On the Lambek embedding and the category of product-preserving presheaves

Peng Fu
Dalhousie University

Kohei Kishida
University of Illinois at Urbana-Champaign

Neil J. Ross
Dalhousie University

Peter Selinger
Dalhousie University

It is well-known that the category of presheaf functors is complete and cocomplete, and that the Yoneda embedding into the presheaf category preserves products. However, the Yoneda embedding does not preserve coproducts. It is perhaps less well-known that if we restrict the codomain of the Yoneda embedding to the full subcategory of limit-preserving functors, then this embedding preserves colimits, while still enjoying most of the other useful properties of the Yoneda embedding. We call this modified embedding the *Lambek embedding*. The category of limit-preserving functors is known to be a reflective subcategory of the category of all functors, i.e., there is a left adjoint for the inclusion functor. In the literature, the existence of this left adjoint is often proved non-constructively, e.g., by an application of Freyd's adjoint functor theorem. In this paper, we provide an alternative, more constructive proof of this fact. We first explain the Lambek embedding and why it preserves coproducts. Then we review some concepts from multi-sorted algebras and observe that there is a one-to-one correspondence between product-preserving presheaves and certain multi-sorted term algebras. We provide a construction that freely turns any presheaf functor into a product-preserving one, hence giving an explicit definition of the left adjoint functor of the inclusion. Finally, we sketch how to extend our method to prove that the subcategory of limit-preserving functors is also reflective.

1 Introduction

Let $\mathbf{A}$ be a small category. Recall that a *presheaf* over $\mathbf{A}$ is just a functor $F : \mathbf{A}^{\mathrm{op}} \to \mathbf{Set}$. The category $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$ of presheaves has many desirable properties. It is complete and cocomplete, cartesian-closed, and even a topos. If $\mathbf{A}$ is monoidal, then $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$ is monoidal closed, where tensor products and exponentials are given by Day's convolution [1]. Also, the Yoneda embedding $y : \mathbf{A} \to \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$ is full and faithful. These properties make the presheaf category a natural candidate for modeling linear functional programming languages [5, 8, 9].

Although the Yoneda embedding preserves all existing products (and more generally, limits), it does not preserve coproducts. In some situations, it is useful to have a version of the Yoneda embedding that also preserves coproducts. For example, in our work on the categorical semantics of quantum programming languages [5], we start with a base category that models quantum operations, which is monoidal but not necessarily monoidal closed. In order to account for lambda abstraction (i.e., *currying*), we can embed the base category into its presheaf category. Sometimes the base category already has coproducts, and it is natural, and often technically necessary, to require the embedding to also preserve these coproducts. Fortunately, in this situation, there is a variant of the Yoneda embedding, which we call the *Lambek embedding* [7], that achieves exactly that.

Before we explain the Lambek embedding, let us first recall why the Yoneda embedding does not preserve coproducts. Let us consider a small category $\mathbf{A}$ with a distinguished object I and a coproduct

$I+I$. Let $y : \mathbf{A} \to \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$ be the Yoneda embedding, given by $y(A) = \mathrm{Hom}(-,A)$. We must show that

$$y(I) \xrightarrow{y(\mathrm{inj}_1)} y(I+I) \xleftarrow{y(\mathrm{inj}_2)} y(I)$$

is not a coproduct cone in $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$. Suppose it is a coproduct cone. Since we know that

$$y(I) \xrightarrow{\mathrm{inj}_1} y(I)+y(I) \xleftarrow{\mathrm{inj}_2} y(I)$$

is also a coproduct cone, there exists a unique isomorphism $f : y(I+I) \to y(I)+y(I)$ such that the following composition is identity:

$$y(I)+y(I) \xrightarrow{[y(\mathrm{inj}_1),y(\mathrm{inj}_2)]} y(I+I) \xrightarrow{f} y(I)+y(I).$$

The Yoneda lemma states that $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}(y(A),F) \cong F(A)$ for all $F : \mathbf{A}^{\mathrm{op}} \to \mathbf{Set}$ and $A \in \mathbf{A}$. In particular, we have

$$\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}(y(I+I),y(I)+y(I)) \cong \mathrm{Hom}(I+I,I)+\mathrm{Hom}(I+I,I).$$

Since an element in the disjoint union $\mathrm{Hom}(I+I,I)+\mathrm{Hom}(I+I,I)$ either belongs to the left component or the right component, it easily follows that $f \in \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}(y(I+I),y(I)+y(I))$ must be a natural transformation that either maps its entire domain to the left component of its codomain, or to the right component. This implies that $f \circ [y(\mathrm{inj}_1),y(\mathrm{inj}_2)] \neq \mathrm{id}$.

