Mathematical Structures in Computer Science (2022), **32**, pp. 1028–1065
doi:10.1017/S0960129522000263

CAMBRIDGE UNIVERSITY PRESS

PAPER

Semantic analysis of normalisation by evaluation for typed lambda calculus*

Marcelo Fiore†

Department of Computer Science and Technology, University of Cambridge, Cambridge CB3 0FD, UK
Email: marcelo.fiore@cl.cam.ac.uk

(Received 18 August 2022; accepted 19 August 2022; first published online 22 November 2022)

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 Artin–Wraith 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. A normalisation-by-evaluation program in a dependently typed functional programming language is synthesised.

Keywords: Typed lambda calculus; normalisation by evaluation; lambda definability; Kripke logical relations; Artin-Wraith glueing; algebraic type theory

1. Introduction

Normalisation by evaluation for typed lambda calculus was first considered by Berger and Schwichtenberg (1991) from a type and proof theoretic viewpoint, and later investigated from the point of view of logic (Berger, 1993), type theory (Coquand, 1994), category theory (Altenkirch et al., 1995; Čubrić et al., 1997; Reynolds, 1998) and partial evaluation (Danvy, 1998; Filinski, 1999). This work gives a new categorical and algebraic perspective on the topic.

Outline. Normalisation by evaluation will be broadly viewed as the technique of giving semantics in (metalanguages for) non-standard models from which normalisation information can be extracted (cf. Martin-Lof 1975). 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 $\mathscr{L}_\tau(\Gamma)$ for the set of terms of type τ in context Γ, to identify a set of normal terms $\mathscr{N}_\tau(\Gamma) \subseteq \mathscr{L}_\tau(\Gamma)$ and show that for every term $t \in \mathscr{L}_\tau(\Gamma)$, there exists a normal term $N \in \mathscr{N}_\tau(\Gamma)$ such that $t =_{\beta\eta} N$.

* This is a slight revision, with an implementation, of the full version, with proofs, of February 2003 for the extended abstract (Fiore, 2002) published in October 2002.

† This research was supported by an EPSRC Advanced Research Fellowship (2000–2005) and partially supported by EPSRC grant EP/V002309/1 (2021–2024).

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

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

satisfying the following three properties.

(1) For all normal terms $N \in \mathscr{N}_\tau(\Gamma)$, the syntactic equality $\mathsf{nf}_{\tau,\Gamma}(N) = N$ holds.
(2) For all terms $t \in \mathscr{L}_\tau(\Gamma)$, the semantic equality $\mathsf{nf}_{\tau,\Gamma}(t) =_{\beta\eta} t$ holds.
(3) For all pair of terms $t, t' \in \mathscr{L}_\tau(\Gamma)$, 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 (Reynolds, 1998). 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 Artin–Wraith glueing construction, finally formalising the observation that normalisation by evaluation is *closely related* to categorical glueing (Coquand and Dybjer, 1997).

More in detail, the paper is organised as follows. Section 2.1 briefly recalls the syntax and categorical semantics of the typed lambda calculus. Section 2.2 presents an analysis of the lambda definability result of Jung and Tiuryn via Kripke logical relations (Jung and Tiuryn, 1993) leading to an extensional normalisation result. Section 3.1 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 3.2 to structure the development of an intensional version of Section 2.2 culminating in the technique of normalisation by evaluation.

Related work. The treatment of extensional normalisation presented here is similar to Tait's approach to strong normalisation via computability predicates (Girard, 1972; Tait, 1967) for the typed lambda calculus and also to Krivine's approach to normalisation (Krivine, 1993, 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 Altenkirch et al. (2001); Altenkirch et al. (1995), Čubrić et al. (1997), Reynolds (1998).

The approach of Čubrić et al. (1997) 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 Artin–Wraith glueing.

In Altenkirch et al. (1995), 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 adopt a purely semantic view, working with intensional logical relations in models given by the traditional categorical-glueing construction (Wraith, 1974).

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

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

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

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

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

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

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

Figure 1. Well-typed terms.

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.

2. Part I

2.1 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., Lambek and Scott (1986), Crole (1994), Taylor (1999).

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

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

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

The grammar for the terms is

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

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 τ, and a context Γ, 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 τ_i. For a context Γ, a variable x, and a type τ, the notation $\Gamma, x : \tau$ 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 τ.

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

Semantics. The appropriate mathematical universes for giving semantics to the typed lambda calculus are cartesian closed categories (Crole, 1994; Lambek and Scott, 1986; Taylor, 1999); i.e., categories with terminal object, binary products and exponentials (for which we, respectively, use the notation 1, $\times$, and $\Rightarrow$).

For an interpretation $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$ of base types in a cartesian closed category, we let $\mathbf{s}[\![\_]\!] : \widetilde{\mathrm{T}} \longrightarrow \mathcal{S}$ be the extension to simple types as prescribed by a chosen cartesian closed structure. That is, $\mathbf{s}[\![\theta]\!] = \mathbf{s}(\theta)$ (for $\theta \in \mathrm{T}$), $\mathbf{s}[\![1]\!] = 1$, and $\mathbf{s}[\![\tau * \tau']\!] = \mathbf{s}[\![\tau]\!] \times \mathbf{s}[\![\tau']\!]$ and $\mathbf{s}[\![\tau \Rightarrow \tau']\!] = \mathbf{s}[\![\tau]\!] \Rightarrow \mathbf{s}[\![\tau']\!]$

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

(for $\tau, \tau' \in \widetilde{\mathrm{T}}$). As usual, the interpretation of types is extended to contexts by setting $\mathbf{s}[\![\Gamma]\!] = \prod_{x \in \mathrm{dom}(\Gamma)} \mathbf{s}[\![\Gamma(x)]\!]$ for all contexts Γ. Finally, the semantics of a term $\Gamma \vdash t : \tau$ as a morphism $\mathbf{s}[\![\Gamma]\!] \longrightarrow \mathbf{s}[\![\tau]\!]$ in $\mathcal{S}$ is denoted $\mathbf{s}[\![\Gamma \vdash t : \tau]\!]$.

2.2 From definability to normalisation

Kripke relations were introduced by Jung and Tiuryn in (1993) for the purpose of characterising lambda definability. We will analyse this result and provide a corresponding extensional normalisation result.

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

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

In other words, a $\mathbb{C}$-Kripke relation R of arity $\boldsymbol{\sigma}$ over an object A is a unary predicate $R \hookrightarrow \mathcal{S}(\boldsymbol{\sigma}(\_), A)$ over the $\mathbb{C}^{\mathrm{op}}$-variable set of A-valued morphisms $\mathcal{S}(\boldsymbol{\sigma}(\_), A) : \mathbb{C}^{\mathrm{op}} \longrightarrow \mathbf{Set}$ in the functor category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$ of $\mathbb{C}^{\mathrm{op}}$-variable sets, referred to as presheaves.

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

Example 1. *The category of $\mathbb{C}$-Kripke relations of arity the unique functor to the terminal category is (isomorphic to) the complete Heyting algebra of subterminal objects of the presheaf topos* $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$.

The following proposition is well-known (see, e.g., Alimohamed 1995; Ma and Reynolds 1992).

Proposition 2. *Let $\mathbb{C}$ be a small category and let $\mathcal{S}$ be a cartesian closed category. For a functor $\boldsymbol{\sigma} : \mathbb{C} \longrightarrow \mathcal{S}$, the category of Kripke relations $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$ is cartesian closed and the forgetful functor $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle \longrightarrow \mathcal{S} : (R, A) \longmapsto A$ preserves the cartesian closed structure strictly.*

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

> (Products) The terminal object is $(\top, 1)$ where 1 is terminal in $\mathcal{S}$ and where $\top(c) = \{ \boldsymbol{\sigma}(c) \longrightarrow 1 \}$ 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 $a : \boldsymbol{\sigma}(c) \longrightarrow A \times A'$ is in $(R \wedge R')(c)$ iff $\pi \circ a : \boldsymbol{\sigma}(c) \longrightarrow A$ is in $R(c)$ and $\pi' \circ a : \boldsymbol{\sigma}(c) \longrightarrow A'$ is in $R'(c)$.

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

The Fundamental Lemma of logical relations is a consequence of Proposition 2.

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

Lemma 3 (Fundamental Lemma (Plotkin 1973; Statman 1985)). *For an interpretation of base types $\mathcal{I} : \mathrm{T} \longrightarrow \underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle : \theta \longmapsto (\mathscr{R}_\theta, \mathcal{I}_0(\theta))$, 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 $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$; that is, for $\mathcal{I}[\![\Gamma]\!] = (\mathscr{R}_\Gamma, \mathcal{I}_0[\![\Gamma]\!])$ and $\mathcal{I}[\![\tau]\!] = (\mathscr{R}_\tau, \mathcal{I}_0[\![\tau]\!])$, the following diagram

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

commutes in $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$ (for a necessarily unique natural map $\mathscr{R}_\Gamma \longrightarrow \mathscr{R}_\tau$).

Definability. The definability result of Jung and Tiuryn (1993) 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.

Definition 4. *For a set of types $\mathcal{T}$, we let $\mathbb{F}{\downarrow}\mathcal{T}$ be the category with objects given by contexts Γ 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 $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$ of base types in a cartesian closed category, we write $\mathbf{s}[\![\_]\!]$ for the canonical semantic functor $\mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}$ interpreting contexts and their renamings. This is explicitly given by

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

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

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

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

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

Lemma 5 (Definability Lemma (Alimohamed 1995; Jung and Tiuryn 1993)). *Let $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$ be an interpretation of base types in a cartesian closed category. Setting $\mathscr{R}_\theta = \mathscr{D}_\theta$ for all base types $\theta \in \mathrm{T}$ and letting $\mathscr{R}_\tau$ be given by the cartesian closed structure of the category of Kripke relations $\underline{\mathrm{K}}\langle\mathbf{s}[\![\_]\!] : \mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}\rangle$ for the other types $\tau \in \widetilde{\mathrm{T}}$, it follows that $\mathscr{R}_\tau = \mathscr{D}_\tau$ 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 Jung and Tiuryn (1993), Alimohamed (1995) (and Fiore and Simpson 1999 for the case with sum types). However, there is a more conceptual argument based on establishing that the definability relations satisfy the following closure properties:

$$\mathscr{D}_1 = \top$$

$$\mathscr{D}_{\tau * \tau'} = \mathscr{D}_\tau \wedge \mathscr{D}_{\tau'}$$

$$\mathscr{D}_{\tau \Rightarrow \tau'} = \mathscr{D}_\tau \supset \mathscr{D}_{\tau'}$$

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

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

The above analysis can be refined further. Indeed, the fact that neither of the following inclusions

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

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 $\mathscr{R}_\tau$ are bounded by possibly distinct Kripke relations (unlike the situation in (2)).

We are thus led to the following Basic Lemma. Notice the mixed-variance treatment of exponentiation. This is akin to Krivine's approach to normalisation for the untyped lambda calculus using *adapted pairs* of subsets of lambda terms (Krivine, 1993, Chapter III, pp. 33–39).

Lemma 6 (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 $\boldsymbol{\sigma} : \mathbb{C} \longrightarrow \mathcal{S}$, let $\langle(\mathscr{A}_\tau, \mathcal{I}_0[\![\tau]\!])\rangle_{\tau\in\widetilde{\mathrm{T}}}$ and $\langle(\mathscr{B}_\tau, \mathcal{I}_0[\![\tau]\!])\rangle_{\tau\in\widetilde{\mathrm{T}}}$ be two families of Kripke relations in $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$ indexed by types such that

$$\mathscr{B}_1 = \top$$

$$\mathscr{A}_{\sigma * \tau} \subseteq (\mathscr{A}_\sigma \wedge \mathscr{A}_\tau) \qquad (\mathscr{B}_\sigma \wedge \mathscr{B}_\tau) \subseteq \mathscr{B}_{\sigma * \tau}$$

$$\mathscr{A}_{\sigma \Rightarrow \tau} \subseteq (\mathscr{B}_\sigma \supset \mathscr{A}_\tau) \qquad (\mathscr{A}_\sigma \supset \mathscr{B}_\tau) \subseteq \mathscr{B}_{\sigma \Rightarrow \tau}$$

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

If $\mathscr{A}_\theta \subseteq \mathscr{R}_\theta \subseteq \mathscr{B}_\theta$ for all base types $\theta \in \mathrm{T}$, then

1. *$\mathscr{A}_\tau \subseteq \mathscr{R}_\tau \subseteq \mathscr{B}_\tau$ 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 $a_i : \boldsymbol{\sigma}(c) \longrightarrow \mathcal{I}_0[\![\tau_i]\!]$ in $\mathscr{A}_{\tau_i}(c)$ $(1 \leq i \leq n, c \in |\mathbb{C}|)$, we have that $\mathcal{I}_0[\![\Gamma \vdash t : \tau]\!] \circ \langle a_1, \ldots, a_n\rangle : \boldsymbol{\sigma}(c) \longrightarrow \mathcal{I}_0[\![\tau]\!]$ is in $\mathscr{B}_\tau(c)$.*

PROOF: The proof of the first part is by induction on the structure of types. This uses the facts that

$$R \subseteq \top \text{ for all } (R, 1) \text{ in } \underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$$

and that, for Kripke relations (R_i, A_i) and (R'_i, A_i) in $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$ $(i = 1, 2)$,

$$\text{if } R_1 \subseteq R'_1 \text{ and } R_2 \subseteq R'_2 \text{ then } (R_1 \wedge R_2) \subseteq (R'_1 \wedge R'_2) \text{ and } (R'_1 \supset R_2) \subseteq (R_1 \supset R'_2)$$

which follows from the functoriality of binary products and exponentials using the observation that, for (R, A) and (R', A) in $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$,

$$R \subseteq R' \Longleftrightarrow \mathrm{id}_A : (R, A) \longrightarrow (R', A) \text{ in } \underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle.$$

The proof of the second part follows from considering the interpretation $\mathcal{I} : \mathrm{T} \longrightarrow \underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$ mapping a base type θ to the Kripke relation $(\mathscr{R}_\theta, \mathcal{I}_0[\![\theta]\!])$ and noticing that, by the first part and the Fundamental Lemma of logical relations, the diagram below in $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$

$$\begin{array}{ccccccc}
\mathscr{A}_\Gamma & \hookrightarrow & \mathscr{R}_\Gamma & & \longrightarrow & & \mathscr{R}_\tau \hookrightarrow \mathscr{B}_\tau \\
 & \searrow & \downarrow & & & & \downarrow \quad \searrow \\
\mathcal{S}(\boldsymbol{\sigma}(_), \mathcal{I}_0[\![\Gamma]\!]) & & & \xrightarrow[\mathcal{I}_0[\![\Gamma\vdash t:\tau]\!]\circ_]{} & & & \mathcal{S}(\boldsymbol{\sigma}(_), \mathcal{I}_0[\![\tau]\!])
\end{array} \tag{3}$$

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

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

The Basic Lemma yields the Definability Lemma by considering $\mathscr{A}_\tau = \mathscr{D}_\tau = \mathscr{B}_\tau$ in the category of Kripke relations $\underline{\mathrm{K}}\langle \mathbf{s}[\![\_]\!] : \mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}\rangle$ for the given interpretation $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$. We will now see that the Basic Lemma can be also applied to obtain an extensional normalisation result (see Lemma 9).

