# Semantic Analysis of Normalisation by Evaluation for Typed Lambda Calculus

(Extended Abstract)

Marcelo Fiore*
Computer Laboratory
University of Cambridge

## ABSTRACT

This paper studies normalisation by evaluation for typed lambda calculus from a categorical and algebraic viewpoint. The first part of the paper analyses the lambda definability result of Jung and Tiuryn via Kripke logical relations and shows how it can be adapted to unify definability and normalisation, yielding an extensional normalisation result. In the second part of the paper the analysis is refined further by considering intensional Kripke relations (in the form of glueing) and shown to provide a function for normalising terms, casting normalisation by evaluation in the context of categorical glueing. The technical development includes an algebraic treatment of the syntax and semantics of the typed lambda calculus that allows the definition of the normalisation function to be given within a simply typed metatheory.

## Categories and Subject Descriptors

D.3.1 [**Programming Languages**]: Formal Definitions and Theory—*syntax, semantics*; F.3.2 [**Logics and Meanings of Programs**]: Semantics of Programming Languages—*algebraic approaches to semantics, denotational semantics, partial evaluation*

## General Terms

Theory, Languages

## Keywords

Typed lambda calculus, lambda definability, logical relations, typed abstract syntax with variable binding, initial algebra semantics, categorical glueing, normalisation by evaluation

*Research supported by an EPSRC Advanced Research Fellowship.


*PPDP'02,* October 6–8, 2002, Pittsburgh, Pennsylvania, USA.


## INTRODUCTION

Normalisation by evaluation for typed lambda calculus was first considered by Berger and Schwichtenberg [5] from a type and proof theoretic viewpoint, and later investigated from the point of view of logic [4], type theory [7], category theory [3, 10, 29], and partial evaluation [11, 14]. This work gives a new categorical and algebraic perspective on the topic.

**Outline.** Normalisation by evaluation will be broadly view as the technique of giving semantics in (metalanguages for) non-standard models from which normalisation information can be extracted (*cf.* [26]). In this light, we will investigate the following problems.

I. *Extensional normalisation problem*: To define normal terms and establish that every term $\beta\eta$-equals one in normal form.

   That is, writing $\mathfrak{L}_\tau(\Gamma)$ for the set of terms of type $\tau$ in context $\Gamma$, to identify a set of normal terms $\mathfrak{N}_\tau(\Gamma) \subseteq \mathfrak{L}_\tau(\Gamma)$ and show that for every term $t \in \mathfrak{L}_\tau(\Gamma)$ there exists a normal term $N \in \mathfrak{N}_\tau(\Gamma)$ such that $t =_{\beta\eta} N$.

II. *Intensional normalisation problem*: To define, and prove the correctness of, a normalisation function associating normal forms to terms.

   More precisely, to construct functions

$$\mathsf{nf}_{\tau,\Gamma} : \mathfrak{L}_\tau(\Gamma) \longrightarrow \mathfrak{N}_\tau(\Gamma)$$

   satisfying the following three properties.

   (1) For all normal terms $N \in \mathfrak{N}_\tau(\Gamma)$, the syntactic equality $\mathsf{nf}_{\tau,\Gamma}(N) = N$ holds.

   (2) For all terms $t \in \mathfrak{L}_\tau(\Gamma)$, the semantic equality $\mathsf{nf}_{\tau,\Gamma}(t) =_{\beta\eta} t$ holds.

   (3) For all pair of terms $t, t' \in \mathfrak{L}_\tau(\Gamma)$, if $t =_{\beta\eta} t'$ then $\mathsf{nf}_{\tau,\Gamma}(t) = \mathsf{nf}_{\tau,\Gamma}(t')$.

   (In the context of normalisation by evaluation, the correctness condition (1) has seldom been considered — the exception being [29]. However, it is both natural and interesting. For instance, together with the correctness condition (3) it implies that $\beta\eta$-equal normal terms are syntactically equal, which in turn, together with the correctness condition (2), entails the stronger version of extensional normalisation that every term $\beta\eta$-equals a unique normal term.)

These problems will be respectively dealt with in Parts I and II of the paper. Part I, provides a unifying view of definability and normalisation leading to an extensional normalisation result. This analysis, besides unifying the two hitherto unrelated problems of definability and normalisation, motivates and elucidates the notions of neutral and normal terms, which are here distilled from semantic considerations. Part II, shows that an intensional view of Part I amounts to the traditional technique of normalisation by evaluation. This development leads to a treatment of normalisation by evaluation via the glueing construction, finally formalising the observation that normalisation by evaluation is closely related to categorical glueing [8].

More in detail, the paper is organised as follows. Section I.1 briefly recalls the syntax and categorical semantics of the typed lambda calculus. Section I.2 presents an analysis of the lambda definability result of Jung and Tiuryn via Kripke logical relations leading to an extensional normalisation result. Section II.1 describes the rudiments of a theory of typed abstract syntax with variable binding which is used to put the typed lambda calculus in an algebraic framework. This algebraic view is exploited in Section II.2 to structure the development of an intensional version of Section I.2 culminating in the technique of normalisation by evaluation.

**Related work.** The treatment of extensional normalisation presented here is similar to the approach to strong normalisation via computability predicates [33, 20] for the typed lambda calculus, and also to the approach to normalisation in [22, Chapter III] for the untyped lambda calculus. The precise relationships need to be investigated.

The analysis of normalisation by evaluation pursued here is categorical and, as such, is related to [3], [10], [29], and [2].

The approach of [10] is in the context of so-called $\mathcal{P}$-category theory; which is, roughly, a version of category theory equipped with an intensional notion of equality formalised by partial equivalence relations. The intensional information needed for the purpose of normalisation will be captured here in the context of traditional category theory via the glueing construction.

In [3], normalisation by evaluation is reconstructed categorically in a model obtained via an ad hoc *twisted glueing* construction. This model embodies objects with both syntactic and semantic components, and translations between them essentially encoding a correctness predicate. In contrast, we will adopt a purely semantic view, working with intensional logical relations in models given by the traditional glueing construction.

Another important point of departure between this work and the other categorical ones is the algebraic treatment of the subject, which led to a deeper understanding of the normalisation function.

# PART I

## I.1 Typed lambda calculus

For the purpose of establishing notation, we briefly recall the syntax and semantics of the typed lambda calculus. For details see, *e.g.*, [23, 9, 34].

**Syntax.** The types of the simply typed lambda calculus are given by the grammar

$$\tau \quad ::= \quad \theta \mid \mathsf{1} \mid \tau_1 * \tau_2 \mid \tau_1 \Rightarrow \tau_2$$

where $\theta$ ranges over base types. We write $\widetilde{\mathrm{T}}$ for the set of simple types generated by a set of base types $\mathrm{T}$.

The grammar for the terms is

$$t \quad ::= \quad x \mid \langle\rangle \mid \pi_1(t) \mid \pi_2(t) \mid \langle t_1, t_2\rangle \mid t(t') \mid \lambda x : \tau.\, t$$

where $x$ ranges over (a countably infinite set of) variables. The notion of free and bound variables are standard. As usual, we will identify terms up to the renaming of bound variables.

Typing contexts, with types in a set $\mathcal{T}$, are defined as functions $V \longrightarrow \mathcal{T}$ where the domain of the context, $V$, is a finite subset of the set of variables. Under this view for a variable $x$, a type $\tau$, and a context $\Gamma$, we let $(x : \tau) \in \Gamma$ stand for $x \in \mathrm{dom}(\Gamma)$ and $\Gamma(x) = \tau$. For distinct variables $x_i$ $(i = 1, n)$, we use the notation $\langle x_i : \tau_i \rangle_{i=1,n}$ for the context $\{ x_1, \ldots, x_n \} \longrightarrow \mathcal{T}$ mapping $x_i$ to $\tau_i$. For a context $\Gamma$, a variable $x$, and a type $\tau$, the notation for the context extension $\Gamma, x : \tau$ presupposes $x \notin \mathrm{dom}(\Gamma)$ and denotes the context $\mathrm{dom}(\Gamma) \cup \{ x \} \longrightarrow \mathcal{T}$ mapping every $y \in \mathrm{dom}(\Gamma)$ to $\Gamma(y)$, and $x$ to $\tau$.

The well-typed terms $\Gamma \vdash t : \tau$ in context (where $\Gamma$ is a typing context, $t$ is a term, and $\tau$ is a type) are given by the usual rules; see Figure 1.

$$\frac{}{\Gamma \vdash x : \tau} \quad (x : \tau) \in \Gamma$$

$$\frac{}{\Gamma \vdash \langle\rangle : \mathsf{1}}$$

$$\frac{\Gamma \vdash t : \tau_1 * \tau_2}{\Gamma \vdash \pi_i(t) : \tau_i} \quad (i = 1, 2)$$

$$\frac{\Gamma \vdash t_i : \tau_i \quad (i = 1, 2)}{\Gamma \vdash \langle t_1, t_2 \rangle : \tau_1 * \tau_2}$$

$$\frac{\Gamma \vdash t : \tau' \Rightarrow \tau \quad \Gamma \vdash t' : \tau'}{\Gamma \vdash t(t') : \tau}$$

$$\frac{\Gamma, x : \tau' \vdash t : \tau}{\Gamma \vdash \lambda x : \tau'.\, t : \tau' \Rightarrow \tau}$$

**Figure 1: Well-typed terms**

**Semantics.** The appropriate mathematical universes for giving semantics to the typed lambda calculus are cartesian closed categories [23, 9, 34]; *i.e.*, categories with terminal object, binary products, and exponentials (for which we respectively use the notation 1, $\times$, and $\Rightarrow$).

For an interpretation $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ of base types in a cartesian closed category, we let $\boldsymbol{s}[\![\_]\!] : \widetilde{\mathrm{T}} \longrightarrow \mathcal{S}$ be the extension to simple types as prescribed by a chosen cartesian closed structure. That is, $\boldsymbol{s}[\![\theta]\!] = \boldsymbol{s}(\theta)$ (for $\theta$ a base type), $\boldsymbol{s}[\![\mathsf{1}]\!] = 1$, $\boldsymbol{s}[\![\tau * \tau']\!] = \boldsymbol{s}[\![\tau]\!] \times \boldsymbol{s}[\![\tau']\!]$, and $\boldsymbol{s}[\![\tau \Rightarrow \tau']\!] = \boldsymbol{s}[\![\tau]\!] \Rightarrow \boldsymbol{s}[\![\tau']\!]$. As usual, the interpretation of types is extended to contexts by setting $\boldsymbol{s}[\![\Gamma]\!] = \prod_{x \in \mathrm{dom}(\Gamma)} \boldsymbol{s}[\![\Gamma(x)]\!]$ for all contexts $\Gamma$. Finally, the semantics of a term $\Gamma \vdash t : \tau$ as a morphism $\boldsymbol{s}[\![\Gamma]\!] \longrightarrow \boldsymbol{s}[\![\tau]\!]$ in $\mathcal{S}$ is denoted $\boldsymbol{s}[\![\Gamma \vdash t : \tau]\!]$.

## I.2 From definability to normalisation

Kripke relations of varying arity were introduced in [21] for the purpose of characterising lambda definability. We will analyse this result and provide a corresponding extensional normalisation result.

**Kripke relations.** For a functor $\varsigma : \mathbb{C} \longrightarrow \mathcal{S}$, a $\mathbb{C}$-*Kripke relation* $R$ of arity $\varsigma$ over an object $A$ of $\mathcal{S}$ is a family $\{ R(c) \subseteq \mathcal{S}(\varsigma(c), A) \}_{c \in |\mathbb{C}|}$ satisfying the following condition.

(Monotonicity) For every $\rho : c' \longrightarrow c$ in $\mathbb{C}$ and every $a : \varsigma(c) \longrightarrow A$ in $R(c)$, the map $a \circ \varsigma(\rho) : \varsigma(c') \longrightarrow A$ is in $R(c')$.

(In other words, a $\mathbb{C}$-Kripke relation of arity $\varsigma$ over an object $A$ is a unary predicate over the $\mathbb{C}^{\mathrm{op}}$-variable set of $A$-valued morphisms $\mathcal{S}(\varsigma(\_), A) : \mathbb{C}^{\mathrm{op}} \longrightarrow \mathbf{Set}$.)

The category of Kripke relations $\underline{\mathrm{K}}\langle\varsigma\rangle$ of arity $\varsigma : \mathbb{C} \longrightarrow \mathcal{S}$ has objects given by pairs $(R, A)$ consisting of an object $A$ of $\mathcal{S}$ and a $\mathbb{C}$-Kripke relation of arity $\varsigma$ over $A$, and morphisms $f : (R, A) \longrightarrow (R', A')$ given by maps $f : A \longrightarrow A'$ in $\mathcal{S}$ such that, for all $a : \varsigma(c) \longrightarrow A$ in $R(c)$, the composite $f \circ a : \varsigma(c) \longrightarrow A$ is in $R(c)$. (Composition and identities are as in $\mathcal{S}$.)

