The category of open simply-typed lambda terms

Daniel Murfet, William Troiani

November 22, 2017

Abstract

We define a category of types and terms in the simply-typed lambda calculus, in which every term (open or closed) is a morphism, function application is composition, and lambda abstraction is given by the universal factorisation of a term through the subset of morphisms with a restriction on their free variable set.

1 Introduction

Lambda calculus is about *variables* and *substitution*, and so the category theory of lambda calculus should be structured around categorical realisations of these concepts. However, in the standard approach to the category of lambda terms [5] it is instead the Cartesian structure arising from *product types* (which, from the point of view of lambda calculus, are optional) which is given central place, and lambda abstraction is interpreted in terms of the adjunction between the functors $\sigma \times (-)$ and $\sigma \Rightarrow (-)$. The purpose of this paper is to remedy this conceptual mismatch, by defining the category of (open) simply-typed lambda terms and explaining how the basic operations of lambda abstraction and function application are represented by natural categorical algebra *without* products.

The guiding philosophy here is Lawvere's thesis [9] that logical constants (in our case, implication) should be characterised by adjoint functors; see also [2, §9]. In this context one way of explaining our approach is that we emphasise the role of adjoints to the functors of the form $\mathcal{L} \longrightarrow \mathcal{L}[x]$, where $\mathcal{L}[x]$ denotes a polynomial category. The existence of a left adjoint to such a functor, often referred to as functional completeness [3], depends on products, but the right adjoint exists more generally and encodes lambda abstraction [1, 2]. The primary advantage of this approach is that it puts open and closed terms on an equal footing, and it gives a monadic interpretation of the structural rules of logic (Section 5.2), which in turn allows for an extension of the Curry-Howard correspondence to open terms.

Acknowledgements. The work presented here was supported by DSTG grant ABC1234.

2 Background

In order to make this paper as accessible as possible, we include a brief introduction to the simply-typed lambda calculus. In the simply-typed lambda calculus [7, Chapter 3] there is a countable set of *atomic types* and the set $\Phi_{\Rightarrow}$ of (*simple*) *types* is built up from the atomic types using the connective $\Rightarrow$. That is, all atomic types are simple types and if σ, τ are simple types then so is $\sigma \Rightarrow \tau$. Following the usual convention we omit brackets, so that $\tau \Rightarrow \rho \Rightarrow \sigma$ means $\tau \Rightarrow (\rho \Rightarrow \sigma)$, see [7, Convention 2.1.2]. For each type σ there is a countable set Y_σ of variables of type σ, and if $\sigma \neq \tau$ then $Y_\sigma \cap Y_\tau = \emptyset$. We write $x : \sigma$ for $x \in Y_\sigma$.

Let Λ' denote the set of (untyped) lambda calculus preterms in the variables $\bigcup_{\sigma \in \Phi_{\Rightarrow}} Y_\sigma$, as defined in [7, Chapter 1]. We define a subset $\Lambda'_{wt} \subseteq \Lambda'$ of *well-typed* preterms, together with a function $t : \Lambda'_{wt} \longrightarrow \Phi_{\Rightarrow}$ by induction:

- all variables $x : \sigma$ are well-typed and $t(x) = \sigma$,
- if $M = (PQ)$ and P, Q are well-typed with $t(P) = \sigma \Rightarrow \tau$ and $t(Q) = \sigma$ for some σ, τ then M is well-typed and $t(M) = \tau$,
- if $M = (\lambda x \,.\, N)$ with N well-typed, then M is well-typed and $T(M) = t(x) \Rightarrow t(N)$.

We define $\Lambda'_\sigma = \{M \in \Lambda'_{wt} \,|\, t(M) = \sigma\}$ and call these *preterms of type* σ. Next we observe that $\Lambda'_{wt} \subseteq \Lambda'$ is closed under the relation of α-equivalence on Λ', as long as we understand α-equivalence type by type, that is, we take

$$\lambda x \,.\, M =_\alpha \lambda y \,.\, M[x := y]$$

as long as $t(x) = t(y)$. Denoting this relation by $=_\alpha$, we may therefore define the sets of *well-typed lambda terms* and *well-typed lambda terms of type* σ, respectively:

$$\Lambda_{wt} = \Lambda'_{wt} / =_\alpha \tag{2.1}$$

$$\Lambda_\sigma = \Lambda'_\sigma / =_\alpha \ . \tag{2.2}$$

Note that Λ_{wt} is the disjoint union over all $\sigma \in \Phi_{\Rightarrow}$ of Λ_σ. We write $M : \sigma$ as a synonym for $[M] \in \Lambda_\sigma$, and call these equivalence classes *terms of type* σ. Since terms are, by definition, α-equivalence classes, the expression $M = N$ henceforth means $M =_\alpha N$ unless indicated otherwise. We denote the set of free variables of a term M by $\mathrm{FV}(M)$. Recall that a term M is called *closed* if $\mathrm{FV}(M) = \emptyset$ and otherwise it is *open*.

We briefly recall the definition of β and η equivalence, for which see [7, Chapter 3]. The equivalence relation $=_\beta$ on Λ_σ is generated by one-step β-reductions of the form

$$((\lambda x \,.\, M)N) \longrightarrow_\beta M[x := N] \tag{2.3}$$

where $N, x : \sigma$ and $M : \sigma \Rightarrow \tau$. The equivalence relation $=_\eta$ on Λ_σ is generated by

$$\lambda x \,.\, (Mx) =_\eta M \tag{2.4}$$

for any $x \notin \mathrm{FV}(M), x : \sigma, M : \sigma \Rightarrow \tau$.

Definition 2.1. $=_{\beta\eta}$ is the smallest equivalence relation on Λ_{wt} containing $=_\beta \cup =_\eta$.

3 The category of simply-typed lambda terms

The basic idea is that the types and terms of lambda calculus can be organised as the objects and morphisms of a category $\mathcal{L}$. To motivate the introduction of category theory into the theory of lambda calculus, let us observe that once we impose $\beta\eta$-equivalence on the set of typed terms, there are distinct types that become "identified" as one can see in the next example.

Definition 3.1. For every type σ let $\mathrm{id}_\sigma = \lambda x^\sigma . x$.

Remark 3.2. Observe that for a term $M : \sigma \Rightarrow \tau$, we have

$$\begin{aligned}\lambda t^\sigma . (M(\mathrm{id}_\sigma\, t)) &= \lambda t^\sigma . (M((\lambda x^\sigma . x)t)) \\ &=_\beta \lambda t . (Mt) \\ &=_\eta M \,,\end{aligned}$$

and similarly $\lambda s^\tau . (\mathrm{id}_\tau(Ms)) =_{\beta\eta} M$.

Example 3.3. Let σ, τ, ρ be types and consider

$$\begin{aligned}T_1 &= \sigma \Rightarrow (\tau \Rightarrow \rho)\,, \\ T_2 &= \tau \Rightarrow (\sigma \Rightarrow \rho)\,.\end{aligned}$$

Consider the following terms M_{12}, M_{21}

$$\begin{aligned}M_{12} &= \lambda u^{\sigma\Rightarrow(\tau\Rightarrow\rho)} v^\tau w^\sigma . ((uw)v) \\ M_{21} &= \lambda u^{\tau\Rightarrow(\sigma\Rightarrow\rho)} w^\sigma v^\tau . ((uv)w)\end{aligned}$$

which are respectively of types $M_{12} : T_1 \Rightarrow T_2$ and $M_{21} : T_2 \Rightarrow T_1$. Recall that the composition of terms F, G is $\lambda x . (F(Gx))$ where x is not in the free variable set of either F or G. Then for a variable $t : T_2$ notice that

$$\begin{aligned}\lambda t . (M_{12}(M_{21}t)) &=_\beta \lambda t . (M_{12}(\lambda w^\sigma v^\tau . ((tv)w))) \\ &=_\beta \lambda t \bar{v}^\tau \bar{w}^\sigma . ((\lambda w^\sigma v^\tau . ((tv)w))\bar{w})\bar{v}) \\ &=_\beta \lambda t \bar{v}^\tau \bar{w}^\sigma . ((\lambda v^\tau . ((tv)\bar{w}))\bar{v}) \\ &=_\beta \lambda t \bar{v}^\tau \bar{w}^\sigma . ((t\bar{v})\bar{w}) \\ &=_\eta \lambda t \bar{v}^\tau . (t\bar{v}) \\ &=_\eta \lambda t . t \\ &= \mathrm{id}_{T_2} \;.\end{aligned}$$

Similarly, one checks that $\lambda s . (M_{21}(M_{12}s)) =_{\beta\eta} \mathrm{id}_{T_1}$. In conclusion, we deduce from this calculation that if we work up to $\beta\eta$-equivalence, M_{12}, M_{21} behave (under the above rule for composition of terms of compatible types) like *isomorphisms* between T_1 and T_2.

