Relating Idioms, Arrows and Monads from Monoidal Adjunctions

Exequiel Rivas
$\pi.r^2$ team, INRIA - IRIF
Paris, France
er@irif.fr

We revisit once again the connection between three notions of computation: *monads*, *arrows* and *idioms* (also called *applicative functors*). We employ monoidal categories of finitary functors and profunctors on finite sets as models of these notions of computation, and develop the connections between them through adjunctions. As a result, we obtain a categorical version of Lindley, Yallop and Wadler's characterisation of monads and idioms as arrows satisfying an isomorphism.

1 Introduction

The semantic study of computational effects has not only provided tools for reasoning about effectful programs, but also introduced a methodology for organising functional code around fundamental interfaces coming from mathematics. This is the story of monads, which in the beginning were introduced by Moggi to model effectful computations, but soon after that Wadler internalised them in a functional programming language to structure code. In this way, monads became an interface capturing a pattern, which was later expressed as a *type-class*, and chosen to be the interface for the basic IO mechanism in Haskell. Over time, new interfaces were defined, each one providing different levels of control on how to combine computations. We are interested in two of these interfaces that emerged: *arrows* and *idioms*. Arrows were introduced by Hughes [7] and idioms, or *applicative functors*, by McBride and Paterson [14]. Even when arrows and idioms differ from monads, they still share some basic form. In previous work [19], we gave an unified presentation of these three interfaces in terms of monoids in monoidal categories. In this paper we take a first step in addressing the connection between these interfaces by looking at their monoidal structures and relating them by adjunctions.

Instead of starting from scratch, we build on previous results relating these structures. Ten years ago, in MSFP 2008, Lindley, Yallop and Wadler presented an article connecting idioms, arrows and monads. In a nutshell, we can summarise their result in the following diagram of embeddings

$$\begin{tikzcd}\text{Idioms} \arrow[r, hook] & \text{Arrows} \arrow[r, hook] & \text{Monads}\end{tikzcd}$$

together with the equations

$$\text{Idiom} = \text{Arrow} + (A \rightsquigarrow B \cong 1 \rightsquigarrow (A \to B)), \tag{1}$$
$$\text{Monad} = \text{Arrow} + (A \rightsquigarrow B \cong A \to (1 \rightsquigarrow B)). \tag{2}$$

These equations explain how to see idioms and monads as arrows in which a particular isomorphism holds. The method they used to establish these formulas was purely syntactical. Basically, they proposed theories for describing different versions of λ-calculus with effects, and used a notion of theory translation to show the relations.

This article is a first step towards understanding these equalities from a semantic point of view. In what follows, we deconstruct these characterisations by using the underlying structure of idioms, monads and arrows: functors and strong profunctors.

Remark 1. *In order to avoid size issues, we restrict ourselves to modelling the notions of computation by finitary functors, and their profunctor counterpart, which are strong endoprofunctors on finite sets. It should be possible to replace finite sets by a general category $\mathbb{C}$ together with an inclusion functor into sets, subject to certain conditions, such as density.*

Denoting the category of finite sets by $\mathbb{F}$ and the category of sets by $\mathbb{S}$, we aim to express Equations (1) and (2) in terms of functor categories

$$[\mathbb{F},\mathbb{S}] \qquad \text{and} \qquad [\mathbb{F}^{\mathrm{op}}\times\mathbb{F},\mathbb{S}]_{\mathrm{s}}$$

where -^{op} denotes the opposite category and the subscript "s" represents the restriction to strong profunctors. The idea that we develop in the article follows from the next observation. Given a strong endoprofunctor P, we can fix its first parameter and obtain a finitary functor:

$$P^{*}(X) = P(1,X), \quad \text{where 1 denotes the terminal object of } \mathbb{F}.$$