The following proposition is well-known (see, *e.g.*, [1, 25]).

Proposition 1. *Let $\mathbb{C}$ be a small category and let $\mathcal{S}$ be a cartesian closed category. For a functor $\varsigma : \mathbb{C} \longrightarrow \mathcal{S}$, the category of Kripke relations $\underline{\mathrm{K}}\langle\varsigma\rangle$ is cartesian closed, and the forgetful functor $\underline{\mathrm{K}}\langle\varsigma\rangle \longrightarrow \mathcal{S} : (R, A) \longmapsto A$ preserves the cartesian closed structure.*

The cartesian closed structure of $\underline{\mathrm{K}}\langle\varsigma\rangle$ is given as follows.

(Products) The terminal object is $(1, \top)$ where $1$ is terminal in $\mathcal{S}$ and where $\top(c) = \{ \varsigma(c) \longrightarrow 1 \}$ for all $c$ in $\mathbb{C}$.

The product $(R, A) \times (R', A')$ of $(R, A)$ and $(R', A')$ is

$$(R, A) \xleftarrow{\pi} (R \wedge R', A \times A') \xrightarrow{\pi'} (R', A')$$

where $A \xleftarrow{\pi} A \times A' \xrightarrow{\pi'} A'$ is the product of $A$ and $A'$ in $\mathcal{S}$, and where $a : \varsigma(c) \longrightarrow A \times A'$ is in $(R \wedge R')(c)$ iff $\pi \circ a : \varsigma(c) \longrightarrow A$ is in $R(c)$ and $\pi' \circ a : \varsigma(c) \longrightarrow A'$ is in $R'(c)$.

(Exponentials) The exponential $(R, A) \Longrightarrow (R', A')$ of $(R, A)$ and $(R', A')$ is

$$(R \supset R', A \Longrightarrow A') \times (R, A) \xrightarrow{\varepsilon} (R', A')$$

where $(A \Longrightarrow A') \times A \xrightarrow{\varepsilon} A'$ is the exponential of $A$ and $A'$ in $\mathcal{S}$, and where $f : \varsigma(c) \longrightarrow A \Longrightarrow A'$ is in $(R \supset R')(c)$ iff, for every $\rho : c' \longrightarrow c$ in $\mathbb{C}$ and $a : \varsigma(c') \longrightarrow A$ in $R(c')$, the composite $\varepsilon \circ \langle f \circ \varsigma(\rho), a \rangle : \varsigma(c') \longrightarrow A'$ is in $R'(c')$.

The Fundamental Lemma of logical relations [28, 30] is a consequence of the above proposition: for an interpretation of base types $\mathcal{I} : \mathrm{T} \longrightarrow \underline{\mathrm{K}}\langle\varsigma\rangle : \theta \longmapsto (\mathcal{R}_\theta, \mathcal{I}_0(\theta))$, the interpretation

$$\mathcal{I}_0[\![\Gamma \vdash t : \tau]\!] : \mathcal{I}_0[\![\Gamma]\!] \longrightarrow \mathcal{I}_0[\![\tau]\!] \text{ in } \mathcal{S}$$

of a term $\Gamma \vdash t : \tau$ yields a morphism $\mathcal{I}[\![\Gamma]\!] \longrightarrow \mathcal{I}[\![\tau]\!]$ in $\underline{\mathrm{K}}\langle\varsigma\rangle$; that is, for $\mathcal{I}[\![\Gamma]\!] = (\mathcal{R}_\Gamma, \mathcal{I}_0[\![\Gamma]\!])$ and $\mathcal{I}[\![\tau]\!] = (\mathcal{R}_\tau, \mathcal{I}_0[\![\tau]\!])$, the following diagram

$$\begin{array}{ccc} \mathcal{R}_\Gamma & \longrightarrow & \mathcal{R}_\tau \\ \downarrow & & \downarrow \\ \mathcal{S}(\varsigma(\_), \mathcal{I}_0[\![\Gamma]\!]) & \xrightarrow{\mathcal{I}_0[\![\Gamma \vdash t : \tau]\!] \circ \_} & \mathcal{S}(\varsigma(\_), \mathcal{I}_0[\![\tau]\!]) \end{array}$$

commutes (for a necessarily unique natural map $\mathcal{R}_\Gamma \longrightarrow \mathcal{R}_\tau$).

**Definability.** The definability result of [21] uses Kripke relations varying over a poset of contexts ordered by context extension. Here, however, to parallel the development with the one to follow in Part II, we will consider Kripke relations varying over a category of contexts and context renamings.

For a set of types $\mathcal{T}$, we let $\mathbb{F} \downarrow \mathcal{T}$ be the category with objects given by contexts $\Gamma$ with types in $\mathcal{T}$, and with morphisms $\Gamma \longrightarrow \Gamma'$ given by type-preserving context renamings; that is, by functions $\rho : \mathrm{dom}(\Gamma) \longrightarrow \mathrm{dom}(\Gamma')$ such that for all variables $x \in \mathrm{dom}(\Gamma)$, the types $\Gamma(x)$ and $\Gamma'(\rho x)$ are equal. We write $\mathbb{F}[\mathcal{T}]$ for $(\mathbb{F} \downarrow \mathcal{T})^{\mathrm{op}}$.

With respect to an interpretation $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ of base types in a cartesian closed category, we write $\boldsymbol{s}[\![\_]\!]$ for the canonical semantic functor $\mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}$ interpreting contexts and their renamings, which is explicitly given by

$$\begin{aligned} \boldsymbol{s}[\![\rho]\!] &= \langle \boldsymbol{s}[\![\Gamma' \vdash \rho x : \tau]\!] \rangle_{(x:\tau) \in \Gamma} \\ &= \langle \pi_{\rho x} \rangle_{x \in \mathrm{dom}(\Gamma)} \;:\; \boldsymbol{s}[\![\Gamma']\!] \longrightarrow \boldsymbol{s}[\![\Gamma]\!] \end{aligned}$$

for all $\rho : \Gamma \longrightarrow \Gamma'$ in $\mathbb{F} \downarrow \widetilde{\mathrm{T}}$.

For every type $\tau \in \widetilde{\mathrm{T}}$, the *definability relation*

$$\mathcal{D}_\tau(\Gamma) = \{ \boldsymbol{s}[\![\Gamma \vdash t : \tau]\!] \mid \Gamma \vdash t : \tau \} \subseteq \mathcal{S}(\boldsymbol{s}[\![\Gamma]\!], \boldsymbol{s}[\![\tau]\!])$$

is an $\mathbb{F}[\widetilde{\mathrm{T}}]$-Kripke relation of arity $\boldsymbol{s}[\![\_]\!] : \mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}$ over $\boldsymbol{s}[\![\tau]\!]$, and the family of definability relations $\{ \mathcal{D}_\tau \}_{\tau \in \widetilde{\mathrm{T}}}$ has the following logical characterisation.

Lemma 2. (Definability Lemma [21, 1]) *Let $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ be an interpretation of base types in a cartesian closed category. Setting $\mathcal{R}_\theta = \mathcal{D}_\theta$ for all base types $\theta \in \mathrm{T}$ and letting $\mathcal{R}_\tau$ be given by the cartesian closed structure of the category of Kripke relations $\underline{\mathrm{K}}\langle \boldsymbol{s}[\![\_]\!] : \mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}\rangle$ for the other types $\tau \in \widetilde{\mathrm{T}}$, it follows that $\mathcal{R}_\tau = \mathcal{D}_\tau$ for all types $\tau \in \widetilde{\mathrm{T}}$.*

The usual proof of the Definability Lemma is by induction on the structure of types using the explicit description of the cartesian closed structure in categories of Kripke relations given above; see [21, 1] (and [18] for the case with sum types). However, there is a more conceptual proof based on the following observation: for $\mathbb{C}$-Kripke relations $R$ and $R'$ of arity $\varsigma$ over $A$,

$$R \subseteq R' \;\text{ iff }\; \mathrm{id}_A : (R, A) \longrightarrow (R', A) \text{ in } \underline{\mathrm{K}}\langle\varsigma\rangle \quad . \tag{1}$$

In this light, to establish the Definability Lemma it is enough to see that the definability relations satisfy the following closure properties

$$\begin{aligned} \mathcal{D}_1 &= \top \\ \mathcal{D}_{\tau * \tau'} &= \mathcal{D}_\tau \wedge \mathcal{D}_{\tau'} \\ \mathcal{D}_{\tau \Rightarrow \tau'} &= \mathcal{D}_\tau \supset \mathcal{D}_{\tau'} \end{aligned}$$

which is, in effect, what the usual calculations amount to.

This analysis can be refined further. Indeed, the fact that neither of the inclusions

$$\mathcal{D}_\tau \subseteq \mathcal{R}_\tau \subseteq \mathcal{D}_\tau \tag{2}$$

in isolation is strong enough to re-establish the inductive hypothesis in the Definability Lemma, suggests considering a more general situation in which the Kripke logical relations $\mathcal{R}_\tau$ are bounded by possibly distinct Kripke relations (unlike the situation in (2)).

The above observations lead to the following basic lemma.

Lemma 3. (Basic Lemma) *Consider an interpretation* $\mathcal{I}_0 : \mathrm{T} \longrightarrow \mathcal{S}$ *of base types in a cartesian closed category* $\mathcal{S}$.

*With respect to a functor* $\varsigma : \mathbb{C} \longrightarrow \mathcal{S}$, *let* $\langle(\mathcal{L}_\tau, \mathcal{I}_0[\![\tau]\!])\rangle_{\tau\in\widetilde{\mathrm{T}}}$ *and* $\langle(\mathcal{U}_\tau, \mathcal{I}_0[\![\tau]\!])\rangle_{\tau\in\widetilde{\mathrm{T}}}$ *be two families of Kripke relations in* $\underline{K}\langle\varsigma\rangle$ *indexed by types such that*

$$\mathcal{U}_1 = \top$$

$$\mathcal{L}_{\sigma * \tau} \subseteq \mathcal{L}_\sigma \wedge \mathcal{L}_\tau \qquad \mathcal{U}_\sigma \wedge \mathcal{U}_\tau \subseteq \mathcal{U}_{\sigma * \tau}$$

$$\mathcal{L}_{\sigma \Rightarrow \tau} \subseteq \mathcal{U}_\sigma \supset \mathcal{L}\tau \qquad \mathcal{L}_\sigma \supset \mathcal{U}_\tau \subseteq \mathcal{U}_{\sigma\Rightarrow\tau}$$

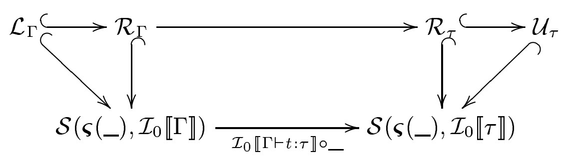
*For a family of Kripke relations* $\langle(\mathcal{R}_\theta, \mathcal{I}_0[\![\theta]\!])\rangle_{\theta\in\mathrm{T}}$ *in* $\underline{K}\langle\varsigma\rangle$ *indexed by base types, let* $\langle(\mathcal{R}_\tau, \mathcal{I}_0[\![\tau]\!])\rangle_{\tau\in\widetilde{\mathrm{T}}}$ *be the family of Kripke relations indexed by types induced by the cartesian closed structure of* $\underline{K}\langle\varsigma\rangle$.

*If* $\mathcal{L}_\theta \subseteq \mathcal{R}_\theta \subseteq \mathcal{U}_\theta$ *for all base types* $\theta \in \mathrm{T}$, *then*

1. $\mathcal{L}_\tau \subseteq \mathcal{R}_\tau \subseteq \mathcal{U}_\tau$ *for all types* $\tau \in \widetilde{\mathrm{T}}$, *and thus*
2. *for all terms* $\Gamma \vdash t : \tau$ *(with* $\Gamma = \langle x_i : \tau_i\rangle_{i=1,n}$*) and morphisms* $a_i : \varsigma(c) \longrightarrow \mathcal{I}_0[\![\tau_i]\!]$ *in* $\mathcal{L}_{\tau_i}(c)$ $(1 \le i \le n$, $c \in |\mathbb{C}|)$, *we have that* $\mathcal{I}_0[\![\Gamma \vdash t : \tau]\!] \circ \langle a_1, \ldots, a_n\rangle : \varsigma(c) \longrightarrow \mathcal{I}_0[\![\tau]\!]$ *is in* $\mathcal{U}_\tau(c)$.

(Notice the mixed variance treatment of exponentiation.)