One consequence of this fact is that there are bijections for any type ω, of the form

$$\Lambda_{\omega \Rightarrow T_1}/{=_{\beta\eta}} \xrightarrow{\cong} \Lambda_{\omega \Rightarrow T_2}/{=_{\beta\eta}}$$
$$\Lambda_{T_1 \Rightarrow \omega}/{=_{\beta\eta}} \xrightarrow{\cong} \Lambda_{T_2 \Rightarrow \omega}/{=_{\beta\eta}}\ .$$

This is the kind of fact which is best understood using a category $\mathcal{L}$ of lambda terms.

Our aim is to define a category of lambda terms, subject to:

Desiderata 3.4. The category $\mathcal{L}$ of lambda terms should satisfy:

A) Every lambda term (open or closed) is represented by a morphism in $\mathcal{L}$.

B) Every simple type is an object of $\mathcal{L}$.

C) The fundamental operations on lambda terms (function application and lambda abstraction) are represented by natural categorical operations on $\mathcal{L}$.

Remark 3.5. Let us record some of the natural expectations beyond the above desiderata about how the category $\mathcal{L}$ should work. Temporarily we use a notation which exaggerates the distinction between a term M in lambda calculus, and the associated arrow $\mathrm{a}(M)$ in our putative category $\mathcal{L}$, which has yet to be defined:

1. We expect a term $G : \sigma \Rightarrow \tau$ to have an associated arrow $\mathrm{a}(G) \in \mathcal{L}(\sigma, \tau)$ and, given another term $F : \tau \Rightarrow \rho$ with associated arrow $\mathrm{a}(F) \in \mathcal{L}(\tau, \rho)$ we expect composition in $\mathcal{L}$ to be as in lambda calculus, that is
$$\mathrm{a}(F) \circ \mathrm{a}(G) = \mathrm{a}\left(\lambda x\,.\,(F(Gx))\right).$$

2. In order to have identity morphisms we should take morphisms to be $\beta\eta$-equivalence classes of terms, since η-equivalence is used in Remark 3.2.

3. In order to fulfill desiderata A, for σ atomic, every term $M : \sigma$ must be interpreted as a morphism $\mathrm{a}(M)$ in $\mathcal{L}$. For this reason we add a new object $\mathbf{1}$ to our category, and take $\mathrm{a}(M) \in \mathcal{L}(\mathbf{1}, \sigma)$.

Definition 3.6. The category $\mathcal{L}$ has objects

$$\mathrm{ob}(\mathcal{L}) = \Phi_{\Rightarrow} \cup \{\mathbf{1}\}$$

and morphisms given for types $\sigma, \tau \in \Phi_{\Rightarrow}$ by

$$\begin{aligned}
\mathcal{L}(\sigma, \tau) &= \Lambda_{\sigma \Rightarrow \tau}/{=_{\beta\eta}}\\
\mathcal{L}(\mathbf{1}, \sigma) &= \Lambda_{\sigma}/{=_{\beta\eta}}\\
\mathcal{L}(\sigma, \mathbf{1}) &= \{\star\}\\
\mathcal{L}(\mathbf{1}, \mathbf{1}) &= \{\star\}\,,
\end{aligned}$$

where $\star$ is a new symbol. For $\sigma, \tau, \rho \in \Phi_{\Rightarrow}$ the composition rule is the function

$$\begin{aligned}\mathcal{L}(\tau, \rho) \times \mathcal{L}(\sigma, \tau) &\longrightarrow \mathcal{L}(\sigma, \rho)\\ (N, M) &\longmapsto \lambda x^{\sigma} \,.\, (N(Mx))\,,\end{aligned}$$

where $x \notin \mathrm{FV}(N) \cup \mathrm{FV}(M)$. We write the composite as $N \circ M$. In the remaining special cases the composite is given by the rules

$$\begin{aligned}\mathcal{L}(\tau, \rho) \times \mathcal{L}(\mathbf{1}, \tau) &\longrightarrow \mathcal{L}(\mathbf{1}, \rho)\,, \qquad & N \circ M &= (NM)\,,\\ \mathcal{L}(\mathbf{1}, \rho) \times \mathcal{L}(\mathbf{1}, \mathbf{1}) &\longrightarrow \mathcal{L}(\mathbf{1}, \rho)\,, \qquad & N \circ \star &= N\,,\\ \mathcal{L}(\mathbf{1}, \rho) \times \mathcal{L}(\sigma, \mathbf{1}) &\longrightarrow \mathcal{L}(\sigma, \rho)\,, \qquad & N \circ \star &= \lambda t^{\sigma} \,.\, N\,,\end{aligned}$$

where in the final rule $t \notin \mathrm{FV}(N)$. All other cases are trivial. Note that these functions, which have been described using a choice of representatives from a $\beta\eta$-equivalence class, are nonetheless well-defined.

For terms $M, N : \sigma \Rightarrow \tau$ the expression $M = N$ is currently ambiguous, since it could either denote an equality of terms up to α-equivalence (as it has done up to now) or an equality of morphisms in $\mathcal{L}$ (which would mean $\beta\eta$-equivalence).

Proposition 3.7. *$\mathcal{L}$ is a category.*

Proof. The calculation in Remark 3.2 shows that $\mathrm{id}_\sigma \in \mathcal{L}(\sigma, \sigma)$ is an identity at σ, and $\star$ is clearly an identity at $\mathbf{1}$. For associativity there are a few cases to check:

- Consider a diagram of objects and morphisms in $\mathcal{L}$ of the form:

$$\delta \xleftarrow{\;P\;} \rho \xleftarrow{\;N\;} \tau \xleftarrow{\;M\;} \sigma\,. \tag{3.1}$$

$$\begin{aligned}P \circ (N \circ M) &= \lambda y^{\sigma} \,.\, (P(N \circ M\, y))\\ &= \lambda y^{\sigma} \,.\, (P((\lambda x^{\sigma} \,.\, (N(Mx)))y))\\ &=_{\beta} \lambda y^{\sigma} \,.\, (P(N(My)))\\ &=_{\beta} (P \circ N) \circ M\,.\end{aligned}$$

- Consider a diagram of objects and morphisms in $\mathcal{L}$ of the form

$$\delta \xleftarrow{\;P\;} \rho \xleftarrow{\;N\;} \tau \xleftarrow{\;M\;} \mathbf{1}\,. \tag{3.2}$$

$$\begin{aligned}P \circ (N \circ M) &= P \circ (NM)\\ &= (P(NM))\\ &= (\lambda y^{\tau} \,.\, (P(Ny))M)\\ &= (P \circ N) \circ M\,.\end{aligned}$$

- Consider a diagram of objects and morphisms in $\mathcal{L}$ of the form

$$\delta \xleftarrow{P} \rho \xleftarrow{N} \mathbf{1} \xleftarrow{\star} \sigma\,. \tag{3.3}$$

$$\begin{aligned}(P \circ N) \circ \star &= (PN) \circ \star \\ &= \lambda t^{\sigma}\,.\,(PN) \\ &= \lambda t^{\sigma}(P((\lambda z^{\sigma}\,.\,N)t)) \\ &= P \circ (N \circ \star)\,.\end{aligned}$$

- Consider a diagram of objects and morphisms in $\mathcal{L}$ of the form

$$\delta \xleftarrow{P} \mathbf{1} \xleftarrow{\star} \tau \xleftarrow{M} \sigma\,. \tag{3.4}$$

$$\begin{aligned}(P \circ \star) \circ M &= (\lambda t^{\tau}\,.\,P) \circ M \\ &= \lambda q^{\sigma}\,.\,((\lambda t^{\tau}\,.\,P)(Mq)) \\ &= \lambda q^{\sigma}\,.\,P \\ &= P \circ (\star \circ M)\,.\end{aligned}$$

The other cases are trivial. $\square$

Definition 3.8. Let $\mathcal{L}^{\neq \mathbf{1}}$ denote the full subcategory of $\mathcal{L}$ whose object set is $\Phi_{\Rightarrow}$. That is, we omit the terminal object $\mathbf{1}$.

So far we have realised Desiderata A and B, and it remains to explain how function application and lambda abstraction are interpreted by natural categorical operations. Function application is clearly present in $\mathcal{L}$, since given terms $M : \tau$ and $N : \tau \Rightarrow \rho$ we have a commutative diagram in $\mathcal{L}$

$$\begin{tikzcd} \mathbf{1} \arrow[r, "M"] \arrow[dr, "(NM)"'] & \tau \arrow[d, "N"] \\ & \rho \end{tikzcd} \tag{3.5}$$

Remark 3.9. Returning now to Example 3.3, we have morphisms $M_{12} \in \mathcal{L}(T_1, T_2)$ and $M_{21} \in \mathcal{L}(T_2, T_1)$ and we have calculated that $M_{12} \circ M_{21} = 1_{T_2}$ and $M_{21} \circ M_{12} = 1_{T_1}$ in $\mathcal{L}$, so that the objects T_1, T_2 are isomorphic.