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

$$(i)\ \ \mathscr{DN}_1 = \top$$

$$(ii)\ \ \mathscr{DM}_{\sigma * \tau} \subseteq (\mathscr{DM}_\sigma \wedge \mathscr{DM}_\tau) \qquad (iii)\ \ (\mathscr{DN}_\sigma \wedge \mathscr{DN}_\tau) \subseteq \mathscr{DN}_{\sigma * \tau}$$

$$(iv)\ \ \mathscr{DM}_{\sigma \Rightarrow \tau} \subseteq (\mathscr{DN}_\sigma \supset \mathscr{DM}_\tau) \qquad (v)\ \ (\mathscr{DM}_\sigma \supset \mathscr{DN}_\tau) \subseteq \mathscr{DN}_{\sigma \Rightarrow \tau}$$

$$(vi)\ \ \mathscr{DM}_\theta \subseteq \mathscr{DN}_\theta \quad (\theta \in \mathrm{T})$$

$$(vii)\ \ \pi_x : \mathbf{s}[\![\Gamma]\!] \longrightarrow \mathbf{s}[\![\tau]\!] \in \mathscr{DM}_\tau(\Gamma) \quad ((x : \tau) \in \Gamma)$$

so that, by the second part of the Basic Lemma, we get (setting $\mathscr{R}_\theta = \mathscr{DM}_\theta$ for all $\theta \in \mathrm{T}$, and $a_i = \pi_i : \mathbf{s}[\![\Gamma]\!] \longrightarrow \mathbf{s}[\![\tau_i]\!]$ for $\Gamma = \langle x_i : \tau_i \rangle_{i=1,n}$) that, for all terms $\Gamma \vdash t : \tau$,

$$\mathbf{s}[\![\Gamma \vdash t : \tau]\!] : \mathbf{s}[\![\Gamma]\!] \longrightarrow \mathbf{s}[\![\tau]\!] \text{ is in } \mathscr{DN}_\tau(\Gamma). \tag{4}$$

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

$$\mathscr{DM}_\tau(\Gamma) = \{\ \mathbf{s}[\![\Gamma \vdash M : \tau]\!] \mid \Gamma \vdash_{\mathscr{M}} M : \tau\ \} \tag{5}$$

$$\mathscr{DN}_\tau(\Gamma) = \{\ \mathbf{s}[\![\Gamma \vdash N : \tau]\!] \mid \Gamma \vdash_{\mathscr{N}} N : \tau\ \} \tag{6}$$

will provide the required Kripke relations (see Proposition 8). The conditions (i)–(vii) amount, roughly, to the following properties.

- The system $\vdash_{\mathscr{M}}$ should contain variables (condition (vii)), and be closed under projections (condition (ii)) and under the application to terms in the system $\vdash_{\mathscr{N}}$ (condition (iv)).
- The system $\vdash_{\mathscr{N}}$ 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 $\vdash_{\mathscr{M}}$ should be in the system $\vdash_{\mathscr{N}}$ (condition (vi)).

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

Thus, from purely semantic considerations, we have synthesised the notions of neutral normal forms (*viz.*, those derivable in the system $\vdash_{\mathscr{M}}$) and of long $\beta\eta$-normal forms (*viz.*, those derivable in the system $\vdash_{\mathscr{N}}$), henceforth, respectively, referred to as *neutral* and *normal* terms, and characterised as follows.

Proposition 7.

1. *(Neutral terms)*

$$\Gamma \vdash_{\mathscr{M}} t : \tau \Longleftrightarrow [\ \exists\, (x : \tau) \in \Gamma.\ t = x\ ]$$
$$\vee\ [\ \exists\, \Gamma \vdash_{\mathscr{M}} M : \tau * \tau'.\ t = \pi_1(M)\ ]\ \ \vee\ [\ \exists\, \Gamma \vdash_{\mathscr{M}} M : \tau' * \tau.\ t = \pi_2(M)\ ]$$
$$\vee\ [\ \exists\, \Gamma \vdash_{\mathscr{M}} M : \tau' \Rightarrow \tau\ ,\ \Gamma \vdash_{\mathscr{N}} N : \tau'.\ t = M(N)\ ]$$

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

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

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

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

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

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

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

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

Figure 2. Neutral and normal terms.

2. *(Normal terms)*
   - $\Gamma \vdash_{\mathscr{N}} t : 1 \Longleftrightarrow t = \langle\rangle$
   - $\Gamma \vdash_{\mathscr{N}} t : \tau_1 * \tau_2 \Longleftrightarrow [\ \exists\, \Gamma \vdash_{\mathscr{N}} N_1 : \tau_1\ , \Gamma \vdash_{\mathscr{N}} N_2 : \tau_2.\ t = \langle N_1, N_2 \rangle\ ]$
   - $\Gamma \vdash_{\mathscr{N}} t : \tau \Rightarrow \tau' \Longleftrightarrow [\ \exists\, \Gamma, x : \tau \vdash_{\mathscr{N}} N : \tau'.\ t = \lambda x : \tau. N\ ]$
   - *For θ a base type,* $\Gamma \vdash_{\mathscr{N}} t : \theta \Longleftrightarrow \Gamma \vdash_{\mathscr{M}} t : \theta$.

Neutral and normal terms are closed under context renamings and thereby semantically induce Kripke relations.

Proposition 8. *Let* $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$ *be an interpretation of base types in a cartesian closed category. For all types* $\tau \in \widetilde{\mathrm{T}}$*, the definitions (5) and (6), respectively, yield* $\mathbb{F}[\widetilde{\mathrm{T}}]$*-Kripke relations* $\mathscr{DM}_\tau$ *and* $\mathscr{DN}_\tau$ *of arity* $\mathbf{s}[\![\_]\!] : \mathbb{F}[\widetilde{\mathrm{T}}] \longrightarrow \mathcal{S}$ *satisfying conditions* (*i*)–(*vii*).

PROOF: The first part is a corollary of the facts that

$$\Gamma \vdash_{\mathscr{M}} M : \tau \Longleftrightarrow \forall\, \rho : \Gamma \to \Gamma' \text{ in } \mathbb{F}{\downarrow}\widetilde{\mathrm{T}}.\ \Gamma' \vdash_{\mathscr{M}} M[{}^{\rho x}/_{x}]_{x \in \mathrm{dom}(\Gamma)} : \tau \tag{7}$$

and

$$\Gamma \vdash_{\mathscr{N}} N : \tau \Longleftrightarrow \forall\, \rho : \Gamma \to \Gamma' \text{ in } \mathbb{F}{\downarrow}\widetilde{\mathrm{T}}.\ \Gamma' \vdash_{\mathscr{N}} N[{}^{\rho x}/_{x}]_{x \in \mathrm{dom}(\Gamma)} : \tau. \tag{8}$$

The second part follows by the construction of the systems $\vdash_{\mathscr{M}}$ and $\vdash_{\mathscr{N}}$. □

From Proposition 8, we have (4) and therefore, from (6) and Proposition 7(2), we obtain the following Extensional Normalisation Lemma.

Lemma 9 (Extensional Normalisation Lemma). *Let* $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$ *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* $\Gamma \vdash_{\mathscr{N}} N : \tau$ *such that*

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

in $\mathcal{S}$.

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

Specialising the Extensional Normalisation Lemma for the canonical interpretation of types in the free cartesian closed category generated by them, we obtain the following syntactic result (cf. Streicher 1998).

Corollary 10. *Every simply typed term is $\beta\eta$-equal to one in long $\beta\eta$-normal form.*

The above 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 development in categories obtained by the Artin–Wraith glueing construction (Wraith 1974). This will be done in Part II. To do it at an appropriate abstract, syntax-independent level, we will first consider the typed lambda calculus algebraically.

3. Part II

3.1 Algebraic typed lambda calculus

We provide an algebraic setting for the syntax and semantics of the typed lambda calculus following and extending the theory of Fiore et al. (1999). 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.

3.1.1 Syntax

Categories of contexts, which we study next, play a crucial role in describing abstract syntax with variable binding; see Fiore et al. (1999) for further details.

Free (co)cartesian categories. The category of untyped contexts and renamings $\mathbb{F}$ 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 in $\mathbb{F}\downarrow\mathcal{T}$ is

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

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 τ to $(1 \xrightarrow{\tau} \mathcal{T})$) and 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 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

$$\mathrm{V}_\tau = \mathbf{y}\langle\tau\rangle \tag{9}$$

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

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

$$\begin{array}{ccc} \mathbb{F}[\widetilde{\mathrm{T}}] & \xhookrightarrow{\mathbf{y}} & \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}} \\ {\scriptstyle \_\times\langle\tau\rangle}\downarrow & \overset{\mathrm{Lan}}{\cong} & {\scriptstyle \_\times\mathbf{y}\langle\tau\rangle}\downarrow \dashv \uparrow{\scriptstyle \mathbf{Set}^{(\_+\langle\tau\rangle)}} \\ \mathbb{F}[\widetilde{\mathrm{T}}] & \xhookrightarrow[\mathbf{y}]{} & \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}} \end{array} \tag{10}$$

Thus, in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$, the exponential P^{V_τ} of the presheaf V_τ and 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 T is a $\widetilde{\mathrm{T}}$-sorted algebra with carrier given by a family $\{\, \mathscr{X}_\tau \,\}_{\tau\in\widetilde{\mathrm{T}}}$ of presheaves in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ equipped with the following operations:

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

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

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

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

In the tradition of categorical algebra, the category of typed lambda algebras can be defined as the category of Σ-algebras for a signature endofunctor Σ on $(\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}})^{\widetilde{\mathrm{T}}}$. This endofunctor is induced by the above operations as follows, for $\theta \in \mathrm{T}$ and $\tau, \tau' \in \widetilde{\mathrm{T}}$:

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

$$\begin{aligned}(\Sigma\mathscr{X})_\theta &= \mathrm{V}_\theta + \mathrm{E}_\theta(\mathscr{X})\\ (\Sigma\mathscr{X})_1 &= \mathrm{V}_1 + 1 + \mathrm{E}_1(\mathscr{X})\\ (\Sigma\mathscr{X})_{\tau*\tau'} &= \mathrm{V}_{\tau*\tau'} + (\mathscr{X}_\tau \times \mathscr{X}_{\tau'}) + \mathrm{E}_{\tau*\tau'}(\mathscr{X})\\ (\Sigma\mathscr{X})_{\tau\Rightarrow\tau'} &= \mathrm{V}_{\tau\Rightarrow\tau'} + (\mathscr{X}_{\tau'})^{\mathrm{V}_\tau} + \mathrm{E}_{\tau\Rightarrow\tau'}(\mathscr{X})\end{aligned}$$

where

$$\mathrm{E}_\tau(\mathscr{X}) = \coprod_{\tau'\in\widetilde{\mathrm{T}}} \mathscr{X}_{\tau*\tau'} + \mathscr{X}_{\tau'*\tau} + (\mathscr{X}_{\tau'\Rightarrow\tau} \times \mathscr{X}_{\tau'})$$

is the signature endofunctor corresponding to the projections and application operations onto τ.

The initial Σ-algebra $\mathscr{L} = \{\, \mathscr{L}_\tau \,\}_{\tau\in\widetilde{\mathrm{T}}}$ with its structure

$$\begin{aligned}\mathrm{V}_\theta + \mathrm{E}_\theta(\mathscr{L}) &\xrightarrow{\cong} \mathscr{L}_\theta\\ \mathrm{V}_1 + 1 + \mathrm{E}_1(\mathscr{L}) &\xrightarrow{\cong} \mathscr{L}_1\\ \mathrm{V}_{\tau*\tau'} + (\mathscr{L}_\tau \times \mathscr{L}_{\tau'}) + \mathrm{E}_{\tau*\tau'}(\mathscr{L}) &\xrightarrow{\cong} \mathscr{L}_{\tau*\tau'}\\ \mathrm{V}_{\tau\Rightarrow\tau'} + (\mathscr{L}_{\tau'})^{\mathrm{V}_\tau} + \mathrm{E}_{\tau\Rightarrow\tau'}(\mathscr{L}) &\xrightarrow{\cong} \mathscr{L}_{\tau\Rightarrow\tau'}\end{aligned} \tag{11}$$

can be explicitly described as the family of presheaves of terms

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

with presheaf action given by variable renaming (that is, 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

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

corresponding to the typing rules in Figure 1.

A full theory of typed abstract syntax with variable binding incorporating substitution along the lines of Fiore et al. (1999) can be developed (see, e.g., Fiore and Hur 2010; Fiore and Szamozvancev 2022). This is however not necessary for the purposes of the paper.

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

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

below:

$$\left\{ (\Sigma_1(\mathscr{X},\mathscr{Y}))_\tau = \mathrm{V}_\tau + \mathrm{E}_\tau(\mathscr{X},\mathscr{Y}) \right. \qquad \left\{ \begin{array}{ll} (\Sigma_2(\mathscr{X},\mathscr{Y}))_\theta & = \mathrm{V}_\theta + \mathrm{E}_\theta(\mathscr{X},\mathscr{Y}) \\ (\Sigma_2(\mathscr{X},\mathscr{Y}))_1 & = 1 \\ (\Sigma_2(\mathscr{X},\mathscr{Y}))_{\tau*\tau'} & = \mathscr{Y}_\tau \times \mathscr{Y}_{\tau'} \\ (\Sigma_2(\mathscr{X},\mathscr{Y}))_{\tau\Rightarrow\tau'} & = (\mathscr{Y}_{\tau'})^{\mathrm{V}_\tau} \end{array} \right.$$

where

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

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

We write $(\mathscr{M},\mathscr{N})$ for the initial $\langle\Sigma_1,\Sigma_2\rangle$-algebra with structure, for $\theta \in \mathrm{T}$ and $\tau,\tau' \in \widetilde{\mathrm{T}}$, as follows:

$$\left\{ \mathrm{V}_\tau + \mathrm{E}_\tau(\mathscr{M},\mathscr{N}) \xrightarrow{\cong} \mathscr{M}_\tau \right. \qquad \left\{ \begin{array}{r} \mathrm{V}_\theta + \mathrm{E}_\theta(\mathscr{M},\mathscr{N}) \xrightarrow{\cong} \mathscr{N}_\theta \\ 1 \xrightarrow{\cong} \mathscr{N}_1 \\ \mathscr{N}_\tau \times \mathscr{N}_{\tau'} \xrightarrow{\cong} \mathscr{N}_{\tau*\tau'} \\ (\mathscr{N}_{\tau'})^{\mathrm{V}_\tau} \xrightarrow{\cong} \mathscr{N}_{\tau\Rightarrow\tau'} \end{array} \right. \tag{12}$$

Note that we have an isomorphism

$$\mathsf{norm} : \mathscr{M}_\theta \cong \mathrm{V}_\theta + \mathrm{E}_\theta(\mathscr{M},\mathscr{N}) \cong \mathscr{N}_\theta \tag{13}$$

for all $\theta \in \mathrm{T}$.

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

$$\mathscr{M}_\tau(\Gamma) = \{\ M \mid \Gamma \vdash_{\mathscr{M}} M : \tau\ \} \qquad \mathscr{N}_\tau(\Gamma) = \{\ N \mid \Gamma \vdash_{\mathscr{N}} N : \tau\ \}$$

with presheaf action given by variable renaming (recall (7) and (8)), and with operations

$$\left\{ \begin{array}{ll} \mathsf{var}_\tau & : \mathrm{V}_\tau \longrightarrow \mathscr{M}_\tau \\ \mathsf{fst}_\tau^{(\tau')} & : \mathscr{M}_{\tau*\tau'} \longrightarrow \mathscr{M}_\tau \\ \mathsf{snd}_\tau^{(\tau')} & : \mathscr{M}_{\tau'*\tau} \longrightarrow \mathscr{M}_\tau \\ \mathsf{app}_\tau^{(\tau')} & : \mathscr{M}_{\tau'\Rightarrow\tau} \times \mathscr{N}_{\tau'} \longrightarrow \mathscr{M}_\tau \end{array} \right. \qquad \left\{ \begin{array}{ll} \mathsf{var}_\theta & : \mathrm{V}_\theta \longrightarrow \mathscr{N}_\theta \\ \mathsf{fst}_\theta^{(\tau')} & : \mathscr{M}_{\theta*\tau'} \longrightarrow \mathscr{N}_\theta \\ \mathsf{snd}_\theta^{(\tau')} & : \mathscr{M}_{\tau'*\theta} \longrightarrow \mathscr{N}_\theta \\ \mathsf{app}_\theta^{(\tau')} & : \mathscr{M}_{\tau'\Rightarrow\theta} \times \mathscr{N}_{\tau'} \longrightarrow \mathscr{N}_\theta \\ \mathsf{unit}_1 & : 1 \xrightarrow{\cong} \mathscr{N}_1 \\ \mathsf{pair}_{\tau*\tau'} & : \mathscr{N}_\tau \times \mathscr{N}_{\tau'} \xrightarrow{\cong} \mathscr{N}_{\tau*\tau'} \\ \mathsf{abs}_{\tau\Rightarrow\tau'} & : (\mathscr{N}_{\tau'})^{\mathrm{V}_\tau} \xrightarrow{\cong} \mathscr{N}_{\tau\Rightarrow\tau'} \end{array} \right. \tag{14}$$

corresponding to the typing rules in Figure 2.

