gamma \vdash M, N : A$ and $M =_{\beta_\eta} N$. Then $\llbracket M \rrbracket_{(\rho, \zeta)} = \llbracket N \rrbracket_{(\rho, \zeta)}$ for every valuation (ρ, ζ) such that $(\rho, \zeta) \models \Gamma$.
2. Assume $\Gamma \vdash M : A$. Then $\llbracket M \rrbracket_{(\rho, \zeta)} = \llbracket M \rrbracket_{(\rho', \zeta)}$ for every valuations (ρ, ζ) and (ρ', ζ) such that $(\rho, \zeta) \models \Gamma$, $(\rho', \zeta) \models \Gamma$ and $\rho =_{\beta_\eta} \rho'$.³

Proof. (1) By confluence and subject reduction of \rightarrow_{β_η} , it is sufficient to consider the case where $M \rightarrow_{\beta_\eta} N$. The proof proceeds by induction on the structure of M . (2) Similarly, it is sufficient to consider the case where $\rho \rightarrow_{\beta_\eta} \rho'$. The proof proceeds by induction on the structure of M .

We can now prove the following result.

Proposition 1. *If $\Gamma \vdash M : A$ then $\Gamma \models M : A$.*

Proof. By induction on the structure of derivations. We treat three cases:

1. assume that $\Gamma \vdash M : A$ is derived using (product). Then $A = s_2$ and $M = \Pi x : B. C$. Moreover the last rule of the derivation is of the form

$$\frac{\Gamma \vdash B : s_1 \quad \Gamma, x : B \vdash C : s_2}{\Gamma \vdash \Pi x : B. C : s_2}$$

Assume $(\rho, \zeta) \models \Gamma$. To show $(\rho, \zeta) \models \Pi x : B. C : s_2$.

- (a) To prove $\mathcal{E} \vdash \llbracket \Pi x : B. C \rrbracket_\rho : s_2$. By induction hypothesis, $(\rho, \zeta) \models B : s_1$. In particular $\mathcal{E} \vdash \llbracket B \rrbracket_\rho : s_1$. Moreover there exists by Lemma 10 a variable z s.t. for every $c \in a(z)$,

$$(\rho(x := z), \zeta(x := c)) \models \Gamma, x : B$$

By induction hypothesis, $\mathcal{E} \vdash \llbracket C \rrbracket_{\rho(x := z)} : s_2$. We conclude by (product).

³ \rightarrow_{β_η} is extended to valuations in the obvious way, i.e. $\rho \rightarrow_{\beta_\eta} \rho'$ if there exists $x \in V$ such that $\rho(x) \rightarrow_{\beta_\eta} \rho'(x)$ and $\rho(y) = \rho'(y)$ if $x \neq y$. Then $=_{\beta_\eta}$ is the reflexive-symmetric-transitive closure of \rightarrow_{β_η} .

(b) To prove $\langle\langle \Pi x : B. C \rangle\rangle_{(\rho, \zeta)} \in a(\langle\langle \Pi x : B. C \rangle\rangle_{\rho})$. Note that it is enough to show that

$$\langle\langle \Pi x : B. C \rangle\rangle_{(\rho, \zeta)} \in \mathbf{SAT}(\langle\langle \Pi x : B. C \rangle\rangle_{\rho})$$

By induction hypothesis, $\langle\langle B \rangle\rangle_{(\rho, \zeta)} \in \mathbf{SAT}(\langle\langle B \rangle\rangle_{\rho})$ and

$$\forall N \in \mathcal{T}. \forall c \in a(N). (\rho(x := N), \zeta(x := c)) \models \Gamma, x : B \Rightarrow \langle\langle C \rangle\rangle_{(\rho(x := N), \zeta(x := c))} \in \mathbf{SAT}(\langle\langle C \rangle\rangle_{\rho(x := N)})$$

By Lemma 10, $\forall N \in \mathcal{T}. \forall c \in a(N). \mathcal{E} \vdash N : \langle\langle B \rangle\rangle_{\rho} \Rightarrow (\rho(x := N), \zeta(x := c)) \models \Gamma, x : B$ and hence $\forall N \in \mathcal{T}. \forall c \in a(N). \mathcal{E} \vdash N : \langle\langle B \rangle\rangle_{\rho} \Rightarrow \langle\langle C \rangle\rangle_{(\rho(x := N), \zeta(x := c))} \in \mathbf{SAT}(\langle\langle C \rangle\rangle_{\rho(x := N)})$. By part 2 of Lemma 6,

$$\forall N \in \mathcal{T}. \mathcal{E} \vdash N : \langle\langle B \rangle\rangle_{\rho} \Rightarrow \bigcap_{c \in a(N)} \langle\langle C \rangle\rangle_{\zeta(x := c)\rho(x := N)} \in \mathbf{SAT}(\langle\langle C \rangle\rangle_{\rho(x := N)})$$

We conclude by part 3 of Lemma 6.