More generally, it is easy to check that

Proposition 3.10. *Given types $\tau_1, \dots, \tau_k, \sigma, \rho$ and a permutation $\theta \in S_k$, the term*

$$P_\theta : \big(\tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho\big) \Rightarrow \big(\tau_{\theta(1)} \Rightarrow \cdots \Rightarrow \tau_{\theta(k)} \Rightarrow \rho\big)$$
$$P_\theta = \lambda U^{\tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho} v_1^{\tau_{\theta(1)}} v_2^{\tau_{\theta(2)}} \cdots v_k^{\tau_{\theta(k)}} \,.\, \big(\cdots ((U v_{\theta^{-1}(1)}) v_{\theta^{-1}(2)}) \cdots v_{\theta^{-1}(k)} \big)$$

is an isomorphism in $\mathcal{L}$ between the objects

$$\big(\tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho\big) \cong \big(\tau_{\theta(1)} \Rightarrow \cdots \Rightarrow \tau_{\theta(k)} \Rightarrow \rho\big)\,.$$

With the notation of the proposition:

Corollary 3.11. *There is a bijection*

$$\Lambda_{\tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho} / {=_{\beta\eta}} \xrightarrow{\ \cong\ } \Lambda_{\tau_{\theta(1)} \Rightarrow \cdots \Rightarrow \tau_{\theta(k)} \Rightarrow \rho} / {=_{\beta\eta}} \ .$$

Proof. We have, by the proposition

$$\begin{aligned} \Lambda_{\tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho} / {=_{\beta\eta}} &= \mathcal{L}(\mathbf{1}, \tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho) \\ &\cong \mathcal{L}(\mathbf{1}, \tau_{\theta(1)} \Rightarrow \cdots \Rightarrow \tau_{\theta(k)} \Rightarrow \rho) \\ &= \Lambda_{\tau_{\theta(1)} \Rightarrow \cdots \Rightarrow \tau_{\theta(k)} \Rightarrow \rho} / {=_{\beta\eta}} \ . \end{aligned}$$

□

By construction $\mathbf{1}$ is a terminal object in $\mathcal{L}$. It is also a generator [**?**, Definition 4.5.1].

Lemma 3.12. *The object $\mathbf{1}$ is a generator in $\mathcal{L}$. That is, if $M, N \in \mathcal{L}(\sigma, \rho)$ are such that $M \circ W = N \circ W$ in $\mathcal{L}$ for all $W \in \mathcal{L}(\mathbf{1}, \sigma)$, then $M = N$ in $\mathcal{L}$.*

Proof. Suppose that $M \circ W = N \circ W$ as morphisms in $\mathcal{L}$ for all W. Then we may take W to be a variable $w : \sigma$ which is not free in any term β-equivalent to either M or N. Then

$$M =_\eta \lambda w \,.\, (Mw) = \lambda w \,.\, (M \circ w) =_{\beta\eta} \lambda w \,.\, (N \circ w) =_\eta N\,.$$

□

4 Lambda abstraction via support

The existence of free variables works somewhat against our natural inclination to think of a term M of type $\sigma \Rightarrow \rho$ as a (set-theoretic) function with domain σ and codomain ρ. We should rather think of the type as a kind of *guarantee* that A will "yield" an output of type ρ on an input of type σ, but where the process involved in this yielding may involve values (via the free variables) from the environment. For example, consider the term

$$M = (\lambda x^\sigma \,.\, y^\tau) : \sigma \Rightarrow \tau\,,$$

which has $\mathrm{FV}(M) = \{y\}$. The dependency of M on y is exposed at the level of the type by forming the lambda abstraction $\lambda q\,.\,M : \tau \Rightarrow (\sigma \Rightarrow \tau)$. This explicit dependency can once again be hidden, by a function application

$$((\lambda q\,.\,M)y) =_\beta M\,.$$

This dynamic between *exposing* and *hiding* dependencies is the subject of this section. First of all, let us observe that the above discussion can be summarised in a commutative diagram. Given a variable $q : \tau$ we write

$$\mathscr{U}^q = \lambda u^{\tau \Rightarrow \rho}\,.\,(uq)\,.$$

There is a commutative diagram in $\mathcal{L}$

(4.1)
$$\begin{tikzcd}
\mathbf{1} \arrow[rr, "M"] \arrow[dr, "\lambda q\,.\,M"'] & & \sigma \Rightarrow \rho \\
& \rho \Rightarrow (\tau \Rightarrow \rho) \arrow[ur, "\mathscr{U}^q"'] &
\end{tikzcd}$$

since

$$\begin{aligned}
\mathscr{U}^q \circ (\lambda q\,.\,M) &= \big(\mathscr{U}^q(\lambda q\,.\,M)\big)\\
&= \big((\lambda u\,.\,(uq))(\lambda q\,.\,M)\big)\\
&=_\beta ((\lambda q\,.\,M)q)\\
&=_\beta M\,.
\end{aligned}$$

As we will show, this diagram is actually the *universal* way of factoring the morphism M through a morphism which, as a term, *does not* contain q in its free variable set (explicitly, the factorisation is the lambda abstraction $\lambda q\,.\,M$). This gives a characterisation of lambda abstraction via a simple universal property. To state and prove this properly, we first need to analyse what we mean by the free variable set of a morphism in $\mathcal{L}$. This is not well-defined, since the free variable set is not invariant under β-reduction.

Example 4.1. For variables $x : \tau$ and $a, y : \sigma$

$$\mathrm{FV}\big(((\lambda a\,.\,x)y)\big) = \{x, y\} \neq \{x\} = \mathrm{FV}(x)\,.$$

The relation $\twoheadrightarrow_\beta$ is multi-step β-reduction [7, Definition 1.3.3].

Lemma 4.2. *If $M \twoheadrightarrow_\beta N$ then* $\mathrm{FV}(N) \subseteq \mathrm{FV}(M)$.

Proof. In a one-step β-reduction $((\lambda x\,.\,P)Q) \rightarrow_\beta P[x := Q]$ the free variables on the reduced term were either already free in P or Q. $\square$

Definition 4.3. Given a term M we define

$$\mathrm{FV}_\beta(M) = \bigcap_{N \in [M]} \mathrm{FV}(N)$$

where the intersection is over all terms N which are β-equivalent to M.

Since the simply-typed lambda calculus is strongly normalising [7, Theorem 3.5.1] and confluent [7, Theorem 3.6.3] there is a unique normal form $\widehat{M}$ in the β-equivalence class of M, and it is equivalent to define

$$\mathrm{FV}_\beta(M) = \mathrm{FV}(\widehat{M})\,.$$

Lemma 4.4. *Given terms $M : \sigma \Rightarrow \rho$ and $N : \sigma$ we have*

$$\mathrm{FV}_\beta((MN)) \subseteq \mathrm{FV}_\beta(M) \cup \mathrm{FV}_\beta(N)\,.$$

Proof. We may assume M, N normal, in which case there is a chain of β-reductions

$$(MN) \twoheadrightarrow_\beta \widehat{(MN)}$$

whence we are done by the previous lemma. □

The inclusion in the previous lemma may be strict, consider $M = (\lambda a\,.\,x)$ and $N = y$ as above. By the same argument, we have

Lemma 4.5. *Given $M : \sigma \Rightarrow \rho$ and $N : \tau \Rightarrow \sigma$ we have*

$$\mathrm{FV}_\beta(M \circ N) \subseteq \mathrm{FV}_\beta(M) \cup \mathrm{FV}_\beta(N)\,. \tag{4.2}$$

Definition 4.6. Let **Var** denote the partially ordered set of finite subsets of

$$Y = \bigcup_{\sigma \in \Phi_\Rightarrow} Y_\sigma\,.$$

At this point we may view $\mathrm{FV}_\beta(-)$ as a function on $\mathcal{L}$, sending morphisms to elements of the partially ordered set **Var** and having a weak functoriality expressed by (4.2). To make sense of this, we adopt the convention that $\mathrm{FV}_\beta(\star) = \emptyset$. In fact, if we view $\mathcal{L}$ as a 2-category with only identity 2-morphisms and **Var** as a 2-category with one object, and composition of 1-morphisms (finite subsets of Y) as union, then FV_β is a colax functor $\mathcal{L} \longrightarrow \mathbf{Var}$.