Conversely, there are two ways in which a finitary functor $F : \mathbb{F} \to \mathbb{S}$ might be presented as a strong endoprofunctor on $\mathbb{F}$:

$$F_{!}(X,Y) = F(X \to Y), \qquad F_{*}(X,Y) = X \to FY.$$

Taking $P(X,Y) = X \rightsquigarrow Y$ (the underlying profunctor of an arrow), we can express the isomorphisms

$$A \rightsquigarrow B \cong 1 \rightsquigarrow (A \to B), \qquad A \rightsquigarrow B \cong A \to (1 \rightsquigarrow B)$$

simply as the equivalences

$$P(A,B) \cong P(1,A \to B), \qquad P(A,B) \cong A \to P(1,B)$$

or directly as profunctor equivalences in terms of -^{*}, $\text{-}_{!}$ and -_{*}

$$P \cong (P^{*})_{!}, \qquad P \cong (P^{*})_{*}.$$

In the following sections, a pair of adjunctions between $\text{-}_{!}$, -^{*} and -_{*} are introduced, and as a result we obtain that the map $P \mapsto (P^{*})_{!}$ extends to a comonad and the map $P \mapsto (P^{*})_{*}$ extends to a monad. The mentioned adjunctions are monoidal, and thus establish equivalences between the categories of monoids. In this way, we get a first approximation to Lindley et al.'s result from a categorical point of view.

This article might be seen as a continuation of previous articles in the spirit of *notions of computation as monoids*. The starting point was an article of the author with Jaskelioff [19]. Later, the adjunctions between $\text{-}_{!}$, -^{*} and -_{*} were exploited in the context of non-monadic effect handlers [17], although their relation to the Equations (1) and (2) by using the derived monad/comonad and their corresponding monoidal structure was not treated.

Contributions. *The main contributions of this paper are:*

- *We give a detailed and more mathematical introduction to the adjunctions presented in previous work [17].*
- *We recognise in these adjunctions the essence of Lindley, Yallop and Wadler's result from a semantic point of view.*

The rest of the article is organised as follows. In Section 2, we review some background material needed to express the adjunctions and relate monoidal structures to arrows, idioms and monads. Then, in Section 3, we present adjunctions between the categories representing effects, and introduce some properties they have. In Section 4 we establish Equations (1) and (2) from a categorical argument, explaining Lindley et al.'s result from a semantic point of view. Finally, in Section 5, we conclude and discuss further work.

2 Background

We assume the reader is familiar with basic concepts of category theory, including adjunctions. We briefly introduce some categorical topics we need, such as monoidal categories, partly to be self-contained and partly to fix notations. An exception to this is the use of coends, which are not explained here mainly for brevity reasons. In case the reader is not familiar with them, coends might be (informally) seen as existential or Σ types. The usual reference for all these topics is Mac Lane's book [13], and a complete reference for (co)ends is Loregian's notes [12]. An introduction to these topics oriented for the functional programming community might be found in [19]. We try to follow standard categorical notation for most of the text. We denote exponentials by $A \to B$, the evaluation morphism by $\mathrm{ev} : (A \to B) \times A \to B$ and the currying of a morphism $f : A \times B \to C$ as $\lfloor f \rfloor : A \to (B \to C)$. Note that exponentials of $\mathbb{F}$ coincide with exponentials of $\mathbb{S}$ and their respective hom-sets, and so we treat them ambiguously.

2.1 Implementation of notions of computation

Before starting with the categorical preliminaries, we remind the reader of the basic interfaces we are treating, as in the rest of the text we will not put emphasis on the programming side.

The first interface we are interested in are monads, which are abstracted by the following type-class.

```
class Functor f ⇒ Monad f where
  return :: a → f a
  (>>=)  :: f a → (a → f b) → f b
```

A monad is a functor (type constructor which is compatible with $- \to -$) which is endowed with two basic primitives. Reading a term of type $f\ a$ as a computation returning a value of type a, then the primitive *return* embeds pure values as computations, while $(\gg\!=)$ (pronounced *bind*) accounts for the sequencing of computations.

The next interface we will model with monoidal categories is that of applicative functors.

```
class Functor f ⇒ Applicative f where
  pure :: a → f a
  (⊛)  :: f (a → b) → f a → f b
```

If we flip arguments for the combinator $(\circledast)$, we see that applicative functors are similar to monads, with the difference that the "following" computation on the combinator $(\circledast)$ cannot depend dynamically on its input parameter a.