2. assume that $\Gamma \vdash M : A$ is derived using (application). Then $M = M_1 M_2$, $A = C\{x := M_2\}$ and the last rule of the derivation is

$$\frac{\Gamma \vdash M_1 : (\Pi x : B. C) \quad \Gamma \vdash M_2 : B}{\Gamma \vdash M_1 M_2 : C\{x := M_2\}}$$

Assume $(\rho, \zeta) \models \Gamma$. To show $(\rho, \zeta) \models M_1 M_2 : C\{x := M_2\}$.

(a) to prove $\mathcal{E} \vdash \langle\langle M_1 M_2 \rangle\rangle_{\rho} : \langle\langle C\{x := M_2\} \rangle\rangle_{\rho}$. It follows directly from the induction hypothesis.

(b) to prove that $\langle\langle M_1 M_2 \rangle\rangle_{(\rho, \zeta)} \in a(\langle\langle M_1 M_2 \rangle\rangle_{\rho})$. There are two cases to distinguish:

i. If $\langle\langle M_1 M_2 \rangle\rangle_{\rho} \notin \mathcal{C}$, then $a(\langle\langle M_1 M_2 \rangle\rangle_{\rho}) = \{\mathbf{1}\}$ and $\langle\langle M_1 M_2 \rangle\rangle_{(\rho, \zeta)} = \mathbf{1}$, so we are done.

ii. If $\langle\langle M \rangle\rangle_{\rho} \in \mathcal{C}$ then by induction hypothesis, $\langle\langle M_1 \rangle\rangle_{(\rho, \zeta)} \in a(\langle\langle M_1 \rangle\rangle_{\rho})$ and $\langle\langle M_2 \rangle\rangle_{(\rho, \zeta)} \in a(\langle\langle M_2 \rangle\rangle_{\rho})$.

Now $\langle\langle M_1 \rangle\rangle_{\rho} \in \mathcal{C}$ and hence $(\langle\langle M_1 \rangle\rangle_{(\rho, \zeta)})_{\langle\langle M_2 \rangle\rangle_{\rho}} \langle\langle M_2 \rangle\rangle_{(\rho, \zeta)} \in a(\langle\langle M_1 M_2 \rangle\rangle_{\rho})$ and we are done.

3. assume that $\Gamma \vdash M : A$ is derived using (abstraction). Necessarily $M = \lambda x : B. N$, $A = \Pi x : B. C$ and the last rule of the derivation is

$$\frac{\Gamma, x : B \vdash N : C \quad \Gamma \vdash (\Pi x : B. C) : s}{\Gamma \vdash \lambda x : B. N : \Pi x : B. C}$$

Assume $(\rho, \zeta) \models \Gamma$. To show $(\rho, \zeta) \models \lambda x : B. N : \Pi x : B. C$.

(a) To prove $\mathcal{E} \vdash \langle\langle \lambda x : B. N \rangle\rangle_{\rho} : \langle\langle \Pi x : B. C \rangle\rangle_{\rho}$. By Lemma 10, there exists $z \in V$ s.t. for every $c \in a(z)$,

$$(\rho(x := z), \zeta(x := c)) \models \Gamma, x : B$$

By induction hypothesis,

$$\begin{aligned} \mathcal{E} &\vdash \langle\langle N \rangle\rangle_{\rho(x := z)} : \langle\langle C \rangle\rangle_{\rho(x := z)} \\ \mathcal{E} &\vdash \langle\langle \Pi x : B. C \rangle\rangle_{\rho} : s \end{aligned}$$

We conclude by (abstraction).

(b) To prove that $\langle\langle \lambda x : B. N \rangle\rangle_{(\rho, \zeta)} \in a(\langle\langle \lambda x : B. N \rangle\rangle_{\rho})$. There are two cases to distinguish:

i. If $\langle\langle \lambda x : B. N \rangle\rangle_{\rho} \notin \mathcal{C}$, then $a(\langle\langle \lambda x : B. N \rangle\rangle_{\rho}) = \{\mathbf{1}\}$ and $\langle\langle \lambda x : B. N \rangle\rangle_{(\rho, \zeta)} = \mathbf{1}$, so we are done.

ii. If $\langle\langle \lambda x : B. N \rangle\rangle_{\rho} \in \mathcal{C}$, then we have to show that for every P such that $\mathcal{E} \vdash P : \langle\langle B \rangle\rangle_{\rho}$:

(α) $(\langle\langle \lambda x : B. N \rangle\rangle_{(\rho, \zeta)})_P Q \in a(\langle\langle \lambda x : B. N \rangle\rangle_{\rho} P)$ for every $Q \in a(P)$;

(β) $(\langle\langle \lambda x : B. N \rangle\rangle_{(\rho, \zeta)})_P = (\langle\langle \lambda x : B. N \rangle\rangle_{(\rho, \zeta)})_{P'}$ for every $P' \in \mathcal{T}$ legal in \mathcal{E} and s.t. $P =_{\beta\eta} P'$.