Definition 4.7. For a subset $Q \subseteq Y$ we define a subcategory $\mathcal{L}_Q \subseteq \mathcal{L}$ by

$$\mathrm{ob}(\mathcal{L}_Q) = \mathrm{ob}(\mathcal{L}) = \Phi_\Rightarrow \cup \{\mathbf{1}\}$$

and for types σ, ρ

$$\begin{aligned}
\mathcal{L}_Q(\sigma,\rho) &= \{M \in \mathcal{L}(\sigma,\rho) \mid \mathrm{FV}_\beta(M) \subseteq Q\}\,,\\
\mathcal{L}_Q(\mathbf{1},\sigma) &= \{M \in \mathcal{L}(\mathbf{1},\sigma) \mid \mathrm{FV}_\beta(M) \subseteq Q\}\,,\\
\mathcal{L}_Q(\sigma,\mathbf{1}) &= \mathcal{L}(\sigma,\mathbf{1}) = \{\star\}\,,\\
\mathcal{L}_Q(\mathbf{1},\mathbf{1}) &= \mathcal{L}(\mathbf{1},\mathbf{1}) = \{\star\}\,.
\end{aligned}$$

Note that the last two lines have the same form as the first two, using the convention that $\mathrm{FV}_\beta(\star) = \emptyset$. We denote the inclusion functor by $I_Q : \mathcal{L}_Q \longrightarrow \mathcal{L}$. We write $\mathcal{L}_{cl}$ for $\mathcal{L}_Q$ when $Q = \emptyset$ and call this the category of *closed* lambda terms.

The fact that $\mathcal{L}_Q$ is a subcategory follows from Lemma 4.5. We claim that the inclusion I_Q has a right adjoint, provided Q is *cofinite*, by which we mean that $Q^c = Y \setminus Q$ is a finite set. Our convention is to use letters $\mathfrak{p}, \mathfrak{q}, \ldots$ for ordered sets of variables, with $\mathfrak{q}$ always denoting an ordering on the finite unordered set of variables Q^c. With this notation, we next define a functor

$$\Gamma_{\mathfrak{q}} : \mathcal{L} \longrightarrow \mathcal{L}_Q$$

which we will prove is right adjoint to I_Q, with counit a natural transformation

$$\mathscr{U}^{\mathfrak{q}} : I_Q \circ \Gamma_{\mathfrak{q}} \longrightarrow 1_{\mathcal{L}}\,.$$

For the rest of this section let Q be a cofinite set of variables and $\mathfrak{q} = (q_1 : \tau_1, \ldots, t_k : \tau_k)$ an ordering of the complement. While the functor $\Gamma_{\mathfrak{q}}$ and natural transformation $\mathscr{U}^{\mathfrak{q}}$ depend on the choice of ordering, by the uniqueness of adjoints they are independent of the ordering up to unique natural isomorphism.

Definition 4.8. For a type ρ we define

$$\Gamma_{\mathfrak{q}}(\rho) = \tau_1 \Rightarrow \tau_2 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho$$

which is ρ if Q is empty. We set $\Gamma_{\mathfrak{q}}(\mathbf{1}) = \mathbf{1}$. For types σ, τ we define a function

$$\Gamma_{\mathfrak{q}} : \mathcal{L}(\sigma,\tau) \longrightarrow \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\sigma, \Gamma_{\mathfrak{q}}\tau) \tag{4.3}$$

on a term $M : \sigma \Rightarrow \tau$ by

$$\Gamma_{\mathfrak{q}}(M) = \lambda U^{\tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \sigma} q_1^{\tau_1} \cdots q_k^{\tau_k} \,.\, \big(M(\cdots(Uq_1)\cdots q_k)\big)\,. \tag{4.4}$$

Since it is clear by inspection that $\mathrm{FV}_\beta(\Gamma_{\mathfrak{q}}M) \subseteq \mathrm{FV}_\beta(M) \setminus Q^c$ we have $\Gamma_{\mathfrak{q}}M \in \mathcal{L}_Q$. In the special cases involving $\mathbf{1}$ we define $\Gamma_{\mathfrak{q}}$ by

$$\begin{aligned}
\mathcal{L}(\sigma,\mathbf{1}) \longrightarrow \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\sigma, \Gamma_{\mathfrak{q}}\mathbf{1}) = \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\sigma, \mathbf{1})\,, &\qquad \star \mapsto \star\\
\mathcal{L}(\mathbf{1},\rho) \longrightarrow \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\mathbf{1}, \Gamma_{\mathfrak{q}}\rho) = \mathcal{L}_Q(\mathbf{1}, \Gamma_{\mathfrak{q}}\rho)\,, &\qquad M \mapsto \lambda q_1^{\tau_1} \cdots q_k^{\tau_k} \,.\, M\\
\mathcal{L}(\mathbf{1},\mathbf{1}) \longrightarrow \mathcal{L}_Q(\Gamma_{\mathfrak{q}}\mathbf{1}, \Gamma_{\mathfrak{q}}\mathbf{1}) = \mathcal{L}_Q(\mathbf{1},\mathbf{1}) &\qquad \star \mapsto \star\,.
\end{aligned}$$

Remark 4.9. It is important in (4.4) that we lambda abstract over the particular variables q_i that belong to Q^c. By α-equivalence the result of a lambda abstraction is independent of the variable we use *if* the term being lambda abstracted does not contain that variable as a free variable. However we are certainly interested in the case where M *does* contain the q_i as free variables, and in these cases $\Gamma_{\mathfrak{q}}(M)$ defined using, say, a sequence of variables $v_1^{\tau_1}, \ldots, v_k^{\tau_k}$ distinct from $\mathfrak{q}$ would be a different morphism in $\mathcal{L}$.

Lemma 4.10. *$\Gamma_{\mathfrak{q}}$ is a functor $\mathcal{L} \longrightarrow \mathcal{L}_Q$.*

With the same notation as in Definition 4.8:

Definition 4.11. For a type ρ we define $\mathscr{U}_\rho^{\mathfrak{q}} \in \mathcal{L}(\Gamma_{\mathfrak{q}}\rho, \rho)$ by

$$\mathscr{U}_\rho^{\mathfrak{q}} = \lambda U^{\Gamma_{\mathfrak{q}}\rho} \,.\, (\cdots((Uq_1)q_2)\cdots q_k)\,. \tag{4.5}$$

Once again, it is significant that we use the sequence of variables $\mathfrak{q}$ to form this term, and not arbitrary variables of the same type. The special case is $\mathscr{U}_{\mathbf{1}}^{\mathfrak{q}} \in \mathcal{L}(\Gamma_{\mathfrak{q}}\mathbf{1}, \mathbf{1}) = \mathcal{L}(\mathbf{1}, \mathbf{1})$ given by $\mathscr{U}_{\mathbf{1}}^{\mathfrak{q}} = \star$.

Lemma 4.12. *$\mathscr{U}^{\mathfrak{q}}$ is a natural transformation $I_Q \circ \Gamma_{\mathfrak{q}} \longrightarrow 1_{\mathcal{L}}$.*

Proof. For simplicity let us assume $Q^c = \{q : \tau\}$. We have to show for $M \in \mathcal{L}(\sigma, \rho)$ that

$$\begin{tikzcd}
\sigma \arrow[r, "M"] & \rho \\
\tau \Rightarrow \sigma \arrow[u, "\mathscr{U}_\sigma^q = \lambda U\,.\,(Uq)"] \arrow[r, "\lambda U' q\,.\,(M(U'q))"'] & \tau \Rightarrow \rho \arrow[u, "\mathscr{U}_\rho^q = \lambda V\,.\,(Vq)"']
\end{tikzcd}$$

commutes. This is the following calculation:

$$\begin{aligned}
\mathscr{U}_\rho^q \circ (\lambda U' q\,.\,(M(U'q))) &=_\beta \lambda t^{\tau\Rightarrow\sigma}\,.\,\big((\lambda V\,.\,(Vq))\,(\lambda q\,.\,(M(tq)))\big)\\
&=_\beta \lambda t\,.\,\big((\lambda q\,.\,(M(tq)))q\big)\\
&=_\beta \lambda t\,.\,\big(M(tq)\big)\\
&=_\beta \lambda t\,.\,\big(M((\lambda U\,.\,(Uq))t)\big)\\
&= M \circ \mathscr{U}_\sigma^q\,.
\end{aligned}$$

There are some other special cases involving $\mathbf{1}$ which are easily checked. □

Theorem 4.13. *The functor $\Gamma_{\mathfrak{q}}$ is right adjoint to I_Q with counit $\mathscr{U}^{\mathfrak{q}}$.*

Remark 4.14. In particular, for types σ, ρ there are natural bijections

$$\mathcal{L}(\sigma, \rho) = \mathcal{L}(I_Q(\sigma), \rho) \cong \mathcal{L}_Q(\sigma, \Gamma_{\mathfrak{q}}\rho)\,, \tag{4.6}$$
$$\mathcal{L}(\mathbf{1}, \rho) = \mathcal{L}(I_Q(\mathbf{1}), \rho) \cong \mathcal{L}_Q(\mathbf{1}, \Gamma_{\mathfrak{q}}\rho)\,. \tag{4.7}$$