We now define the Lambek embedding. Let $[\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times$ be the full subcategory of $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$ consisting of product-preserving functors. Note that a product in $\mathbf{A}^{\mathrm{op}}$ is a coproduct in $\mathbf{A}$, so that a product-preserving functor $F : \mathbf{A}^{\mathrm{op}} \to \mathbf{Set}$ is the same thing as a contravariant functor that maps coproducts of $\mathbf{A}$ to products of $\mathbf{Set}$. Every functor of the form $y(A) = \mathrm{Hom}(-,A)$ is product-preserving in this sense, because we have $\mathrm{Hom}(B+C,A) \cong \mathrm{Hom}(B,A) \times \mathrm{Hom}(C,A)$ for all $B,C \in \mathbf{A}$. Therefore, the image of the Yoneda embedding $y : \mathbf{A} \to \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$ is entirely contained in the subcategory $[\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times$. The *Lambek embedding* $\overline{y} : \mathbf{A} \to [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times$ is defined to be the restriction of y to this codomain, i.e., the unique functor making the following diagram commute.

$$\begin{array}{ccc} \mathbf{A} & & \\ \downarrow \overline{y} & \searrow^{y} & \\ [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times & \xhookrightarrow{i} & \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}} \end{array}$$

We can show that the Lambek embedding preserves all coproducts that exist in $\mathbf{A}$. For example, if $A+B$ is a coproduct in $\mathbf{A}$, by the Yoneda lemma, for any $G \in [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times$, we have

$$\begin{array}{rcl} [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times(\overline{y}(A+B),G) & = & [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times(\mathrm{Hom}(-,A+B),G) \\ & \cong & \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}(\mathrm{Hom}(-,A+B),G) \\ & \cong & G(A+B) \\ & \cong^* & G(A) \times G(B) \\ & \cong & \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}(\mathrm{Hom}(-,A),G) \times \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}(\mathrm{Hom}(-,B),G) \\ & \cong & [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times(\mathrm{Hom}(-,A),G) \times [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times(\mathrm{Hom}(-,B),G) \\ & \cong & [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times(\overline{y}(A)+\overline{y}(B),G), \end{array}$$

and therefore $\overline{y}(A+B) \cong \overline{y}(A)+\overline{y}(B)$. Note that the step marked "*" uses the assumption that G is product-preserving.

Like the category of presheaves, the full subcategory $[\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times$ has many desirable properties. It is complete and cocomplete, and it is monoidal closed if $\mathbf{A}$ is monoidal [2]. The proofs of these properties rely on the fact that $[\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times$ is a reflective subcategory of $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$, i.e., there is an adjunction $L \dashv i : [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times \to \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$. The existence of the left adjoint functor L is not at all obvious. For example, Kennison's original proof [6] uses Freyd's adjoint functor theorem [3], which requires a solution set condition and the axiom of choice.

In this paper, we give an explicit construction of the left adjoint functor $L : \mathbf{Set}^{\mathbf{A}^{\mathrm{op}}} \to [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times$. The methods of this paper are probably already familiar to researchers who specialize in categorical algebra. However, we believe that outside this immediate field, the connection between presheaves and multi-sorted type theories is perhaps not as well-known as it should be, and that an explicit description, such as the one we give here, will be beneficial.

The paper is organized as follows. In Section 2, we briefly review some concepts from multi-sorted algebras. In Section 3, we show that there is a one-to-one correspondence between presheaves and certain multi-sorted algebras. In Section 4, we consider multi-sorted algebras corresponding to product-preserving functors. In Section 5, we define a functor L by constructing a multi-sorted term algebra, and we show that L is the left adjoint of the inclusion functor i. In Section 6, we sketch how to extend our method to limit-preserving functors.

2 Background on multi-sorted algebras

The definition of a multi-sorted algebra starts by assuming we are given a collection of *sorts*, usually denoted A,B,C, etc. An *arity* is an $(n+1)$-tuple of sorts. A *signature* is a set of *function symbols*, together with an assignment of an arity to each function symbol. We usually write $f : A_1,\ldots,A_n \to B$ to indicate that the function symbol f has arity $\langle A_1,\ldots,A_n,B\rangle$. When the sorts $A_1,\ldots,A_n,B$ are not important, we also sometimes say that f is an n-ary function. In case $n = 0$, if c is a function symbol of arity $\langle B\rangle$, we also write $c : B$ and call c a *constant symbol* of sort B.

Definition 2.1. Let Σ be a signature. A Σ-algebra $\mathscr{T}$ consists of the following data:

- For each sort A, a set $\mathscr{T}(A)$. The sets $\mathscr{T}(A)$ are called the *carriers* of the algebra.
- For each function symbol $f : A_1,\ldots,A_n \to B$ in Σ, a function $\mathscr{T}(f) : \mathscr{T}(A_1)\times\ldots\times\mathscr{T}(A_n) \to \mathscr{T}(B)$.