By definition, $(\langle\langle \lambda x : B. N \rangle\rangle_{(\rho, \zeta)})_P Q = \langle\langle N \rangle\rangle_{(\rho(x := P), \zeta(x := Q))}$. We conclude the proof of (α) by Lemma 10 and induction hypothesis. As for (β), we conclude by Lemma 11.

We now turn to soundness, which states that under suitable conditions, $\langle\langle M \rangle\rangle_{\rho} \in \langle\langle A \rangle\rangle_{(\rho, \zeta)}$ whenever $\Gamma \vdash M : A$. We begin with some definitions and preliminary results.

Definition 14.

1. A valuation (ρ, ζ) semantically entails $M : A$, written $(\rho, \zeta) \models^s M : A$, if $(\rho, \zeta) \models M : A$ and $\langle\langle M \rangle\rangle_{\rho} \in a(\langle\langle A \rangle\rangle_{(\rho, \zeta)})$.

2. A valuation (ρ, ζ) semantically entails Γ , written $(\rho, \zeta) \models^s \Gamma$, where Γ is a pseudo-context, if $(\rho, \zeta) \models^s x : A$ for every $(x : A) \in \Gamma$.
3. A judgment $\Gamma \vdash M : A$ is sound, written $\Gamma \models M : A$, if $(\rho, \zeta) \models^s M : A$ for every valuation (ρ, ζ) s.t. $(\rho, \zeta) \models^s \Gamma$.

The following lemma shows that valuations can always be extended in such a way that satisfaction is preserved.

Lemma 12. *Let $\Gamma, x : A$ be a legal context and let (ρ, ζ) be a valuation s.t. $(\rho, \zeta) \models^s \Gamma, A : s$.*

1. $(\rho(x := N), \zeta(x := c)) \models^s \Gamma, x : A$ for every $N \in \mathcal{T}$ s.t. $\mathcal{E} \vdash N : \langle A \rangle_\rho$ and $c \in a(N)$ and $N \in \langle\langle A \rangle\rangle_{(\rho, \zeta)}$.
2. There exists $z \in V$ s.t. $(\rho(x := z), \zeta(x := c)) \models^s \Gamma, x : A$ for every $c \in a(z)$.

Proof. By definition of \models^s .

Finally, we prove that the model construction is sound.

Proposition 2 (Soundness). *If $\Gamma \vdash M : A$ then $\Gamma \models^s M : A$.*

Proof. Note that we only have to show that $\langle M \rangle_\rho \in \langle\langle A \rangle\rangle_{(\rho, \zeta)}$ whenever $(\zeta, \rho) \models^s \Gamma$. We treat three cases:

1. assume that $\Gamma \vdash M : A$ is derived using (product). Then $A = s_2$ and $M = \Pi x : B. C$. Moreover the last rule of the derivation is of the form

$$\frac{\Gamma \vdash B : s_1 \quad \Gamma, x : B \vdash C : s_2}{\Gamma \vdash \Pi x : B. C : s_2}$$

Assume $(\rho, \zeta) \models^s \Gamma$. To show $\langle \Pi x : B. C \rangle_\rho \in \langle\langle s_2 \rangle\rangle_{(\rho, \zeta)}$. Note that $\langle\langle s_2 \rangle\rangle_{(\rho, \zeta)} = \text{SN}(s_2)$ so we need to prove that $\langle \Pi x : B. C \rangle_\rho$ is strongly normalizing. By induction hypothesis, both $\langle B \rangle_\rho$ and $\langle C \rangle_{\rho'}$ (where $(\rho', \zeta') \models^s \Gamma, x : B$) are strongly normalizing. By Lemma 12, there exists $z \in V$ and $c \in a(z)$ s.t. $(\rho(x := z), \zeta(x := c)) \models^s \Gamma, x : B$. Hence $\langle C \rangle_{\rho(x := z)}$ is strongly normalizing. Thus $\langle \Pi x : B. C \rangle_\rho$ is strongly normalizing.