These are the only cases with any content. The special cases involving **1** are just equalities

$$\begin{aligned}\mathcal{L}(I_Q(\mathbf{1}),\mathbf{1}) = \mathcal{L}(\mathbf{1},\mathbf{1}) = \{\star\} = \mathcal{L}_Q(\mathbf{1},\mathbf{1}) = \mathcal{L}_Q(\mathbf{1},\Gamma_{\mathfrak{q}}(\mathbf{1}))\\ \mathcal{L}(I_Q(\sigma),\mathbf{1}) = \mathcal{L}(\sigma,\mathbf{1}) = \{\star\} = \mathcal{L}_Q(\sigma,\mathbf{1}) = \mathcal{L}_Q(\sigma,\Gamma_{\mathfrak{q}}(\mathbf{1}))\,.\end{aligned}$$

Proof. We give the proof of (4.6), and (4.7) is the same. So we have to prove that

$$\Psi : \mathcal{L}_Q(\sigma, \Gamma_{\mathfrak{q}}\rho) \longrightarrow \mathcal{L}(\sigma,\rho), \qquad \Psi(M) = \mathscr{U}^{\mathfrak{q}}_{\rho} \circ M$$

is a bijection. We define a function Ψ', which we will prove is an inverse to Ψ, by taking a term $N : \sigma \Rightarrow \rho$ and forming

$$\lambda q_1^{\tau_1} \cdots q_k^{\tau_k} \,.\, N : \tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \sigma \Rightarrow \rho\,.$$

Under the bijection of Corollary 3.11 this term corresponds to a term of type $\sigma \Rightarrow \tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho$ and we define $\Psi'(N)$ to be this term, which we will denote $P[\lambda q_1^{\tau_1} \cdots q_k^{\tau_k} \,.\, N]$ in what follows. Observe that by construction this term does not have any of the q_i in its free variable set, so this is a morphism in $\mathcal{L}_Q$.

Observe that

$$\begin{aligned}\Psi\Psi'(N) &= \mathscr{U}^{\mathfrak{q}}_{\rho} \circ P[\lambda q_1 \cdots q_k \,.\, N]\\ &= \lambda t^{\sigma} \,.\, (\mathscr{U}^{\mathfrak{q}}_{\rho}(P[\lambda q_1 \cdots q_k \,.\, N]t))\\ &= \lambda t \,.\, \Big(\lambda U^{\Gamma_{\mathfrak{q}}\rho} \,.\, (\cdots((U q_1)q_2)\cdots q_k) \;\; (P[\lambda q_1 \cdots q_k \,.\, N]t)\Big)\\ &=_{\beta} \lambda t \,.\, (\cdots(((P[\lambda q_1 \cdots q_k \,.\, N]t)q_1)q_2)\cdots q_k)\\ &=_{\beta\eta} \lambda t \,.\, ((\cdots(((\lambda q_1 \cdots q_k \,.\, N)q_1)q_2)\cdots q_k)t)\\ &=_{\beta} \lambda t \,.\, (Nt)\\ &=_{\eta} N\,.\end{aligned}$$

Let $M : \sigma \Rightarrow \tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho$ be given without any of the q_i in $\mathrm{FV}_{\beta}(M)$ and therefore also not in $\mathrm{FV}(M)$. Then

$$\begin{aligned}\Psi'\Psi(M) &= P[\lambda q_1 \ldots q_k \,.\, \Psi(M)]\\ &= P\Big[\lambda q_1 \ldots q_k t^{\sigma} \,.\, \Big(\lambda U^{\Gamma_{\mathfrak{q}}\rho} \,.\, (\cdots((U q_1)q_2)\cdots q_k) \;\; (Mt)\Big)\Big]\\ &=_{\beta} P\Big[\lambda q_1 \ldots q_k t^{\sigma} \,.\, (\cdots(((Mt)q_1)q_2)\cdots q_k)\Big]\\ &= \lambda t q_1 \ldots q_k \,.\, (\cdots(((Mt)q_1)q_2)\cdots q_k)\\ &=_{\eta} M\,.\end{aligned}$$

Note that the hypothesis that $Q^c \cap \mathrm{FV}_{\beta}(M) = \emptyset$ is used to justify the application of the η-rules in the final line. $\square$

In fact the same proof shows something more general:

Theorem 4.15. *Given $P \subseteq Q \subseteq Y$ with $A = Q \setminus P$ finite the inclusion $I : \mathcal{L}_P \longrightarrow \mathcal{L}_Q$ has a right adjoint.*

Proof. The adjoint $\Gamma_{\mathfrak{a}}$ is determined by choosing an ordering on A, and using the same formulas from Definition 4.8, and the counit of adjunction $\mathscr{U}^{\mathfrak{a}}$ as in Definition 4.11. $\square$

In conclusion, we have fulfilled Desiderata C by showing that function application and lambda abstraction in the simply-typed lambda calculus are realised as natural categorical algebra in $\mathcal{L}$. Function application is composition, and lambda abstraction is given by a universal property involving factorisation of morphisms in $\mathcal{L}$ through the collection of morphisms satisfying a "support" condition.

Concretely, given a morphism $M \in \mathcal{L}(\sigma, \rho)$ and a cofinite subset $Q \subseteq Y$ we consider the condition on a morphism f of $\mathcal{L}$ that $\mathrm{FV}_\beta(f) \subseteq Q$ (that is, the morphism f is "supported" on the set of variables Q). We can consider the set of all factorisations of M through morphisms of this form, that is, the set of all commutative diagrams

$$\begin{tikzcd} \sigma \arrow[rr, "M"] \arrow[dr, "f"'] & & \rho \\ & \kappa \arrow[ur] & \end{tikzcd} \tag{4.8}$$

where $\mathrm{FV}_\beta(f) \subseteq Q$. We have shown that among these commutative diagrams there is a universal one (that is, the category of such diagrams has an initial object) and it is

$$\begin{tikzcd} \sigma \arrow[rr, "M"] \arrow[dr, "\lambda q_1 \cdots q_k \,.\, M"'] & & \rho \\ & \tau_1 \Rightarrow \cdots \Rightarrow \tau_k \Rightarrow \rho \arrow[ur, "\mathscr{U}^{\mathfrak{q}}"'] & \end{tikzcd} \tag{4.9}$$

for any ordering $\mathfrak{q}$ of Q^c.

Remark 4.16. The above is related to the idea of *functional completeness*, but is different. Observe that for a term $M : \rho$ and variable $q_q : \tau_1$ there is a commutative diagram in $\mathcal{L}$

$$\begin{tikzcd} \mathbf{1} \arrow[rr, "M"] \arrow[dr, "q_1"'] & & \rho \\ & \tau_1 \arrow[ur, "\lambda q_1 \,.\, M"'] & \end{tikzcd} \tag{4.10}$$

which factors M as an arrow via a morphism $\lambda q_1 \,.\, M$ which lies in $\mathcal{L}_{\{q_1\}^c}$. However if we perform another lambda abstraction, we obtain

$$\lambda q_2^{\tau_2} q_1^{\tau_1} \,.\, M : \tau_2 \Rightarrow (\tau_1 \Rightarrow \rho) \tag{4.11}$$

which does not have an interpretation as an arrow *into* ρ, and therefore cannot be the second morphism in a pair factorising M in $\mathcal{L}$. If we add product types to the underlying language, then we can interpret the term in (4.11) as being of type $\tau_2 \times \tau_1 \Rightarrow \rho$ and there is a factorisation of the desired kind

$$
\begin{tikzcd}
\mathbf{1} \arrow[rr, "M"] \arrow[dr, "\langle q_1, q_2 \rangle"'] & & \rho \\
& \tau_1 \times \tau_2 \arrow[ur, "\lambda q_2 q_1 \, . \, M"'] &
\end{tikzcd} \tag{4.12}
$$

Note also that the diagram (4.10) cannot be generalised to arrows $M \in \mathcal{L}(\sigma, \rho)$ without products, since what we want to write is the commutative diagram

$$
\begin{tikzcd}
\sigma \arrow[rr, "M"] \arrow[dr, "q_1"'] & & \rho \\
& \tau_1 \times \sigma \arrow[ur, "\lambda q_1 \, . \, M"'] &
\end{tikzcd} \tag{4.13}
$$

where q_1 here denotes the composite

$$\sigma \cong \mathbf{1} \times \sigma \xrightarrow{\begin{pmatrix} q_1 \\ 1_\sigma \end{pmatrix}} \tau_1 \times \sigma$$

and we identify $\lambda q_1 \, . \, M : \tau_1 \Rightarrow (\sigma \Rightarrow \rho)$ with a term of type $\tau_1 \times \sigma \Rightarrow \rho$ in the natural way. In this way one can show that if we add products, the inclusion $\mathcal{L}_Q \longrightarrow \mathcal{L}$ (again in the case where Q is cofinite) has also a *left* adjoint given by $\sigma \mapsto \tau_1 \times \cdots \times \tau_k \times \sigma$, see [1, 2, 8]. This distinction between left and right adjoints is referred to as context completeness versus functional completeness in [8, Definition 2.1]. The structural rules of weakening and contraction can be interpreted in terms of the counit and comultiplication of the comonad, in the case where $\mathcal{L}_{cl} \longrightarrow \mathcal{L}_Q$ has a left adjoint [8, p.505].