The third interface we cover is arrows.

```
class Arrow (⇝) where
  arr    :: (a → b) → (a ⇝ b)
```

$$
\begin{array}{ll}
(\ggg) & :: (a \rightsquigarrow b) \to (b \rightsquigarrow c) \to (a \rightsquigarrow c) \\
\mathit{first} & :: (a \rightsquigarrow b) \to ((a,c) \rightsquigarrow (b,c))
\end{array}
$$

This interface builds on profunctors rather than on functors. Terms of type $a \rightsquigarrow b$ are read as computations that ask for input of type a and return output of type b. Using the *arr* combinator, one can construct a computation from a function. The combinator $(\ggg)$ sequences two computations. Finally, the combinator *first* represents the *strength* of the arrow: it is used to construct computations that operate on an extra parameter without modifying it. The *first* operation is handling the compatibility of the arrow with the cartesian structure.

Remark 2. *Morally, one can think that there is nothing corresponding to the first operation in the typeclasses Monad and Applicative because a strength can be obtained for free in the case of functors:*

$$
\begin{array}{l}
\sigma :: f\ a \to b \to f\ (a,b) \\
\sigma\ v\ b = \mathit{fmap}\ (\lambda a \to (a,b))\ v
\end{array}
$$

We return later to this point in our discussion of strong profunctors.

2.2 Monoidal categories

We recall briefly the basic definitions of monoidal categories. A monoidal category is a category $\mathbb{C}$ together with a bifunctor $\otimes : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$, a distinguished object I, and natural isomorphisms

$$\lambda : I \otimes A \cong A, \quad \rho : A \otimes I \cong A, \quad \alpha : A \otimes (B \otimes C) \cong (A \otimes B) \otimes C$$

such that certain coherences hold. The bifunctor $\otimes$ is referred to as the *tensor*, and the distinguished object I as the *unit*. Moreover, we refer to the monoidal category by the triple $(\mathbb{C}, \otimes, I)$, omitting λ, ρ and α. When it is clear from the context, we even omit the monoidal structure and just speak of $\mathbb{C}$ being a monoidal category.

Monoidal categories generalise categories with finite products: the bifunctor is given by the binary product $\times$, the unit by the terminal object, and the isomorphisms are defined as

$$
\begin{array}{lll}
\lambda = \pi_2 & \rho = \pi_1 & \alpha = \langle\langle \pi_1, \pi_1 \circ \pi_2 \rangle, \pi_2 \circ \pi_2 \rangle \\
\lambda^{-1} = \langle !_A, \mathrm{id} \rangle & \rho^{-1} = \langle \mathrm{id}, !_A \rangle & \alpha^{-1} = \langle \pi_1 \circ \pi_1, \langle \pi_2 \circ \pi_1, \pi_2 \rangle\rangle
\end{array}
$$

where $\langle f, g \rangle : A \to B \times C$ is defined by the universal property of products as the unique morphism such that $\pi_1 \circ \langle f, g \rangle = f$ and $\pi_2 \circ \langle f, g \rangle = g$.

Another popular example of monoidal category is the category of endofunctors $[\mathbb{C}, \mathbb{C}]$ on any category $\mathbb{C}$. The tensor is defined as composition of functors, and the identity functor Id is the unit. This a *strict* monoidal category, i.e. a monoidal category in which the isomorphisms λ, ρ and α are identities.