Note that every Σ-algebra $\mathscr{X}$ induces a canonical $\langle\Sigma_1,\Sigma_2\rangle$-algebra structure on the pair $(\mathscr{X},\mathscr{X})$ and hence, by initiality, homomorphic interpretations $(\mathscr{M},\mathscr{N}) \longrightarrow (\mathscr{X},\mathscr{X})$. Applying

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

this observation to the initial Σ-algebra $\mathscr{L}$, we obtain the embeddings $\mathscr{M} \rightarrowtail \mathscr{L}$ and $\mathscr{N} \rightarrowtail \mathscr{L}$ of neutral and normal terms into terms.

Structural induction. Initial algebras have the following associated structural induction principle (Lehmann and Smyth, 1981).

Let $\alpha : FA \longrightarrow A$ be an initial algebra for an endofunctor F, if the subobject $m : P \rightarrowtail A$ satisfies the closure property of being a sub F-algebra of A, in the sense that the diagram

$$\begin{array}{ccc} FP & \dashrightarrow & P \\ {\scriptstyle Fm}\downarrow & & \downarrow{\scriptstyle m} \\ FA & \xrightarrow[\alpha]{\cong} & A \end{array} \tag{15}$$

commutes for a (necessarily unique) map $FP \longrightarrow P$, then $m : P \rightarrowtail A$ is an isomorphism.

For the initial Σ-algebra $\mathscr{L}$ (resp. the initial $\langle \Sigma_1, \Sigma_2 \rangle$-algebra $(\mathscr{M}, \mathscr{N})$), the structural induction principle corresponds, in elementary terms, to proving a property of terms (resp. of neutral and normal terms) by induction (resp. simultaneous induction) on their derivation. The structural induction principle for the initial $\langle \Sigma_1, \Sigma_2 \rangle$-algebra $(\mathscr{M}, \mathscr{N})$ features in the proof of Theorem 21.

3.1.2 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.

Nerve functor. Every functor $\boldsymbol{\sigma} : \mathbb{C} \longrightarrow \mathcal{S}$ induces the following situation

$$\begin{array}{ccc} \mathbb{C} & \xrightarrow{\mathbf{y}} & \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \\ & {\scriptstyle \boldsymbol{\sigma}}\searrow \quad \underline{\boldsymbol{\sigma}} \Downarrow_{\mathrm{Lan}} \quad \nearrow{\scriptstyle \langle \boldsymbol{\sigma} \rangle} & \\ & \mathcal{S} & \end{array} \tag{16}$$

where $\langle \boldsymbol{\sigma} \rangle(A) = \mathcal{S}(\boldsymbol{\sigma}(\_), A)$ and where $(\underline{\boldsymbol{\sigma}}_\Gamma)_{\Gamma'} = \boldsymbol{\sigma}_{\Gamma', \Gamma} : \mathbb{C}(\Gamma', \Gamma) \longrightarrow \mathcal{S}(\boldsymbol{\sigma}(\Gamma'), \boldsymbol{\sigma}(\Gamma))$. We refer to $\langle \boldsymbol{\sigma} \rangle : \mathcal{S} \longrightarrow \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$ as the *$\boldsymbol{\sigma}$-nerve functor* and to the presheaf $\langle \boldsymbol{\sigma} \rangle(A)$ as the *$\boldsymbol{\sigma}$-nerve* of A.

Two important properties of nerve functors follow.

Proposition 11. *For a functor $\boldsymbol{\sigma} : \mathbb{C} \longrightarrow \mathcal{S}$ where $\mathbb{C}$ is small, the nerve functor $\langle \boldsymbol{\sigma} \rangle : \mathcal{S} \longrightarrow \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$ preserves limits. Further, for $\boldsymbol{\sigma}$ and $\mathbb{C}$ cartesian and $\mathcal{S}$ cartesian closed, it also commutes with exponentiation by representables in the sense that there is a canonical natural isomorphism*

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

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

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

PROOF: The first part is well-known and follows from the canonical natural isomorphism

$$\begin{array}{rcl} \mathcal{S}(\boldsymbol{\sigma}(\Gamma), L) & \cong & \lim_{\Delta\in\mathbb{D}}\ \mathcal{S}(\boldsymbol{\sigma}(\Gamma), D(\Delta)) \\ f & \longmapsto & \langle\, \pi_\Delta \circ f \,\rangle_{\Delta\in\mathbb{D}} \end{array} \qquad (\Gamma \in \mathbb{C})$$

available for any diagram $D : \mathbb{D} \longrightarrow \mathcal{S}$ with limit $\langle\, \pi_\Delta : L \longrightarrow D(\Delta) \,\rangle_{\Delta\in\mathbb{D}}$ in $\mathcal{S}$.

For the second part, note that for $\Gamma \in |\mathbb{C}|$ we have the following situation (generalising (10))

$$\begin{array}{ccc} \mathbb{C} & \xrightarrow{\ \mathbf{y}\ } & \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \\ {\scriptstyle -\times\Gamma}\downarrow & \overset{\mathrm{Lan}}{\cong} & {\scriptstyle -\times\mathbf{y}(\Gamma)}\downarrow \dashv \uparrow{\scriptstyle \mathbf{Set}^{(-\times\Gamma)^{\mathrm{op}}}} \\ \mathbb{C} & \xrightarrow[\ \mathbf{y}\ ]{} & \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \end{array}$$

from which it follows that $(P^{\mathbf{y}(\Gamma)})(\Delta) \cong P(\Delta \times \Gamma)$ naturally in $\Delta \in \mathbb{C}$. We thus obtain a canonical isomorphism

$$\begin{array}{rcl} (\langle\boldsymbol{\sigma}\rangle A)^{\mathbf{y}(\Gamma)}(\Delta) \cong (\langle\boldsymbol{\sigma}\rangle A)(\Delta \times \Gamma) & = & \mathcal{S}(\boldsymbol{\sigma}(\Delta \times \Gamma), A) \\ & \cong & \mathcal{S}(\boldsymbol{\sigma}(\Delta) \times \boldsymbol{\sigma}(\Gamma), A) \\ & \cong & \mathcal{S}(\boldsymbol{\sigma}(\Delta), \boldsymbol{\sigma}(\Gamma) \Rightarrow A) \ = \ \langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}(\Gamma) \Rightarrow A)(\Delta) \end{array}$$

natural in $\Gamma, \Delta \in \mathbb{C}$ and $A \in \mathcal{S}$. □

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

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

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

$$\mathscr{C} = \{\ \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau]\!])\ \}_{\tau\in\widetilde{\mathrm{T}}} = \{\ \langle\mathbf{s}\rangle(\mathbf{s}[\![\tau]\!])\ \}_{\tau\in\widetilde{\mathrm{T}}} \tag{17}$$

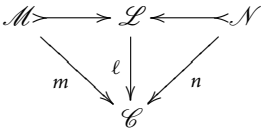
The operations are as follows:

1. $\mathrm{V}_\tau \xrightarrow{\ \mathbf{s}[\![\_]\!]\ } \langle\mathbf{s}\rangle(\mathbf{s}[\![\tau]\!])$
2. $1 \xrightarrow{\ \cong\ } \langle\mathbf{s}\rangle(\mathbf{s}[\![1]\!])$
3. $\langle\mathbf{s}\rangle(\mathbf{s}[\![\tau * \tau']\!]) \xrightarrow{\ \langle\mathbf{s}\rangle(\pi_1)\ } \langle\mathbf{s}\rangle(\mathbf{s}[\![\tau]\!])$
4. $\langle\mathbf{s}\rangle(\mathbf{s}[\![\tau' * \tau]\!]) \xrightarrow{\ \langle\mathbf{s}\rangle(\pi_2)\ } \langle\mathbf{s}\rangle(\mathbf{s}[\![\tau]\!])$ (18)
5. $\langle\mathbf{s}\rangle(\mathbf{s}[\![\tau]\!]) \times \langle\mathbf{s}\rangle(\mathbf{s}[\![\tau']\!]) \xrightarrow{\ \cong\ } \langle\mathbf{s}\rangle(\mathbf{s}[\![\tau * \tau']\!])$
6. $\langle\mathbf{s}\rangle(\mathbf{s}[\![\tau' \Rightarrow \tau]\!]) \times \langle\mathbf{s}\rangle(\mathbf{s}[\![\tau']\!]) \xrightarrow{\ \cong\ } \langle\mathbf{s}\rangle((\mathbf{s}[\![\tau']\!] \Rightarrow \mathbf{s}[\![\tau]\!]) \times \mathbf{s}[\![\tau']\!]) \xrightarrow{\ \langle\mathbf{s}\rangle(\varepsilon)\ } \langle\mathbf{s}\rangle(\mathbf{s}[\![\tau]\!])$
7. $(\langle\mathbf{s}\rangle(\mathbf{s}[\![\tau']\!]))^{\mathrm{V}_\tau} \xrightarrow{\ \cong\ } \langle\mathbf{s}\rangle(\mathbf{s}[\![\tau \Rightarrow \tau']\!])$

(Note that item 1 relies on diagram (16) while items 2, 5, 6, and 7 rely on Proposition 11. Similar applications of this proposition will be used throughout without further reference.)

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

By initiality, the semantic typed lambda algebra induces semantic homomorphic interpretations $\ell : \mathscr{L} \longrightarrow \mathscr{C}$ and $(m, n) : (\mathscr{M}, \mathscr{N}) \longrightarrow (\mathscr{C}, \mathscr{C})$. These are related as shown below



(19)

Indeed, by the initiality of $(\mathscr{M}, \mathscr{N})$, (19) directly follows from the fact that the homomorphism property of $\ell : \mathscr{L} \longrightarrow \mathscr{C}$ amounts to the commutativity of the diagrams in Appendix A and that the homorphism property of $(m, n) : (\mathscr{M}, \mathscr{N}) \longrightarrow (\mathscr{C}, \mathscr{C})$ amounts to the commutativity of the diagrams in Appendix B.

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

$$t \in \mathscr{L}_\tau(\Gamma) \xmapsto{\ell_\tau} \mathbf{s}[\![\Gamma \vdash t : \tau]\!] \in \mathcal{S}(\mathbf{s}[\![\Gamma]\!], \mathbf{s}[\![\tau]\!]) \tag{20}$$

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

3.2 Normalisation by evaluation via categorical glueing

We will now see how, by working with *intensional Kripke relations*, the analysis of normalisation given in Section 2.2 amounts to normalisation by evaluation. As in that section, we will work with semantic models of (covariant) presheaves in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ over nerves induced by interpretations $\mathbf{s}$ of the set of base types T in arbitrary cartesian closed categories (see (24)). This level of generality allows the definition of normalisation functions $\mathbf{s}\text{-}\mathsf{nf}_\tau : \mathscr{L}_\tau \longrightarrow \mathscr{N}_\tau$ $(\tau \in \widetilde{\mathrm{T}})$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ over the $\mathbf{s}[\![\_]\!]$-nerve of $\mathbf{s}[\![\tau]\!]$ (Corollary 20) that are parametric on the interpretation $\mathbf{s}$. Crucially, the normalisation functions will be shown to be parametrically polymorphic, in the sense of being interpretation independent (Corollary 22). This is methodologically important. Firstly, as in Corollary 10, the consideration of the universal interpretation of base types into the free cartesian closed category over them leads to our solution of the *intensional normalisation problem* (see the discussions after Corollaries 20 and 22 in § *Normalisation function* below) stated in Introduction. Secondly, the consideration of the trivial interpretation of base types in the trivial cartesian closed category leads to a normalisation algorithm from which a normalisation program is synthesised (see § *Normalisation algorithm* below).

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

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

commutes.

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

Example 12. *The category of intensional $\mathbb{C}$-Kripke relations of arity the unique functor to the terminal category is (isomorphic to) the presheaf topos* $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$.

As it is well-known (see, e.g., Crole 1994; Lambek and Scott 1986; Taylor 1999), for $\mathcal{S}$ cartesian closed, the glueing category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$ is also cartesian closed. Indeed, the cartesian closed structure of $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$ is given as follows.

(Products) The terminal object is $(1, t, 1)$ where t is the unique map $1 \xrightarrow{\cong} \langle\boldsymbol{\sigma}\rangle(1)$.
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 $P \times Q \xrightarrow{p\times q} \langle\boldsymbol{\sigma}\rangle(A) \times \langle\boldsymbol{\sigma}\rangle(B) \xrightarrow{\cong} \langle\boldsymbol{\sigma}\rangle(A \times B)$.

(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

$$\begin{array}{ccccc} R & & \longrightarrow & & Q^P \\ {\scriptstyle r}\downarrow & & \mathrm{pb} & & \downarrow{\scriptstyle q^P} \\ \langle\boldsymbol{\sigma}\rangle(A \Rightarrow B) & \longrightarrow & (\langle\boldsymbol{\sigma}\rangle B)^{(\langle\boldsymbol{\sigma}\rangle A)} & \xrightarrow[(\langle\boldsymbol{\sigma}\rangle B)^p]{} & (\langle\boldsymbol{\sigma}\rangle B)^P \end{array} \tag{21}$$

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

$$\langle\boldsymbol{\sigma}\rangle(A \Rightarrow B) \times \langle\boldsymbol{\sigma}\rangle(A) \xrightarrow{\cong} \langle\boldsymbol{\sigma}\rangle((A \Rightarrow B) \times A) \xrightarrow{\langle\boldsymbol{\sigma}\rangle(\varepsilon)} \langle\boldsymbol{\sigma}\rangle(B).$$

Explicitly, one may take $R(c)$ to be

$$\left\{ (f : \boldsymbol{\sigma}(c) \to A \Rightarrow B\,,\ \varphi : \mathbf{y}(c) \times P \to Q) \;\middle|\; \begin{array}{l} \forall\, \rho : c' \to c.\ \forall\, a \in P(c'). \\ \quad q_{c'}(\varphi_{c'}(\rho, a)) = \varepsilon \circ \langle f \circ \boldsymbol{\sigma}(\rho), p_{c'}(a)\rangle \end{array} \right\} \tag{22}$$

with r projecting pairs onto their first component.