Definition 2.2. Let $\mathscr{A},\mathscr{B}$ be Σ-algebras. A Σ-homomorphism $\varphi : \mathscr{A} \to \mathscr{B}$ consists of:

- A function $\varphi_A : \mathscr{A}(A) \to \mathscr{B}(A)$ for each sort A, such that
- for each function symbol $f : A_1,\ldots,A_n \to B \in \Sigma$ and all $a_i \in \mathscr{A}(A_i)$, we have

$$\varphi_B(\mathscr{A}(f)(a_1,\ldots,a_n)) = \mathscr{B}(f)(\varphi_{A_1}(a_1),\ldots,\varphi_{A_n}(a_n)).$$

One example of a Σ-algebra is a term algebra. We first define well-sorted terms. For each sort A, we assume that we are given a countable set X_A of *variables*. We further assume that these sets of variables are pairwise disjoint, i.e., $X_A \cap X_B = \emptyset$ when $A \neq B$. The set of Σ-terms is freely generated from variables and function symbols, in the obvious sort-respecting way.

Definition 2.3. The set of Σ-*terms of sort* A is defined inductively as follows. We write $\Sigma \vdash t : A$ to mean that t is a Σ-term of sort A.

$$\frac{x \in X_A}{\Sigma \vdash x : A} \qquad \frac{f : A_1,\ldots,A_n \to B \in \Sigma \qquad \Sigma \vdash t_1 : A_1 \quad \ldots \quad \Sigma \vdash t_n : A_n}{\Sigma \vdash f(t_1,\ldots,t_n) : B.}$$

We say a term is *closed* if it contains no variables. An *equation* over a signature Σ is a triple $\langle s,t,A\rangle$, where s,t are Σ-terms of sort A. We usually write an equation as $s \approx t : A$. If E is a set of equations, we write $E \vdash s \approx t : A$ to mean that the equation $s \approx t : A$ follows from the equations in E.

Definition 2.4. The relation $E \vdash s \approx t : A$ is inductively defined by the following rules.

$$\frac{(s \approx t : A) \in E}{E \vdash s \approx t : A}\ \textit{(ax)} \qquad \frac{x \in X_A \quad \Sigma \vdash r : A \quad E \vdash s \approx t : B}{E \vdash s[r/x] \approx t[r/x] : B}\ \textit{(subst)}$$

$$\frac{\Sigma \vdash s : A}{E \vdash s \approx s : A}\ \textit{(refl)} \qquad \frac{E \vdash t \approx s : B}{E \vdash s \approx t : B}\ \textit{(symm)} \qquad \frac{E \vdash r \approx s : B \quad E \vdash s \approx t : B}{E \vdash r \approx t : B}\ \textit{(trans)}$$

$$\frac{E \vdash s_i \approx t_i : A_i \text{ for all } i \qquad f : A_1,\ldots,A_n \to B}{E \vdash f(s_1,\ldots,s_n) \approx f(t_1,\ldots,t_n) : B}\ \textit{(cong)}$$

In the rule *(subst)*, $s[r/x]$ denotes the term obtained from s by replacing all occurrences of the variable x by the term r. This rule ensures that variables are generic, i.e., if an equation holds for a variable, then it holds for any term.

Definition 2.5. Given a signature Σ, the *open term algebra* $\mathscr{T} = \mathrm{Term}(\Sigma)$ is defined as follows: $\mathscr{T}(A)$ is the set of Σ-terms of sort A, and for each function symbol $f : A_1,\ldots,A_n \to B$ in Σ, the function $\mathscr{T}(f) : \mathscr{T}(A_1) \times \ldots \times \mathscr{T}(A_n) \to \mathscr{T}(B)$ is defined by $\mathscr{T}(f)(t_1,\ldots,t_n) = f(t_1,\ldots,t_n)$. The *closed term algebra* $\mathrm{Term}_0(\Sigma)$ is defined similarly, except that the carriers consist only of closed terms.

If we are also given a set of equations E, we can define the open and closed *quotient term algebras* $\mathrm{Term}(\Sigma)/E$ and $\mathrm{Term}_0(\Sigma)/E$, which are defined in the same way except that the carriers consist of $\approx$-equivalence classes of (open or closed) terms. The *(cong)* rule ensures that the function $\mathscr{T}(f)$ is well-defined on such equivalence classes.

Definition 2.6. We say that a Σ-algebra $\mathscr{T}$ *satisfies* an equation $s \approx t : A$, or equivalently, that the equation is *valid* in $\mathscr{T}$, if for any Σ-homomorphism $\varphi : \mathrm{Term}(\Sigma) \to \mathscr{T}$, we have $\varphi_A(s) = \varphi_A(t)$. If E is a set of equations, we say that $\mathscr{T}$ satisfies E if it satisfies all the equations in E. If $\mathscr{T}$ is a Σ-algebra satisfying a set of equations E, we also say that $\mathscr{T}$ is a (Σ,E)-*algebra*.

By a multi-sorted *theory*, we mean all of the above data, i.e., a collection of sorts, a signature, and a set of equations. Note that $\mathrm{Term}(\Sigma)$ and $\mathrm{Term}_0(\Sigma)$ are Σ-algebras and $\mathrm{Term}(\Sigma)/E$ and $\mathrm{Term}_0(\Sigma)/E$ are (Σ,E)-algebras.