2. assume that $\Gamma \vdash M : A$ is derived using (application). Then $M = M_1 M_2$, $A = C\{x := M_2\}$ and the last rule of the derivation is

$$\frac{\Gamma \vdash M_1 : (\Pi x : B. C) \quad \Gamma \vdash M_2 : B}{\Gamma \vdash M_1 M_2 : C\{x := M_2\}}$$

Assume $(\rho, \zeta) \models^s \Gamma$. To show $\langle M_1 M_2 \rangle_\rho \in \langle\langle C\{x := M_2\} \rangle\rangle_{(\rho, \zeta)}$. By induction hypothesis,

$$\langle M_1 \rangle_\rho \in \langle\langle \Pi x : B. C \rangle\rangle_{(\rho, \zeta)} \quad \langle M_2 \rangle_\rho \in \langle\langle B \rangle\rangle_{(\rho, \zeta)} \quad \langle M_2 \rangle_\rho \in a(\langle M_2 \rangle_\rho)$$

By definition of $\langle\langle \Pi x : B. C \rangle\rangle_{(\rho, \zeta)}$, $\langle M_1 M_2 \rangle_\rho \in \bigcap_{c \in a(\langle M_2 \rangle_\rho)} \langle\langle C \rangle\rangle_{(\rho(x := \langle M_2 \rangle_\rho), \zeta(x := c))}$. A fortiori

$$\langle M_1 M_2 \rangle_\rho \in \langle\langle C \rangle\rangle_{(\rho(x := \langle M_2 \rangle_\rho), \zeta(x := \langle M_2 \rangle_{(\rho, \zeta)}))}$$

By Lemma 9, $\langle M_1 M_2 \rangle_\rho \in \langle\langle C\{x := M_2\} \rangle\rangle_{(\rho, \zeta)}$.

3. assume that $\Gamma \vdash M : A$ is derived using (abstraction). Necessarily $M = \lambda x : B. N$, $A = \Pi x : B. C$ and the last rule of the derivation is

$$\frac{\Gamma, x : B \vdash N : C \quad \Gamma \vdash (\Pi x : B. C) : s}{\Gamma \vdash \lambda x : B. N : \Pi x : B. C}$$

Assume $(\rho, \zeta) \models^s \Gamma$. To show that $\langle \lambda x : B. N \rangle_\rho \in \langle\langle \Pi x : B. C \rangle\rangle_{(\rho, \zeta)}$, or by definition of $\langle\langle \Pi x : B. C \rangle\rangle_{(\rho, \zeta)}$ that

$$\forall P \in \langle\langle B \rangle\rangle_{(\rho, \zeta)}. \langle \lambda x : B. N \rangle_\rho P \in \bigcap_{c \in a(P)} \langle\langle B \rangle\rangle_{(\rho(x := P), \zeta(x := c))} \quad (*)$$

By induction hypothesis and Lemma 12, $\langle\langle B \rangle\rangle_{(\rho(x := P), \zeta(x := c))}$ is a saturated set whenever $P \in \langle\langle B \rangle\rangle_{(\rho, \zeta)}$ and $c \in a(P)$. By definition of saturated sets, (*) follows from:

- (a) for every $P \in \langle\langle B \rangle\rangle_{(\rho, \zeta)}$, $\langle \lambda x : B. N \rangle_\rho P \in \text{SN}(\langle \Pi x : B. C \rangle_\rho)$;

(b) for every $P \in \langle\langle B \rangle\rangle_{(\rho, \zeta)}$, $\langle\langle N \rangle\rangle_{\rho(x:=P)} \in \bigcap_{c \in \alpha(P)} \langle\langle B \rangle\rangle_{(\rho(x:=P), \zeta(x:=c))}$.

The latter is a direct consequence of the induction hypothesis and part 2 of Lemma 6. As for (a), we need to prove by Lemma 3 that $\langle\langle B \rangle\rangle_{\rho}$ is strongly normalizing and that $\langle\langle N \rangle\rangle_{\rho(x:=P')}$ for every P' such that $P \rightarrow_{\beta\bar{\eta}} P'$. As $P' \in \langle\langle B \rangle\rangle_{(\rho, \zeta)}$ for $P \rightarrow_{\beta\bar{\eta}} P'$, both follow from the induction hypothesis.

It follows that legal terms of the systems of the λ -cube are strongly normalizing w.r.t. β -reduction+ η -expansion.