The proof of the first part of the Basic Lemma is again by induction on the structure of types, using the observation (1) and the functoriality of the type constructors. The proof of the second part follows from considering the interpretation $\mathcal{I} : \mathrm{T} \longrightarrow \underline{K}\langle\varsigma\rangle$ mapping a base type $\theta$ to the Kripke relation $(\mathcal{R}_\theta, \mathcal{I}_0[\![\theta]\!])$ and noticing that, by the first part and the Fundamental Lemma of logical relations, the diagram




commutes, where for $\Gamma = \langle x_i : \tau_i\rangle_{i=1,n}$, $\mathcal{L}_\Gamma = \mathcal{L}_{\tau_1} \wedge \ldots \wedge \mathcal{L}_{\tau_n}$ and $\mathcal{R}_\Gamma = \mathcal{R}_{\tau_1} \wedge \ldots \wedge \mathcal{R}_{\tau_n}$.

The Basic Lemma yields the Definability Lemma by considering $\mathcal{L}_\tau = \mathcal{D}_\tau = \mathcal{U}_\tau$ in the category of Kripke relations $\underline{K}\langle \boldsymbol{s}[\![\_]\!] : \mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}\rangle$ for the given interpretation $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$. We will now see that the Basic Lemma can be also applied to obtain an extensional normalisation result.

**Normalisation.** For an interpretation $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ of base types in a cartesian closed category we aim at defining families $\{\,(\mathcal{M}_\tau, \boldsymbol{s}[\![\tau]\!])\,\}_{\tau\in\widetilde{\mathrm{T}}}$ and $\{\,(\mathcal{N}_\tau, \boldsymbol{s}[\![\tau]\!])\,\}_{\tau\in\widetilde{\mathrm{T}}}$ of $\mathbb{F}[\widetilde{\mathrm{T}}]$-Kripke relations of arity $\boldsymbol{s}[\![\_]\!] : \mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}$ of definable morphisms such that

$$(i)\ \mathcal{N}_1 = \top$$

$$(ii)\ \mathcal{M}_{\sigma * \tau} \subseteq \mathcal{M}_\sigma \wedge \mathcal{M}_\tau \qquad (iii)\ \mathcal{N}_\sigma \wedge \mathcal{N}_\tau \subseteq \mathcal{N}_{\sigma * \tau}$$

$$(iv)\ \mathcal{M}_{\sigma\Rightarrow\tau} \subseteq \mathcal{N}_\sigma \supset \mathcal{M}_\tau \qquad (v)\ \mathcal{M}_\sigma \supset \mathcal{N}_\tau \subseteq \mathcal{N}_{\sigma\Rightarrow\tau}$$

$$(vi)\ \mathcal{M}_\theta \subseteq \mathcal{N}_\theta \quad (\theta \in \mathrm{T})$$

$$(vii)\ \pi_x : \boldsymbol{s}[\![\Gamma]\!] \longrightarrow \boldsymbol{s}[\![\tau]\!] \in \mathcal{M}_\tau(\Gamma) \quad ((x : \tau) \in \Gamma)$$

so that, by the second part of the Basic Lemma, we get

$$\frac{}{\Gamma \vdash_{\mathcal{M}} x : \tau} \quad (x : \tau) \in \Gamma$$

$$\frac{\Gamma \vdash_{\mathcal{M}} M : \tau_1 * \tau_2}{\Gamma \vdash_{\mathcal{M}} \pi_i(M) : \tau_i} \quad (i = 1, 2)$$

$$\frac{\Gamma \vdash_{\mathcal{M}} M : \tau \Rightarrow \tau' \quad \Gamma \vdash_{\mathcal{N}} N : \tau}{\Gamma \vdash_{\mathcal{M}} M(N) : \tau'}$$

$$\frac{}{\Gamma \vdash_{\mathcal{N}} \langle\rangle : 1}$$

$$\frac{\Gamma \vdash_{\mathcal{N}} N_i : \tau_i \quad (i = 1, 2)}{\Gamma \vdash_{\mathcal{N}} \langle N_1, N_2\rangle : \tau_1 * \tau_2}$$

$$\frac{\Gamma, x : \tau \vdash_{\mathcal{N}} N : \tau'}{\Gamma \vdash_{\mathcal{N}} \lambda x : \tau.\, N : \tau \Rightarrow \tau'}$$

$$\frac{\Gamma \vdash_{\mathcal{M}} M : \theta}{\Gamma \vdash_{\mathcal{N}} M : \theta} \quad (\theta \text{ a base type})$$

**Figure 2: Neutral and normal terms**

(setting $\mathcal{R}_\theta = \mathcal{M}_\theta$ for all $\theta \in \mathrm{T}$, and $a_i = \pi_i : \boldsymbol{s}[\![\Gamma]\!] \longrightarrow \boldsymbol{s}[\![\tau_i]\!]$ for $\Gamma = \langle x_i : \tau_i\rangle_{i=1,n}$) that, for all terms $\Gamma \vdash t : \tau$,

$$\boldsymbol{s}[\![\Gamma \vdash t : \tau]\!] : \boldsymbol{s}[\![\Gamma]\!] \longrightarrow \boldsymbol{s}[\![\tau]\!] \in \mathcal{N}_\tau(\Gamma) \quad .$$

The above will be achieved by distilling the semantic closure properties $(i)$–$(vii)$ into two syntactic typing systems $\vdash_{\mathcal{M}}$ and $\vdash_{\mathcal{N}}$ with respect to which the definitions

$$\mathcal{M}_\tau(\Gamma) \;=\; \{\, \boldsymbol{s}[\![\Gamma \vdash M : \tau]\!] \mid \Gamma \vdash_{\mathcal{M}} M : \tau \,\}$$
$$\mathcal{N}_\tau(\Gamma) \;=\; \{\, \boldsymbol{s}[\![\Gamma \vdash N : \tau]\!] \mid \Gamma \vdash_{\mathcal{N}} N : \tau \,\}$$

will provide the required Kripke relations. The conditions $(i)$–$(vii)$ amount, roughly, to the following properties.

- The system $\vdash_{\mathcal{M}}$ should contain variables (condition $(vii)$), and be closed under projections (condition $(ii)$) and under the application to terms in the system $\vdash_{\mathcal{N}}$ (condition $(iv)$).
- The system $\vdash_{\mathcal{N}}$ should contain the unit (condition $(i)$), and should be closed under pairing (condition $(iii)$) and under abstraction (condition $(v)$).
- Every term of base type in the system $\vdash_{\mathcal{M}}$ should be in the system $\vdash_{\mathcal{N}}$ (condition $(vi)$).

Formally, the systems are given by the rules in Figure 2.

Thus, from purely semantic considerations, we have obtained the well-known notions of *neutral* terms (*viz.*, those derivable in the system $\vdash_{\mathcal{M}}$) and of *long* $\beta\eta$*-normal* terms (*viz.*, those derivable in the system $\vdash_{\mathcal{N}}$) together with the following result.

Lemma 4. (Extensional Normalisation Lemma) *Let* $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ *be an interpretation of base types in a cartesian closed category. For every term* $\Gamma \vdash t : \tau$ *there exists a long* $\beta\eta$*-normal term* $\Gamma \vdash_{\mathcal{N}} N : \tau$ *such that*

$$\boldsymbol{s}[\![\Gamma \vdash t : \tau]\!] \;=\; \boldsymbol{s}[\![\Gamma \vdash N : \tau]\!] \;:\; \boldsymbol{s}[\![\Gamma]\!] \longrightarrow \boldsymbol{s}[\![\tau]\!]$$

*in* $\mathcal{S}$.

Specialising the Extensional Normalisation Lemma for the canonical interpretation of types in the free cartesian closed category generated by them we have the following syntactic result (*cf.* [31]).

COROLLARY 5. *Every simply typed term $\beta\eta$-equals one in long $\beta\eta$-normal form.*

The above development does not give information about the long $\beta\eta$-normal form associated to a term because Kripke relations are extensional predicates. What is needed instead for this purpose is a notion of intensional Kripke relation in which the extension of the predicate is witnessed (or realised). Technically, this amounts to revisiting the above in categories obtained by the glueing construction [35]. This will be done in Part II, where to do it at an appropriate abstract, syntax-independent level we will first consider the typed lambda calculus algebraically.

# PART II

## II.1 Algebraic typed lambda calculus

We provide an algebraic setting for the syntax and semantics of the typed lambda calculus following the theory of [17]. In particular, we describe the typed abstract syntax of simply typed and of neutral and normal terms as initial algebras, and show how the usual semantics corresponds to unique algebra homomorphisms from the initial (term) algebras to suitable semantic algebras.

### II.1.1 Syntax

Categories of contexts, which we study next, play a crucial role in describing abstract syntax with variable binding; see [17] for further details.

**Free (co)cartesian categories.** The category of untyped contexts $\mathbb{F}$ with objects given by finite subsets of (the countably infinite set of) variables and morphisms given by all functions is the free cocartesian category on one generator.

More generally, the free cocartesian category over a set $\mathcal{T}$ can be described as the comma category $\mathbb{F}\downarrow\mathcal{T}$ of contexts with types in the set $\mathcal{T}$ and type-preserving context renamings. (That is, $\mathbb{F}\downarrow\mathcal{T}$ is the category with objects given by maps $\Gamma : V \longrightarrow \mathcal{T}$ where $V$ is in $\mathbb{F}$, and with morphisms $\rho : \Gamma \longrightarrow \Gamma'$ given by functions $\rho : \mathrm{dom}(\Gamma) \longrightarrow \mathrm{dom}(\Gamma')$ such that $\Gamma = \Gamma' \circ \rho$.) The initial object $(0 \longrightarrow \mathcal{T})$ in $\mathbb{F}\downarrow\mathcal{T}$ is the empty context; whilst the coproduct

$$(V \xrightarrow{\Gamma} \mathcal{T}) + (V' \xrightarrow{\Gamma'} \mathcal{T}) \quad = \quad (V + V' \xrightarrow{[\Gamma,\Gamma']} \mathcal{T})$$

in $\mathbb{F}\downarrow\mathcal{T}$ amounts to the operation of context extension.

As before, we write $\mathbb{F}[\mathcal{T}]$ for $(\mathbb{F}\downarrow\mathcal{T})^{\mathrm{op}}$. Further, we write $\langle\_\rangle : \mathcal{T} \longrightarrow \mathbb{F}[\mathcal{T}]$ for the universal embedding (mapping $\tau$ to $(1 \xrightarrow{\tau} \mathcal{T})$) exhibiting $\mathbb{F}[\mathcal{T}]$ as the free cartesian category over $\mathcal{T}$.

**Typed abstract syntax with variable binding.** The semantic universe on which to consider the algebras for the typed lambda calculus over a set of base types $\mathrm{T}$ is the functor category $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ of $\mathbb{F}\downarrow\widetilde{\mathrm{T}}$-variable sets, referred to as (covariant) presheaves. (Recall that $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ has objects given by functors $\mathbb{F}\downarrow\widetilde{\mathrm{T}} \longrightarrow \mathbf{Set}$ and morphisms $\varphi : P \longrightarrow P'$ given by natural transformations; that is, families of functions $\varphi = \{\, \varphi_\Gamma : P(\Gamma) \longrightarrow P'(\Gamma) \,\}_{\Gamma\in|\mathbb{F}\downarrow\widetilde{\mathrm{T}}|}$ such that $\varphi_{\Gamma'} \circ P(\rho) = P'(\rho) \circ \varphi_\Gamma$ for all $\rho : \Gamma \longrightarrow \Gamma'$ in $\mathbb{F}\downarrow\widetilde{\mathrm{T}}$.)

The structure of $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ allowing the interpretation of variables and binding operators is described below.

- The presheaf of variables of type $\tau \in \widetilde{\mathrm{T}}$ is $\mathrm{V}_\tau = \mathbf{y}\langle\tau\rangle$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ where

$$\begin{array}{ccc} \mathbb{F}[\widetilde{\mathrm{T}}] & \xhookrightarrow{\mathbf{y}} & \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}} \\ \Gamma & \longmapsto & (\mathbb{F}\downarrow\widetilde{\mathrm{T}})(\Gamma,\_) \end{array}$$

  is the Yoneda embedding.

  Hence, $\mathrm{V}_\tau(\Gamma) \cong \{\, x \mid (x:\tau) \in \Gamma \,\}$.