Proposition 13. *Let $\mathbb{C}$ be a small category and let $\mathcal{S}$ be a cartesian closed category. For a functor $\boldsymbol{\sigma} : \mathbb{C} \longrightarrow \mathcal{S}$, the glueing category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$ is cartesian closed and the forgetful functor $\pi : \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle \longrightarrow \mathcal{S} : (P, p, A) \longmapsto A$ preserves the cartesian closed structure strictly.*

Remark. The category of $\mathbb{C}$-Kripke relations $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$ is a full subcategory of the glueing category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$ via the mapping $(R, A) \longmapsto (R, R \hookrightarrow \langle\boldsymbol{\sigma}\rangle(A), A)$. 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)

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

and the mapping $\mathrm{im} : (P, f, A) \longmapsto (\mathrm{im}(f), A)$ exhibits $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$ as a reflective subcategory of $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$. For $\mathcal{S}$ cartesian closed, as can be readily seen from the explicit descriptions of finite products in $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$ and $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$, the reflection $\mathrm{im} : \underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle \to \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$ preserves the cartesian structure and, therefore, $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$ is an exponential ideal of $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$ (as can also be readily seen

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

from the descriptions of exponentials in $\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle$ and $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\langle\boldsymbol{\sigma}\rangle$). Thus, for (P,p,A) and (Q,q,A) in $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\langle\boldsymbol{\sigma}\rangle$, there are inclusions

$$\mathrm{im}(\,(P,p,A)\Rightarrow(Q,q,B)\,)(c)\ \subseteq\ (\,\mathrm{im}(P,p,A)\supset\mathrm{im}(Q,q,B)\,)(c)\qquad(\,c\in|\mathbb{C}|\,)$$

where

$$\mathrm{im}(\,(P,p,A)\Rightarrow(Q,q,B)\,)(c)$$
$$=\left\{f:\boldsymbol{\sigma}(c)\to A\Rightarrow B\;\middle|\;\begin{array}{c}\exists\,\varphi:\mathbf{y}(c)\times P\to Q.\,\forall\,\rho:c'\to c.\,\forall\,a\in P(c').\\ q_{c'}(\varphi_{c'}(\rho,a))=\varepsilon\circ\langle f\circ\boldsymbol{\sigma}(\rho),p_{c'}(a)\rangle\end{array}\right\}$$

and

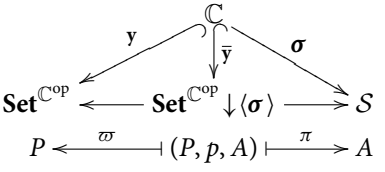
$$(\,\mathrm{im}(P,p,A)\supset\mathrm{im}(Q,q,B)\,)(c)$$
$$=\left\{f:\boldsymbol{\sigma}(c)\to A\Rightarrow B\;\middle|\;\begin{array}{c}\forall\,\rho:c'\to c.\,\forall\,a\in P(c').\,\exists\,b\in Q(c').\\ q_{c'}(b)=\varepsilon\circ\langle f\circ\boldsymbol{\sigma}(\rho),p_{c'}(a)\rangle\end{array}\right\}.$$

These inclusions may be strict; as it happens, for instance, when $\mathbb{C}^{\mathrm{op}}=\mathbb{F}$ (the category of untyped contexts and renamings), $\boldsymbol{\sigma}$ is the unique functor to the trivial cartesian closed category, $Q=\mathbf{y}(1)$ (for a singleton context 1), $P=\mathrm{im}(Q\to\langle\boldsymbol{\sigma}\rangle(1))$, and $c=0$ (the empty context). Indeed, in this situation, $(P\Rightarrow Q)(c)\cong\mathbf{Set}^{\mathbb{F}}(P,Q)=\emptyset$ whilst $(\mathrm{im}(p)\supset\mathrm{im}(q))(c)=(P\supset P)(c)=\{\,\mathrm{id}\,\}$. Thus, in general, the reflection $\mathrm{im}:\underline{\mathrm{K}}\langle\boldsymbol{\sigma}\rangle\to\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\langle\boldsymbol{\sigma}\rangle$ does not preserve exponentials.

Now, note that (16) induces the embedding

$$\begin{array}{ccc}\mathbb{C}&\xrightarrow{\ \bar{\mathbf{y}}\ }&\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\langle\boldsymbol{\sigma}\rangle\\ \Gamma&\longmapsto&(\,\mathbf{y}(\Gamma)\,,\,\mathbf{y}(\Gamma)\xrightarrow{\boldsymbol{\sigma}_\Gamma}\langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}\Gamma)\,,\,\boldsymbol{\sigma}(\Gamma)\,)\end{array}$$

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



and satisfying the following extended form of the Yoneda Lemma (which we will use in § *Normalisation function* below).

Lemma 14 (Extended Yoneda Lemma). *For a functor $\boldsymbol{\sigma}:\mathbb{C}\longrightarrow\mathcal{S}$ where $\mathbb{C}$ is a small category, the natural transformation*

$$\left[\bar{\mathbf{y}}(\_),(P,p,A)\right]\longrightarrow P(\_):\ (\varphi,f)\longmapsto\varphi(\mathrm{id}),$$

where $\left[\_,\_\right]$ denotes the hom-functor of the glueing category $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\!\downarrow\langle\boldsymbol{\sigma}\rangle$, is an isomorphism making the following diagram

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

$$\begin{array}{ccc} \left[\overline{\mathbf{y}}(\_), (P, p, A)\right] & \xrightarrow{\cong} & P(\_) \\ & {\scriptstyle\pi}\searrow \quad \swarrow{\scriptstyle p} & \\ & \mathcal{S}(\boldsymbol{\sigma}(\_), A) & \end{array}$$

commute.

PROOF: Follows from the fact that, for $\varphi : \mathbf{y}(\Gamma) \longrightarrow P$ in $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$ and $f : \boldsymbol{\sigma}(\Gamma) \longrightarrow A$ in $\mathcal{S}$, the diagram

$$\begin{array}{ccc} \mathbf{y}(\Gamma) & \xrightarrow{\varphi} & P \\ {\scriptstyle \underline{\boldsymbol{\sigma}}_\Gamma}\downarrow & & \downarrow{\scriptstyle p} \\ \mathcal{S}(\boldsymbol{\sigma}(\_), \boldsymbol{\sigma}(\Gamma)) & \xrightarrow[f\circ\_]{} & \mathcal{S}(\boldsymbol{\sigma}(\_), A) \end{array}$$

commutes if and only if $f = p_\Gamma(\varphi_\Gamma(\mathrm{id}_\Gamma))$. □

Proposition 15. *For a functor $\boldsymbol{\sigma} : \mathbb{C} \longrightarrow \mathcal{S}$ where $\mathbb{C}$ is small, $\mathbb{C}$ and $\boldsymbol{\sigma}$ are cartesian, and $\mathcal{S}$ is cartesian closed, we have that $\overline{\mathbf{y}} : \mathbb{C} \hookrightarrow \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$ preserves products and that the exponential $(P, p, A)^{\overline{\mathbf{y}}(\Gamma)}$ in $\mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}\downarrow\langle\boldsymbol{\sigma}\rangle$ can be described as $(P^{\mathbf{y}(\Gamma)}, p', \boldsymbol{\sigma}(\Gamma) \Rightarrow A)$ where p' is the composite*

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

PROOF: The first part follows from the commutativity of

$$\begin{array}{ccc} \mathbf{y}(\Gamma \times \Delta) & \xrightarrow{\cong} & \mathbf{y}(\Gamma) \times \mathbf{y}(\Delta) \\ & & \downarrow{\scriptstyle \underline{\boldsymbol{\sigma}}_\Gamma \times \underline{\boldsymbol{\sigma}}_\Delta} \\ {\scriptstyle \underline{\boldsymbol{\sigma}}_{\Gamma\times\Delta}}\downarrow & & \langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}\Gamma) \times \langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}\Delta) \\ & & \downarrow\cong \\ \langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}(\Gamma \times \Delta)) & \xrightarrow[\cong]{} & \langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}(\Gamma) \times \boldsymbol{\sigma}(\Delta)) \end{array}$$

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

For the second part, since the exponential $(P, p, A)^{\overline{\mathbf{y}}(\Gamma)}$ is given by pulling back the map $p^{\mathbf{y}(\Gamma)} : P^{\mathbf{y}(\Gamma)} \longrightarrow (\langle\boldsymbol{\sigma}\rangle A)^{\mathbf{y}(\Gamma)}$ along the composite

$$\langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}\Gamma \Rightarrow A) \xrightarrow{f} (\langle\boldsymbol{\sigma}\rangle A)^{\langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}\Gamma)} \xrightarrow{(\langle\boldsymbol{\sigma}\rangle A)^{\underline{\boldsymbol{\sigma}}_\Gamma}} (\langle\boldsymbol{\sigma}\rangle A)^{\mathbf{y}(\Gamma)}$$

(recall (21)), where f is the exponential transpose of

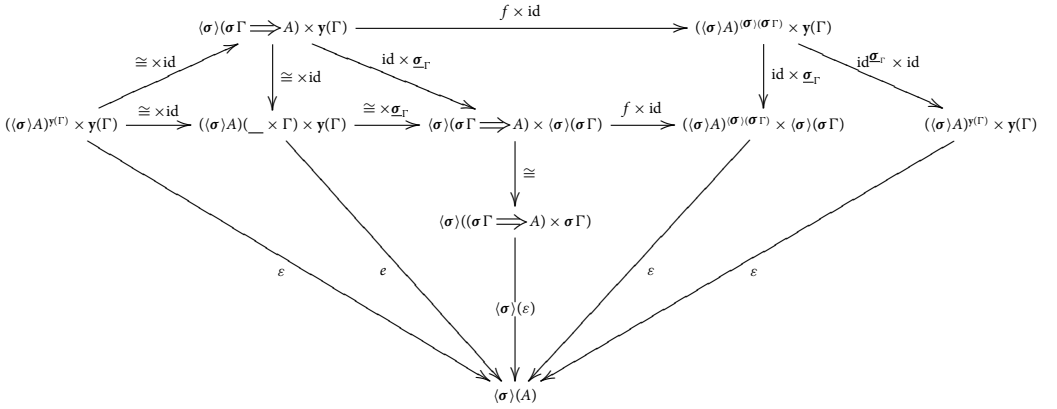
$$\langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}\Gamma \Rightarrow A) \times \langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}\Gamma) \xrightarrow{\cong} \langle\boldsymbol{\sigma}\rangle((\boldsymbol{\sigma}\Gamma \Rightarrow A) \times \boldsymbol{\sigma}\Gamma) \xrightarrow{\langle\boldsymbol{\sigma}\rangle(\varepsilon)} \langle\boldsymbol{\sigma}\rangle(A),$$

it will be enough to show that the composite

$$(\langle\boldsymbol{\sigma}\rangle A)^{\mathbf{y}(\Gamma)} \xrightarrow{\cong} \langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}\Gamma \Rightarrow A) \xrightarrow{f} (\langle\boldsymbol{\sigma}\rangle A)^{\langle\boldsymbol{\sigma}\rangle(\boldsymbol{\sigma}\Gamma)} \xrightarrow{(\langle\boldsymbol{\sigma}\rangle A)^{\underline{\boldsymbol{\sigma}}_\Gamma}} (\langle\boldsymbol{\sigma}\rangle A)^{\mathbf{y}(\Gamma)}$$

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

is the identity. This is indeed the case as follows from the commutativity of the diagram below



where

$$e_P \,:\, P(\_ \times \Gamma) \times \mathbf{y}(\Gamma) \longrightarrow P(\_) \,:\, (x, \rho) \longmapsto (P\langle \mathrm{id}, \rho\rangle)(x) \tag{23}$$

denotes the counit of the adjunction $\_ \times \mathbf{y}(\Gamma) \dashv \mathbf{Set}^{(\_\times\Gamma)^{\mathrm{op}}} : \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}} \longrightarrow \mathbf{Set}^{\mathbb{C}^{\mathrm{op}}}$. □

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

$$\nu_\tau \;=\; \overline{\mathbf{y}}\langle\tau\rangle \;=\; (\,\mathrm{V}_\tau\,,\; \mathrm{V}_\tau \xrightarrow{\mathbf{s}[\![\_]\!]} \mathscr{C}_\tau\,,\; \mathbf{s}[\![\tau]\!]\,) \qquad \text{in } \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}} \downarrow \langle \mathbf{s}[\![\_]\!]\rangle$$

glueing the syntax and semantics of variables. In the same spirit, glueing the syntax and semantics of neutral and normal terms (see (19)), we obtain the glued objects

$$\mu_\tau = (\,\mathscr{M}_\tau\,,\; \mathscr{M}_\tau \xrightarrow{m_\tau} \mathscr{C}_\tau\,,\; \mathbf{s}[\![\tau]\!]\,)$$

$$\eta_\tau = (\,\mathscr{N}_\tau\,,\; \mathscr{N}_\tau \xrightarrow{n_\tau} \mathscr{C}_\tau\,,\; \mathbf{s}[\![\tau]\!]\,)$$

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

Having constructed the $\langle \Sigma_1, \Sigma_2\rangle$-algebra structure on $(\mathscr{C}, \mathscr{C})$ by lifting the semantic operations in $\mathcal{S}$ (recall (17) and (18)), the homomorphism property of the semantic interpretation $(m, n) : (\mathscr{M}, \mathscr{N}) \longrightarrow (\mathscr{C}, \mathscr{C})$ (see Appendix B) entails the two propositions below, which show how the algebraic operations on the initial $\langle \Sigma_1, \Sigma_2\rangle$-algebra $(\mathscr{M}, \mathscr{N})$ and on the semantic $\langle \Sigma_1, \Sigma_2\rangle$-algebra $(\mathscr{C}, \mathscr{C})$ can be glued to yield operations in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}} \downarrow \langle \mathbf{s}[\![\_]\!]\rangle$ on the pair of families of glued objects $(\{\,\mu_\tau\,\}_{\tau\in\widetilde{\mathrm{T}}}\,,\; \{\,\eta_\tau\,\}_{\tau\in\widetilde{\mathrm{T}}})$.

Proposition 16. *Let $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$ be an interpretation of base types in a cartesian closed category.*

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

$$(\,\mathsf{var}_\tau : \mathrm{V}_\tau \longrightarrow \mathscr{M}_\tau\,,\; \mathrm{id}_{\mathbf{s}[\![\tau]\!]}\,)$$

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

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

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

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

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

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

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

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

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

$$(\ \mathsf{app}_{\tau}^{(\tau')} : \mathscr{M}_{\tau' \Rightarrow \tau} \times \mathscr{N}_{\tau'} \longrightarrow \mathscr{M}_{\tau}\ ,\ \varepsilon : (\mathbf{s}[\![\tau']\!] \Rightarrow \mathbf{s}[\![\tau]\!]) \times \mathbf{s}[\![\tau']\!] \longrightarrow \mathbf{s}[\![\tau]\!]\ )$$

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

PROOF: Items 1, 2, 3, and 4, respectively, follow from (B1), (B2), (B3), and (B4) in Appendix B. □

Proposition 17. *Let* $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$ *be an interpretation of base types in a cartesian closed category.*

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

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

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

2. *The pair of isomorphisms*

$$(\ \mathsf{unit}_1 : 1 \xrightarrow{\cong} \mathscr{N}_1\ ,\ \mathrm{id}_1\ )$$

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

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

$$(\ \mathsf{pair}_{\tau * \tau'} : \mathscr{N}_{\tau} \times \mathscr{N}_{\tau'} \xrightarrow{\cong} \mathscr{N}_{\tau * \tau'}\ ,\ \mathrm{id}_{\mathbf{s}[\![\tau]\!] \times \mathbf{s}[\![\tau']\!]}\ )$$

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

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

$$(\ \mathsf{abs}_{\tau \Rightarrow \tau'} : \mathscr{N}_{\tau'}{}^{\mathrm{V}_{\tau}} \xrightarrow{\cong} \mathscr{N}_{\tau \Rightarrow \tau'}\ ,\ \mathrm{id}_{\mathbf{s}[\![\tau]\!] \Rightarrow \mathbf{s}[\![\tau']\!]}\ )$$

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

PROOF: Items 1, 2, 3, and 4, respectively, follow from (B1–B8), (B9), (B10), and (B11) (relying on Proposition 15) in Appendix B. □

Note that the above operations on glued objects are given by pairs of syntactic operations together with their associated semantic meaning in the case of neutral terms (Proposition 16) and together with the identity in the case of normal terms (Proposition 17).

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

$$\begin{array}{ccc} \mathrm{T} & \xrightarrow{\ \bar{\mathbf{s}}\ } & \mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\downarrow\langle \mathbf{s}[\![\_]\!]\rangle \\ \theta & \longmapsto & \mu_{\theta} \end{array} \tag{24}$$

By Proposition 13, the semantics of terms induced by $\bar{\mathbf{s}}$ in $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\downarrow\langle \mathbf{s}[\![\_]\!]\rangle$ extends the semantics induced by $\mathbf{s}$ in $\mathcal{S}$; that is, the denotation $\bar{\mathbf{s}}[\![\Gamma \vdash t : \tau]\!]$ is a pair of the form

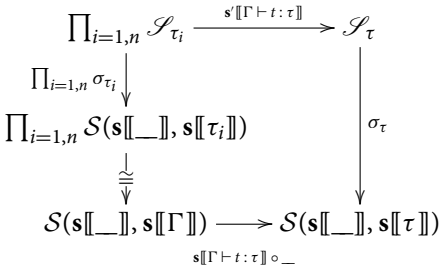
$$(\ \mathbf{s}'[\![\Gamma \vdash t : \tau]\!]\ ,\ \mathbf{s}[\![\Gamma \vdash t : \tau]\!])$$

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

such that, letting

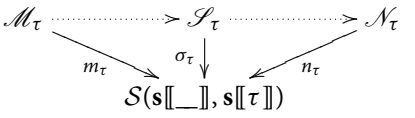
$$\bar{\mathbf{s}}[\![\tau]\!] = (\mathscr{S}_\tau\ ,\ \sigma_\tau\ ,\ \mathbf{s}[\![\tau]\!])\ ,$$

the diagram



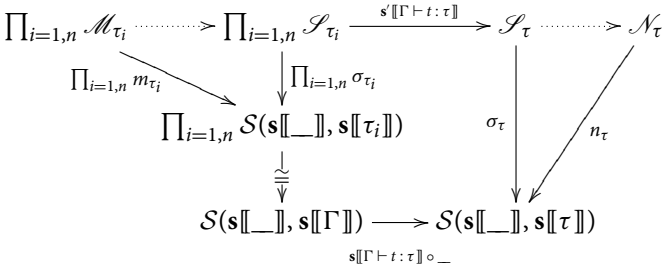
commutes for all $\Gamma = \langle x_i : \tau_i\rangle_{i=1,n}$.