Theorem 3. *For every system of the λ -cube: $\Gamma \vdash M : A \Rightarrow M \in \text{SN}(\beta\bar{\eta}(\Gamma))$*

Proof. Without loss of generality, we can assume $\Gamma = \langle \rangle$, as

$$\begin{aligned} x_1 : A_1, \dots, x_n : A_n \vdash M : B &\Rightarrow \langle \rangle \vdash \lambda x_1 : A_1. \dots \lambda x_n : A_n. M : \Pi x_1 : A_1. \dots \Pi x_n : A_n. M \\ \lambda x_1 : A_1. \dots \lambda x_n : A_n. M \in \text{SN}(\beta\bar{\eta}) &\Rightarrow M \in \text{SN}(\beta\bar{\eta}) \end{aligned}$$

Then for every valuation (ρ, ξ) , $\langle\langle M \rangle\rangle_{\rho} \in \langle\langle A \rangle\rangle_{(\rho, \xi)} \in \text{SAT}(\langle\langle A \rangle\rangle_{\rho})$. In particular, $\langle\langle M \rangle\rangle_{\rho} \in \text{SN}(\beta\bar{\eta})$. But $\langle\langle M \rangle\rangle_{\rho} = M$ so we are done.

4 Strong normalization of the algebraic λ -cube

In this section, we introduce the algebraic λ -cube and show that strong normalization is a modular property of the algebraic λ -cube, provided the underlying term-rewriting system is non-duplicating.

4.1 The algebraic λ -cube

The algebraic λ -cube is obtained from the λ -cube by aggregating many-sorted rewriting systems to the type systems. As in [7, 8, 10], we shall only consider first-order term-rewriting systems. Note however that there are other presentations of the algebraic λ -cube based on higher-order rewriting systems, see e.g. [3].

Definition 15. *A signature Σ consists of a pair $(A, (F_{w,s})_{w \in \text{List}(A), s \in A})$ where A is a set of universes and $(F_{w,s})_{w \in \text{List}(A), s \in A}$ is an indexed family of pairwise disjoint sets of function symbols. In the sequel, we let $F = \bigcup_{w \in \text{List}(A), s \in A} F_{w,s}$.*

Below we let Σ be a fixed signature and assume given a fixed countably infinite set of variables V_{τ} for every universe τ .

Definition 16. *The set $T_{\Sigma}(\tau)$ of τ -terms is defined by the clauses:*

1. if $x \in V_{\tau}$, then $x \in T_{\Sigma}(\tau)$;
2. if $f \in F_{(\tau_1, \dots, \tau_n), \tau}$ and $t_i \in T_{\Sigma}(\tau_i)$ for $i = 1, \dots, n$, then $f(t_1, \dots, t_n) \in T_{\Sigma}(\tau)$.

As usual, we let $\text{var}(t)$ and $\text{mvar}(t)$ denote the set and multiset of variables of t respectively.

Finally, we define term-rewriting systems.

Definition 17. *A term-rewriting system \mathcal{R} is an indexed set $(R_{\tau})_{\tau \in A}$ such that $R_{\tau} \subseteq T_{\Sigma}(\tau) \times T_{\Sigma}(\tau)$ and $\text{var}(l) \subseteq \text{var}(r)$ for every $(l, r) \in R_{\tau}$. We say \mathcal{R} is non-duplicating if $\text{mvar}(l) \subseteq \text{mvar}(r)$ for every $(l, r) \in R_{\tau}$.*

Basic concepts, such as the rewriting relation and termination, are defined in the standard way, see e.g. [27].

Throughout this section, we let \mathcal{R} be a fixed term-rewriting system. Every rule set \mathbf{S} yields an algebraic type system $\lambda\mathbf{S} + \mathcal{R}$ as follows.

Definition 18 (Algebraic Type Systems of the algebraic λ -cube).

1. The set \mathcal{T} of pseudo-terms is defined by the abstract syntax

$$\mathcal{T} = V^* \mid V^{\square} \mid * \mid \square \mid \mathcal{T}\mathcal{T} \mid \lambda V : \mathcal{T}. \mathcal{T} \mid \Pi V : \mathcal{T}. \mathcal{T} \mid A \mid F$$

2. Algebraic reduction $\rightarrow_{\mathcal{R}}$ is defined by the rules $C[\theta l] \rightarrow_{\mathcal{R}} C[\theta r]$ where (l, r) is a rule, $C[\cdot]$ is a context (in the terminology of rewriting) and θ is a substitution.
3. β -reduction, η -reduction... are defined as in Section 2.
4. The derivability relation \vdash is given by the rules of Figure 1 except for the conversion rule, and the rules of Figure 4.