- For every type $\tau \in \widetilde{\mathrm{T}}$, the parameterisation functor $\_ \times \langle\tau\rangle : \mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathbb{F}[\widetilde{\mathrm{T}}]$ induces the following situation

$$\begin{array}{ccc} \mathbb{F}[\widetilde{\mathrm{T}}] & \xhookrightarrow{\quad\mathbf{y}\quad} & \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}} \\ {\scriptstyle \_\times\langle\tau\rangle}\downarrow & \overset{\mathrm{Lan}}{\cong} & {\scriptstyle \_\times\mathbf{y}\langle\tau\rangle}\downarrow \dashv \uparrow{\scriptstyle \mathbf{Set}^{(\_+\langle\tau\rangle)}} \\ \mathbb{F}[\widetilde{\mathrm{T}}] & \xhookrightarrow[\quad\mathbf{y}\quad]{} & \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}} \end{array}$$

  Thus, in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$, the exponential $P^{\mathrm{V}_\tau}$ of a presheaf $P$ can be explicitly described as $P(\_ + \langle\tau\rangle)$.

  Hence, $P^{\mathrm{V}_\tau}(\Gamma) \cong P(\Gamma + \langle\tau\rangle)$.

A *typed lambda algebra* over a set of base types $\mathrm{T}$ is a $\widetilde{\mathrm{T}}$-sorted algebra with carrier given by a family $\{\, \mathfrak{X}_\tau \,\}_{\tau\in\widetilde{\mathrm{T}}}$ of presheaves in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ equipped with the operations

| | |
|---|---|
| (Variables) | $\mathrm{V}_\tau \longrightarrow \mathfrak{X}_\tau$ |
| (Unit) | $1 \longrightarrow \mathfrak{X}_1$ |
| (First Projection) | $\mathfrak{X}_{\tau*\tau'} \longrightarrow \mathfrak{X}_\tau$ |
| (Second Projection) | $\mathfrak{X}_{\tau'*\tau} \longrightarrow \mathfrak{X}_\tau$ |
| (Pairing) | $\mathfrak{X}_\tau \times \mathfrak{X}_{\tau'} \longrightarrow \mathfrak{X}_{\tau*\tau'}$ |
| (Application) | $\mathfrak{X}_{\tau'\Rightarrow\tau} \times \mathfrak{X}_{\tau'} \longrightarrow \mathfrak{X}_\tau$ |
| (Abstraction) | $(\mathfrak{X}_{\tau'})^{\mathrm{V}_\tau} \longrightarrow \mathfrak{X}_{\tau\Rightarrow\tau'}$ |

Informally, one thinks of the sets $\mathfrak{X}_\tau(\Gamma)$ ($\tau \in \widetilde{\mathrm{T}}$, $\Gamma \in |\mathbb{F}\downarrow\widetilde{\mathrm{T}}|$) as the $\tau$-sorted elements of the algebra $\mathfrak{X}$ in the context $\Gamma$. Note that under this interpretation the abstraction operation corresponds to a natural family of mappings

$$\mathfrak{X}_{\tau'}(\Gamma + \langle\tau\rangle) \longrightarrow \mathfrak{X}_{\tau\Rightarrow\tau'}(\Gamma)$$

associating an element of sort $\tau'$ in the context $\Gamma + \langle\tau\rangle$ (that is, the context $\Gamma$ extended with a fresh variable of type $\tau$) with an element of sort $\tau\Rightarrow\tau'$ in context $\Gamma$.

In the tradition of categorical algebra, the category of typed lambda algebras can be defined as the category of $\Sigma$-algebras for a signature endofunctor $\Sigma$ on $(\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}})^{\widetilde{\mathrm{T}}}$. This endofunctor is induced by the above operations as follows

$$\begin{aligned} (\Sigma\mathfrak{X})_\theta &= \mathrm{V}_\theta + \mathrm{E}_\theta(\mathfrak{X}) \\ (\Sigma\mathfrak{X})_1 &= \mathrm{V}_1 + 1 + \mathrm{E}_1(\mathfrak{X}) \end{aligned}$$

$$(\Sigma\mathfrak{X})_{\tau*\tau'} = \mathrm{V}_{\tau*\tau'} + (\mathfrak{X}_\tau \times \mathfrak{X}_{\tau'}) + \mathrm{E}_{\tau*\tau'}(\mathfrak{X})$$
$$(\Sigma\mathfrak{X})_{\tau\Rightarrow\tau'} = \mathrm{V}_{\tau\Rightarrow\tau'} + (\mathfrak{X}_{\tau'})^{\mathrm{V}_\tau} + \mathrm{E}_{\tau\Rightarrow\tau'}(\mathfrak{X})$$

with

$$\mathrm{E}_\tau(\mathfrak{X}) = \coprod_{\tau'} \mathfrak{X}_{\tau*\tau'} + \mathfrak{X}_{\tau'*\tau} + (\mathfrak{X}_{\tau'\Rightarrow\tau} \times \mathfrak{X}_{\tau'})$$

where $\theta \in \mathrm{T}$ and $\tau, \tau' \in \widetilde{\mathrm{T}}$.

The initial $\Sigma$-algebra $\mathfrak{L} = \{ \mathfrak{L}_\tau \}_{\tau \in \widetilde{\mathrm{T}}}$ with its structure

$$\mathrm{V}_\theta + \mathrm{E}_\theta(\mathfrak{L}) \xrightarrow{\cong} \mathfrak{L}_\theta$$
$$\mathrm{V}_1 + 1 + \mathrm{E}_1(\mathfrak{L}) \xrightarrow{\cong} \mathfrak{L}_1$$
$$\mathrm{V}_{\tau*\tau'} + (\mathfrak{L}_\tau \times \mathfrak{L}_{\tau'}) + \mathrm{E}_{\tau*\tau'}(\mathfrak{L}) \xrightarrow{\cong} \mathfrak{L}_{\tau*\tau'}$$
$$\mathrm{V}_{\tau\Rightarrow\tau'} + (\mathfrak{L}_{\tau'})^{\mathrm{V}_\tau} + \mathrm{E}_{\tau\Rightarrow\tau'}(\mathfrak{L}) \xrightarrow{\cong} \mathfrak{L}_{\tau\Rightarrow\tau'}$$

can be explicitly described as the family of presheaves of terms

$$\mathfrak{L}_\tau(\Gamma) = \{ t \mid \Gamma \vdash t : \tau \}$$

with presheaf action given by variable renaming (*i.e.*, by the mapping associating $\Gamma \vdash t : \tau$ to $\Gamma' \vdash t[{}^{\rho x}/_{x}]_{x \in \mathrm{dom}(\Gamma)} : \tau$ for any $\rho : \Gamma \longrightarrow \Gamma'$ in $\mathbb{F}{\downarrow}\widetilde{\mathrm{T}}$), and with operations

$$\begin{array}{lcl}
\mathsf{var}_\tau & : & \mathrm{V}_\tau \longrightarrow \mathfrak{L}_\tau \\
\mathsf{unit}_1 & : & 1 \longrightarrow \mathfrak{L}_1 \\
\mathsf{fst}_\tau^{(\tau')} & : & \mathfrak{L}_{\tau*\tau'} \longrightarrow \mathfrak{L}_\tau \\
\mathsf{snd}_\tau^{(\tau')} & : & \mathfrak{L}_{\tau'*\tau} \longrightarrow \mathfrak{L}_\tau \\
\mathsf{pair}_{\tau*\tau'} & : & \mathfrak{L}_\tau \times \mathfrak{L}_{\tau'} \longrightarrow \mathfrak{L}_{\tau*\tau'} \\
\mathsf{app}_\tau^{(\tau')} & : & \mathfrak{L}_{\tau'\Rightarrow\tau} \times \mathfrak{L}_{\tau'} \longrightarrow \mathfrak{L}_\tau \\
\mathsf{abs}_{\tau\Rightarrow\tau'} & : & (\mathfrak{L}_{\tau'})^{\mathrm{V}_\tau} \longrightarrow \mathfrak{L}_{\tau\Rightarrow\tau'}
\end{array}$$

corresponding to the typing rules in Figure 1.

A full theory of typed abstract syntax with variable binding incorporating substitution along the lines of [17] can be developed. This is not necessary for the purposes of the paper and hence will not be pursued here.

The notions of neutral and normal terms are given by mutual induction (see Figure 2) and, as such, the associated algebraic notion corresponds to considering a signature endofunctor on the product category $(\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}})^{\widetilde{\mathrm{T}}} \times (\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}})^{\widetilde{\mathrm{T}}}$. This endofunctor, with components $\langle \Sigma_1, \Sigma_2 \rangle$, is defined below.

$$(\Sigma_1(\mathfrak{X},\mathfrak{Y}))_\tau = \mathrm{V}_\tau + \mathrm{E}_\tau(\mathfrak{X},\mathfrak{Y})$$
$$(\Sigma_2(\mathfrak{X},\mathfrak{Y}))_\theta = \mathrm{V}_\theta + \mathrm{E}_\theta(\mathfrak{X},\mathfrak{Y})$$

where $\mathrm{E}_\tau(\mathfrak{X},\mathfrak{Y}) = \coprod_{\tau' \in \widetilde{\mathrm{T}}} \mathfrak{X}_{\tau*\tau'} + \mathfrak{X}_{\tau'*\tau} + (\mathfrak{X}_{\tau'\Rightarrow\tau} \times \mathfrak{Y}_{\tau'})$ and

$$(\Sigma_2(\mathfrak{X},\mathfrak{Y}))_1 = 1$$
$$(\Sigma_2(\mathfrak{X},\mathfrak{Y}))_{\tau*\tau'} = \mathfrak{Y}_\tau \times \mathfrak{Y}_{\tau'}$$
$$(\Sigma_2(\mathfrak{X},\mathfrak{Y}))_{\tau\Rightarrow\tau'} = (\mathfrak{Y}_{\tau'})^{\mathrm{V}_\tau}$$

We write $(\mathfrak{M},\mathfrak{N})$ for the initial $\langle \Sigma_1, \Sigma_2 \rangle$-algebra

$$\left\{ \mathrm{V}_\tau + \mathrm{E}_\tau(\mathfrak{M},\mathfrak{N}) \xrightarrow{\cong} \mathfrak{M}_\tau \right.$$

$$\left\{ \begin{array}{rcl}
\mathrm{V}_\theta + \mathrm{E}_\theta(\mathfrak{M},\mathfrak{N}) & \xrightarrow{\cong} & \mathfrak{N}_\theta \\
1 & \xrightarrow{\cong} & \mathfrak{N}_1 \\
\mathfrak{N}_\tau \times \mathfrak{N}_{\tau'} & \xrightarrow{\cong} & \mathfrak{N}_{\tau*\tau'} \\
(\mathfrak{N}_{\tau'})^{\mathrm{V}_\tau} & \xrightarrow{\cong} & \mathfrak{N}_{\tau\Rightarrow\tau'}
\end{array} \right.$$

Explicitly, the presheaves $\mathfrak{M}_\tau$ and $\mathfrak{N}_\tau$ can be described as the neutral and normal terms

$$\mathfrak{M}_\tau(\Gamma) = \{ M \mid \Gamma \vdash_{\mathcal{M}} M : \tau \}$$
$$\mathfrak{N}_\tau(\Gamma) = \{ N \mid \Gamma \vdash_{\mathcal{N}} N : \tau \}$$

with presheaf action given by variable renaming, and with operations

$$\left\{ \begin{array}{lcl}
\mathsf{var}_\tau & : & \mathrm{V}_\tau \longrightarrow \mathfrak{M}_\tau \\
\mathsf{fst}_\tau^{(\tau')} & : & \mathfrak{M}_{\tau*\tau'} \longrightarrow \mathfrak{M}_\tau \\
\mathsf{snd}_\tau^{(\tau')} & : & \mathfrak{M}_{\tau'*\tau} \longrightarrow \mathfrak{M}_\tau \\
\mathsf{app}_\tau^{(\tau')} & : & \mathfrak{M}_{\tau'\Rightarrow\tau} \times \mathfrak{N}_{\tau'} \longrightarrow \mathfrak{M}_\tau
\end{array} \right.$$

$$\left\{ \begin{array}{lcl}
\mathsf{var}_\theta & : & \mathrm{V}_\theta \longrightarrow \mathfrak{N}_\theta \\
\mathsf{fst}_\theta^{(\tau')} & : & \mathfrak{M}_{\theta*\tau'} \longrightarrow \mathfrak{N}_\theta \\
\mathsf{snd}_\theta^{(\tau')} & : & \mathfrak{M}_{\tau'*\theta} \longrightarrow \mathfrak{N}_\theta \\
\mathsf{app}_\theta^{(\tau')} & : & \mathfrak{M}_{\tau'\Rightarrow\theta} \times \mathfrak{N}_{\tau'} \longrightarrow \mathfrak{N}_\theta \\
\mathsf{unit}_1 & : & 1 \xrightarrow{\cong} \mathfrak{N}_1 \\
\mathsf{pair}_{\tau*\tau'} & : & \mathfrak{N}_\tau \times \mathfrak{N}_{\tau'} \xrightarrow{\cong} \mathfrak{N}_{\tau*\tau'} \\
\mathsf{abs}_{\tau\Rightarrow\tau'} & : & (\mathfrak{N}_{\tau'})^{\mathrm{V}_\tau} \xrightarrow{\cong} \mathfrak{N}_{\tau\Rightarrow\tau'}
\end{array} \right.$$

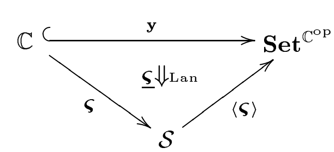
corresponding to the typing rules in Figure 2.