3 The theory associated to a category A

In the rest of this paper, we will work with a small category $\mathbf{A}$. For convenience and without loss of generality, we will work with the functor category $\mathbf{Set}^{\mathbf{A}}$ rather than $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$ unless otherwise noted.

Definition 3.1. To the category $\mathbf{A}$, we associate a multi-sorted theory as follows. The sorts are the objects of $\mathbf{A}$. The signature $\Sigma_{\mathbf{A}}$ contains a unary function symbol $f : A \to B$ for every morphism $f : A \to B$ in $\mathbf{A}$. The set of equations $E_{\mathbf{A}}$ consists of the following:

- *(id)* $\mathrm{id}(x) \approx x : A$, whenever $\mathrm{id} : A \to A$ is an identity morphism and x is a variable of sort A.
- *(comp)* $(f \circ g)(x) \approx f(g(x)) : C$, whenever $f : A \to B$ and $g : B \to C$ are morphisms and x is a variable of sort A.

Remark 3.2. There is a one-to-one correspondence between $(\Sigma_{\mathbf{A}}, E_{\mathbf{A}})$-algebras and functors $T : \mathbf{A} \to \mathbf{Set}$. Indeed, if $T : \mathbf{A} \to \mathbf{Set}$ is functor, then T is a $\Sigma_{\mathbf{A}}$-algebra because for each $A \in \mathbf{A}$, there is a set $T(A)$. For each function symbol $f : A \to B$ in $\mathbf{A}$, since f is also a morphism of $\mathbf{A}$, there is a function $T(f) : T(A) \to T(B)$. The equations *(id)* and *(comp)* are satisfied by functoriality. Conversely, every $(\Sigma_{\mathbf{A}}, E_{\mathbf{A}})$-algebra gives rise to a functor, and the two assignments are mutually inverse.

4 Product-preserving functors

Recall from the introduction that we are interested in taking a functor $F : \mathbf{A} \to \mathbf{Set}$, and constructing another functor $L(F) : \mathbf{A} \to \mathbf{Set}$ that is product-preserving. Rather than requiring $L(F)$ to necessarily preserve *all* products, we will consider the slightly more general problem of preserving some *distinguished* class of products. Therefore, we will assume that we are given a small category $\mathbf{A}$ and some particular collection $\mathscr{C}$ of product cones in $\mathbf{A}$. We say that a functor is $\mathscr{C}$-product-preserving when it preserves all of the product cones in $\mathscr{C}$.

On one extreme, we may of course assume that $\mathbf{A}$ has all products, and that $\mathscr{C}$ is the set of all product cones in $\mathbf{A}$. In that case, a $\mathscr{C}$-product-preserving functor is just the same thing as a product-preserving functor in the ordinary sense. On the other extreme, $\mathscr{C}$ could just consist of a single product cone $A \leftarrow A \times B \to B$, in which case a functor is $\mathscr{C}$-product-preserving if it preserves just that one product.

For simplicity and ease of exposition, we will assume in the following that $\mathscr{C}$ is a collection of binary product cones. However, the same construction works for n-ary products for any finite n. It also works for infinite products, assuming that we extend the notion of term algebras to allow terms of infinite arity (which causes no problems). Product cones for $n = 0$, i.e., terminal objects, are of course also a special case.

We first define a theory of product-preserving functors.

Definition 4.1. To a small category $\mathbf{A}$ with a distinguished collection $\mathscr{C}$ of (binary) product cones, we associate a multi-sorted theory as follows. The signature $\Sigma_{\mathscr{C}}$ has the same function symbols as $\Sigma_{\mathbf{A}}$, and additionally, for each product cone $A \xleftarrow{\mathsf{fst}} C \xrightarrow{\mathsf{snd}} B$ in $\mathscr{C}$, we add a function symbol

$$\mathsf{pair} : A, B \to C.$$

Note that there are already function symbols for all the morphisms of $\mathbf{A}$, including $\mathsf{fst} : C \to A$ and $\mathsf{snd} : C \to B$. The set of equations $E_{\mathscr{C}}$ has the same equations as $E_{\mathbf{A}}$, and additionally, for each product cone $A \xleftarrow{\mathsf{fst}} C \xrightarrow{\mathsf{snd}} B$ in $\mathscr{C}$, we add the following three equations:

(fst) $\mathsf{fst}(\mathsf{pair}(x,y)) \approx x : A$, whenever x and y are variables of sorts A and B, respectively.

(snd) $\mathsf{snd}(\mathsf{pair}(x,y)) \approx y : A$, whenever x and y are variables of sorts A and B, respectively.

(pair) $\mathsf{pair}(\mathsf{fst}(z),\mathsf{snd}(z)) \approx z : C$, whenever z is a variable of sort C.