5 Comparison to standard approach

In the standard approach to associating a category to the simply-typed lambda calculus, due to Lambek and Scott [5, §I.11], one extends the lambda calculus to include product types, denoted $\sigma \times \tau$, and the objects of the category $\mathcal{C}_{\Rightarrow,\times}$ are the types of the extended calculus (which includes an empty product $\mathbf{1}$) and the set $\mathcal{C}_{\Rightarrow,\times}(\sigma, \rho)$ is a set of equivalence classes of pairs $(x : \sigma, M : \rho)$ where x is a variable and M is a term with $\mathrm{FV}(M) \subseteq \{x\}$.

Despite appearances, $\mathcal{C}_{\Rightarrow,\times}$ should really be thought of as the category of *closed* lambda terms, as the following theorem makes clear. Let $\mathcal{C}_{\Rightarrow}$ denote the full subcategory of $\mathscr{C}_{\Rightarrow,\times}$ whose objects are elements of the set $\Phi_{\Rightarrow}$.

Theorem 5.1. *There is an equivalence of categories*

$$F : \mathcal{C}_{\Rightarrow} \to \mathcal{L}_{cl}^{\neq \mathbf{1}}$$

which is the identity $F(\tau) = \tau$ on objects, and is defined on morphisms by

$$F_{\sigma,\rho} : \mathcal{C}_{\Rightarrow}(\sigma, \rho) \longrightarrow \mathcal{L}_{cl}^{\neq \mathbf{1}}(\sigma, \rho)\,, \qquad (x : \sigma, M : \rho) \mapsto \lambda x \,.\, M\,.$$

Proof. Note that by definition if $(x : \sigma, M : \rho) \in \mathcal{C}_{\Rightarrow}(\sigma, \rho)$ then $\mathrm{FV}(M) \subseteq \{x\}$ and so $\lambda x \,.\, M$ is a closed term. To see that F is a functor, consider two morphisms

$$(x : \sigma, M : \rho) \in \mathcal{C}_{\Rightarrow}(\sigma, \rho)\,, \qquad (y : \rho, N : \tau) \in \mathcal{C}_{\Rightarrow}(\rho, \tau)\,.$$

Then

$$\begin{aligned} F((y : \rho, N : \tau) \circ (x : \sigma, M : \rho)) &= F(x : \sigma, ((\lambda y^{\rho} \,.\, N)M)) \\ &= \lambda x^{\sigma} \,.\, ((\lambda y^{\rho} \,.\, N)M) \\ &= \lambda z^{\sigma} \,.\, \big((\lambda y^{\rho} \,.\, N)\; ((\lambda x^{\sigma} \,.\, M)z)\big) \\ &= (\lambda y^{\rho} \,.\, N) \circ (\lambda x^{\sigma} \,.\, M) \\ &= F(y : \rho, N : \tau) \circ F(x : \sigma, M : \rho) \end{aligned}$$

where in the equality marked by $*$, z is such that $z \notin \{\mathrm{FV}(\lambda y^{\rho}.N) \cup \mathrm{FV}(\lambda x^{\tau}.M)\}$. The proof that F preserves identities is similarly straightforward.

The function $F_{\sigma,\rho}$ is injective, since if $\lambda x^{\sigma} \,.\, M^{\rho} = \lambda y^{\sigma} \,.\, N^{\rho}$ in $\mathcal{L}$ then

$$M =_{\beta} ((\lambda x \,.\, M)x) =_{\beta\eta} ((\lambda y \,.\, N)x) =_{\beta} N[y := x]\,.$$

But this is the definition of the equivalence relation on morphisms in $\mathcal{C}_{\Rightarrow}$, see [5, p.78].

To see that $F_{\sigma,\rho}$ is surjective for all σ, ρ, let us suppose otherwise for a contradiction. Then among all pairs (σ, ρ) for which $F_{\sigma,\rho}$ fails to be surjective, and among all $N \in \mathcal{L}_{cl}^{\neq \mathbf{1}}(\sigma, \rho)$ not in the image of $F_{\sigma,\rho}$, we may choose an N in normal form and of minimal length. By hypothesis there is no variable $x : \sigma$ and term $M : \rho$ with $\mathrm{FV}(M) \subseteq \{x\}$ and $N =_{\beta\eta} \lambda x \,.\, M$, and consequently N must either be a variable or a function application. It cannot be a variable since it is closed, and if it were a function application $N = (PQ)$ then $P : \alpha \Rightarrow (\sigma \Rightarrow \rho)$ and $Q : \alpha$ for some type α. But now P is shorter than N and cannot be a lambda abstraction (else N would not be normal), which is a contradiction. □

5.1 Monads

As above, let $\mathcal{L}_{cl}$ denote the category of closed lambda terms. Throughout this section, $A \subseteq Y$ is finite and so in particular the inclusion $\emptyset \subseteq A$ satisfies the conditions of Theorem 4.15 and there is a right adjoint $\Gamma_{\mathfrak{a}}$ to the inclusion I for any ordering $\mathfrak{a}$ of A:

(5.1)
$$\begin{tikzcd} \mathcal{L}_{cl} \arrow[r, shift left, "I"] & \mathcal{L}_{A} \arrow[l, shift left, "\Gamma_{\mathfrak{a}}"] \end{tikzcd}\,.$$

Definition 5.2. Denote by $T_{\mathfrak{a}}$ the monad $\Gamma_{\mathfrak{a}} \circ I$ on $\mathcal{L}_{cl}$.

It is easy to see that

Proposition 5.3. *$\mathcal{L}_A$ is equivalent to the Kleisli category of $T_{\mathfrak{a}}$.*

In the case where $\mathfrak{a} = \{x : \alpha\}$ the monad $T_{\mathfrak{a}}$ has multiplication μ given by

$$\mu_\sigma = \lambda u^{\alpha \Rightarrow (\alpha \Rightarrow \sigma)} x^\alpha.((ux)x) : (\alpha \Rightarrow (\alpha \Rightarrow \sigma)) \Rightarrow \sigma$$

and unit ξ given by

$$\xi_\sigma = \lambda w^\sigma x^\alpha.w : \sigma \Rightarrow (\alpha \Rightarrow \sigma)\,.$$

Remark 5.4. We write $\mathcal{L}_{cl}^{\neq \mathbf{1}}$ for the category whose objects are types $\Phi_{\Rightarrow}$ and whose morphisms are closed lambda terms, and observe that the monad $T_{\mathfrak{q}}$ on $\mathcal{L}_{cl}$ for a finite set of ordered variables $\mathfrak{q}$ restricts to a monad on this subcategory. We define a $\mathfrak{q}$-*environment* to be a natural transformation

$$c^E : T_{\mathfrak{q}}|_{\mathcal{L}_{cl}^{\neq \mathbf{1}}} \longrightarrow 1_{\mathcal{L}_{cl}^{\neq \mathbf{1}}}\,.$$

Any closed term $E : \tau$ gives rise to an environment c^E defined by

$$\begin{aligned} c^E &: T_{\{y:\tau\}}|_{\mathcal{L}_{cl}^{\neq \mathbf{1}}} \longrightarrow 1_{\mathcal{L}_{cl}^{\neq \mathbf{1}}} \\ c^E_\rho &= \lambda U^{\tau \Rightarrow \rho}\,.\,(UE)\,. \end{aligned}$$

Given $M \in \mathcal{L}_{\{y:\tau\}}(\sigma, \rho)$ and an environment c we can determine the evaluation of M in the environment by taking the composite $c_\rho \circ (\lambda q\,.\,M)$ as in the diagram

(5.2)
$$\begin{tikzcd} \sigma \arrow[rr, "M"] \arrow[dr, "\lambda q\,.\,M"'] & & \rho \\ & \tau \Rightarrow \rho \arrow[ur, "\mathscr{U}^{\mathfrak{q}}"'] \arrow[d, "c_\rho"] & \\ & \rho & \end{tikzcd}$$

In the case $c = c^E$ observe that

$$\begin{aligned} c^E_\rho \circ (\lambda q\,.\,M) &= \lambda z^\sigma\,.\,\big((\lambda U\,.\,(UE))\;((\lambda q\,.\,M)z)\big) \\ &=_{\beta\eta} \lambda z\,.\,(((\lambda q\,.\,M)E)z) \\ &=_\beta \lambda z\,.\,(M[q := E]z) \\ &=_\eta M[q := E]\,, \end{aligned}$$

which is indeed the evaluation of M in the context $E : \tau$.