In the conversion rule, one could have used η -expansion instead of η -reduction. The important point is not to allow conversion, especially algebraic conversion, as it would lead to an inconsistent system e.g. for the system \mathcal{R} defined by the rules $\text{or}(x, y) \rightarrow x$ and $\text{or}(x, y) \rightarrow y$, see e.g. [7].

| | | |
|--------------|---|--|
| (universe) | $\langle \rangle \vdash \sigma : *$ | if $\sigma \in A$ |
| (function) | $\frac{\Gamma \vdash M_1 : \sigma_1 \quad \dots \quad \Gamma \vdash M_n : \sigma_n}{\Gamma \vdash f(M_1, \dots, M_n) : \tau}$ | if $f \in F_{((\sigma_1, \dots, \sigma_n), \tau)}$ |
| (conversion) | $\frac{\Gamma \vdash A : B \quad \Gamma \vdash B' : s}{\Gamma \vdash A : B'}$ | if $B \downarrow_{\beta\eta R} B'$ |

Fig. 4. NEW RULES FOR THE ALGEBRAIC λ -CUBE

4.2 Proof-irrelevance

One major obstacle with the algebraic λ -cube is subject reduction. As shown in [3], subject reduction can be proved directly, but the proof is extremely complex and lengthy. A simpler alternative, suggested by the author in [7], is to define a *proof-irrelevant* algebraic λ -cube, for which subject reduction is trivial to establish, to show that the algebraic λ -cube is a subsystem of the proof-irrelevant algebraic λ -cube, and to conclude by showing the latter to be strongly normalizing. We begin by defining the proof-irrelevant algebraic λ -cube. As a preliminary, we define pseudo-objects, pseudo-constructors and pseudo-kinds. The definition is almost identical to the one of Section 3.

Definition 19 (Pseudo-objects, pseudo-constructors and pseudo-kinds). *The classes \mathcal{O} , \mathcal{C} and \mathcal{K} of pseudo-objects, pseudo-constructors and pseudo-kinds are given by the abstract syntaxes*

$$\begin{aligned} \mathcal{O} &= V^* \mid F \mid \lambda V^*. \mathcal{C}. \mathcal{O} \mid \lambda V^\square. \mathcal{K}. \mathcal{O} \mid \mathcal{O} \mathcal{O} \mid \mathcal{O} \mathcal{C} \\ \mathcal{C} &= V^\square \mid \Lambda \mid \Pi V : \mathcal{C}. \mathcal{C} \mid \Pi V : \mathcal{K}. \mathcal{C} \mid \lambda V^*. \mathcal{C}. \mathcal{C} \mid \lambda V^\square. \mathcal{K}. \mathcal{C} \mid \mathcal{C} \mathcal{C} \mid \mathcal{C} \mathcal{O} \\ \mathcal{K} &= * \mid \Pi V : \mathcal{C}. \mathcal{K} \mid \Pi V : \mathcal{K}. \mathcal{K} \end{aligned}$$

Intuitively, proof-irrelevant type systems are obtained by identifying all the pseudo-objects in the conversion rule. This motivates the definition below. For technical convenience, we assume the term-rewriting system under consideration to contain a specific constant \bullet .

Definition 20.

1. The proof-irrelevant skeleton $|\cdot| : \mathcal{T} \rightarrow \mathcal{T}$ is defined inductively as follows:

$$\begin{aligned} |M| &= \bullet && \text{if } M \in \mathcal{O} \\ |v| &= v && \text{if } x \in V^\square \cup \Lambda \cup \mathcal{S} \\ |M N| &= |M| |N| && \text{if } (M N) \notin \mathcal{O} \\ |\lambda x.A. M| &= \lambda x.|A|. |M| && \text{if } (\lambda x.A. M) \notin \mathcal{O} \\ |\Pi x.A. B| &= \Pi x.|A|. |B| \end{aligned}$$

2. The proof-irrelevant conversion \simeq is defined by $M \simeq N \Leftrightarrow |M| =_{\beta\eta} |N|$.
3. The relation $\Gamma \vdash^{pi} M : A$ is obtained from the rules of the algebraic λ -cube by replacing the conversion rule by

$$\frac{\Gamma \vdash^{pi} M : A \quad \Gamma \vdash^{pi} B : s}{\Gamma \vdash^{pi} M : B} \quad \text{if } A \simeq B$$

As in [7], one can show the systems of the algebraic λ -cube to be *sound w.r.t. proof-irrelevance*, i.e. for every system of the algebraic λ -cube

$$\Gamma \vdash M : A \quad \Rightarrow \quad \Gamma \vdash^{pi} M : A$$

To prove strong normalization of the algebraic λ -cube, it is thus enough to prove

$$\Gamma \vdash^{pi} M : A \quad \Rightarrow \quad M \in \text{SN}(\beta\bar{\eta}(\Gamma)\mathcal{R})$$

The gain is that proof-irrelevant systems enjoy soundness w.r.t. convertibility and subject reduction. Consequently the model construction of Section 3 only needs to be modified slightly.