We then have:

Remark 4.2. There is a one-to-one correspondence between $(\Sigma_{\mathscr{C}}, E_{\mathscr{C}})$-algebras and $\mathscr{C}$-product-preserving functors $F : \mathbf{A} \to \mathbf{Set}$. The proof is basically the same as that of Remark 3.2. The point is that the equations *(fst)*, *(snd)*, and *(pair)* are exactly what is required to ensure that the cone $F(A) \xleftarrow{F(\mathsf{fst})} F(C) \xrightarrow{F(\mathsf{snd})} F(B)$ is a product cone in $\mathbf{Set}$.

5 The construction of the functor L

Given a small category $\mathbf{A}$ with a distinguished collection $\mathscr{C}$ of product cones as in the previous section, we write $[\mathbf{Set}^{\mathbf{A}}]_{\mathscr{C}}$ for the full subcategory of $\mathbf{Set}^{\mathbf{A}}$ consisting of $\mathscr{C}$-product-preserving functors. The notation $[\mathbf{Set}^{\mathbf{A}}]_{\times}$ used in the introduction is a special case, where $\mathscr{C}$ is the collection of all product cones.

In this section, we will focus on defining the functor $L : \mathbf{Set}^{\mathbf{A}} \to [\mathbf{Set}^{\mathbf{A}}]_{\mathscr{C}}$. By Remark 4.2, we know that a product-preserving presheaf corresponds to a $(\Sigma_{\mathscr{C}}, E_{\mathscr{C}})$-algebra. Thus, given any functor $F : \mathbf{A} \to \mathbf{Set}$, we must construct a $(\Sigma_{\mathscr{C}}, E_{\mathscr{C}})$-algebra $L(F)$. This can be done freely, as we now show.

Definition 5.1. To any functor $F : \mathbf{A} \to \mathbf{Set}$, we associate a multi-sorted theory as follows. The sorts, signature, and equations are the same as in Definitions 3.1 and 4.1, except for the following:

- For each object $A \in \mathbf{A}$ and each element $c \in F(A)$, we add a constant symbol $c : A$ to the signature.
- For each morphism $f : A \to B \in \mathbf{A}$ and each $c \in F(A)$, let $d = F(f)(c)$. We add an equation $d \approx f(c) : B$.

We write the resulting signature and equation as Σ_F and E_F, respectively.

The quotient term algebra $\mathrm{Term}_0(\Sigma_F)/E_F$ still satisfies the equations *(id)*, *(comp)*, *(fst)*, *(snd)*, and *(pair)*, and therefore, by Remark 3.2, it is a product-preserving functor. Hence we define the functor $L : \mathbf{Set}^{\mathbf{A}} \to [\mathbf{Set}^{\mathbf{A}}]_{\mathscr{C}}$ to be the following.

Definition 5.2. We define the functor $L : \mathbf{Set}^{\mathbf{A}} \to [\mathbf{Set}^{\mathbf{A}}]_{\mathscr{C}}$ by

$$L(F) = \mathrm{Term}_0(\Sigma_F)/E_F.$$

We note that L is a well-defined functor, because for every natural transformation $\alpha : F \to G \in \mathbf{Set}^{\mathbf{A}}$, we have a $\Sigma_{\mathscr{C}}$-homomorphism $L(\alpha) : \mathrm{Term}_0(\Sigma_F)/E_F \to \mathrm{Term}_0(\Sigma_G)/E_G$. This homomorphism is defined inductively on the structure of the terms in $\mathrm{Term}_0(\Sigma_F)/E_F$, using α for the constant symbols. In other words, we have $L(\alpha)(f(t_1,\ldots,t_n)) = f(L(\alpha)(t_1),\ldots,L(\alpha)(t_n))$, where f is any function symbol from $\Sigma_{\mathscr{C}}$, and $L(\alpha)(c) = \alpha(c)$, where c is one of the constant symbols introduced in Definition 5.1.

Theorem 5.3. *We have an adjunction $L \dashv i$, where $i : \mathbf{Set}^{\mathbf{A}} \to [\mathbf{Set}^{\mathbf{A}}]_{\mathscr{C}}$ is the inclusion functor.*

Proof. First we will define, for any $F : \mathbf{A} \to \mathbf{Set}$, a natural transformation $\eta_F : F \to iL(F)$, i.e., $\eta_F : F \to \mathrm{Term}_0(\Sigma_F)/E_F$. For any $A \in \mathbf{A}$ and $a \in F(A)$, we define $\eta_{F,A}(a) = a$. Note that η_F is natural in A because for any $f : A \to B \in \mathbf{A}$ and $a \in F(A)$, we have

$$(\mathrm{Term}_0(\Sigma_F)/E_F)(f)(\eta_{F,A}(a)) = f(a) \approx b = \eta_{F,B}(F(f)(a)),$$

where $b = F(f)(a)$. Moreover, η is natural in F because for any $\alpha_A : F(A) \to G(A)$ and $a \in F(A)$, we have

$$(L\alpha)_A(\eta_{F,A}(a)) = (L\alpha)_A(a) = \alpha_A(a) = \eta_{G,A}(\alpha_A(a)).$$

Next, we will show that for any natural transformation $\gamma : F \to i(G)$, there exists a unique natural transformation $\hat{\gamma} : L(F) \to G$ such that $\hat{\gamma} \circ \eta_F = \gamma$. For all $A \in \mathbf{A}$, we define $\hat{\gamma}_A : (\mathrm{Term}_0(\Sigma_F)/E_F)(A) \to G(A)$ by induction on the structure of the terms in $(\mathrm{Term}_0(\Sigma_F)/E_F)(A)$.