Note that every $\Sigma$-algebra $\mathfrak{X}$ induces a canonical $\langle \Sigma_1, \Sigma_2 \rangle$-algebra structure on the pair $(\mathfrak{X},\mathfrak{X})$ and hence, by initiality, homomorphic interpretations $(\mathfrak{M},\mathfrak{N}) \longrightarrow (\mathfrak{X},\mathfrak{X})$. Applying this observation to $\mathfrak{L}$ we obtain the embeddings $\mathfrak{M} \rightarrowtail \mathfrak{L}$ and $\mathfrak{N} \rightarrowtail \mathfrak{L}$ of neutral and normal terms into terms.

### II.1.2 Semantics

As we will see below, every interpretation of base types in a cartesian closed category induces a canonical semantic typed lambda algebra with respect to which the unique algebra homomorphism from the initial (term) algebra is the usual semantics of simply typed terms.

**Relative hom-functor.** Every functor $\varsigma : \mathbb{C} \longrightarrow \mathcal{S}$ induces the following situation



(3)

where $\langle\varsigma\rangle(A) = \mathcal{S}(\varsigma(\_), A)$ and where $(\underline{\varsigma}_\Gamma)_{\Gamma'} = \varsigma_{\Gamma',\Gamma} : \mathbb{C}(\Gamma', \Gamma) \longrightarrow \mathcal{S}(\varsigma(\Gamma'), \varsigma(\Gamma))$.

Two important properties of the *relative hom-functor* $\langle\varsigma\rangle$ are that it preserves limits and that, for $\varsigma$ and $\mathbb{C}$ cartesian and $\mathcal{S}$ cartesian closed, it commutes with exponentiation by representables in the sense that there is a canonical natural isomorphism

$$\begin{array}{ccc} \mathcal{S} & \xrightarrow{\varsigma(\Gamma)\Rightarrow(\_)} & \mathcal{S} \\ {\scriptstyle\langle\varsigma\rangle}\downarrow & \cong & \downarrow{\scriptstyle\langle\varsigma\rangle} \\ \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} & \xrightarrow[(\_)^{\mathbf{y}(\Gamma)}]{} & \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \end{array}$$

for all $\Gamma \in |\mathbb{C}|$.

**Initial algebra semantics.** Using the relative hom-functor $\langle \boldsymbol{s}\rangle : \mathcal{S} \longrightarrow \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ induced by the cartesian extension $\boldsymbol{s}[\![\_]\!] : \mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}$ of an interpretation $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ of base types in a cartesian closed category, the operations

$$\begin{array}{lll} \pi_1 & : & \boldsymbol{s}[\![\tau]\!] \times \boldsymbol{s}[\![\tau']\!] \longrightarrow \boldsymbol{s}[\![\tau]\!] \\ \pi_2 & : & \boldsymbol{s}[\![\tau']\!] \times \boldsymbol{s}[\![\tau]\!] \longrightarrow \boldsymbol{s}[\![\tau]\!] \\ \varepsilon & : & (\boldsymbol{s}[\![\tau]\!] \Rightarrow \boldsymbol{s}[\![\tau']\!]) \times \boldsymbol{s}[\![\tau]\!] \longrightarrow \boldsymbol{s}[\![\tau']\!] \end{array}$$

in $\mathcal{S}$ can be lifted to $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ to provide a semantic typed lambda algebra structure on the family

$$\mathfrak{C} \;=\; \{\, \mathcal{S}(\boldsymbol{s}[\![\_]\!], \boldsymbol{s}[\![\tau]\!]) \,\}_{\tau\in\widetilde{\mathrm{T}}} \;=\; \{\, \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau]\!]) \,\}_{\tau\in\widetilde{\mathrm{T}}} \quad .$$

Indeed, the operations are as follows.

- $\mathrm{V}_\tau \xrightarrow{\boldsymbol{s}[\![\_]\!]} \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau]\!])$
- $1 \xrightarrow{\cong} \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![1]\!])$
- $\langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau * \tau']\!]) \xrightarrow{\langle \boldsymbol{s}\rangle(\pi_1)} \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau]\!])$
- $\langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau' * \tau]\!]) \xrightarrow{\langle \boldsymbol{s}\rangle(\pi_2)} \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau]\!])$
- $\langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau]\!]) \times \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau']\!]) \xrightarrow{\cong} \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau * \tau']\!])$
- $\langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau' \Rightarrow \tau]\!]) \times \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau']\!])$
  $\cong \langle \boldsymbol{s}\rangle((\boldsymbol{s}[\![\tau']\!] \Rightarrow \boldsymbol{s}[\![\tau]\!]) \times \boldsymbol{s}[\![\tau']\!]) \xrightarrow{\langle \boldsymbol{s}\rangle(\varepsilon)} \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau]\!])$
- $(\langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau']\!]))^{\mathrm{V}_\tau} \xrightarrow{\cong} \langle \boldsymbol{s}\rangle(\boldsymbol{s}[\![\tau \Rightarrow \tau']\!])$

The above algebra structure induces semantic homomorphic interpretations $\ell : \mathfrak{L} \longrightarrow \mathfrak{C}$ and $(m, n) : (\mathfrak{M}, \mathfrak{N}) \longrightarrow (\mathfrak{C}, \mathfrak{C})$ related as shown below

$$\begin{array}{ccccc} \mathfrak{M} & \rightarrowtail & \mathfrak{L} & \leftarrowtail & \mathfrak{N} \\ & {\scriptstyle m}\searrow & \downarrow{\scriptstyle \ell} & \swarrow{\scriptstyle n} & \\ & & \mathfrak{C} & & \end{array} \tag{4}$$

Explicitly, for $\tau \in \widetilde{\mathrm{T}}$, the mapping $\ell_\tau : \mathfrak{L}_\tau \longrightarrow \mathfrak{C}_\tau$ is the semantic interpretation of terms

$$t \in \mathfrak{L}_\tau(\Gamma) \;\overset{\ell_\tau}{\longmapsto}\; \boldsymbol{s}[\![\Gamma \vdash t : \tau]\!] \in \mathcal{S}(\boldsymbol{s}[\![\Gamma]\!], \boldsymbol{s}[\![\tau]\!]) \quad ,$$

whilst $m_\tau : \mathfrak{M}_\tau \longrightarrow \mathfrak{C}_\tau$ and $n_\tau : \mathfrak{N}_\tau \longrightarrow \mathfrak{C}_\tau$ are, respectively, the semantic interpretations of neutral and normal terms.

## II.2 Normalisation by evaluation via glueing

We will now see how by working with intensional Kripke relations, the analysis of normalisation given in Section I.2 amounts to normalisation by evaluation.

**Intensional Kripke relations.** The category of intensional $\mathbb{C}$-Kripke relations of arity $\varsigma : \mathbb{C} \longrightarrow \mathcal{S}$ is defined as the *glueing* of $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$ and $\mathcal{S}$ along the relative hom-functor $\langle\varsigma\rangle : \mathcal{S} \longrightarrow \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$. That is, as the comma category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \downarrow \langle\varsigma\rangle$ of objects given by triples $(P, p, A)$ with $P \in |\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}|$, $A \in |\mathcal{S}|$, and $p : P \longrightarrow \langle\varsigma\rangle(A)$ in $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$, and of morphisms $(P, p, A) \longrightarrow (P', p', A')$ given by pairs

$$(\ \varphi : P \longrightarrow P' \text{ in } \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}},\ \ f : A \longrightarrow A' \text{ in } \mathcal{S}\ )$$

such that the square

$$\begin{array}{ccc} P & \xrightarrow{\varphi} & P' \\ {\scriptstyle p}\downarrow & & \downarrow{\scriptstyle p'} \\ \langle\varsigma\rangle(A) & \xrightarrow[\langle\varsigma\rangle(f)]{} & \langle\varsigma\rangle(A') \end{array} \qquad \text{in } \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$$

commutes.

The category of Kripke relations $\underline{\mathrm{K}}\langle\varsigma\rangle$ is a full subcategory of the glueing category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \downarrow \langle\varsigma\rangle$ via the mapping $(R, A) \longmapsto (R, R \hookrightarrow \langle\varsigma\rangle(A), A)$. On the other hand, every glued object $(P, f, A)$ has an associated Kripke relation given by the extension of the map $f$ (as shown in the diagram below, where $\mathrm{im}(f)$ denotes the image of $f$)

$$\begin{array}{ccc} P & & \\ {\scriptstyle f}\downarrow & \searrow & \\ & & \mathrm{im}(f) \\ \downarrow & \swarrow & \\ \langle\varsigma\rangle(A) & & \end{array}$$

and the mapping $(P, f, A) \longmapsto (\mathrm{im}(f), A)$ exhibits $\underline{\mathrm{K}}\langle\varsigma\rangle$ as a reflective subcategory of $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \downarrow \langle\varsigma\rangle$.

As it is well-known (see, *e.g.*, [23, 9, 34]), for $\mathcal{S}$ cartesian closed, the glueing category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \downarrow \langle\varsigma\rangle$ is also cartesian closed, and the category of Kripke relations $\underline{\mathrm{K}}\langle\varsigma\rangle$ is an exponential ideal of it. Indeed, the cartesian closed structure of $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \downarrow \langle\varsigma\rangle$ is given as follows.

(Products) The terminal object is $(1, t, 1)$ where $t$ is the unique map $1 \xrightarrow{\cong} \langle\varsigma\rangle(1)$.

The binary product $(P, p, A) \times (Q, q, B)$ of $(P, p, A)$ and $(Q, q, B)$ is $(P \times Q, r, A \times B)$ where $r$ is the composite $P \times Q \xrightarrow{p \times q} \langle\varsigma\rangle(A) \times \langle\varsigma\rangle(B) \xrightarrow{\cong} \langle\varsigma\rangle(A \times B)$.

(Exponentials) The exponential $(P, p, A) \Rightarrow (Q, q, B)$ of $(P, p, A)$ and $(Q, q, B)$ is $(R, r, A \Rightarrow B)$ in the pullback diagram

$$\begin{array}{ccc} R & \xrightarrow{\qquad\qquad} & Q^P \\ {\scriptstyle r}\downarrow & \text{pb} & \downarrow{\scriptstyle q^P} \\ \langle\varsigma\rangle(A \Rightarrow B) \longrightarrow (\langle\varsigma\rangle B)^{(\langle\varsigma\rangle A)} & \xrightarrow[(\langle\varsigma\rangle B)^p]{} & (\langle\varsigma\rangle B)^P \end{array}$$

where the map $\langle\varsigma\rangle(A \Rightarrow B) \longrightarrow (\langle\varsigma\rangle B)^{(\langle\varsigma\rangle A)}$ is the exponential transpose of the composite

$$\begin{aligned}&\langle\varsigma\rangle(A \Rightarrow B) \times \langle\varsigma\rangle(A)\\ &\cong \langle\varsigma\rangle((A \Rightarrow B) \times A) \xrightarrow{\langle\varsigma\rangle(\varepsilon)} \langle\varsigma\rangle(B) \quad .\end{aligned}$$

PROPOSITION 6. *Let $\mathbb{C}$ be a small category and let $\mathcal{S}$ be a cartesian closed category. For a functor $\varsigma : \mathbb{C} \longrightarrow \mathcal{S}$, the glueing category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\!\langle\varsigma\rangle$ is cartesian closed, and the forgetful functor $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\!\langle\varsigma\rangle \longrightarrow \mathcal{S} : (P, p, A) \longmapsto A$ preserves the cartesian closed structure.*

Notice that the situation (3) induces the embedding

$$\begin{array}{ccc}\mathbb{C} & \overset{\overline{\mathbf{y}}}{\hookrightarrow} & \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\!\langle\varsigma\rangle\\ \Gamma & \longmapsto & (\ \mathbf{y}(\Gamma)\ ,\ \mathbf{y}(\Gamma) \xrightarrow{\varsigma_\Gamma} \langle\varsigma\rangle(\varsigma\Gamma)\ ,\ \varsigma(\Gamma)\ )\end{array}$$

extending the Yoneda embedding $\mathbf{y} : \mathbb{C} \hookrightarrow \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$ and the functor $\varsigma : \mathbb{C} \longrightarrow \mathcal{S}$

$$\begin{array}{ccccc} & & \mathbb{C} & & \\ & \swarrow^{\mathbf{y}} & \downarrow \overline{\mathbf{y}} & \searrow^{\varsigma} & \\ \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} & \longleftarrow & \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\!\langle\varsigma\rangle & \longrightarrow & \mathcal{S}\\ P & \xleftarrow{\varpi} & (P, p, A) & \xmapsto{\pi} & A\end{array}$$

and satisfying the following form of the Yoneda lemma

$$\begin{array}{ccc}(\varphi, f) & \longmapsto & \varphi(\mathrm{id})\\ {}[\overline{\mathbf{y}}(\_), (P, p, A)] & \xrightarrow{\cong} & P(\_)\\ & \searrow^{\pi} \quad \swarrow_{p} & \\ & \mathcal{S}(\varsigma(\_), A) & \end{array}$$

where $[\_,=]$ denotes the hom-functor of the glueing category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\!\langle\varsigma\rangle$.