We now aim at defining maps $\mathscr{M}_\tau \dashrightarrow \mathscr{S}_\tau \dashrightarrow \mathscr{N}_\tau$ $(\tau \in \widetilde{\mathrm{T}})$ such that



(25)

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



will commute (cf. diagram (3) of the Basic Lemma (Lemma 6)) and, hence, the evaluation of the horizontal top composite at the tuple $\langle \mathsf{var}_{\tau_i}(x_i)\rangle_{i=1,n}$ of the variables in the context Γ will yield a normal term in $\mathscr{N}_\tau(\Gamma)$ with the same semantics as the given term t (compare the Extensional Normalisation Lemma (Lemma 9) and see Corollary 20 below). Moreover, as we will show below (see Corollary 19), the long $\beta\eta$-normal forms associated to two $\beta\eta$-equal terms will be the same.

The abstract way to define the maps in (25) – which in the literature on normalisation by evaluation are either referred to as *unquote* and *quote* or as *reflect* and *reify* – is by defining maps

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

that project in $\mathcal{S}$ onto identities (see Proposition 18 below). The definition of these maps is by induction on the structure of types relying on Propositions 16 and 17 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

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

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

as the pairing of the maps

$$\mu_{\tau*\tau'} \xrightarrow{(\mathsf{fst}_\tau^{(\tau')},\pi_1)} \mu_\tau \xrightarrow{\mathrm{u}_\tau} \bar{\mathbf{s}}[\![\tau]\!] \text{ and } \mu_{\tau*\tau'} \xrightarrow{(\mathsf{snd}_{\tau'}^{(\tau)},\pi_2)} \mu_{\tau'} \xrightarrow{\mathrm{u}_{\tau'}} \bar{\mathbf{s}}[\![\tau']\!],$$

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

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

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

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

as the exponential transpose of the map

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

and let $\mathrm{q}_{\tau\Rightarrow\tau'} : \bar{\mathbf{s}}[\![\tau']\!]^{\bar{\mathbf{s}}[\![\tau]\!]} \longrightarrow \eta_{\tau\Rightarrow\tau'}$ be the composite

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

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

Proposition 18 below yields (25) as a corollary.

Proposition 18. *For every type $\tau \in \widetilde{\mathrm{T}}$, we have the identities*

$$\pi(u_\tau) \;=\; \mathrm{id}_{\mathbf{s}[\![\tau]\!]} \;=\; \pi(q_\tau)$$

for π the forgetful functor $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\downarrow\langle \mathbf{s}[\![\_]\!]\rangle \longrightarrow \mathcal{S}$.

PROOF: The proof is by induction on the structure of types.

(1) For a base type $\theta \in \mathrm{T}$, $\pi(\mathrm{u}_\theta) = \pi(\mathrm{q}_\theta) = \mathrm{id}_{\mathbf{s}[\![\theta]\!]}$ by definition of u_θ and q_θ.
(2) $\pi(\mathrm{u}_1) = \pi(\mathrm{q}_1) = \mathrm{id}_1$ by definition of u_1 and q_1.
(3) For types $\tau, \tau' \in \widetilde{\mathrm{T}}$,

$$\begin{aligned}
\pi(\mathrm{u}_{\tau*\tau'}) &= \langle\, \pi(\mathrm{u}_\tau) \circ \pi_1\,,\; \pi(\mathrm{u}_{\tau'}) \circ \pi_2 \,\rangle \;\text{, by definition of } \mathrm{u}_{\tau*\tau'} \\
&= \langle\, \pi_1\,,\; \pi_2 \,\rangle \qquad\qquad\qquad\qquad \text{, by induction} \\
&= \mathrm{id}_{\mathbf{s}[\![\tau]\!]\times\mathbf{s}[\![\tau']\!]}
\end{aligned}$$

and

$$\begin{aligned}
\pi(\mathrm{q}_{\tau*\tau'}) &= \pi(\mathrm{q}_\tau) \times \pi(\mathrm{q}_{\tau'}) \;\text{, by definition of } \mathrm{q}_{\tau*\tau'} \\
&= \mathrm{id}_{\mathbf{s}[\![\tau]\!]} \times \mathrm{id}_{\mathbf{s}[\![\tau']\!]} \;\text{, by induction} \\
&= \mathrm{id}_{\mathbf{s}[\![\tau]\!]\times\mathbf{s}[\![\tau']\!]}.
\end{aligned}$$

(4) For types $\tau, \tau' \in \widetilde{\mathrm{T}}$,

$$\begin{aligned}
&\varepsilon \circ (\pi(\mathrm{u}_{\tau\Rightarrow\tau'}) \times \mathrm{id}_{\mathbf{s}[\![\tau]\!]}) \\
&\quad= \pi(\mathrm{u}_{\tau'}) \circ \varepsilon \circ (\mathrm{id}_{\mathbf{s}[\![\tau]\!]\Rightarrow\mathbf{s}[\![\tau']\!]} \times \pi(\mathrm{q}_\tau)) \;\text{, by definition of } \mathrm{u}_{\tau\Rightarrow\tau'} \\
&\quad= \varepsilon \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{, by induction}
\end{aligned}$$

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

and hence

$$\pi(\mathrm{u}_{\tau\Rightarrow\tau'}) = \mathrm{id}_{\mathbf{s}[\![\tau]\!]\Rightarrow\mathbf{s}[\![\tau']\!]} \quad ;$$

further

$$\begin{aligned}\pi(\mathrm{q}_{\tau\Rightarrow\tau'}) &= (\pi(\mathrm{u}_\tau)\circ\pi(\mathrm{v}_\tau))\Rightarrow(\pi(\mathrm{q}_{\tau'})) \text{ , by definition of } \mathrm{q}_{\tau\Rightarrow\tau'}\\ &= \mathrm{id}_{\mathbf{s}[\![\tau]\!]}\Rightarrow\mathrm{id}_{\mathbf{s}[\![\tau']\!]} \qquad \text{, by induction and definition of } \mathrm{v}_\tau\\ &= \mathrm{id}_{\mathbf{s}[\![\tau]\!]\Rightarrow\mathbf{s}[\![\tau']\!]}.\end{aligned}$$

□

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

$$\mathscr{L}_\tau \xrightarrow{\bar{\ell}_\tau} \left[\bar{\mathbf{s}}[\![\_]\!],\bar{\mathbf{s}}[\![\tau]\!]\right] \xrightarrow{[\mathrm{uv},\mathrm{q}_\tau]} \left[\bar{\mathbf{y}}(\_),\eta_\tau\right] \xrightarrow{\cong} \mathscr{N}_\tau$$

where $\bar{\ell}$ denotes the semantics of terms induced by the interpretation $\bar{\mathbf{s}}:\mathrm{T}\longrightarrow\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\downarrow\langle\mathbf{s}[\![\_]\!]\rangle$ of (24) and where

$$(\mathrm{uv})_\Gamma = \bar{\mathbf{y}}(\Gamma)\xrightarrow{\mathrm{v}_\Gamma}\mu[\![\Gamma]\!]\xrightarrow{\mathrm{u}_\Gamma}\bar{\mathbf{s}}[\![\Gamma]\!]$$

for

$$\begin{aligned}\mu[\![\Gamma]\!] &= \textstyle\prod_{(x:\tau)\in\Gamma}\mu_\tau \quad ,\\ \mathrm{v}_\Gamma \quad &= \bar{\mathbf{y}}(\Gamma)\xrightarrow{\cong}\textstyle\prod_{(x:\tau)\in\Gamma}\nu_\tau \xrightarrow{\prod_{(x:\tau)\in\Gamma}\mathrm{v}_\tau} \mu[\![\Gamma]\!] \quad ,\\ \mathrm{u}_\Gamma \quad &= \textstyle\prod_{(x:\tau)\in\Gamma}\mathrm{u}_\tau \quad .\end{aligned}$$

Explicitly,

$$\mathbf{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(t) \;=\; (\mathrm{q}_\tau\;\bar{\mathbf{s}}[\![\Gamma\vdash t:\tau]\!]\;(\mathrm{uv})_\Gamma)(\mathrm{id}_\Gamma) \;\in\mathscr{N}_\tau(\Gamma)$$

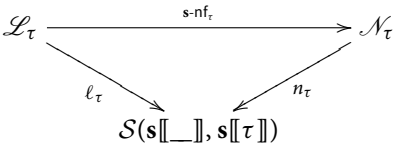
for all terms $t\in\mathscr{L}_\tau(\Gamma)$.

Having the same denotation, $\beta\eta$-equal terms are identified by the normalisation function.

Corollary 19. *Let $\mathbf{s}:\mathrm{T}\longrightarrow\mathcal{S}$ be an interpretation of base types in a cartesian closed category. For every pair of terms t,t' in $\mathscr{L}_\tau(\Gamma)$, if $t=_{\beta\eta}t'$ then $\mathbf{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(t)=\mathbf{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(t')$ in $\mathscr{N}_\tau(\Gamma)$.*

Further, as a consequence of Proposition 18 (see also (25)), we have that a term and its associated normal form have the same semantics.

Corollary 20. *For every interpretation $\mathbf{s}:\mathrm{T}\longrightarrow\mathcal{S}$ of base types in a cartesian closed category, the diagram*



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

Considering the universal interpretation $\mathbf{f}:\mathrm{T}\longrightarrow\mathcal{F}_{\mathrm{ccc}}[\mathrm{T}]$ of the set of base types T into the free cartesian closed category $\mathcal{F}_{\mathrm{ccc}}[\mathrm{T}]$ over them, by Corollary 20, we have that

$$t =_{\beta\eta}\mathbf{f}\text{-}\mathsf{nf}_{\tau,\Gamma}(t) \tag{26}$$

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

and hence, by Corollary 19, that

$$\mathbf{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(t) = \mathbf{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(\mathbf{f}\text{-}\mathsf{nf}_{\tau,\Gamma}(t))$$

for all terms $t \in \mathscr{L}_\tau(\Gamma)$. Thus, the normalisation function $\mathbf{f}\text{-}\mathsf{nf}_\tau$ is idempotent and therefore fixes some normal terms. In fact, as we will see below (see (29) in Theorem 21), all normalisation functions $\mathbf{s}\text{-}\mathsf{nf}_\tau$ fix all normal terms: that is,

$$\text{for all } N \in \mathscr{N}_\tau(\Gamma),\ \mathbf{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(N) = N. \tag{27}$$

This fixed-point property is important: from it and Corollary 19 it follows that

- for all terms $t \in \mathscr{L}_\tau(\Gamma)$ and normal terms $N \in \mathscr{N}_\tau(\Gamma)$, if $t =_{\beta\eta} N$ then $\mathbf{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(t) = N$, and
- for every pair of normal terms $N, N' \in \mathscr{N}_\tau(\Gamma)$, if $N =_{\beta\eta} N'$ then $N = N'$;

so that, further using Corollary 20 in the form (26), we have that

- for all terms $t \in \mathscr{L}_\tau(\Gamma)$, $\mathbf{s}\text{-}\mathsf{nf}_{\tau,\Gamma}(t) = \mathbf{f}\text{-}\mathsf{nf}_{\tau,\Gamma}(t)$.

Thus, the fixed-point property (27) allows one to conclude that:

> all interpretations induce the same normalisation function $\mathsf{nf}_\tau : \mathscr{L}_\tau \longrightarrow \mathscr{N}_\tau$ such that, for every term $t \in \mathscr{L}_\tau(\Gamma)$, one has that $\mathsf{nf}_{\tau,\Gamma}(t) \in \mathscr{N}_\tau(\Gamma)$ is the unique normal term $\beta\eta$-equal to t.

We now establish (27). The appropriate induction hypothesis to proceed by induction on the structure of neutral and normal terms is stated in the theorem below.

Theorem 21. *For every interpretation* $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$ *of base types in a cartesian closed category, the diagrams*

$$\begin{array}{ccc} \mathscr{M}_\tau \cong \left[\overline{\mathbf{y}}(\_), \mu_\tau\right] & \xrightarrow{\ [\mathrm{id},u_\tau]\ } & \left[\overline{\mathbf{y}}(\_), \overline{\mathbf{s}}[\![\tau]\!]\right] \\ & {\scriptstyle \overline{m}_\tau} \searrow \quad \nearrow {\scriptstyle [uv,\mathrm{id}]} & \\ & \left[\overline{\mathbf{s}}[\![\_]\!], \overline{\mathbf{s}}[\![\tau]\!]\right] & \end{array} \tag{28}$$

and

$$\begin{array}{ccc} \mathscr{N}_\tau & \xrightarrow{\ \cong\ } & \left[\overline{\mathbf{y}}(\_), \eta_\tau\right] \\ & {\scriptstyle \overline{n}_\tau} \searrow \quad \nearrow {\scriptstyle [uv,q_\tau]} & \\ & \left[\overline{\mathbf{s}}[\![\_]\!], \overline{\mathbf{s}}[\![\tau]\!]\right] & \end{array} \tag{29}$$

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

PROOF: The proof uses the induction principle associated to the initial $\langle \Sigma_1, \Sigma_2 \rangle$-algebra $(\mathscr{M}, \mathscr{N})$ (see (15)) by considering the equalisers

$$\mathscr{P}_\tau \rightarrowtail^{\iota_\tau} \mathscr{M}_\tau \quad \text{and} \quad \mathscr{Q}_\tau \rightarrowtail^{\jmath_\tau} \mathscr{N}_\tau$$

of (28) and (29), respectively, and showing that the family

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

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

is a sub $\langle \Sigma_1, \Sigma_2 \rangle$-algebra, from which it follows that ι_τ and J_τ are isomorphisms and hence that (28) and (29) commute. The details are spelled out in Appendix C. □

Remark. In elementary terms, the above categorical proof amounts to establishing the identities

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

and

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

for $M \in \mathscr{M}_\tau(\Gamma)$ and $N \in \mathscr{N}_\tau(\Gamma)$, by simultaneous induction on the derivation of neutral and normal terms (cf. Reynolds 1998).

The commutativity of diagram (29) amounts to property (27) and hence, as explained above, all normalisation functions coincide.

Corollary 22. *For every interpretation* $\mathbf{s} : \mathrm{T} \longrightarrow \mathcal{S}$ *of base types in a cartesian closed category and for the universal interpretation* $\mathbf{f} : \mathrm{T} \longrightarrow \mathcal{F}_{\mathrm{ccc}}[\mathrm{T}]$ *of base types into the free cartesian closed category over them, the identity*

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

holds.

Summarising, we have obtained normalisation functions

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

satisfying the correctness properties below.

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

Normalisation algorithm. The simplest description of the normalisation function from which to extract an algorithm is the one induced by the trivial interpretation $\mathbf{t}$ of base types in the trivial cartesian closed category as, in this case, the glueing category $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}\downarrow\langle \mathbf{t}[\![\_]\!] \rangle$ is simply (isomorphic to) the presheaf category $\mathbf{Set}^{\mathbb{F}\downarrow\widetilde{\mathrm{T}}}$ (recall Example 12). In fact, previous categorical analysis of normalisation by evaluation have centred around this interpretation (Altenkirch et al., 1995; Reynolds, 1998).