Following the *microcosm principle*, the notion of monoid can be defined in any monoidal category $(\mathbb{C}, \otimes, I)$. A monoid is a triple (M, m, e) where M is an object of $\mathbb{C}$, and $m : M \otimes M \to M$, $e : I \to M$ are morphisms of $\mathbb{C}$ such that the equations

$$\lambda = m \circ (e \otimes \mathrm{id}), \quad \rho = m \circ (\mathrm{id} \otimes e), \quad m \circ (m \otimes \mathrm{id}) \circ \alpha = m \circ (\mathrm{id} \otimes m)$$

hold. For example, a monoid in $(\mathbb{S}, \times, 1)$ is a monoid in the usual mathematical sense, and a monoid in $[\mathbb{C}, \mathbb{C}]$ is a monad on $\mathbb{C}$. Monoids on a monoidal category $(\mathbb{C}, \otimes, I)$ form a category, where an morphism from a monoid (M, m_M, e_M) to a monoid (N, m_N, e_N) is a morphism $f : M \to N$ such that

$$f \circ e_M = e_N \qquad \text{and} \qquad f \circ m_M = m_N \circ (f \otimes f).$$

We denote this category by $\mathrm{Mon}(\mathbb{C})$.

Let $(\mathbb{C},\otimes,I)$ and $(\mathbb{D},\oplus,J)$ be monoidal categories. It is natural to consider functors that are compatible with the monoidal structures. A *strong monoidal functor* is a triple (F,γ,γ_0) where $F:\mathbb{C}\to\mathbb{D}$ is a functor, $\gamma_0 : J\cong FI$ is an isomorphism and $\gamma_{A,B} : FA\oplus FB\cong F(A\otimes B)$ is family of isomorphisms natural in A and B such that certain coherences hold. If instead of isomorphisms we just consider morphisms, then we have *monoidal functors* if we ask the morphisms to go from left to right, or *opmonoidal functors* if the morphisms go from right to left. A monoidal or opmonoidal functor can be *normal* if the morphism relating the units (I and J) is invertible. We usually omit the monoidal structure from (F,γ,γ_0) and speak of F being a (strong, op)monoidal functor. However, it is important to have in mind that a functor could have different monoidal structures.

2.3 Profunctors and their strengths

An endoprofunctor on the category of finite sets is a functor $\mathbb{F}^{\mathrm{op}}\times\mathbb{F}\to\mathbb{S}$. By fixing arguments, each profunctor on $\mathbb{F}$ comes with two canonical strengths. If we fix the contravariant argument, we obtain a functor with a strength σ^P in the usual sense (see Remark 2), while in the case of fixing the covariant argument, we obtain a strength in the sense of Brady and Trimble [4]. That is, we obtain a contravariant functor $F:\mathbb{F}\to\mathbb{S}$ with a family of maps

$$\varsigma_{A,B} : F(A\times B)\times A\to FB, \qquad (v,a)\mapsto F(\lambda z.(a,z))(v)$$

natural in A and B, such that certain coherence conditions hold. We denote these two strengths for an endoprofunctor $P:\mathbb{F}^{\mathrm{op}}\times\mathbb{F}\to\mathbb{S}$ by

$$\sigma^P : P(X,Y)\times Z\to P(X\times Z,Y), \qquad\qquad \varsigma^P : P(X\times Y,Z)\times X\to P(Y,Z).$$

Remark 3. *Note that such strengths are always available for functors $\mathbb{F}\to\mathbb{S}$. When we consider these strengths on the component $Z=1$, the strength morphism is actually invertible.*

The strengths σ^P and ς^P are strengths independent in each variable. Alternatively, we might consider strength for endoprofunctors which act on both variables at the same time. This gives the notion of *strong endoprofunctor*, which is a form of the notion of *Tambara module* defined by Pastro and Street [16].

Definition 1. *A endoprofunctor $P:\mathbb{F}^{\mathrm{op}}\times\mathbb{F}\to\mathbb{S}$ is said to be* (right) strong *if it comes equipped with a family of morphisms*

$$\mathrm{str}_{X,Y,Z} : P(X,Y)\to P(X\times Z,Y\times Z)$$

natural in X, Y and dinatural in Z such that the equations

$$P(\mathrm{id},\pi_1)\circ\mathrm{str}_{X,Y,1} = P(\pi_1,\mathrm{id}),$$
$$\mathrm{str}_{X,Y,W}\circ\mathrm{str}_{X,Y,V} = P(\alpha^{-1},\alpha)\circ\mathrm{str}_{X,Y,V\times W}$$

hold. Since we work with the cartesian monoidal structure, there is no need to define left strong and bistrong endoprofunctors.