Further, for $\mathbb{C}$ and $\varsigma$ cartesian and $\mathcal{S}$ cartesian closed, we have that $\overline{\mathbf{y}}$ preserves products and that the exponential $(P, p, A)^{\overline{\mathbf{y}}(\Gamma)}$ in $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\!\langle\varsigma\rangle$ can be simply described as $(P^{\mathbf{y}(\Gamma)}, p', \varsigma(\Gamma) \Rightarrow A)$ where $p'$ is the composite

$$P^{\mathbf{y}(\Gamma)} \xrightarrow{p^{\mathbf{y}(\Gamma)}} (\langle\varsigma\rangle A)^{\mathbf{y}(\Gamma)} \xrightarrow{\cong} \langle\varsigma\rangle(\varsigma(\Gamma) \Rightarrow A) \quad .$$

**Glueing syntax and semantics.** Let $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ be an interpretation of base types in a cartesian closed category. The embedding $\overline{\mathbf{y}} : \mathbb{F}[\widetilde{\mathrm{T}}] \hookrightarrow \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$ restricted to types $\tau \in \widetilde{\mathrm{T}}$, yields the object

$$\nu_\tau \ = \ \overline{\mathbf{y}}\langle\tau\rangle \ = \ (\ \mathrm{V}_\tau\ ,\ \mathrm{V}_\tau \xrightarrow{\boldsymbol{s}[\![\_]\!]} \mathfrak{C}_\tau\ ,\ \boldsymbol{s}[\![\tau]\!]\ )$$

in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$ glueing the syntax and semantics of variables. In the same spirit, glueing the syntax and semantics of neutral and normal terms (see (4)) we obtain the glued objects

$$\begin{aligned}\mu_\tau &= (\ \mathfrak{M}_\tau\ ,\ \mathfrak{M}_\tau \xrightarrow{m_\tau} \mathfrak{C}_\tau\ ,\ \boldsymbol{s}[\![\tau]\!]\ )\\ \eta_\tau &= (\ \mathfrak{N}_\tau\ ,\ \mathfrak{N}_\tau \xrightarrow{n_\tau} \mathfrak{C}_\tau\ ,\ \boldsymbol{s}[\![\tau]\!]\ )\end{aligned}$$

in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$.

Having constructed the $\langle\Sigma_1, \Sigma_2\rangle$-algebra structure on $(\mathfrak{C}, \mathfrak{C})$ as the lifting of operations in $\mathcal{S}$, the homomorphism property of the semantic interpretation $(m, n) : (\mathfrak{M}, \mathfrak{N}) \longrightarrow (\mathfrak{C}, \mathfrak{C})$ entails the two propositions below, which show how the algebraic operations on the initial $\langle\Sigma_1, \Sigma_2\rangle$-algebra $(\mathfrak{M}, \mathfrak{N})$ and on the semantic $\langle\Sigma_1, \Sigma_2\rangle$-algebra $(\mathfrak{C}, \mathfrak{C})$ can be glued to yield operations in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$ on the family of glued objects $(\{\,\mu_\tau\,\}_{\tau\in\widetilde{\mathrm{T}}}\ ,\ \{\,\eta_\tau\,\}_{\tau\in\widetilde{\mathrm{T}}})$.

PROPOSITION 7. *Let $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ be an interpretation of base types in a cartesian closed category.*

1. *For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the pair of maps*

   $$\mathsf{var}_\tau : \mathrm{V}_\tau \longrightarrow \mathfrak{M}_\tau\ ,\ \mathrm{id}_{\boldsymbol{s}[\![\tau]\!]}$$

   *constitute a map $\nu_\tau \longrightarrow \mu_\tau$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$.*

2. *For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the pair of maps*

   $$\mathsf{fst}^{(\tau')}_\tau : \mathfrak{M}_{\tau*\tau'} \longrightarrow \mathfrak{M}_\tau\ ,\ \pi_1 : \boldsymbol{s}[\![\tau]\!] \times \boldsymbol{s}[\![\tau']\!] \longrightarrow \boldsymbol{s}[\![\tau]\!]$$

   *constitute a map $\mu_{\tau*\tau'} \longrightarrow \mu_\tau$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$.*

3. *For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the pair of maps*

   $$\mathsf{snd}^{(\tau')}_\tau : \mathfrak{M}_{\tau'*\tau} \longrightarrow \mathfrak{M}_\tau\ ,\ \pi_2 : \boldsymbol{s}[\![\tau']\!] \times \boldsymbol{s}[\![\tau]\!] \longrightarrow \boldsymbol{s}[\![\tau]\!]$$

   *constitute a map $\mu_{\tau'*\tau} \longrightarrow \mu_\tau$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$.*

4. *For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the pair of maps*

   $$\begin{aligned}&\mathsf{app}^{(\tau')}_\tau : \mathfrak{M}_{\tau'\Rightarrow\tau} \times \mathfrak{N}_{\tau'} \longrightarrow \mathfrak{M}_\tau\ ,\\ &\varepsilon : (\boldsymbol{s}[\![\tau']\!] \Rightarrow \boldsymbol{s}[\![\tau]\!]) \times \boldsymbol{s}[\![\tau']\!] \longrightarrow \boldsymbol{s}[\![\tau]\!]\end{aligned}$$

   *constitute a map $\mu_{\tau'\Rightarrow\tau} \times \eta_{\tau'} \longrightarrow \mu_\tau$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$.*

PROPOSITION 8. *Let $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ be an interpretation of base types in a cartesian closed category.*

1. *For a base type $\theta \in \mathrm{T}$, the pair of isomorphisms*

   $$\mathfrak{M}_\theta \cong \mathrm{V}_\theta + \mathrm{E}_\theta(\mathfrak{M}, \mathfrak{N}) \cong \mathfrak{N}_\theta\ ,\ \mathrm{id}_{\boldsymbol{s}(\theta)}$$

   *constitute an isomorphism $\mu_\theta \cong \eta_\theta$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$.*

2. *The pair of isomorphisms*

   $$\mathsf{unit}_1 : 1 \xrightarrow{\cong} \mathfrak{N}_1\ ,\ \mathrm{id}_1$$

   *constitute an isomorphism $1 \xrightarrow{\cong} \eta_1$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$.*

3. *For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the pair of isomorphisms*

   $$\mathsf{pair}_{\tau*\tau'} : \mathfrak{N}_\tau \times \mathfrak{N}_{\tau'} \xrightarrow{\cong} \mathfrak{N}_{\tau*\tau'}\ ,\ \mathrm{id}_{\boldsymbol{s}[\![\tau]\!] \times \boldsymbol{s}[\![\tau']\!]}$$

   *constitute an isomorphism $\eta_\tau \times \eta_{\tau'} \xrightarrow{\cong} \eta_{\tau*\tau'}$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$.*

4. *For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the pair of isomorphisms*

   $$\mathsf{abs}_{\tau\Rightarrow\tau'} : {\mathfrak{N}_{\tau'}}^{\mathrm{V}_\tau} \xrightarrow{\cong} \mathfrak{N}_{\tau\Rightarrow\tau'}\ ,\ \mathrm{id}_{\boldsymbol{s}[\![\tau]\!] \Rightarrow \boldsymbol{s}[\![\tau']\!]}$$

   *constitute an isomorphism ${\eta_{\tau'}}^{\nu_\tau} \xrightarrow{\cong} \eta_{\tau\Rightarrow\tau'}$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\!\downarrow\!\langle\boldsymbol{s}\rangle$.*

Note that the above operations on glued objects are given by pairs of syntactic operations over their corresponding semantic meaning in the case of the type destructors (Proposition 7) and over the identity for the type constructors (Proposition 8).

**Normalisation by evaluation.** Let $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ be an interpretation of base types in a cartesian closed category. Consider the interpretation

$$\begin{array}{ccc} \mathrm{T} & \xrightarrow{\;\overline{\boldsymbol{s}}\;} & \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\downarrow\langle \boldsymbol{s}\rangle \\ \theta & \longmapsto & \mu_\theta \end{array} \qquad (5)$$

By Proposition 6, the semantics of terms induced by $\overline{\boldsymbol{s}}$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\downarrow\langle \boldsymbol{s}\rangle$ extends the semantics induced by $\boldsymbol{s}$ in $\mathcal{S}$; that is, the denotation $\overline{\boldsymbol{s}}[\![\Gamma \vdash t : \tau]\!]$ is a pair of the form $(\boldsymbol{s}'[\![\Gamma \vdash t : \tau]\!], \boldsymbol{s}[\![\Gamma \vdash t : \tau]\!])$.

Writing $(\mathfrak{S}_\tau, \sigma_\tau, \boldsymbol{s}[\![\tau]\!])$ for $\overline{\boldsymbol{s}}[\![\tau]\!]$, we now aim at defining maps as shown below

$$\begin{array}{ccccc} \mathfrak{M}_\tau & \dashrightarrow & \mathfrak{S}_\tau & \dashrightarrow & \mathfrak{N}_\tau \\ & {\scriptstyle m_\tau}\searrow & \downarrow{\scriptstyle \sigma_\tau} & \swarrow{\scriptstyle n_\tau} & \\ & & \mathcal{S}(\boldsymbol{s}[\![\_]\!], \boldsymbol{s}[\![\tau]\!]) & & \end{array} \qquad (6)$$

so that, for all terms $\Gamma \vdash t : \tau$ ($\Gamma = \langle x_i : \tau_i\rangle_{i=1,n}$), the diagram

$$\begin{array}{ccccccc} \prod_{i=1,n}\mathfrak{M}_{\tau_i} & \dashrightarrow & \prod_{i=1,n}\mathfrak{S}_{\tau_i} & \xrightarrow{\boldsymbol{s}'[\![\Gamma \vdash t : \tau]\!]} & \mathfrak{S}_\tau & \dashrightarrow & \mathfrak{N}_\tau \\ & {\scriptstyle \prod_{i=1,n} m_{\tau_i}}\searrow & \downarrow{\scriptstyle \prod_{i=1,n}\sigma_{\tau_i}} & & \downarrow{\scriptstyle\sigma_\tau} & \swarrow{\scriptstyle n_\tau} & \\ & & \prod_{i=1,n}\mathcal{S}(\boldsymbol{s}[\![\_]\!], \boldsymbol{s}[\![\tau_i]\!]) & & & & \\ & & \downarrow\cong & & & & \\ & & \mathcal{S}(\boldsymbol{s}[\![\_]\!], \boldsymbol{s}[\![\Gamma]\!]) & \xrightarrow[\boldsymbol{s}[\![\Gamma \vdash t : \tau]\!] \circ \_]{} & \mathcal{S}(\boldsymbol{s}[\![\_]\!], \boldsymbol{s}[\![\tau]\!]) & & \end{array}$$

will commute, and hence the evaluation of the top composite at the tuple $\langle \mathsf{var}_{\tau_i}(x_i)\rangle_{i=1,n}$ of the variables in the context $\Gamma$ will yield a normal term in $\mathfrak{N}_\tau(\Gamma)$ with the same semantics as the given term $t$ (*cf.* the Extensional Normalisation Lemma in Section I.2). Moreover, as $\beta\eta$-equivalent terms have the same denotation, the normal forms associated to two such terms will be the same.

The abstract way to define the maps in (6) —which in the literature on normalisation by evaluation are either referred to as *unquote* and *quote* or as *reflect* and *reify*— is by defining translations

$$\mu_\tau \xrightarrow{\;\mathrm{u}_\tau\;} \overline{\boldsymbol{s}}[\![\tau]\!] \xrightarrow{\;\mathrm{q}_\tau\;} \eta_\tau \quad \text{in } \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\downarrow\langle \boldsymbol{s}\rangle$$

that project in $\mathcal{S}$ onto identities. The definitions are by induction on the structure of types as follows.