- $\hat{\gamma}_A(a) = \gamma_A(a)$ for any $a \in F(A)$.
- $\hat{\gamma}_B(f(t)) = G(f)(\hat{\gamma}_A(t))$ for any $f : A \to B \in \mathbf{A}, \Sigma_F \vdash t : A$.
- $\hat{\gamma}_C(\mathsf{pair}(s,t)) = G(\mathsf{pair})(\hat{\gamma}_A(s), \hat{\gamma}_B(t))$ for any cone $A \leftarrow C \to B$ in $\mathscr{C}$, $\Sigma_F \vdash s : A$, and $\Sigma_F \vdash t : B$.

We must show that $\hat{\gamma}_A : (\mathrm{Term}_0(\Sigma_F)/E_F)(A) \to G(A)$ is a well-defined function for all $A \in \mathbf{A}$. We can show this by induction on $E_F \vdash s \approx t : B$.

- Case

$$\frac{\Sigma_F \vdash a : A \quad b = F(f)(a) \quad f : A \to B \in \mathbf{A}}{E_F \vdash b \approx f(a) : B.}$$

  In this case, we have

$$\hat{\gamma}_B(b) = \gamma_B(b) = \gamma_B(F(f)(a)) \overset{(*)}{=} G(f)(\gamma_A(a)) = \hat{\gamma}_B(f(a)).$$

  The equality $(*)$ is by naturality of γ.

- Case
$$\frac{\Sigma_F \vdash s : A \quad \Sigma_F \vdash t : B}{E_F \vdash \mathsf{fst}(\mathsf{pair}(s,t)) \approx s : A.}$$
We need to show $\hat{\gamma}_B(\mathsf{fst}(\mathsf{pair}(s,t))) = G(\mathsf{fst})(G(\mathsf{pair})(\hat{\gamma}_A(s), \hat{\gamma}_B(t))) = \hat{\gamma}_A(s)$. This equality holds because G is a product-preserving functor.
- Case
$$\frac{\Sigma_F \vdash s : C}{E_F \vdash \mathsf{pair}(\mathsf{fst}(s), \mathsf{snd}(s)) \approx s : C.}$$
We need to show $\hat{\gamma}_C(\mathsf{pair}(\mathsf{fst}(s), \mathsf{snd}(s))) = G(\mathsf{pair})(G(\mathsf{fst})(\hat{\gamma}_C(s)), G(\mathsf{snd})(\hat{\gamma}_C(s))) = \hat{\gamma}_C(s)$. This equality holds because G is a product-preserving functor.
- Case
$$\frac{E_F \vdash s_1 \approx t_1 : A_1 \quad \ldots \quad E_F \vdash s_n \approx t_n : A_n}{E_F \vdash f(s_1, \ldots, s_n) \approx f(t_1, \ldots, t_n) : B.}$$
We have
$$\hat{\gamma}_B(f(s_1, \ldots, s_n)) \overset{(*)}{=} G(f)(\hat{\gamma}_{A_1}(s_1), \ldots, \hat{\gamma}_{A_n}(s_n)) = G(f)(\hat{\gamma}_{A_1}(t_1), \ldots, \hat{\gamma}_{A_n}(t_n)) = \hat{\gamma}_B(f(t_1, \ldots, t_n)),$$
where the equality $(*)$ uses the induction hypothesis.
- All the other cases are proved similarly.

Now we show that $\hat{\gamma}$ is a natural transformation. Suppose $f : A \to B \in \mathbf{A}$. We need to show
$$G(f)(\hat{\gamma}_A(t)) = \hat{\gamma}_B((\mathrm{Term}_0(\Sigma_F)/E_F)(f)(t)).$$
This is true because by the definition of $\hat{\gamma}$, we have
$$\hat{\gamma}_B((\mathrm{Term}_0(\Sigma_F)/E_F)(f)(t)) = \hat{\gamma}_B(f(t)) = G(f)(\hat{\gamma}_A(t))$$
for any $\Sigma_F \vdash t : A$.

It is obvious to verify that $\hat{\gamma} \circ \eta = \gamma$.

Lastly, we must show that $\hat{\gamma}$ is unique. Consider a natural transformation $\hat{\gamma}'$ such that $\hat{\gamma}' \circ \eta = \gamma$. We must show $\hat{\gamma}'_A(t) = \hat{\gamma}_A(t)$ for all $\Sigma_F \vdash t : A$. This can be shown by induction on $\Sigma_F \vdash t : A$.