Accordingly, a notion of morphism between strong endoprofunctors is defined.

Definition 2. *A* strong natural transformation *from (P,str^P) to (Q,str^Q) is a natural transformation $\tau : P\to Q$ such that the equation*

$$\tau_{X\times Z,Y\times Z}\circ\mathrm{str}^P_{X,Y,Z} = \mathrm{str}^Q_{X,Y,Z}\circ\tau_{X,Y}$$

holds.

Strong endoprofunctors and strong natural transformations between them form a category, which we denote by $[\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]_{\mathrm{s}}$. There is a forgetful functor

$$U : [\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]_{\mathrm{s}} \to [\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]$$

which forgets the strength on a strong endoprofunctor. The functor U has left and right adjoints, providing a way to construct free and cofree strong endoprofunctors from an endoprofunctor [19, 16].

2.4 Monoidal Structures on Finitary Functors

We have already mentioned that the category of endofunctors forms a monoidal category with composition, and that its monoids are monads. In particular, a monoid in the monoidal category $([\mathbb{S}, \mathbb{S}], \circ, \mathrm{Id})$ is a monad on $\mathbb{S}$. A finitary functor $F : \mathbb{F} \to \mathbb{S}$ might be seen as a functor $F : \mathbb{S} \to \mathbb{S}$ by the following construction:

$$\bar{F}(Z) = \int^{Y} F(Y) \times (i\, Y \to Z)$$

where $i : \mathbb{F} \to \mathbb{S}$ denotes the inclusion functor from finite sets to sets.

When we consider a finitary functors, i.e. the category $[\mathbb{F}, \mathbb{S}]$, we obtain an alternative presentation of the composition structure, which is defined by the coend formula

$$(F \circ G)(X) = \int^{C} F\, C \times (C \to GX)$$

and the unit is given by the inclusion functor i. A monoid for this monoidal structure is a finitary monad on $\mathbb{S}$, i.e. a finitary functor F together with morphisms

$$e : i \to F, \qquad m : F \circ F \to F$$

such that certain diagrams commute. The morphism e corresponds to the *return* operation in the *Monad* type-class, while m corresponds to the operation $(\gg\!=)$.

There is another monoidal structure on finitary functors. As finitary functors are presheaves, they have a canonical monoidal structure given by the Day convolution. The tensor of finitary functors F and G as objects in $[\mathbb{F}, \mathbb{S}]$ might be expressed as

$$(F \star G)(X) = \int^{C} F\, C \times G\, (C \to X),$$

and the inclusion functor $i : \mathbb{F} \to \mathbb{S}$ acts again as the unit. A monoid for this monoidal structure is given by a finitary functor F together with morphisms

$$e : i \to F, \qquad m : F \star F \to F$$

such that certain diagrams commute. The morphism e represents the *pure* operation of an idiom, and m represents the combinator $(\circledast)$.

Remark 4. *The categorical presentation of idioms as* monoidal functors with a strength *is well known [14]. We have chosen a different (but equivalent) presentation for the Day convolution as it is closer to the traditional Applicative interface. As the monoidal structure considered is cartesian, both constructions are equivalent.*

In contexts where confusion might arise, to distinguish between $([\mathbb{F}, \mathbb{S}], \circ, i)$ and $([\mathbb{F}, \mathbb{S}], \star, i)$ without having to mention all the monoidal structure, we simply refer to them as $[\mathbb{F}, \mathbb{S}]_{\circ}$ and $[\mathbb{F}, \mathbb{S}]_{\star}$.

2.5 Monoidal Structure on Strong Profunctors

Profunctors in general, and endoprofunctors in particular, can be composed by Bénabou's tensor, giving rise to a monoidal structure on the category of endoprofunctors $[\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]$. Indeed, this monoidal structure can be lifted to strong endoprofunctors. We make this structure precise in the following definition.