Explicitly, the unquote and quote maps

$$\mathscr{M}_\tau \xrightarrow{\mathrm{u}_\tau} \mathbf{s}[\![\tau]\!] \xrightarrow{\mathrm{q}_\tau} \mathscr{N}_\tau \quad (\tau \in \widetilde{\mathrm{T}}) \tag{30}$$

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

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

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

   $\mathrm{q}_\theta(M) = \mathsf{norm}(M)$, where $\mathsf{norm} : \mathscr{M}_\theta \xrightarrow{\cong} \mathscr{N}_\theta$ (see (13))

2. $\mathrm{u}_1(M) = (\,)$

   $\mathrm{q}_1(\,) = \mathsf{unit}_1(\,)$

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

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

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

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

and the normalisation function is given by

$$\mathsf{nf}_{\tau,\Gamma}(t) = \mathrm{q}_\tau(\mathbf{s}[\![\Gamma \vdash t : \tau]\!]\ \langle \mathrm{u}_{\tau_i}(\mathsf{var}_{\tau_i}\, x_i)\rangle_{i=1,n}) \tag{31}$$

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

These functions can be directly implemented, for instance, in metalanguages supporting abstract syntax with variable binding, like HOAS (Pfenning and Elliot, 1988), `Fresh O'Caml` (Fresh O'Caml, *www.cl.cam.ac.uk/∼amp12/fresh − ocaml/*), the Scope-and-Type Safe Universe of Syntaxes with Binding (Allais et al., 2021), and the SOAS Framework (Fiore and Szamozvancev, 2022). Indeed, for concreteness, we here synthesise an elementary implementation in Agda considered as a dependently typed functional programming language.[1]

Syntax. We consider simple types over a countably infinite set of base types (see (1)):

```
-- base types
T : Set
T = Nat

-- simple types
data T̃ : Set where
  θ : T → T̃
  1 : T̃
  _*_ : T̃ → T̃ → T̃
  _⇒_ : T̃ → T̃ → T̃
```

Typing contexts (Definition 4) are inductively generated by context extension from an empty context:

```
-- typing contexts
data 𝔽↓ : Set → Set where
  · : {𝒯 : Set} → 𝔽↓𝒯
  _::_ : {𝒯 : Set} → 𝔽↓𝒯 → 𝒯 → 𝔽↓𝒯
```

We then have a family of variable indices (recall (9)) given as follows:

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

```
-- variable indices
data V : {𝒯 : Set} → 𝒯 → 𝔽↓𝒯 → Set where
  • : {𝒯 : Set} {τ : 𝒯} {Γ : 𝔽↓𝒯} → V τ (Γ :: τ)
  ↑ : {𝒯 : Set} {τ σ : 𝒯} {Γ : 𝔽↓𝒯} → V σ Γ → V σ (Γ :: τ)
```

for which context renamings (Definition 4) are considered:

```
-- context renamings
𝔽↓ : (T̃ : Set) → 𝔽↓T̃ × 𝔽↓T̃ → Set
𝔽↓T̃ (Δ , Γ ) = {τ : T̃} → V τ Δ → V τ Γ
```

The abstract syntax of simply typed terms (see (11)) is implemented by the inductive family below:

```
-- simply typed terms
data ℒ : T̃ → 𝔽↓T̃ → Set where
  var : {τ : T̃} {Γ : 𝔽↓T̃} → V τ Γ → ℒ τ Γ
  unit : {Γ : 𝔽↓T̃} → ℒ 1 Γ
  pair : {τ σ : T̃} {Γ : 𝔽↓T̃} → ℒ τ Γ → ℒ σ Γ → ℒ (τ ∗ σ) Γ
  fst : {τ σ : T̃} {Γ : 𝔽↓T̃} → ℒ (τ ∗ σ) Γ → ℒ τ Γ
  snd : {τ σ : T̃} {Γ : 𝔽↓T̃} → ℒ (τ ∗ σ) Γ → ℒ σ Γ
  abs : {τ σ : T̃} {Γ : 𝔽↓T̃} → ℒ σ (Γ :: τ ) → ℒ (τ ⇒ σ) Γ
  app : {τ σ : T̃} {Γ : 𝔽↓T̃} → ℒ (τ ⇒ σ) Γ → ℒ τ Γ → ℒ σ Γ
```

Analogously, the abstract syntax of neutral and normal terms (see (12) and (14)) is implemented by the following mutually inductive families:

```
-- neutral and normal terms
mutual
  data ℳ : T̃ → 𝔽↓T̃ → Set where
    varₘ : {τ : T̃} {Γ : 𝔽↓T̃} → V τ Γ → ℳ τ Γ
    fstₘ : {τ σ : T̃} {Γ : 𝔽↓T̃} → ℳ (τ ∗ σ) Γ → ℳ τ Γ
    sndₘ : {τ σ : T̃} {Γ : 𝔽↓T̃} → ℳ (τ ∗ σ) Γ → ℳ σ Γ
    appₘ : {τ σ : T̃} {Γ : 𝔽↓T̃} → ℳ (τ ⇒ σ) Γ → 𝒩 τ Γ → ℳ σ Γ

  data 𝒩 : T̃ → 𝔽↓T̃ → Set where
    varₙ : {i : T} {Γ : 𝔽↓T̃} → V (θ i) Γ → 𝒩 (θ i) Γ
    fstₙ : {i : T} {σ : T̃} {Γ : 𝔽↓T̃} → ℳ (θ i ∗ σ) Γ → 𝒩 (θ i) Γ
    sndₙ : {i : T} {τ : T̃} {Γ : 𝔽↓T̃} → ℳ (τ ∗ θ i) Γ → 𝒩 (θ i) Γ
    appₙ : {i : T} {τ : T̃} {Γ : 𝔽↓T̃} → ℳ (τ ⇒ θ i) Γ → 𝒩 τ Γ → 𝒩 (θ i) Γ
    unitₙ : {Γ : 𝔽↓T̃} → 𝒩 1 Γ
    pairₙ : {τ σ : T̃} {Γ : 𝔽↓T̃} → 𝒩 τ Γ → 𝒩 σ Γ → 𝒩 (τ ∗ σ) Γ
    absₙ : {τ σ : T̃} {Γ : 𝔽↓T̃} → 𝒩 σ (Γ :: τ) → 𝒩 (τ ⇒ σ) Γ
```

Their presheaf actions (recall (7) and (8)) will be needed:

```
-- neutral and normal presheaf actions
mutual
  _[_]ₘ : {τ : T̃} {Δ Γ : 𝔽↓T̃} → ℳ τ Δ → 𝔽↓T̃( Δ , Γ ) → ℳ τ Γ
  varₘ x [ ρ ]ₘ = varₘ ( ρ x )
  fstₘ m [ ρ ]ₘ = fstₘ ( m [ ρ ]ₘ )
  sndₘ m [ ρ ]ₘ = sndₘ ( m [ ρ ]ₘ )
  appₘ m n [ ρ ]ₘ = appₘ ( m [ ρ ]ₘ ) ( n [ ρ ]ₙ )
```

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

```
_[_]ₙ : {τ : T̃} {Δ Γ : F↓T̃} → Nˢ τ Δ → F↓T̃( Δ , Γ ) → Nˢ τ Γ
varₙ x [ ρ ]ₙ = varₙ ( ρ x )
fstₙ m [ ρ ]ₙ = fstₙ ( m [ ρ ]ₘ )
sndₙ m [ ρ ]ₙ = sndₙ ( m [ ρ ]ₘ )
appₙ m n [ ρ ]ₙ = appₙ ( m [ ρ ]ₘ ) ( n [ ρ ]ₙ )
unitₙ [ ρ ]ₙ = unitₙ
pairₙ n₁ n₂ [ ρ ]ₙ = pairₙ ( n₁ [ ρ ]ₙ ) ( n₂ [ ρ ]ₙ )
absₙ n [ ρ ]ₙ = absₙ ( n [ lift ρ ]ₙ )
                where
                  lift : {τ : T̃} {Δ Γ : F↓T̃} → F↓T̃( Δ , Γ ) → F↓T̃( Δ :: τ , Γ :: τ )
                  lift _ • = •
                  lift ρ (↑ x) = ↑(ρ x)
```

Note the treatment of abstraction guaranteeing fresh bindings.

Semantics. We implement the presheaf semantics of types induced by the interpretation of base types as neutral terms (see (24)). Note that for higher types, the implementation glosses over the naturality condition required of presheaf exponentials (see (22)).

```
-- type semantics
⟦_⟧ : T̃ → F↓T̃ → Set
⟦ θ i ⟧ Γ = Mˢ (θ i) Γ
⟦ 1 ⟧ _ = ⊤
⟦ τ * σ ⟧ Γ = ⟦ τ ⟧ Γ × ⟦ σ ⟧ Γ
⟦ τ ⇒ σ ⟧ Γ = {Δ : F↓T̃} → F↓T̃( Γ , Δ ) → ⟦ τ ⟧ Δ → ⟦ σ ⟧ Δ

_[_] : {τ : T̃} {Δ Γ : F↓T̃} → ⟦ τ ⟧ Δ → F↓T̃( Δ , Γ ) → ⟦ τ ⟧ Γ
_[_] {θ _} = _[_]ₘ
_[_] {1} = _
_[_] {_ * _} ( x₁ , x₂ ) ρ = ( x₁ [ ρ ] , x₂ [ ρ ] )
_[_] {_ ⇒ _} f ρ ρ′ = f (ρ′ ∘ ρ)
```

The semantic interpretation of terms (see (20)) follows:

```
-- term semantics
Π : F↓T̃ → (T̃ → F↓T̃ → Set) → F↓T̃ → Set
Π • _ _ = ⊤
Π (Γ :: τ) P Δ = (Π Γ P Δ) × (P τ Δ)

⟦_⟧ : {τ : T̃} {Γ : F↓T̃} → ℒ τ Γ → {Δ : F↓T̃} → Π Γ ⟦_⟧ Δ → ⟦ τ ⟧ Δ
⟦ var x ⟧ ε = ε ⟨ x ⟩
          where
            _⟨_⟩ : {τ : T̃} {Γ Δ : F↓T̃} → Π Γ ⟦_⟧ Δ → V τ Γ → ⟦ τ ⟧ Δ
            ε ⟨ • ⟩ = π₂ ε
            ε ⟨ ↑ x ⟩ = π₁ ε ⟨ x ⟩
⟦ unit ⟧ _ = _
⟦ pair t₁ t₂ ⟧ ε = ( ⟦ t₁ ⟧ ε , ⟦ t₂ ⟧ ε )
⟦ fst t ⟧ ε = π₁ ( ⟦ t ⟧ ε )
⟦ snd t ⟧ ε = π₂ ( ⟦ t ⟧ ε )
⟦ abs t ⟧ ε f x = ⟦ t ⟧ ( ε [ f ]Π , x )
                 where
                   _[_]Π : {Ξ Δ Γ : F↓T̃} → Π Ξ ⟦_⟧ Δ → F↓T̃( Δ , Γ ) → Π Ξ ⟦_⟧ Γ
                   _[_]Π {•} = _
                   _[_]Π {_ :: _} ( ε , x ) ρ = ( ε [ ρ ]Π , x [ ρ ] )
⟦ app t₁ t₂ ⟧ ε = ⟦ t₁ ⟧ ε (λ x → x) ( ⟦ t₂ ⟧ ε )
```

Normalisation by evaluation. The unquote and quote functions (see (30)) are implemented:

```
-- unquote and quote
mutual
```

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

```
u : {τ : T̃} {Γ : F↓T̃} → M τ Γ → ⟦ τ ⟧ Γ
u {θ _} m = m
u {1} _ = _
u {_ * _} m = ( u (fst_m m) , u (snd_m m) )
u {_ ⇒ _} m ρ x = u ( app_m (m [ ρ ]_m) (q x) )

q : {τ : T̃} {Γ : F↓T̃} → ⟦ τ ⟧ Γ → N τ Γ
q {θ _} (var_m x) = var_n x
q {θ _} (fst_m m) = fst_n m
q {θ _} (snd_m m) = snd_n m
q {θ _} (app_m m n) = app_n m n
q {1} _ = unit_n
q {_ * _} (x1 , x2) = pair_n (q x1) (q x2)
q {_ ⇒ _} f = abs_n( q( f ↑ (u (var_m •)) ) )
```

Remark. A technical point to note is that the implementation of `q {_ ⇒ _} f` arises from the functorial action of presheaf exponentiation (in particular with respect to the evaluation map (23)), which in this case instatiates to the equivalent expression $\mathsf{abs_n}(\ \mathsf{q}(\ \mathsf{f}\ [\uparrow]\ (\lambda\, x \to x)\ (\mathsf{u}\ (\mathsf{var_m}\ \bullet))))$.

Finally, the normalisation function (see (31)) is

```
-- nbe
nf : {τ : T̃} {Γ : F↓T̃} → L τ Γ → N τ Γ
nf t = q ( ⟦ t ⟧ ( Π (u ∘ var_m) xs ) )
    where
      Π : (f : {τ : T̃} {Δ : F↓T̃} → V τ Δ → ⟦_⟧ τ Δ) {Γ Δ : F↓T̃} → Π Γ V Δ → Π Γ ⟦_⟧ Δ
      Π _ {·} _ = _
      Π f {_ :: _} ( xs , x ) = ( Π f xs , f x )

      xs : {Γ : F↓T̃} → Π Γ V Γ
      xs {·} = _
      xs {_ :: _} = ( xs [ ↑ ]_v , • )
                where
                  _[_]_v : {Ξ Δ Γ : F↓T̃} → Π Ξ V Δ → F↓T̃( Δ , Γ ) → Π Ξ V Γ
                  _[_]_v {·} = _
                  _[_]_v {_ :: _} ( xs , x ) ρ = ( xs [ ρ ]_v , ρ x )
```

4. 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 Fiore and Simpson (1999) via Grothendiek logical relations led to syntactic counterparts of the normal forms of Altenkirch et al. (2001) and has been applied to establish extensional normalisation for the typed lambda calculus with empty and sum types (Balat et al., 2004). 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 (Streicher, 2000), 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.

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

- 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 metatheory.
- The synthesis of a normalisation-by-evaluation program in a dependently typed functional programming language.

The obtained abstract normalisation algorithm synthesises various concrete implementations. Its specialisation to particular implementations of abstract syntax directly yields normalisation programs for concrete syntactic representations. In particular, we have provided a normalisation-by-evaluation program for the type-and-scope safe, intrinsically typed encoding of typed lambda terms (Allais et al., 2021; Altenkirch and Reus, 1999; Benton et al., 2012). How the abstract setting is related to representations of binding based on generating globally unique identifiers, say as in Filinski (2001), needs to be investigated.

The role of categorical glueing in our analysis is reminiscent of realisability. It would be interesting to understand whether there are connections to the modified realisability approach of Berger (1993).

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 Pierre-Louis Curien and Vincent Danos are gratefully acknowledged.

Note

1 The code is available from `www.cl.cam.ac.uk/~mpf23/Notes/Notes.html`.

References

Allais, G., Atkey, R., Chapman, J., McBride, C. and McKinna, J. (2021). A type- and scope-safe universe of syntaxes with binding: Their semantics and proofs. *Journal of Functional Programming* **31** E22.

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

Altenkirch, T., Dybjer, P., Hofmann, M. and Scott, P. (2001). Normalization by evaluation for typed lambda calculus with coproducts. In: *Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science*, 203–210.

Altenkirch, T., Hofmann, M. and Streicher, T. (1995). Categorical reconstruction of a *reduction-free* normalization proof. In: *Category Theory and Computer Science*, Lecture Notes in Computer Science, vol. 953, Springer-Verlag, 182–199.

Altenkirch, T. and Reus, B. (1999). Monadic presentations of lambda terms using generalized inductive types. In: *Proceedings of the 13th International Workshop on Computer Science Logic*, Lecture Notes in Computer Science, vol. 1683, Springer-Verlag, 453–468.

Balat, V., Di Cosmo, R. and Fiore, M. (2004). Extensional normalisation and type-directed partial evaluation for typed lambda calculus with sums. In: *Proceedings of the 31st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'04)*, 64–76.