- Case
$$\frac{a \in F(A)}{\Sigma_F \vdash a : A.}$$
In this case, we have $\hat{\gamma}'_A(a) = \hat{\gamma}'_A(\eta_{F,A}(a)) = \gamma_A(a) = \hat{\gamma}_A(a)$.
- Case
$$\frac{\Sigma_F \vdash s : A \quad \Sigma_F \vdash t : B}{\Sigma_F \vdash \mathsf{pair}(s,t) : C.}$$
By induction hypothesis, we have $\hat{\gamma}'_A(s) = \hat{\gamma}_A(s)$ and $\hat{\gamma}'_B(t) = \hat{\gamma}_B(t)$. We need to show
$$\hat{\gamma}'_{A\times B}(\mathsf{pair}(s,t)) = \hat{\gamma}_{A\times B}(\mathsf{pair}(s,t)) = G(\mathsf{pair})(\hat{\gamma}_A(s), \hat{\gamma}_B(t)) = G(\mathsf{pair})(\hat{\gamma}'_A(s), \hat{\gamma}'_B(t)).$$
This is the case because by the naturality of $\hat{\gamma}'$, we have
$$G(\mathsf{fst})(\hat{\gamma}'_C(\mathsf{pair}(s,t))) = \hat{\gamma}'_A(\mathsf{fst}(\mathsf{pair}(s,t)))) = \hat{\gamma}'_A(s)$$
and
$$G(\mathsf{snd})(\hat{\gamma}'_C(\mathsf{pair}(s,t))) = \hat{\gamma}'_B(t).$$
This shows that $\hat{\gamma}'_C(\mathsf{pair}(s,t))$ is indeed the pair $G(\mathsf{pair})(\hat{\gamma}'_A(s), \hat{\gamma}'_B(t))$.

- Case

$$\frac{\Sigma_F \vdash t : A \quad f : A \to B}{\Sigma_F \vdash f(t) : B.}$$

By naturality of $\hat{\gamma}', \hat{\gamma}$ and the induction hypothesis, we have

$$\hat{\gamma}'_B(f(t)) = G(f)(\hat{\gamma}'_A(t)) = G(f)(\hat{\gamma}_A(t)) = \hat{\gamma}_B(f(t)). \qquad \square$$

Remark 5.4. In the introduction, we showed that the Lambek embedding $\overline{y} : \mathbf{A} \to [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times$ preserves all coproducts, where $[\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_\times$ is the subcategory of $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$ consisting of product-preserving functors. This result can easily be relativized, without changing the proof, to the case of a chosen collection of cones. Namely, if $\mathscr{C}$ is a collection of product cones in $\mathbf{A}^{\mathrm{op}}$ (or, equivalently, coproduct cones in $\mathbf{A}$), then the Lambek embedding $\overline{y} : \mathbf{A} \to [\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}]_{\mathscr{C}}$ preserves all of the coproduct cones in $\mathscr{C}$. Thus, the construction can be customized to preserve coproducts "of interest".

6 The subcategory of limit-preserving presheaves

Our method of showing that the subcategory of $\mathscr{C}$-product-preserving functors is reflective can also be adapted to functors that preserve a given class of limits (not necessarily products). A small complication is that, in Freyd's terminology, the notion of general limits is not *algebraic* but only *essentially algebraic* [4]. This means that the domain of some operations is defined in terms of equations. In the following, we sketch the proof in the case of equalizers, but the same method works for other kinds of limits as well.

Definition 6.1. Consider a small category $\mathbf{A}$ with a distinguished collection $\mathscr{C}$ of limits (we focus on equalizers for simplicity). To this, we associate a multi-sorted theory as follows. The signature $\Sigma_{\mathscr{C}}$ has the same function symbols as $\Sigma_{\mathbf{A}}$, and additionally, for each equalizer $E \xrightarrow{e} A \overset{f,g}{\rightrightarrows} B$ in $\mathscr{C}$, we add a function symbol

$$\mathsf{eql} : A \to E.$$

Its term formation rule is slightly different than that of other function symbols (and for this reason, the resulting theory is not strictly speaking a multi-sorted algebraic theory in the sense of Section 2, but rather what should be called a multi-sorted *essentially algebraic* theory):

$$\frac{\Sigma_{\mathscr{C}} \vdash t : A \quad f, g : A \to B \quad E_{\mathscr{C}} \vdash f(t) \approx g(t) : B}{\Sigma_{\mathscr{C}} \vdash \mathsf{eql}(t) : E}$$

The set of equations $E_{\mathscr{C}}$ has the same equations as $E_{\mathbf{A}}$, and additionally, for each equalizer $E \xrightarrow{e} A \overset{f,g}{\rightrightarrows} B$ in $\mathscr{C}$, we add the following two equations:

(beta) $e(\mathsf{eql}(t)) \approx t : A$, whenever $\Sigma_{\mathscr{C}} \vdash t : A$ and $E_{\mathscr{C}} \vdash f(t) \approx g(t) : B$.