1. For a base type $\theta \in \mathrm{T}$, we define $\mathrm{u}_\theta = \mathrm{id}_{\mu_\theta}$ and $\mathrm{q}_\theta = (\mu_\theta \xrightarrow{\cong} \eta_\theta)$.

2. We let $\mathrm{u}_1 = (\mu_1 \longrightarrow 1)$ and $\mathrm{q}_1 = 1 \xrightarrow[\cong]{(\mathsf{unit}_1, \mathrm{id})} \eta_1$.

3. For types $\tau, \tau' \in \widetilde{\mathrm{T}}$, we define

$$\mathrm{u}_{\tau * \tau'} : \mu_{\tau*\tau'} \longrightarrow \overline{\boldsymbol{s}}[\![\tau]\!] \times \overline{\boldsymbol{s}}[\![\tau']\!]$$

as the pairing of the maps

$$\mu_{\tau*\tau'} \xrightarrow{(\mathsf{fst}^{(\tau')}_\tau, \pi_1)} \mu_\tau \xrightarrow{\;\mathrm{u}_\tau\;} \overline{\boldsymbol{s}}[\![\tau]\!]$$

and

$$\mu_{\tau*\tau'} \xrightarrow{(\mathsf{snd}^{(\tau)}_{\tau'}, \pi_2)} \mu_{\tau'} \xrightarrow{\;\mathrm{u}_{\tau'}\;} \overline{\boldsymbol{s}}[\![\tau']\!] \quad ,$$

and let $\mathrm{q}_{\tau*\tau'} : \overline{\boldsymbol{s}}[\![\tau]\!] \times \overline{\boldsymbol{s}}[\![\tau']\!] \longrightarrow \eta_{\tau*\tau'}$ be the composite

$$\overline{\boldsymbol{s}}[\![\tau]\!] \times \overline{\boldsymbol{s}}[\![\tau']\!] \xrightarrow{\mathrm{q}_\tau \times \mathrm{q}_{\tau'}} \eta_\tau \times \eta_{\tau'} \xrightarrow[\cong]{(\mathsf{pair}_{\tau*\tau'}, \mathrm{id})} \eta_{\tau*\tau'} \ .$$

4. For types $\tau, \tau' \in \widetilde{\mathrm{T}}$, we define

$$\mathrm{u}_{\tau\Rightarrow\tau'} : \mu_{\tau\Rightarrow\tau'} \longrightarrow \overline{\boldsymbol{s}}[\![\tau']\!]^{\overline{\boldsymbol{s}}[\![\tau]\!]}$$

as the exponential transpose of the map

$$\mu_{\tau\Rightarrow\tau'} \times \overline{\boldsymbol{s}}[\![\tau]\!] \xrightarrow{\mathrm{id}\times\mathrm{q}_\tau} \mu_{\tau\Rightarrow\tau'} \times \eta_\tau \xrightarrow{(\mathsf{app}^{(\tau)}_{\tau'}, \varepsilon)} \mu_{\tau'} \xrightarrow{\mathrm{u}_{\tau'}} \overline{\boldsymbol{s}}[\![\tau']\!] \ ,$$

and let $\mathrm{q}_{\tau\Rightarrow\tau'} : \overline{\boldsymbol{s}}[\![\tau']\!]^{\overline{\boldsymbol{s}}[\![\tau]\!]} \longrightarrow \mu_{\tau\Rightarrow\tau'}$ be the composite

$$\overline{\boldsymbol{s}}[\![\tau']\!]^{\overline{\boldsymbol{s}}[\![\tau]\!]} \xrightarrow{\mathrm{q}_{\tau'}{}^{\mathrm{u}_\tau \mathrm{v}_\tau}} \eta_{\tau'}{}^{\nu_\tau} \xrightarrow[\cong]{(\mathsf{abs}_{\tau\Rightarrow\tau'}, \mathrm{id})} \mu_{\tau\Rightarrow\tau'}$$

where $\mathrm{v}_\tau = (\mathsf{var}_\tau, \mathrm{id}) : \nu_\tau \longrightarrow \mu_\tau$.

The proposition below establishes the situation (6).

PROPOSITION 9. *For every type $\tau \in \widetilde{\mathrm{T}}$, the identities*

$$\pi(\mathrm{u}_\tau) \ = \ \mathrm{id}_{\boldsymbol{s}[\![\tau]\!]} \ = \ \pi(\mathrm{q}_\tau)$$

*hold.*

**Normalisation function.** Every interpretation $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ of base types in a cartesian closed category, induces a *normalisation function* $\boldsymbol{s}\text{-}\mathsf{nf}_\tau : \mathfrak{L}_\tau \longrightarrow \mathfrak{N}_\tau$ defined as the composite

$$\mathfrak{L}_\tau \xrightarrow{\;\overline{\ell}_\tau\;} [\overline{\boldsymbol{s}}[\![\_]\!], \overline{\boldsymbol{s}}[\![\tau]\!]] \xrightarrow{[\mathrm{uv}, \mathrm{q}_\tau]} [\overline{\mathbf{y}}(\_), \eta_\tau] \xrightarrow{\;\cong\;} \mathfrak{N}_\tau$$

where $\overline{\ell}$ denotes the semantics of terms induced by the interpretation $\overline{\boldsymbol{s}} : \mathrm{T} \longrightarrow \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\downarrow\langle \boldsymbol{s}\rangle$ of (5) and where

$$\begin{array}{rcl} (\mathrm{uv})_\Gamma & = & \overline{\mathbf{y}}(\Gamma) \xrightarrow{\mathrm{v}_\Gamma} \mu[\![\Gamma]\!] \xrightarrow{\mathrm{u}_\Gamma} \overline{\boldsymbol{s}}[\![\Gamma]\!] \ , \\ \mu[\![\Gamma]\!] & = & \prod_{(x:\tau)\in\Gamma} \mu_\tau \ , \\ \mathrm{v}_\Gamma & = & \overline{\mathbf{y}}(\Gamma) \xrightarrow{\cong} \prod_{(x:\tau)\in\Gamma} \nu_\tau \xrightarrow{\prod_{(x:\tau)\in\Gamma} \mathrm{v}_\tau} \mu[\![\Gamma]\!] \ , \\ \mathrm{u}_\Gamma & = & \prod_{(x:\tau)\in\Gamma} \mathrm{u}_\tau \ . \end{array}$$

That is,

$$\boldsymbol{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(t) \ = \ (\mathrm{q}_\tau \ \overline{\boldsymbol{s}}[\![\Gamma \vdash t : \tau]\!] \ (\mathrm{uv})_\Gamma)(\mathrm{id}_\Gamma)$$

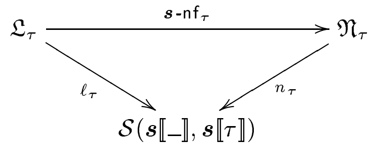
for all terms $t \in \mathfrak{L}_\tau(\Gamma)$.

As $\beta\eta$-equivalent terms have the same denotation, the corollary below follows directly from the definition of the normalisation function.

COROLLARY 10. *Let $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ be an interpretation of base types in a cartesian closed category. For every pair of terms $t, t' \in \mathfrak{L}_\tau(\Gamma)$, if $t =_{\beta\eta} t'$ then $\boldsymbol{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(t) = \boldsymbol{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(t')$.*

Further, as a consequence of Proposition 9, we have that a term and its associated normal form have the same semantics.

COROLLARY 11. *For every interpretation* $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ *of base types in a cartesian closed category, the diagram*




*commutes for all types* $\tau \in \widetilde{\mathrm{T}}$.

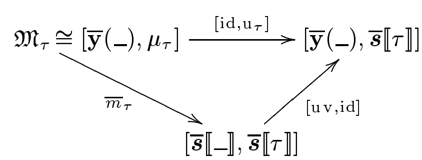
Applying the above two corollaries to the universal interpretation $\boldsymbol{f} : \mathrm{T} \longrightarrow \mathcal{F}[\mathrm{T}]$ of the base types T into the free cartesian closed category $\mathcal{F}[\mathrm{T}]$ over them, we have that $t =_{\beta\eta} \boldsymbol{f}\text{-nf}_{\tau,\Gamma}(t)$ and that $\boldsymbol{s}\text{-nf}_{\tau,\Gamma}(t) = \boldsymbol{s}\text{-nf}_{\tau,\Gamma}(\boldsymbol{f}\text{-nf}_{\tau,\Gamma}(t))$, for all terms $t \in \mathfrak{L}_\tau(\Gamma)$. It follows that the normalisation function $\boldsymbol{s}\text{-nf}_\tau$ fixes some normal terms. In fact, as we will see below, it fixes them all: that is,

$$\text{for all } N \in \mathfrak{N}_\tau(\Gamma),\ \boldsymbol{s}\text{-nf}_{\tau,\Gamma}(N) = N \quad . \tag{7}$$

This property is important. From it and Corollary 10, we have that, for all terms $t \in \mathfrak{L}_\tau(\Gamma)$ and normal terms $N \in \mathfrak{N}_\tau(\Gamma)$, if $t =_{\beta\eta} N$ then $\boldsymbol{s}\text{-nf}_{\tau,\Gamma}(t) = N$; and hence that (*i*) for every pair of normal terms $N, N' \in \mathfrak{N}_\tau(\Gamma)$, if $N =_{\beta\eta} N'$ then $N = N'$ and that (*ii*) for all terms $t \in \mathfrak{L}_\tau(\Gamma)$, $\boldsymbol{s}\text{-nf}_{\tau,\Gamma}(t) = \boldsymbol{f}\text{-nf}_{\tau,\Gamma}(t)$. Concluding thus, that every interpretation induces the same normalisation function $\mathrm{nf}_\tau : \mathfrak{L}_\tau \longrightarrow \mathfrak{N}_\tau$ such that, for every $t \in \mathfrak{L}_\tau(\Gamma)$ there exists a unique $N \in \mathfrak{N}_\tau(\Gamma)$ (namely $\mathrm{nf}_{\tau,\Gamma}(t)$, by Corollary 11) such that $t =_{\beta\eta} N$.

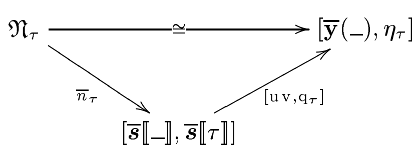
The appropriate induction hypothesis for establishing (7) is stated in the theorem below.

THEOREM 12. *For every interpretation* $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ *of base types in a cartesian closed category, the diagrams*




(8)

*and*




(9)

*commute for all types* $\tau \in \widetilde{\mathrm{T}}$.

The proof uses the induction principle associated to initial algebras [24] by considering the equalisers

$$\mathfrak{P}_\tau \overset{\imath_\tau}{\rightarrowtail} \mathfrak{M}_\tau \quad \text{and} \quad \mathfrak{Q}_\tau \overset{\jmath_\tau}{\rightarrowtail} \mathfrak{N}_\tau$$

of (8) and (9) respectively, and showing that the family

$$(\imath_\tau, \jmath_\tau) : (\mathfrak{P}_\tau, \mathfrak{Q}_\tau) \rightarrowtail (\mathfrak{M}_\tau, \mathfrak{N}_\tau) \qquad (\tau \in \widetilde{\mathrm{T}})$$

is a sub $\langle \Sigma_1, \Sigma_2 \rangle$-algebra, from which, by initiality, it follows that $\imath_\tau$ and $\jmath_\tau$ are isomorphisms. (In elementary terms, this categorical proof amounts to establishing the identities

$$\overline{\boldsymbol{s}}[\![\Gamma \vdash M : \tau]\!]\ (\mathrm{uv})_\Gamma \;=\; \mathrm{u}_\tau\ (M[\_], \boldsymbol{s}[\![\Gamma \vdash M : \tau]\!])$$

and

$$\mathrm{q}_\tau\ \overline{\boldsymbol{s}}[\![\Gamma \vdash N : \tau]\!]\ (\mathrm{uv})_\Gamma \;=\; (N[\_], \boldsymbol{s}[\![\Gamma \vdash N : \tau]\!])$$

for $M \in \mathfrak{M}_\tau(\Gamma)$ and $N \in \mathfrak{N}_\tau(\Gamma)$, by simultaneous induction on the derivation of neutral and normal terms (*cf.* [29]).)

The commutativity of diagram (9) amounts to property (7) and hence, as observed before, all normalisation functions coincide.