Benton, N., Hur, C.-K., Kennedy, A. and McBride, C. (2012). Strongly typed term representations in Coq. *Journal of Automated Reasoning* **49** (2) 141–159.

Berger, U. (1993). Program extraction from normalization proofs. In: *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science, vol. 664, Springer-Verlag, 91–10.

Berger, U. and Schwichtenberg, H. (1991). An inverse of the evaluation functional for typed λ-calculus. In: *Proceedings of the 6th Annual IEEE Symposium on Logic in Computer Science*, 203–211.

Bezem, M. and Groote, J. (eds.) (1993). *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science, vol. 664, Springer, Springer-Verlag.

Coquand, C. (1994). From semantics to rules: A machine assisted analysis. In: Börger, E., Gurevich, Y. and Meinke, K. (eds.) *Proceedings of the Computer Science Logic'93*, Lecture Notes in Computer Science, vol. 832, Springer-Verlag.

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

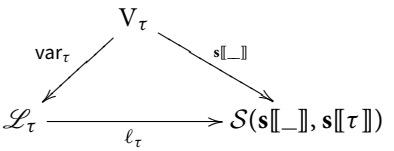
Crole, R. (1994). *Categories for Types*, Cambridge, Cambridge University Press.

Čubrić, D., Dybjer, P. and Scott, P. (1997). Normalization and the Yoneda embedding. *Mathematical Structures in Computer Science* **8** 153–192.

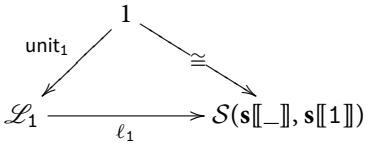
https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

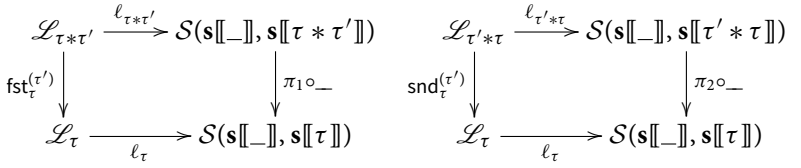
Danvy, O. (1998). Type-directed partial evaluation. In: *Partial Evaluation — Practise and Theory, Proceedings of the 1998 DIKU Summer School*, Lecture Notes in Computer Science, vol. 1706, Springer-Verlag, 367–411.
Danvy, O. and Dybjer, P. (eds.) (1998). *Proceedings of the 1998 APPSEM Workshop on Normalization by Evaluation (NBE'98)*, BRICS Note NS-98-8. Department of Computer Science, University of Aarhus.
Filinski, A. (1999). A semantic account of type-directed partial evaluation. In: *Principles and Practice of Declarative Programming*, Lecture Notes in Computer Science, vol. 1702, Springer-Verlag, 378–395.
Filinski, A. (2001). Normalization by evaluation for the computational lambda-calculus. In: *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science, Springer-Verlag.
Fiore, M. (2002). Semantic analysis of normalisation by evaluation for typed lambda calculus. In: *Proceedings of the 4th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming (PPDP'02)*, 26–37.
Fiore, M. and Hur, C.-K. (2010). Second-order equational logic. In: *Proceedings of the 19th EACSL Annual Conference on Computer Science Logic (CSL 2010)*, Lecture Notes in Computer Science, vol. 6247, Springer-Verlag, 320–335.
Fiore, M., Plotkin, G. and Turi, D. (1999). Abstract syntax and variable binding. In: *Proceedings of the 14th Annual IEEE Symposium on Logic in Computer Science*, 193–202.
Fiore, M. and Simpson, A. (1999). Lambda definability with sums via Grothendieck logical relations. In: *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science, vol. 1581, Springer-Verlag, 147–161.
Fiore, M. and Szamozvancev, D. (2022). Formal metatheory of second-order abstract syntax. In: *Proceedings of the 49th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL 2022)*.
`Fresh O'Caml`. In: `www.cl.cam.ac.uk/~amp12/fresh-ocaml/`.
Girard, J.-Y. (1972). Interprétation fonctionnelle et élimination des coupures dans l'arithmétique d'ordre supérieur. Thèse de doctorat d'état, Université Paris 7.
Jung, A. and Tiuryn, J. (1993). A new characterization of lambda definability. In: *Typed Lambda Calculi and Applications*, Lecture Notes in Computer Science, vol. 664, Springer-Verlag, 245–257.
Krivine, J. (1993). *Lambda-Calculus, Types and Models*, Computers and their Applications, Masson and Ellis Horwood.
Lambek, J. and Scott, P. (1986). *Introduction to Higher Order Categorical Logic*, Cambridge Studies in Advanced Mathematics, vol. 7, Cambridge, Cambridge University Press.
Lehmann, D. and Smyth, M. (1981). Algebraic specification of data types: A synthetic approach. *Mathematical Systems Theory* **14** 97–139.
Ma, Q. and Reynolds, J. (1992). Types, abstraction and parametric polymorphism, part 2. In: *Mathematical Foundations of Programming Semantics*, Lecture Notes in Computer Science, vol. 598, Springer-Verlag, 1–40.
Martin-Löf, P. (1975). About models for intuitionistic type theories and the notion of definitional equality. In: *Proceedings of the 3rd Scandinavian Logic Symposium*, 81–109.
Pfenning, F. and Elliot, C. (1988). Higher-order abstract syntax. In: *Proceedings of the ACM SIGPLAN'88 Symposium on Language Design and Implementation*.
Plotkin, G. (1973). Lambda-definability and logical relations. Technical report, School of Artificial Intelligence, University of Edinburgh.
Reynolds, J. (1998). Normalization and functor categories. In: *Proceedings of the 1998 APPSEM Workshop on Normalization by Evaluation (NBE'98)*, BRICS Note NS-98-8. Department of Computer Science, University of Aarhus, 33–36.
Statman, R. (1985). Logical relations and the typed lambda calculus. *Information and Control* **65** 85–97.
Streicher, T. (1998). Categorical intuitions underlying semantic normalisation proofs. In: *Proceedings of the 1998 APPSEM Workshop on Normalization by Evaluation (NBEâĂŹ98)*, BRICS Note NS-98-8. Department of Computer Science, University of Aarhus, 9–10.
Streicher, T. (2000). Denotational completeness revisited. In: *Electronic Notes in Theoretical Computer Science*, vol. 29, Elsevier Science Publishers, 288–300.
Tait, W. (1967). Intensional interpretation of functionals of finite type I. *Journal of Symbolic Logic* **32** (2) 198–212.
Taylor, P. (1999). *Practical Foundations of Mathematics*, Cambridge Studies in Advanced Mathematics, vol. 59, Cambridge, Cambridge University Press.
Wraith, G. (1974). Artin glueing. *Journal of Pure and Applied Algebra* **4** 345–348.

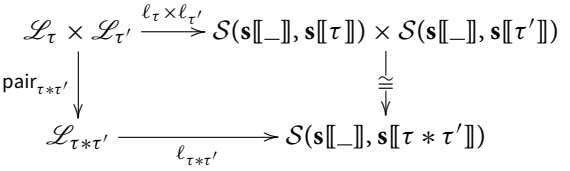
Appendix A. Homomorphism property of $\ell : \mathscr{L} \to \mathscr{C}$

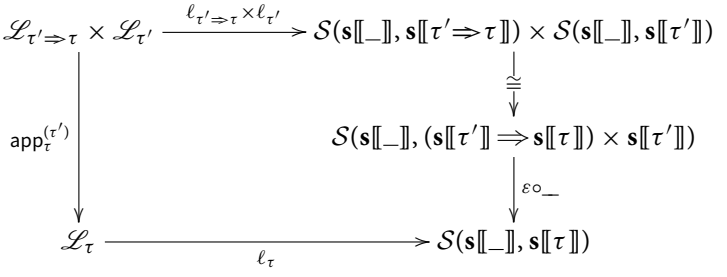


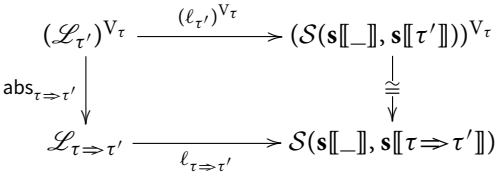
https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press



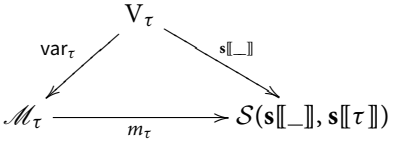




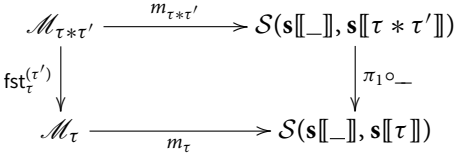




Appendix B. Homomorphism property of $(m, n) : (\mathscr{M}, \mathscr{N}) \to (\mathscr{C}, \mathscr{C})$



(B1)



(B2)

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

$$\begin{array}{ccc} \mathscr{M}_{\tau' * \tau} & \xrightarrow{m_{\tau' * \tau}} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau' * \tau]\!]) \\ {\scriptstyle \mathsf{snd}_{\tau}^{(\tau')}}\downarrow & & \downarrow{\scriptstyle \pi_2 \circ \_} \\ \mathscr{M}_{\tau} & \xrightarrow[m_{\tau}]{} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau]\!]) \end{array} \tag{B3}$$

$$\begin{array}{ccc} \mathscr{M}_{\tau' \Rightarrow \tau} \times \mathscr{N}_{\tau'} & \xrightarrow{m_{\tau' \Rightarrow \tau} \times n_{\tau'}} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau' \Rightarrow \tau]\!]) \times \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau']\!]) \\ & & \downarrow \cong \\ {\scriptstyle \mathsf{app}_{\tau}^{(\tau')}}\downarrow & & \mathcal{S}(\mathbf{s}[\![\_]\!], (\mathbf{s}[\![\tau']\!] \Rightarrow \mathbf{s}[\![\tau]\!]) \times \mathbf{s}[\![\tau']\!]) \\ & & \downarrow{\scriptstyle \varepsilon \circ \_} \\ \mathscr{M}_{\tau} & \xrightarrow[m_{\tau}]{} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau]\!]) \end{array} \tag{B4}$$

$$\begin{array}{ccc} & \mathrm{V}_{\theta} & \\ {\scriptstyle \mathsf{var}_{\theta}}\swarrow & & \searrow{\scriptstyle \mathbf{s}[\![\_]\!]} \\ \mathscr{N}_{\theta} & \xrightarrow[n_{\theta}]{} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\theta]\!]) \end{array} \tag{B5}$$

$$\begin{array}{ccc} \mathscr{M}_{\theta * \tau'} & \xrightarrow{m_{\theta * \tau'}} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\theta * \tau']\!]) \\ {\scriptstyle \mathsf{fst}_{\theta}^{(\tau')}}\downarrow & & \downarrow{\scriptstyle \pi_1 \circ \_} \\ \mathscr{N}_{\theta} & \xrightarrow[n_{\theta}]{} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\theta]\!]) \end{array} \tag{B6}$$

$$\begin{array}{ccc} \mathscr{M}_{\tau' * \theta} & \xrightarrow{m_{\tau' * \theta}} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau' * \theta]\!]) \\ {\scriptstyle \mathsf{snd}_{\theta}^{(\tau')}}\downarrow & & \downarrow{\scriptstyle \pi_2 \circ \_} \\ \mathscr{N}_{\theta} & \xrightarrow[n_{\theta}]{} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\theta]\!]) \end{array} \tag{B7}$$

$$\begin{array}{ccc} \mathscr{M}_{\tau' \Rightarrow \theta} \times \mathscr{N}_{\tau'} & \xrightarrow{m_{\tau' \Rightarrow \theta} \times n_{\tau'}} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau' \Rightarrow \theta]\!]) \times \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau']\!]) \\ & & \downarrow \cong \\ {\scriptstyle \mathsf{app}_{\theta}^{(\tau')}}\downarrow & & \mathcal{S}(\mathbf{s}[\![\_]\!], (\mathbf{s}[\![\tau']\!] \Rightarrow \mathbf{s}[\![\theta]\!]) \times \mathbf{s}[\![\tau']\!]) \\ & & \downarrow{\scriptstyle \varepsilon \circ \_} \\ \mathscr{N}_{\theta} & \xrightarrow[n_{\theta}]{} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\theta]\!]) \end{array} \tag{B8}$$

$$\begin{array}{ccc} & 1 & \\ {\scriptstyle \mathsf{unit}_1}\swarrow\cong & & \cong\searrow \\ \mathscr{N}_1 & \xrightarrow[n_1]{} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![1]\!]) \end{array} \tag{B9}$$

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

$$\begin{array}{ccc} \mathcal{N}_\tau \times \mathcal{N}_{\tau'} & \xrightarrow{n_\tau \times n_{\tau'}} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau]\!]) \times \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau']\!]) \\ \downarrow{\scriptstyle \mathsf{pair}_{\tau*\tau'}}\cong & & \downarrow\cong \\ \mathcal{N}_{\tau*\tau'} & \xrightarrow[n_{\tau*\tau'}]{} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau * \tau']\!]) \end{array} \tag{B10}$$

$$\begin{array}{ccc} (\mathcal{N}_{\tau'})^{\mathrm{V}_\tau} & \xrightarrow{(n_{\tau'})^{\mathrm{V}_\tau}} & (\mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau']\!]))^{\mathrm{V}_\tau} \\ \downarrow{\scriptstyle \mathsf{abs}_{\tau\Rightarrow\tau'}}\cong & & \downarrow\cong \\ \mathcal{N}_{\tau\Rightarrow\tau'} & \xrightarrow[n_{\tau\Rightarrow\tau'}]{} & \mathcal{S}(\mathbf{s}[\![\_]\!], \mathbf{s}[\![\tau\Rightarrow\tau']\!]) \end{array} \tag{B11}$$

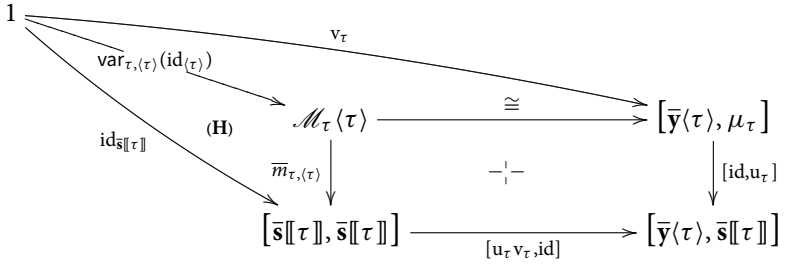
Appendix C. Proof of Theorem 21

For $\iota_\tau : \mathscr{P}_\tau \rightarrowtail \mathscr{M}_\tau$ and $\jmath_\tau : \mathscr{Q}_\tau \rightarrowtail \mathscr{N}_\tau$ the equalisers of (28) and (29), respectively, we show that $(\{\, \mathscr{P}_\tau \,\}_{\tau\in\widetilde{\mathrm{T}}}, \{\, \mathscr{Q}_\tau \,\}_{\tau\in\widetilde{\mathrm{T}}})$ is a sub $\langle\Sigma_1, \Sigma_2\rangle$-algebra of $(\mathscr{M}, \mathscr{N})$. That is, that we have the following situation

$$\begin{array}{ccc} \Sigma_1(\mathscr{P}, \mathscr{Q}) & \dashrightarrow & \mathscr{P} \\ \downarrow{\scriptstyle \Sigma_1(\iota,\jmath)} & & \downarrow{\scriptstyle \iota} \\ \Sigma_1(\mathscr{M}, \mathscr{N}) & \xrightarrow{\cong} & \mathscr{M} \end{array} \qquad \begin{array}{ccc} \Sigma_2(\mathscr{P}, \mathscr{Q}) & \dashrightarrow & \mathscr{Q} \\ \downarrow{\scriptstyle \Sigma_2(\iota,\jmath)} & & \downarrow{\scriptstyle \jmath} \\ \Sigma_2(\mathscr{M}, \mathscr{N}) & \xrightarrow{\cong} & \mathscr{N} \end{array}$$

Below, we will use the following conventions: **(H)** indicates commutativity by the homomorphism property; **(I)** and **(J)**, respectively, indicate commutativity by the definition of ι and $\jmath$; and **(Q)** and **(U)**, respectively, indicate commutativity by the definition of q and u.