Let $\mathfrak{a}, \mathfrak{b}$ be *disjoint* finite ordered sets of variables, and $T_\mathfrak{a}, T_\mathfrak{b}$ the associated monads on $\mathcal{L}_{cl}$. There is a distributive law between these two monads, and their composition as functors is therefore naturally equipped with the structure of a monad (this is not true for any two monads, in general). For simplicity, we write down the formulas only in the case where $\mathfrak{a} = \{x : \alpha\}$ and $\mathfrak{b} = \{y : \beta\}$ are singletons.

Theorem 5.5. *The monads $T_\mathfrak{a}, T_\mathfrak{b}$ admit a distributive law*

$$\xi : T_\mathfrak{a} T_\mathfrak{b} \longrightarrow T_\mathfrak{b} T_\mathfrak{a}$$
$$\chi_\sigma = \lambda z^{\alpha \Rightarrow (\beta \Rightarrow \sigma)} y^\beta x^\alpha.((zx)y)\,.$$

With the induced monad structure the composite $T_\mathfrak{a} T_\mathfrak{b}$ is isomorphic, as a monad, to $T_{\mathfrak{a}:\mathfrak{b}}$ where $\mathfrak{a} : \mathfrak{b}$ denotes concatenation of sequences.

Proof. The following diagram commutes

$$\begin{tikzcd} & \alpha \Rightarrow \sigma \arrow[dl, "T_{\{x:\alpha\}}\xi_\sigma^{\{y:\beta\}}"'] \arrow[dr, "\xi_{\alpha \Rightarrow \sigma}^{\{y:\beta\}}"] & \\ \alpha \Rightarrow (\beta \Rightarrow \sigma) \arrow[rr, "\chi_\sigma"'] & & \beta \Rightarrow (\alpha \Rightarrow \sigma) \end{tikzcd}$$

since

$$\begin{aligned} \chi_\sigma \circ T_{\{x:\alpha\}}\xi_\sigma^{\{y:\beta\}} &= (\lambda z^{\alpha \to \beta \to \sigma} y^\beta x^\alpha.zxy) \circ (\lambda u^{\alpha \to \sigma} x^\alpha.(\lambda w^\sigma y^\beta.w)(ux)) \\ &= (\lambda z^{\alpha \to \beta \to \sigma} y^\beta x^\alpha.zxy) \circ (\lambda u^{\alpha \to \sigma} x^\alpha y^\beta.ux) \\ &= \lambda v^{\alpha \to \sigma}.(\lambda z^{\alpha \to \beta \to \sigma} y^\beta x^\alpha.zxy)((\lambda u^{\alpha \to \sigma} x^\alpha y^\beta.ux)v) \\ &= \lambda v^{\alpha \to \sigma}.(\lambda z^{\alpha \to \beta \to \sigma} y^\beta x^\alpha.zxy)(\lambda x^\alpha y^\beta.vx) \\ &= \lambda v.\lambda yx.(\lambda xy.ux)xy \\ &= \lambda v.\lambda yx.(vx) \\ &=_\eta \lambda vy.v = \xi_{\alpha \to \sigma}^{\{y:\beta\}} \end{aligned}$$

Next, observe that

$$\begin{tikzcd} & \beta \Rightarrow \sigma \arrow[dl, "\xi_{\beta \Rightarrow \sigma}^{\{x:\alpha\}}"'] \arrow[dr, "T_{\{y:\beta\}}\xi_{\alpha \Rightarrow \sigma}^{\{x:\alpha\}}"] & \\ \alpha \Rightarrow (\beta \Rightarrow \sigma) \arrow[rr, "\chi_\sigma"'] & & \beta \Rightarrow (\alpha \Rightarrow \sigma) \end{tikzcd}$$

commutes since

$$\begin{aligned} T_{\{y:\beta\}}\xi_{\alpha \to \sigma}^{\{x:\alpha\}} &= \lambda u^{\beta \to \sigma} y^\beta.(\lambda w^\sigma x^\alpha.w)(uy) \\ &= \lambda uyx.uy \end{aligned}$$

and,

$$\begin{aligned}
\chi_\sigma \circ \xi^{\{x:\alpha\}}_{\beta\to\sigma} &= (\lambda z^{\alpha\to\beta\to\sigma} y^\beta x^\alpha . zxy) \circ (\lambda w^{\beta\to\sigma} x^\alpha . w) \\
&= \lambda v^{\beta\to\sigma} . (\lambda z^{\alpha\to\beta\to\sigma} y^\beta x^\alpha . zxy)((\lambda w^{\beta\to\sigma} x^\alpha . w)v) \\
&= \lambda v . (\lambda zyx . zxy)(\lambda x . v) \\
&= \lambda v . (\lambda yx . (\lambda x . v)xy) \\
&= \lambda vyx . vy
\end{aligned}$$

Finally

$$\begin{tikzcd}
\alpha \Rightarrow \alpha \Rightarrow \beta \Rightarrow \sigma \arrow[r, "T^{\{x:\alpha\}}\chi_\sigma"] \arrow[d, "\mu^{\{x:\alpha\}}_{\beta\Rightarrow\sigma}"'] & \alpha \Rightarrow \beta \Rightarrow \alpha \Rightarrow \sigma \arrow[r, "\chi_{\alpha\Rightarrow\sigma}"] & \beta \Rightarrow \alpha \Rightarrow \alpha \Rightarrow \sigma \arrow[d, "T^{\{y:\beta\}}\mu_{x:\alpha}"] \\
\alpha \Rightarrow \beta \Rightarrow \sigma \arrow[rr, "\chi_\sigma"'] & & \beta \Rightarrow \alpha \Rightarrow \sigma
\end{tikzcd}$$

commutes since

$$\begin{aligned}
&(T^{\{y:\beta\}}\mu_{x:\alpha}) \circ (\chi_{\alpha\to\sigma}) \circ (T^{\{x:\alpha\}}\chi_\sigma) \\
&= (\lambda u^{\beta\to\alpha\to\alpha\to\sigma} y^\beta . (\lambda v^{\alpha\to\alpha\to\sigma} x^\alpha . vxx)(uy)) \circ (\lambda z^{\alpha\to\beta\to\alpha\to\sigma} yx . zxy) \\
&\quad \circ (\lambda u^{\alpha\to\alpha\to\beta\to\sigma} x^\alpha . (\lambda z^{\alpha\to\beta\to\sigma} . \lambda y^\beta . \lambda \hat{x}^\alpha . z\hat{x}y)(ux))
\end{aligned}$$

$$\begin{aligned}
&= (\lambda uyx . uyxx) \circ (\lambda zyx . zxy) \circ (\lambda uxy\hat{x} . ux\hat{x}y) \\
&= (\lambda uyx . uyxx) \circ (\lambda v^{\alpha\to\alpha\to\beta\to\sigma} . (\lambda zyx . zxy)((\lambda uxy\hat{x} . ux\hat{x}y)v)) \\
&= (\lambda uyx . uyxx) \circ (\lambda v . (\lambda zyx . zxy)(\lambda xy\hat{x} . vx\hat{x}y)) \\
&= (\lambda uyx . uyxx) \circ (\lambda vyx . (\lambda xy\hat{x} . vx\hat{x}y)xy)) \\
&= (\lambda uyx . uyxx) \circ (\lambda vyx\hat{x} . vx\hat{x}y) \\
&= \lambda w . (\lambda uyx . uyxx)((\lambda vyx\hat{x} . vx\hat{x}y)w) \\
&= \lambda w . (\lambda uyx . uyxx)(\lambda yx\hat{x} . wx\hat{x}y) \\
&= \lambda w . (\lambda yx . (\lambda yx\hat{x} . wx\hat{x}y)yxx) \\
&= \lambda w . (\lambda yx . wxxy) \\
&= \lambda wyx . wxxy
\end{aligned}$$

and,

$$\begin{aligned}
\chi_\sigma \circ \mu^{\{x:\alpha\}}_{\beta\to\sigma} &= (\lambda z^{\alpha\to\beta\to\sigma} y^\beta x^\alpha . zxy) \circ (\lambda u^{\alpha\to\alpha\to\beta\to\sigma} . ux) \\
&= \lambda v^{\alpha\to\alpha\to\beta\to\sigma}(\lambda zyx . zxy)((\lambda u . ux)v) \\
&= \lambda v(\lambda zyx . zxy)(vx) \\
&= \lambda vyx . vxxy
\end{aligned}$$