COROLLARY 13. *For every interpretation* $\boldsymbol{s} : \mathrm{T} \longrightarrow \mathcal{S}$ *of base types in a cartesian closed category and for the universal interpretation* $\boldsymbol{f} : \mathrm{T} \longrightarrow \mathcal{F}[\mathrm{T}]$ *of base types into the free cartesian closed category over them, the identity*

$$\boldsymbol{s}\text{-nf}_\tau \;=\; \boldsymbol{f}\text{-nf}_\tau \;: \mathfrak{L}_\tau \longrightarrow \mathfrak{N}_\tau \quad \textit{in } \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$$

*holds.*

Summarising, we have obtain normalisation functions

$$\mathrm{nf}_{\tau,\Gamma} : \mathfrak{L}_\tau(\Gamma) \longrightarrow \mathfrak{N}_\tau(\Gamma) \quad (\tau \in \widetilde{\mathrm{T}}, \Gamma \in |\mathbb{F}\downarrow\widetilde{\mathrm{T}}|)$$

satisfying the correctness properties below.

- For all context renamings $\rho : \Gamma \longrightarrow \Gamma'$ in $F\downarrow\widetilde{\mathrm{T}}$,
  $$(\mathrm{nf}_{\tau,\Gamma}\ t)[\rho] = \mathrm{nf}_{\tau,\Gamma'}(t[\rho])$$
  for every term $t \in \mathfrak{L}_\tau(\Gamma)$.
- For all normal terms $N \in \mathfrak{N}_\tau(\Gamma)$,
  $$\mathrm{nf}_{\tau,\Gamma}(N) = N \quad .$$
- For all terms $t \in \mathfrak{L}_\tau(\Gamma)$,
  $$\mathrm{nf}_{\tau,\Gamma}(t) =_{\beta\eta} t \quad .$$
- For all terms $t, t' \in \mathfrak{L}_\tau(\Gamma)$,
  $$\text{if } t =_{\beta\eta} t' \text{ then } \mathrm{nf}_{\tau,\Gamma}(t) = \mathrm{nf}_{\tau,\Gamma}(t') \quad .$$

**Normalisation algorithm.** The simplest normalisation function from which to extract an algorithm is the one induced by the trivial interpretation $\boldsymbol{t} : \mathrm{T} \longrightarrow \mathbf{1}$ of base types in the trivial cartesian closed category, as in this case the glueing category $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}} \downarrow \langle \boldsymbol{t} \rangle$ is simply (isomorphic to) the presheaf category $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$. (In fact, previous categorical analysis of normalisation by evaluation have centred around this interpretation [3, 29].)

Explicitly, the unquote and quote maps

$$\mathfrak{M}_\tau \xrightarrow{\mathrm{u}_\tau} \boldsymbol{s}[\![\tau]\!] \xrightarrow{\mathrm{q}_\tau} \mathfrak{N}_\tau \quad (\tau \in \widetilde{\mathrm{T}})$$

in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$, with respect to the interpretation of base types $\boldsymbol{s} : \theta \longmapsto \mathfrak{M}_\theta$, are (in the internal language of $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$) as follows

1. $\mathrm{u}_\theta(M) = M$

   $\mathrm{q}_\theta(M) = \mathsf{n}(M)$, where $\mathsf{n} : \mathfrak{M}_\theta \xrightarrow{\cong} \mathfrak{N}_\theta$
2. $\mathrm{u}_1(M) = (\,)$

   $\mathrm{q}_1(\,) = \mathsf{unit}_1(\,)$
3. $\mathrm{u}_{\tau*\tau'}(M) = (\ \mathrm{u}_\tau(\mathsf{fst}_\tau^{(\tau')}\ M)\ ,\ \mathrm{u}_{\tau'}(\mathsf{snd}_{\tau'}^{(\tau)}\ M)\ )$

   $\mathrm{q}_{\tau*\tau'}(x) = \mathsf{pair}_{\tau*\tau'}(\ \mathrm{q}_\tau(\pi_1\, x)\ ,\ \mathrm{q}_{\tau'}(\pi_2\, x)\ )$

4. $\mathrm{u}_{\tau \Rightarrow \tau'}(M) = \boldsymbol{\lambda}\, x^{\boldsymbol{s}[\![\tau]\!]}.\, \mathrm{u}_{\tau'}(\mathsf{app}^{(\tau)}_{\tau'}(M, \mathrm{q}_\tau\, x))$

$\mathrm{q}_{\tau \Rightarrow \tau'}(f) = \mathsf{abs}_{\tau \Rightarrow \tau'}(\boldsymbol{\lambda}\, v^{\mathrm{V}_\tau}.\, \mathrm{q}_{\tau'}(f(\mathrm{u}_\tau(\mathsf{var}_\tau\, v))))$

and the normalisation function is given by

$$\mathsf{nf}_{\tau,\Gamma}(t) = \mathrm{q}_\tau(\boldsymbol{s}[\![\Gamma \vdash t : \tau]\!]\, \langle \mathrm{u}_{\tau_i}(\mathsf{var}_{\tau_i}\, x_i)\rangle_{i=1,n})$$

for all terms $t \in \mathfrak{L}_\tau(\Gamma)$ where $\Gamma = \langle x_i : \tau_i \rangle_{i=1,n}$.

These functions coincide with the abstract implementations of normalisation by evaluation for typed lambda calculus (see, *e.g.*, [13]), and can be directly implemented in metalanguages supporting abstract syntax for terms with variable binding, like HOAS [27] and `FreshML` [19].

# CONCLUSION

We have given a new categorical view of normalisation by evaluation for typed lambda calculus, both for extensional and intensional normalisation problems.

Extensional normalisation was obtained from a basic lemma unifying definability and normalisation. Our analysis has the important methodological consequence of providing guidance when looking for normal forms. Indeed, a basic lemma based on the definability result of [18] via Grothendiek logical relations led to syntactic counterparts of the normal forms of [2] and has been applied to establish extensional normalisation for the typed lambda calculus with empty and sum types [16]. Along this line of research, one can study normalisation for other calculi for which definability results based on Kripke relations have been obtained —as classical linear logic [32], for instance.

The approach to normalisation by evaluation presented in the paper is novel, chiefly, in the following respects.

- The refinement from the extensional setting to the intensional one leading to the formalisation of normalisation by evaluation via categorical glueing.
- The use of an algebraic framework to structure both the development and proofs culminating in the definition of the normalisation function within a simply typed meta-theory.

The obtained abstract normalisation algorithm synthesises various concrete implementations. Its specialisation to particular implementations of abstract syntax directly yields normalisation programs for concrete representations of terms. In particular, as explained in [17], this can be easily done for representations of binding by de Bruijn levels or indices. How the abstract setting is related to representations of binding based on generating globally unique identifiers, say as in [15], needs to be investigated.

The role of glueing in our analysis is reminiscent of realisability. It would be interesting to understand whether there are connections to the modified realisability approach of [4].

**Acknowledgements.** The basis for this work, which was motivated by a question of Roberto Di Cosmo, was done during a visit to PPS, Université Paris 7 in July 2001 organised by Paul-André Melliès and supported by the CNRS. Discussions with Vincent Danos are gratefully acknowledged.

# REFERENCES

[1] M. Alimohamed. A characterization of lambda definability in categorical models of implicit polymorphism. *Theoretical Computer Science*, 146(1–2):5–23, 1995.

[2] T. Altenkirch, P. Dybjer, M. Hofmann, and P. Scott. Normalization by evaluation for typed lambda calculus with coproducts. In *Proceedings of the 16$^{th}$ Annual IEEE Symposium on Logic in Computer Science*, pages 203–210, 2001.

[3] T. Altenkirch, M. Hofmann, and T. Streicher. Categorical reconstruction of a *reduction-free* normalization proof. In *Category Theory and Computer Science*, volume 953 of *Lecture Notes in Computer Science*, pages 182–199. Springer-Verlag, 1995.

[4] U. Berger. Program extraction from normalization proofs. In *[6]*, pages 91–106, 1993.

[5] U. Berger and H. Schwichtenberg. An inverse of the evaluation functional for typed $\lambda$-calculus. In *Proceedings of the 6$^{th}$ Annual IEEE Symposium on Logic in Computer Science*, pages 203–211, 1991.

[6] M. Bezem and J. Groote, editors. *Typed Lambda Calculi and Applications*, volume 664 of *Lecture Notes in Computer Science*. Springer-Verlag, 1993.

[7] C. Coquand. From semantics to rules: A machine assisted analysis. In E. Börger, Y. Gurevich, and K. Meinke, editors, *Proc. Computer Science Logic'93*, volume 832 of *Lecture Notes in Computer Science*. Springer-Verlag, 1994.

[8] T. Coquand and P. Dybjer. Intuitionistic model constructions and normalization proofs. *Mathematical Structures in Computer Science*, 7:75–94, 1997. (Preliminary version in *Preliminary Proceedings of the 1993 TYPES Workshop*.).

[9] R. Crole. *Categories for Types*. Cambridge University Press, 1994.

[10] D. Čubrić, P. Dybjer, and P. Scott. Normalization and the Yoneda embedding. *Mathematical Structures in Computer Science*, 8:153–192, 1997.

[11] O. Danvy. Type-directed partial evaluation. In *Partial Evaluation — Practise and Theory, Proceedings of the 1998 DIKU Summer School*, volume 1706 of *Lecture Notes in Computer Science*, pages 367–411. Springer-Verlag, 1998.

[12] O. Danvy and P. Dybjer, editors. *Preliminary Proceedings of the APPSEM Workshop on Normalisation by Evaluation*, BRICS Note NS-98-1. Department of Computer Science, University of Aarhus, 1998.

[13] O. Danvy, P. Dybjer, and A. Filinski. Normalization and partial evaluation. Preliminary lecture notes for the International Summer School on Applied Semantics, 2000.

[14] A. Filinski. A semantic account of type-directed partial evaluation. In *Principles and Practice of Declarative Programming*, volume 1702 of *Lecture Notes in Computer Science*, pages 378–395. Springer-Verlag, 1999.

[15] A. Filinski. Normalization by evaluation for the computational lambda-calculus. In *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science. Springer-Verlag, 2001.

[16] M. Fiore, R. Di Cosmo, and V. Balat. Extensional normalisation for typed lambda calculus with sums via Grothendieck logical relations. Manuscript, 2002.

[17] M. Fiore, G. Plotkin, and D. Turi. Abstract syntax and variable binding. In *Proceedings of the $14^{th}$ Annual IEEE Symposium on Logic in Computer Science*, pages 193–202, 1999.

[18] M. Fiore and A. Simpson. Lambda definability with sums via Grothendieck logical relations. In *Typed Lambda Calculi and Applications*, volume 1581 of *Lecture Notes in Computer Science*, pages 147–161. Springer-Verlag, 1999.

[19] `FreshML` — A Fresh Approach to Name Binding in Metaprogramming Languages. In http://www.cl.cam.ac.uk/~amp12/research/freshml/, 2001.

[20] J.-Y. Girard. Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur. Thèse de doctorat d'état, Université Paris 7, 1972.

[21] A. Jung and J. Tiuryn. A new characterization of lambda definability. In *[6]*, pages 245–257, 1993.

[22] J. Krivine. *Lambda-Calculus, Types and Models.* Computers and their Applications. Masson and Ellis Horwood, 1993.

[23] J. Lambek and P. Scott. *Introduction to higher order categorical logic*, volume 7 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1986.

[24] D. Lehmann and M. Smyth. Algebraic specification of data types: A synthetic approach. *Math. Systems Theory*, 14:97–139, 1981.

[25] Q. Ma and J. Reynolds. Types, abstraction and parametric polymorphism, part 2. In *Mathematical Foundations of Programming Semantics*, volume 598 of *Lecture Notes in Computer Science*, pages 1–40. Springer-Verlag, 1992.

[26] P. Martin-Löf. About models for intuitionistic type theories and the notion of definitional equality. In *Proceedings of the $3^{rd}$ Scandinavian Logic Symposium*, pages 81–109, 1975.

[27] F. Pfenning and C. Elliot. Higher-order abstract syntax. In *Proc. of the ACM SIGPLAN '88 Symposium on Language Design and Implementation*, 1988.

[28] G. Plotkin. Lambda-definability and logical relations. Technical report, School of Artificial Intelligence, University of Edinburgh, 1973.

[29] J. Reynolds. Normalization and functor categories. In *[12]*, pages 33–36, 1998.

[30] R. Statman. Logical relations and the typed lambda calculus. *Inf. and Control*, 65:85–97, 1985.

[31] T. Streicher. Categorical intuitions underlying semantic normalisation proofs. In *[12]*, pages 9–10, 1998.

[32] T. Streicher. Denotational completeness revisited. In *Electronic Notes in Theoretical Computer Science*, volume 29. Elsevier Science Publishers, 2000.

[33] W. Tait. Intensional interpretation of functionals of finite type I. *Journal of Symbolic Logic*, 32, 1967.

[34] P. Taylor. *Practical Foundations of Mathematics*, volume 59 of *Cambridge studies in advanced mathematics*. Cambridge University Press, 1999.

[35] G. Wraith. Artin glueing. *Journal of Pure and Applied Algebra*, 4:345–348, 1974.