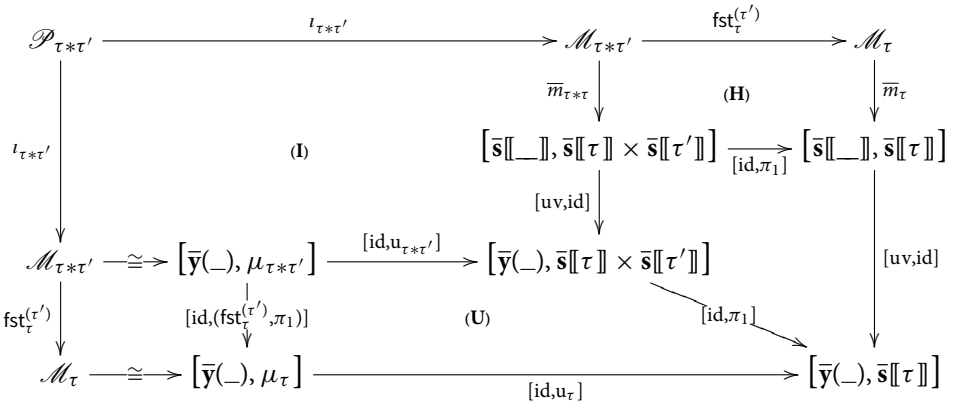
(1) For $\tau \in \widetilde{\mathrm{T}}$, the map $\mathrm{V}_\tau \xrightarrow{\mathsf{var}_\tau} \mathscr{M}_\tau$ equalises diagram (28), and hence factors through $\mathscr{P}_\tau \xrightarrow{\iota_\tau} \mathscr{M}_\tau$, because the diagram



commutes.

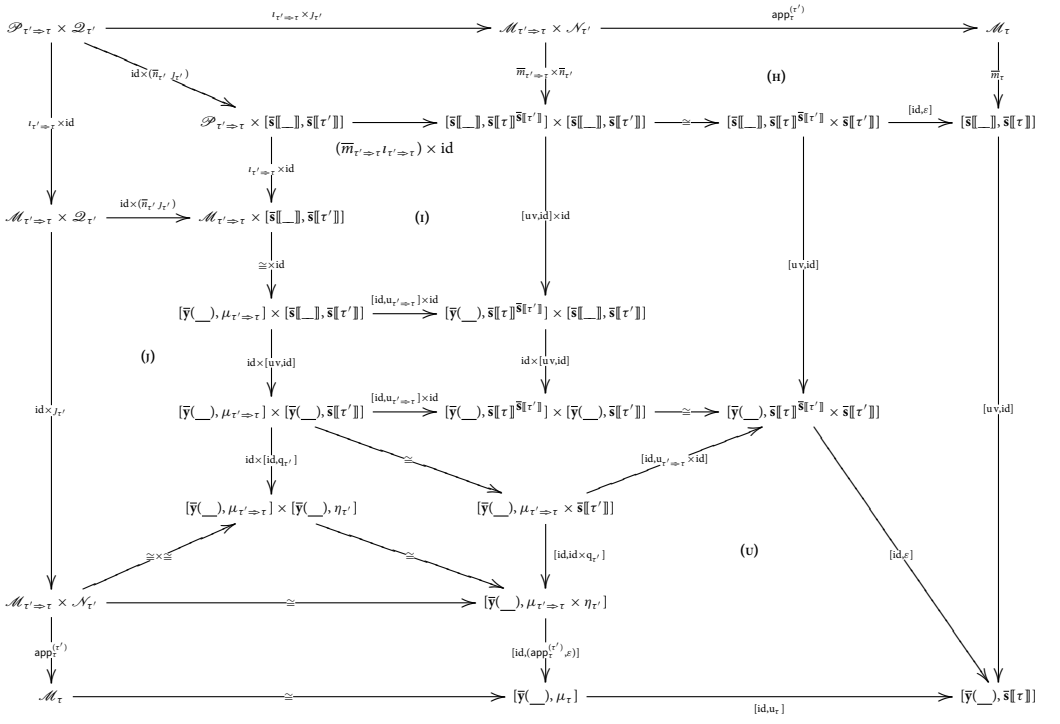
(2) For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the map $\mathscr{P}_{\tau*\tau'} \xrightarrow{\iota_{\tau*\tau'}} \mathscr{M}_{\tau*\tau'} \xrightarrow{\mathsf{fst}_\tau^{(\tau')}} \mathscr{M}_\tau$ equalises diagram (28), and hence factors through $\mathscr{P}_\tau \xrightarrow{\iota_\tau} \mathscr{M}_\tau$, as shown by the diagram below.

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press



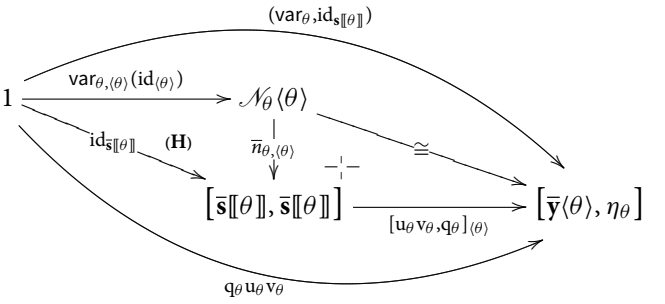
Analogously, for $\tau, \tau' \in \mathrm{T}$, the map $\mathscr{P}_{\tau'*\tau} \xrightarrow{\iota_{\tau'*\tau}} \mathscr{M}_{\tau'*\tau} \xrightarrow{\mathsf{snd}_\tau^{(\tau')}} \mathscr{M}_\tau$ equalises diagram (28), and hence also factors through $\mathscr{P}_\tau \xrightarrow{\iota_\tau} \mathscr{M}_\tau$.

(3) For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the map $\mathscr{P}_{\tau' \Rightarrow \tau} \times \mathscr{Q}_{\tau'} \xrightarrow{\iota_{\tau' \Rightarrow \tau} \times J_{\tau'}} \mathscr{M}_{\tau' \Rightarrow \tau} \times \mathscr{N}_{\tau'} \xrightarrow{\mathsf{app}_\tau^{(\tau')}} \mathscr{M}_\tau$ equalises diagram (28), and hence factors through $\mathscr{P}_\tau \xrightarrow{\iota_\tau} \mathscr{M}_\tau$, as shown by the diagram below.



(4) For $\theta \in \mathrm{T}$, the map $\mathrm{V}_\theta \xrightarrow{\mathsf{var}_\theta} \mathscr{N}_\theta$ equalises diagram (29) with $\tau = \theta$, and hence factors through $\mathscr{Q}_\theta \xrightarrow{J\theta} \mathscr{N}_\theta$ because the diagram

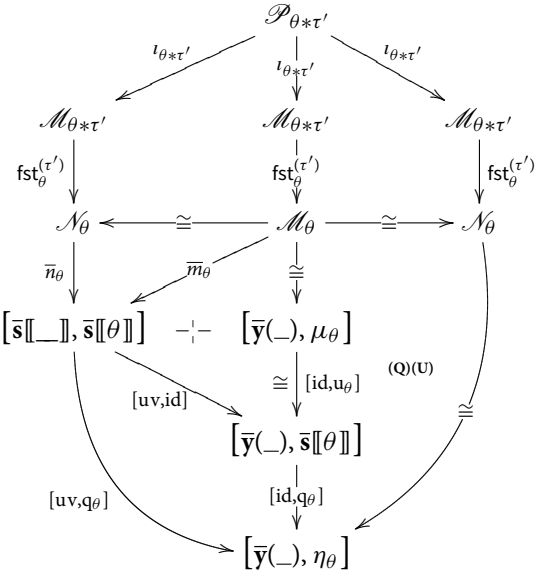
https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press



in **Set**

commutes.

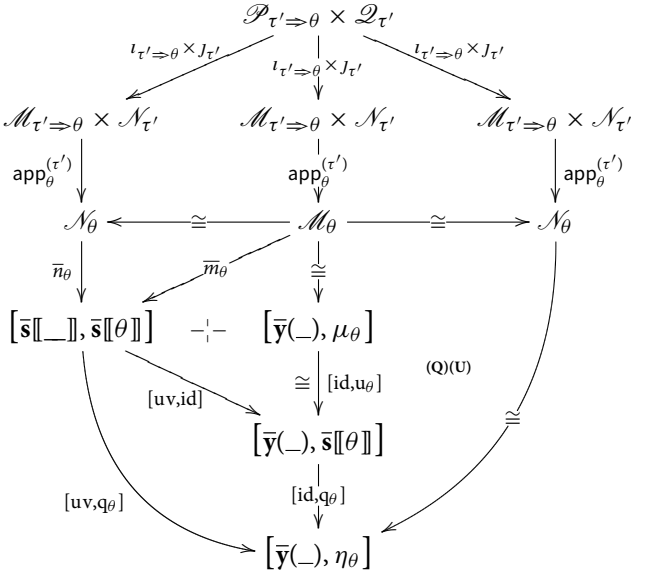
(5) For $\theta \in \mathrm{T}$ and $\tau' \in \widetilde{\mathrm{T}}$, the map $\mathscr{P}_{\theta * \tau'} \xrightarrow{\iota_{\theta*\tau'}} \mathscr{M}_{\theta*\tau'} \xrightarrow{\mathsf{fst}_\theta^{(\tau')}} \mathscr{N}_\theta$ equalises diagram (29) with $\tau = \theta$, and hence factors through $\mathscr{Q}_\theta \xrightarrow{J\theta} \mathscr{N}_\theta$, as shown by the diagram below (which depends on the diagram in item 2 above with $\tau = \theta$).



Analogously, for $\theta \in \mathrm{T}$ and $\tau' \in \widetilde{\mathrm{T}}$, the map $\mathscr{P}_{\tau' * \theta} \xrightarrow{\iota_{\tau'*\theta}} \mathscr{M}_{\tau'*\theta} \xrightarrow{\mathsf{snd}_\theta^{(\tau')}} \mathscr{N}_\theta$ equalises diagram (29) for $\tau = \theta$, and hence also factors through $\mathscr{Q}_\theta \xrightarrow{J\theta} \mathscr{N}_\theta$.

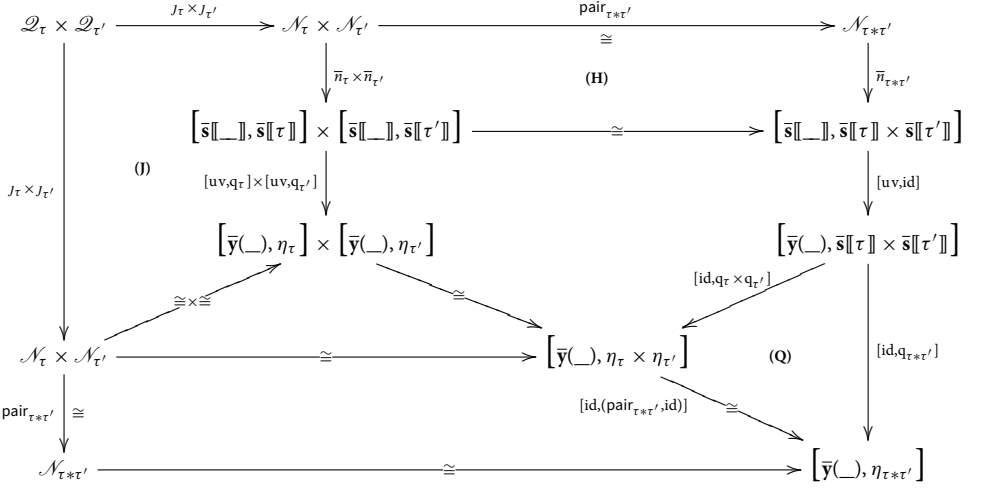
(6) For $\theta \in \mathrm{T}$ and $\tau' \in \widetilde{\mathrm{T}}$, the map $\mathscr{P}_{\tau' \Rightarrow \theta} \times \mathscr{Q}_{\tau'} \xrightarrow{\iota_{\tau' \Rightarrow \theta} \times J_{\tau'}} \mathscr{M}_{\tau' \Rightarrow \theta} \times \mathscr{N}_{\tau'} \xrightarrow{\mathsf{app}_\theta^{(\tau')}} \mathscr{N}_\theta$ equalises diagram (29) with $\tau = \theta$, and hence factors through $\mathscr{Q}_\theta \xrightarrow{J\theta} \mathscr{N}_\theta$, as shown by the diagram below (which depends on the diagram in item 3 above with $\tau = \theta$).

https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press



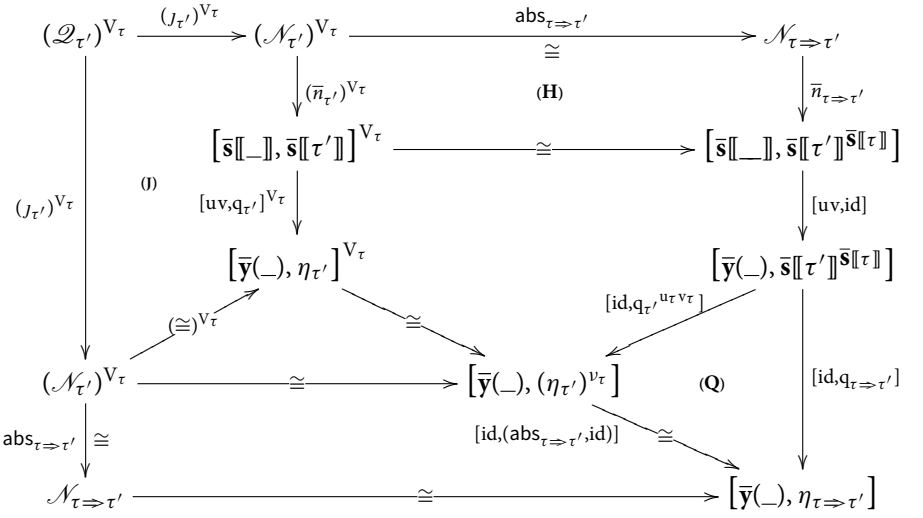
(7) Diagram (29) with $\tau = 1$ commutes, and hence the map $1 \xrightarrow[\cong]{\mathsf{unit}_1} \mathscr{N}_1$ factors through the equaliser $\mathscr{Q}_1 \rightarrowtail[\cong] \mathscr{N}_1$.

(8) For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the map $\mathscr{Q}_\tau \times \mathscr{Q}_{\tau'} \xrightarrow{J_\tau \times J_{\tau'}} \mathscr{N}_\tau \times \mathscr{N}_{\tau'} \xrightarrow[\cong]{\mathsf{pair}_{\tau * \tau'}} \mathscr{N}_{\tau * \tau'}$ equalises diagram (29), and hence factors through $\mathscr{Q}_{\tau*\tau'} \rightarrowtail[J_{\tau*\tau'}] \mathscr{N}_{\tau*\tau'}$, as shown by the diagram below.



https://doi.org/10.1017/S0960129522000263 Published online by Cambridge University Press

(9) For $\tau, \tau' \in \widetilde{\mathrm{T}}$, the map $(\mathscr{Q}_{\tau'})^{\mathrm{V}_\tau} \xrightarrow{(J_{\tau'})^{\mathrm{V}_\tau}} (\mathscr{N}_{\tau'})^{\mathrm{V}_\tau} \xrightarrow[\cong]{\mathsf{abs}_{\tau\Rightarrow\tau'}} \mathscr{N}_{\tau\Rightarrow\tau'}$ equalises diagram (29), and hence factors through $\mathscr{Q}_{\tau\Rightarrow\tau'} \xrightarrow{J_{\tau\Rightarrow\tau'}} \mathscr{N}_{\tau\Rightarrow\tau'}$, as shown by the diagram below.



Cite this article: Fiore M (2022). Semantic analysis of normalisation by evaluation for typed lambda calculus. *Mathematical Structures in Computer Science* **32**, 1028–1065. https://doi.org/10.1017/S0960129522000263