Definition 3. *The category* $[\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]_{\mathrm{s}}$ *has a monoidal structure, with tensor product of* (P, str^P) *and* (Q, str^Q) *defined as* $(P \otimes Q, \mathrm{str}^{(P\otimes Q)})$ *where*

$$(P \otimes Q)(X,Y) = \int^W P(X,W) \times Q(W,Y),$$

$$\mathrm{str}^{(P\otimes Q)}_{X,Y,Z} : \int^W P(X,W) \times Q(W,Y) \longrightarrow \int^W P(X \times Z, W) \times Q(W, Y \times Z).$$

The strength $\mathrm{str}^{(P\otimes Q)}_{X,Y,Z}$ *is defined by the universal property of coends as the unique morphism such that*

$$\mathrm{str}^{(P\otimes Q)}_{X,Y,Z} \circ \iota_W = \iota_{W\times Z} \circ (\mathrm{str}^P_Z \times \mathrm{str}^Q_Z).$$

with ι_W *being the coend injection for* W*. The unit for the monoidal tensor is* $(\mathrm{Hom}, \mathrm{str}^{\mathrm{Hom}})$*, where*

$$\begin{array}{rccc} \mathrm{str}^{\mathrm{Hom}}_{X,Y,Z} : & \mathrm{Hom}(X,Y) & \longrightarrow & \mathrm{Hom}(X \times Z, Y \times Z) \\ & f & \longmapsto & f \times \mathrm{id} \end{array}$$

As observed by Jacobs et al. [8], arrows correspond to monoids in this category (disregarding the fact they consider a general category $\mathbb{C}$ instead of $\mathbb{F}$). A monoid in this category is a strong profunctor (A, str^A) together with morphisms

$$e : \mathrm{Hom} \to A, \qquad m : A \otimes A \to A$$

such that certain diagrams commute. The morphism e represents the *arr* operation of the arrow, m represents sequential composition $\ggg$, and the strength str^A of the profunctor gives the *first* operation.

In what follows, we sometimes refer to a strong profunctor (P, str^P) just by P, but the reader has to have in mind that a profunctor could have more than one strength, and that we expect natural transformations to be strong.

3 The Cayley and Kleisli Adjunctions

The Equations (1) and (2) relate the categories of monoids corresponding to

$$[\mathbb{F}, \mathbb{S}]_\star, \qquad [\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]_{\mathrm{s}}, \qquad [\mathbb{F}, \mathbb{S}]_\circ .$$

To obtain this result, we first construct adjunctions between the underlying categories, and then see how they interact with the monoidal structures. The central objects of our adjunctions are strong profunctors, and how they can be recast into a functor. As we explained in the introduction, the contravariant argument of the profunctor might be fixed to the terminal object 1 of $\mathbb{F}$, obtaining thus a finitary functor. We capture the re-casting with the functor -*.

$$\begin{array}{rcl} -^* & : & [\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]_{\mathrm{s}} \longrightarrow [\mathbb{F}, \mathbb{S}] \\ P^* & = & Z \mapsto P(1, Z) \\ \tau^*{}_Z & = & \tau_{1,Z} \end{array}$$

The next step is to find left and right adjoints for the re-casting functor. These adjoints might be derived using the left and right Kan extensions [13], although we do a direct presentation in order to minimise the background material.

We begin describing the left adjoint to -^*. This functor was called *Cayley* in previous work [16, 19, 17]. We now refer to it as $\text{-}_!$, but keep the name Cayley to refer to the adjunction.

Theorem 1. *The functor -^* has a left adjoint $\text{-}_!$ given by*

$$\begin{array}{lcl} \text{-}_! & : & [\mathbb{F},\mathbb{S}] \longrightarrow [\mathbb{F}^{\mathrm{op}} \times \mathbb{F},\mathbb{S}]_{\mathrm{s}} \\ F_! & = & ((X,Y) \mapsto F(X \to Y), v \mapsto F(\lfloor (\mathrm{ev} \times \mathrm{id}) \circ \alpha \rfloor)(v)) \\ (\tau_!)_{X,Y} & = & \tau_{X \to Y} \end{array}$$

Moreover, the unit of the adjunction is a natural isomorphism.