(eta) $x \approx \mathsf{eql}(e(x)) : E$, whenever x is a variable of sort E.

Note that Definition 6.1 introduces an apparent circularity, because unlike Definition 3.1 and 4.1, the well-sortedness judgement $\Sigma_{\mathscr{C}} \vdash t : E$ and the set of equations $E_{\mathscr{C}}$ are now defined in terms of each other. Of course this is not an actual circularity; it just means that these two items are defined by simultaneous induction. Similarly, the notion of a $(\Sigma_{\mathscr{C}}, E_{\mathscr{C}})$-algebras must be adjusted so that $\mathscr{T}(\mathsf{eql}) : \mathscr{T}(A) \to \mathscr{T}(E)$ is a partial function that is defined and satisfies $\mathscr{T}(e)(\mathscr{T}(\mathsf{eql})(x)) = x$ for those elements $x \in \mathscr{T}(A)$ satisfying $\mathscr{T}(f)(x) = \mathscr{T}(g)(x)$. With these adjustments, we have the following:

Remark 6.2. Similarly to Remark 4.2, we can show that there is a one-to-one correspondence between $(\Sigma_{\mathscr{C}}, E_{\mathscr{C}})$-algebras and $\mathscr{C}$-limit-preserving functors $F : \mathbf{A} \to \mathbf{Set}$. The equations *(beta)* and *(eta)* ensure that

$$F(E) \stackrel{F(e)}{\to} F(A) \stackrel{F(f),F(g)}{\rightrightarrows} F(B)$$

is an equalizer in **Set**. To show that the full subcategory $[\mathbf{Set}^{\mathbf{A}}]_{\mathscr{C}}$ of $\mathscr{C}$-limit-preserving functors is reflective, we first define the functor $L : \mathbf{Set}^{\mathbf{A}} \to [\mathbf{Set}^{\mathbf{A}}]_{\mathscr{C}}$, similarly to Definition 5.2. Then we can show that L is a left adjoint of the inclusion functor, by adapting the proof of Theorem 5.3.

7 Conclusion

We gave a brief introduction to the Lambek embedding, a version of the Yoneda embedding that preserves coproducts (or more generally, a chosen class of distinguished coproducts or colimits). The Lambek embedding is obtained by restricting the codomain of the Yoneda embedding to a suitable full subcategory of $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$, namely, the full subcategory of product-preserving functors (or more generally, functors that preserve the distinguished class of products or limits). This is a reflective subcategory of $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$. Like $\mathbf{Set}^{\mathbf{A}^{\mathrm{op}}}$ itself, this subcategory is complete and cocomplete, as well as monoidal closed provided that $\mathbf{A}$ is monoidal. Our method uses concepts from multi-sorted algebras. In particular, we observed that there is a one-to-one correspondence between product-preserving functors and certain multi-sorted algebras. We gave a direct syntactic construction of the functor L and proved that it is left adjoint to the inclusion functor.

Acknowledgements

This work was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) and by the Air Force Office of Scientific Research under Award No. FA9550-21-1-0041.

References

[1] Brian Day (1970): *On closed categories of functors*. In: *Reports of the Midwest Category Seminar IV, Springer Lecture Notes in Mathematics* 137, pp. 1–38, doi:`10.1007/BFb0060438`.

[2] Brian Day (1972): *A reflection theorem for closed categories*. *Journal of Pure and Applied Algebra* 2(1), pp. 1–11, doi:`10.1016/0022-4049(72)90021-7`.

[3] Peter J. Freyd (1964): *Abelian categories*. Harper & Row New York.

[4] Peter J. Freyd (1972): *Aspects of topoi*. *Bulletin of the Australian Mathematical Society* 7(1), pp. 1–76, doi:`10.1017/S0004972700044828`.

[5] Peng Fu, Kohei Kishida, Neil J. Ross & Peter Selinger (2022): *A biset-enriched categorical model for Proto-Quipper with dynamic lifting*. Preprint available from `arXiv:2204.13039`.

[6] J. F. Kennison (1968): *On limit-preserving functors*. *Illinois Journal of Mathematics* 12(4), pp. 616–619, doi:`10.1215/ijm/1256053963`.

[7] Joachim Lambek (1966): *Completions of categories: Seminar lectures given 1966 in Zürich*. *Lecture Notes in Mathematics* 24, Springer, doi:`10.1007/BFb0077265`.

[8] Octavio Malherbe (2013): *Categorical Models of Computation: Partially Traced Categories and Presheaf Models of Quantum Computation*. Ph.D. thesis, University of Ottawa, Department of Mathematics and Statistics. Available from `arXiv:1301.5087`.

[9] Francisco Rios & Peter Selinger (2018): *A categorical model for a quantum circuit description language. Extended Abstract*. In: *Proceedings of the 14th International Conference on Quantum Physics and Logic, QPL 2017, Nijmegen*, *Electronic Proceedings in Theoretical Computer Science* 266, pp. 164–178, doi:10.4204/EPTCS.266.11.