4.3 The model construction

The model construction is extended by setting $(\sigma)_\rho = \text{SN}(\sigma)$ for every universe σ . The proof of soundness proceeds as before, except for:

1. the conversion rule: one needs to strengthen Lemma 11 by replacing $=_{\beta\eta}$ by \simeq . One proceeds as in [7];
2. the function rule: one needs to use the following result.

Lemma 13. *If \mathcal{R} is non-duplicating and terminating, $f \in F_{((\sigma_1, \dots, \sigma_n), \tau)}$ and $\mathcal{E} \vdash t_i : \sigma_i$ for $1 \leq i \leq n$, then*

$$t_i \in \text{SN}(\sigma_i) \text{ for } 1 \leq i \leq n \quad \Rightarrow \quad f(t_1, \dots, t_n) \in \text{SN}(\tau)$$

Proof. (Sketch) Termination is a persistent property of non-duplicating term-rewriting systems [27] thus $\rightarrow_{\mathcal{R}}$ is terminating on algebraic pseudo-terms and more generally on pseudo-terms, see [8]. To conclude, proceed exactly as in e.g. [2], using the notions of cap and aliens.

It follows that strong normalization (under β -reduction+ η -expansion+algebraic reduction) is a modular property of the systems of the algebraic λ -cube, provided the underlying term-rewriting system is non-duplicating.

Theorem 4. *For every system $\lambda\mathbf{S} + \mathcal{R}$ of the algebraic λ -cube: $\Gamma \vdash M : A \quad \Rightarrow \quad M \in \text{SN}(\beta\bar{\eta}(\Gamma)\mathcal{R})$ provided \mathcal{R} is terminating and non-duplicating.*

Theorem 4 partially solves a conjecture by Di Cosmo and Ghani [10]. Yet Theorem 4 is not fully satisfactory since one would like to drop the condition of the term-rewriting system being non-duplicating. The key is to prove Lemma 13 for an arbitrary terminating term-rewriting system. One possibility, which needs to be further explored, is to use the techniques of [8].

5 Conclusion

This paper brings new light to η -expansion in systems of dependent types. In particular, we have shown:

1. β -reduction+ η -expansion is strongly normalizing for the systems of the λ -cube;
2. β -reduction+ η -expansion+algebraic reduction is strongly normalizing for the systems of the algebraic λ -cube, provided the underlying term-rewriting system is terminating and non-duplicating.

In the future, it would be interesting to study strong normalization of β -reduction+ η -expansion for Pure Type Systems. We therefore conclude with the following conjecture:⁴

Conjecture 1. For every specification \mathbf{S} such that $\lambda_\beta\mathbf{S}$ is strongly normalizing w.r.t. \rightarrow_β , $\lambda\mathbf{S}$ is strongly normalizing w.r.t. $\rightarrow_{\beta\bar{\eta}}$.

A similar conjecture for $\rightarrow_{\beta\eta}$ -reduction has been formulated by Geuvers [14].

Conjecture 2 (Geuvers). For every specification \mathbf{S} such that $\lambda_\beta\mathbf{S}$ is strongly normalizing w.r.t. \rightarrow_β , $\lambda\mathbf{S}$ is strongly normalizing w.r.t. $\rightarrow_{\beta\eta}$.

Our conjecture implies his, as

$$\begin{aligned} \lambda\mathbf{S} \text{ is strongly normalizing w.r.t. } \rightarrow_{\beta\bar{\eta}} &\Rightarrow \lambda\mathbf{S} \text{ is strongly normalizing w.r.t. } \rightarrow_\beta \\ &\Rightarrow \lambda\mathbf{S} \text{ is strongly normalizing w.r.t. } \rightarrow_{\beta\eta} \end{aligned}$$

⁴ $\lambda_\beta\mathbf{S}$ denotes the usual Pure Type System $\lambda\mathbf{S}$ associated to a specification \mathbf{S} .