Finally, it remains to show that

$$\begin{tikzcd}
\alpha \Rightarrow \beta \Rightarrow \beta \Rightarrow \sigma \arrow[r, "\chi_{\beta\Rightarrow\sigma}"] \arrow[d, "T_{\{x:\alpha\}}\mu^{\{y:\beta\}}"'] & \beta \Rightarrow \alpha \Rightarrow \beta \Rightarrow \sigma \arrow[r, "T^{\{y:\beta\}}\chi_\sigma"] & \beta \Rightarrow \beta \Rightarrow \alpha \Rightarrow \sigma \arrow[d, "\mu^{\{y:\beta\}}_{\alpha\Rightarrow\sigma}"] \\
\alpha \Rightarrow \beta \Rightarrow \sigma \arrow[rr, "\chi_\sigma"'] & & \beta \Rightarrow \alpha \Rightarrow \sigma
\end{tikzcd}$$

commutes;

$$\begin{aligned}
&(\mu^{\{y:\beta\}}_{\alpha\to\sigma}) \circ (T^{\{y:\beta\}}\chi_\sigma) \circ (\chi_{\beta\to\sigma}) \\
&= (\lambda u^{\beta\to\beta\to\alpha\to\sigma} y^\beta.uyy) \circ (\lambda v^{\beta\to\alpha\to\beta\to\sigma} y^\beta.(\lambda z^{\alpha\to\beta\to\sigma}.\lambda\hat{y}^\beta.\lambda x^\alpha.zx\hat{y})(vy)) \\
&\quad \circ(\lambda z^{\alpha\to\beta\to\beta\to\sigma} yx.zxy)
\end{aligned}$$

$$\begin{aligned}
&= (\lambda uy.uyy) \circ (\lambda vy\hat{y}x.vyx\hat{y}) \circ (\lambda zyx.zxy) \\
&= (\lambda w^{\beta\to\alpha\to\beta\to\sigma}.(\lambda uy.uyy)((\lambda vy\hat{y}x.vyx\hat{y})w)) \circ (\lambda zyx.zxy) \\
&= (\lambda wy.(\lambda uy.uyy)(\lambda y\hat{y}x.wyx\hat{y})) \circ (\lambda zyx.zxy) \\
&= (\lambda wy.(\lambda y\hat{y}x.wyx\hat{y})yy) \circ (\lambda zyx.zxy) \\
&= (\lambda wyx.wyxy) \circ (\lambda zyx.zxy) \\
&= \lambda u^{\alpha\to\beta\to\beta\to\sigma}(\lambda wyx.wyxy)((\lambda zyx.zxy)u) \\
&= \lambda u(\lambda wyx.wyxy)(\lambda yx.uxy) \\
&= \lambda u(\lambda yx.(\lambda yx.uxy)yxy) \\
&= \lambda uyx.uxyy
\end{aligned}$$

and,

$$\begin{aligned}
\chi_\sigma \circ T_{\{x:\alpha\}}\mu^{\{y:\beta\}} &= (\lambda z^{\alpha\to\beta\to\sigma} y^\beta x^\alpha.zxy) \circ (\lambda u^{\alpha\to\beta\to\beta\to\sigma} x^\alpha.(\lambda v^{\beta\to\beta\to\sigma} y^\beta.vyy)(ux)) \\
&= (\lambda zyx.zxy) \circ (\lambda uxy(uxyy)) \\
&= \lambda w^{\alpha\to\beta\to\beta\to\sigma}.(\lambda zyx.zxy)((\lambda uxy.uxyy)w) \\
&= \lambda w.(\lambda zyx.zxy)(\lambda xy.wxyy) \\
&= \lambda w.(\lambda yx.(\lambda xy.wxyy)xy) \\
&= \lambda w.(\lambda yx.wxyy) \\
&= \lambda wyx.wxyy
\end{aligned}$$

$\square$

5.2 The structural rules are monadic

The *structural rules* of the sequent calculus of intuitionistic logic are

$$\text{(Contraction):}\quad \frac{\mathfrak{b}, q:\tau, q':\tau \vdash M(q,q'):\sigma}{\mathfrak{b}, q:\tau \vdash M(q,q):\sigma}\,\text{ctr}$$

$$\text{(Weakening):}\quad \frac{\mathfrak{b} \vdash M:\sigma}{\mathfrak{b}, q:\tau \vdash M:\sigma}\,\text{weak}$$

$$\text{(Exchange):}\quad \frac{\mathfrak{b}, q_1:\tau_1, q_2:\tau_2 \vdash M:\sigma}{\mathfrak{b}, q_2:\tau_2, q_1:\tau_1 \vdash M:\sigma}\,\text{ex}$$

where $\mathfrak{b}$ denotes a typing context, which is an ordered list of typed variables. From the point of view of lambda calculus these rules correspond respectively to the identification of two free variables (contraction) the introduction of a spurious dependence on a free variable (weakening) and the exchange of the order of two free variables in the context which is viewed as an ordered list (exchange). These structural rules can be recognised in the categorical presentation of lambda calculus given in this paper, using the structure presented on the category $\mathcal{L}_{cl}$ by the monads $T_{\mathfrak{a}}$ discussed above.

Let us first explain the interpretation $[\![-]\!]$ of typing judgements for open lambda terms in $\mathcal{L}$. If there is a typing judgement for $M:\sigma$ of the form

$$\mathfrak{a} \vdash M:\sigma$$

then the denotation is just the term M, but with the context $\mathfrak{a}$ recorded either by the monad $T_{\mathfrak{a}}$ or by working in the category $\mathcal{L}_A$ where A is the underlying set of $\mathfrak{a}$:

$$[\![\mathfrak{a} \vdash M:\sigma]\!] = M \in \mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{a}}\sigma) = \mathcal{L}_A(\mathbf{1},\sigma)\,.$$

Now, observe that

$$[\![\mathfrak{b}, q:\tau, q':\tau \vdash M:\sigma]\!] = M \in \mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b},q,q'}\sigma)\,.$$

But there is an obvious isomorphism of monads $T_{\{q:\tau\}} \cong T_{\{q':\tau\}}$ on $\mathcal{L}_{cl}$, so that we have, using the multiplication on the monad $T_{\{q:\tau\}}$, the map

$$\begin{aligned}\mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b},q,q'}\sigma) &\cong \mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b}}T_{\{q:\tau\}}T_{\{q':\tau\}}\sigma)\\ &\cong \mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b}}T_{\{q:\tau\}}T_{\{q:\tau\}}\sigma)\\ &\longrightarrow \mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b}}T_{\{q:\tau\}}\sigma)\\ &\cong \mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b},q:\tau}\sigma)\,.\end{aligned}$$

The image under this map of $[\![\mathfrak{b}, q:\tau, q':\tau \vdash M:\sigma]\!]$ is precisely the result of applying the contraction rule to the typing judgement.

Similarly, given

$$[\![\mathfrak{b} \vdash M:\sigma]\!] = M \in \mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b}}\sigma)$$

and a variable $q : \tau$ the unit map on the monad $T_{\{q:\tau\}}$ gives a function

$$\mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b}}\sigma) \longrightarrow \mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b}}T_{\{q:\tau\}}\sigma) \cong \mathcal{L}_{cl}(\mathbf{1}, T_{\mathfrak{b},q:\tau}\sigma)$$

the image of $[\![\mathfrak{b} \vdash M : \sigma]\!]$ under which is the result of applying the weakening rule. The exchange rule has a similarly obvious interpretation, using the distributive law.

References

[1] K. Došen, *Deductive completeness*, Bulletin of symbolic logic 2.3 (1996): 243–283.

[2] K. Došen, *Abstraction and application in adjunction*, arXiv preprint math/0111061 (2001).

[3] J. Lambek, *Functional completeness of cartesian categories*, Annals of Mathematical Logic **6.3-4** (1974): 259–292.

[4] S. Mac Lane, *The Lambda Calculus and Adjoint Functors*, Logic, Meaning and Computation, Springer Netherlands, (2001) 181–184.

[5] J. Lambek and P.J. Scott, *Introduction to higher-order categorical logic*, (Vol. 7). Cambridge University Press, (1988).

[6] P. Taylor, *Practical foundations of mathematics*, No. 71. Cambridge University Press, 1999.

[7] M. Sørensen and P. Urzyczyn, *Lectures on the Curry-Howard isomorphism*, Vol. 149. Elsevier, 2006.

[8] C. Hermida and B. Jacobs, *Fibrations with indeterminates: Contextual and functional completeness for polymorphic lambda calculi*, Math. Structures Comput. Sci. 5 (1995), 501–531.

[9] F.W. Lawvere, *Adjointness in foundations*, Dialectica 23 (1969), 281-296.