References

1. T. Altenkirch. *Constructions, inductive types and strong normalisation*. PhD thesis, Laboratory for the Foundations of Computer Science, University of Edinburgh, 1994.
2. F. Barbanera and M. Fernández. Modularity of termination and confluence in combinations of rewrite systems with λ_ω . In A. Lingas, R. Karlsson, and S. Karlsson, editors, *Proceedings of ICALP'93*, volume 700 of *Lecture Notes in Computer Science*, pages 657–668. Springer-Verlag, 1993.
3. F. Barbanera, M. Fernández, and H. Geuvers. Modularity of strong normalisation and confluence in the algebraic λ -cube. *Journal of Functional Programming*, 7(6):613–660, November 1997.
4. H. Barendregt. Introduction to Generalised Type Systems. *Journal of Functional Programming*, 1(2):125–154, April 1991.
5. H. Barendregt. Lambda calculi with types. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, pages 117–309. Oxford Science Publications, 1992. Volume 2.
6. G. Barthe. Existence and uniqueness of normal forms in pure type systems with $\beta\eta$ -conversion. Manuscript, 1998.
7. G. Barthe. The relevance of proof-irrelevance. In K.G. Larsen, S. Skyum, and G. Winskel, editors, *Proceedings of ICALP'98*, volume 1443 of *Lecture Notes in Computer Science*, pages 755–768. Springer-Verlag, 1998.
8. G. Barthe and F. van Raamsdonk. Termination of algebraic type systems: the syntactic approach. In M. Hanus and J. Heering, editors, *Proceedings of ALP '97 - HOA '97*, volume 1298 of *Lecture Notes in Computer Science*, pages 174–193. Springer-Verlag, 1997.
9. R. Di Cosmo. A brief history of rewriting with extensionality. Presented at the International Summer School on Type Theory and Term Rewriting, Glasgow, September 1996.
10. R. Di Cosmo and N. Ghani. On modular properties of higher order extensional lambda calculi. In P. Degano, R. Gorrieri, and A. Marchetti-Spaccamela, editors, *Proceedings of ICALP'97*, volume 1256 of *Lecture Notes in Computer Science*, pages 237–247. Springer-Verlag, 1997.
11. R. Di Cosmo and D. Kesner. Simulating expansions without expansions. *Mathematical Structures in Computer Science*, 4(3):315–362, September 1994.
12. D. van Daalen. *The language theory of Automath*. PhD thesis, Technical University of Eindhoven, 1980.
13. G. Dowek, G. Huet, and B. Werner. On the existence of long $\beta\eta$ -normal forms in the cube. In H. Geuvers, editor, *Informal Proceedings of TYPES'93*, pages 115–130, 1993.
14. H. Geuvers. *Logics and type systems*. PhD thesis, University of Nijmegen, 1993.
15. H. Geuvers. A short and flexible proof of strong normalisation for the Calculus of Constructions. In P. Dybjer, B. Nordström, and J. Smith, editors, *Proceedings of TYPES'94*, volume 996 of *Lecture Notes in Computer Science*, pages 14–38. Springer-Verlag, 1995.
16. H. Geuvers and B. Werner. On the Church-Rosser property for expressive type systems and its consequence for their metatheoretic study. In *Proceedings of LICS'94*, pages 320–329. IEEE Computer Society Press, 1994.
17. N. Ghani. Eta-expansions in dependent type theory—the calculus of constructions. In P. de Groote and J. Hindley, editors, *Proceedings of TLCA'97*, volume 1210 of *Lecture Notes in Computer Science*, pages 164–180. Springer-Verlag, 1997.
18. F. Joachimski. η -expansion in Gödel's T, Pure Type Systems and calculi with permutative conversions. Manuscript, 1997.
19. J.W. Klop. Term-rewriting systems. In S. Abramsky, D. Gabbay, and T. Maibaum, editors, *Handbook of Logic in Computer Science*, pages 1–116. Oxford Science Publications, 1992. Volume 2.
20. P.-A. Mellès and B. Werner. A generic proof of strong normalisation for pure type systems. In E. Giménez and C. Paulin-Mohring, editors, *Proceedings of TYPES'96*, volume 1512 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.
21. R. Nederpelt. *Strong normalisation in a typed lambda calculus with lambda structured types*. PhD thesis, Technical University of Eindhoven, 1973.
22. G. Pottinger. Strong normalisation for terms of the theory of constructions. Technical Report TR 11-7, Odissey Research Associates, 1987.
23. D. Prawitz. Ideas and results of proof theory. In J.E. Fenstad, editor, *The 2nd Scandinavian Logical Symposium*, pages 235–307. North-Holland, 1970.
24. A. Salvesen. The Church-Rosser property for $\beta\eta$ -reduction. Manuscript, 1991.
25. J. Terlouw. Strong normalization in type systems: a model theoretic approach. *Annals of Pure and Applied Logic*, 73(1):53–78, May 1995.
26. H. Xi. Simulating η -expansions with β -reductions in the second-order polymorphic lambda-calculus. In S. Adian and A. Nerode, editors, *Proceedings of LFCS'97*, volume 1234 of *Lecture Notes in Computer Science*, pages 399–409. Springer-Verlag, 1997.
27. H. Zantema. Termination of term rewriting: Interpretation and type elimination. *Journal of Symbolic Computation*, 17(1):23–50, January 1994.