Proof. The unit and counit of the adjunction are

$$\begin{array}{lcl} \eta^!_F & : & F\,Z \longrightarrow F_!{}^*\,Z \\ \eta^!_F & = & F(\lfloor \rho \rfloor) \end{array} \qquad \begin{array}{lcl} \varepsilon^!_{(P,\mathrm{str})} & : & P^*{}_!(X,Y) \longrightarrow P(X,Y) \\ \varepsilon^!_{(P,\mathrm{str})} & = & P(\lambda^{-1},\mathrm{ev}) \circ \mathrm{str} \end{array}$$

where these definitions are parametric over Z and (X,Y) respectively. After some calculations, it can be shown that the triangular identities hold. The inverse of the unit is simply defined as $\eta^{!-1}_F = F(\mathrm{ev} \circ \rho^{-1})$. □

The right adjoint to -_* is defined by a construction similar to the direct image of a functor. In previous work we have called this functor *Kleisli*, and therefore we now refer to the resulting adjunction as the *Kleisli adjunction*.

Theorem 2. *The functor -^* has a right adjoint -_* given by*

$$\begin{array}{lcl} \text{-}_* & : & [\mathbb{F},\mathbb{S}] \longrightarrow [\mathbb{F}^{\mathrm{op}} \times \mathbb{F},\mathbb{S}]_{\mathrm{s}} \\ F_* & = & ((X,Y) \mapsto i\,X \to F\,Y, v \mapsto F(\lfloor \sigma \circ (\mathrm{ev} \times \mathrm{id}) \circ \alpha \rfloor)(v)) \\ (\tau_*)_{X,Y} & = & \lfloor \tau_Y \circ \mathrm{ev} \rfloor \end{array}$$

Moreover, the counit of the adjunction is a natural isomorphism.

Proof. The unit and counit of the adjunction are

$$\begin{array}{lcl} \eta^*_{(P,\mathrm{str})} & : & P(X,Y) \longrightarrow P^*{}_*(X,Y) \\ \eta^*_{(P,\mathrm{str})} & = & \lfloor \varsigma^P \circ P(\rho,\mathrm{id}) \rfloor \end{array} \qquad \begin{array}{lcl} \varepsilon^*_F & : & F_*{}^*\,Z \longrightarrow F\,Z \\ \varepsilon^*_F & = & \mathrm{ev} \circ \rho^{-1} \end{array}$$

where these definitions are parametric over (X,Y) and Z respectively.

Again, it is routine to show that the triangular identities hold. The inverse of the counit is $\varepsilon^{*-1}_F = \lfloor \rho \rfloor$. □

Summing up, we obtain the following situation, which we depict in a diagram.

$$\begin{tikzcd} {[\mathbb{F},\mathbb{S}]} \arrow[r, bend left, "\text{-}_!"] \arrow[r, bend right, "\text{-}_*"'] & {[\mathbb{F}^{\mathrm{op}} \times \mathbb{F},\mathbb{S}]_{\mathrm{s}}} \arrow[l, "\text{-}^*" description] \end{tikzcd}$$

$\perp$ $\perp$

The fact that the unit (counit) is invertible can be expressed in a number of equivalent statements. It will prove useful to ave these equivalences around, and so we state the following theorem [6].

Theorem 3. *Let $L \vdash R : \mathbb{C} \to \mathbb{D}$ be an adjunction.*
The following statements are equivalent:

- *The counit $\varepsilon : LR \to \mathrm{Id}$ is a natural isomorphism.*
- *R is fully faithful.*
- *The monad associated with the adjunction is idempotent.*

Dually, the following are also equivalent:

- *The unit $\eta : \mathrm{Id} \to RL$ is a natural isomorphism.*
- *L is fully faithful.*
- *The comonad associated with the adjunction is idempotent.*

We can apply Theorem 3 to the adjunctions $\text{-}^* \dashv \text{-}_*$ and $\text{-}_! \dashv \text{-}^*$, and obtain the following propositions.

Proposition 1. *The functor $\text{-}_!$ is fully faithful and the comonad $\Box P = (P^*)_!$ is idempotent.*

Proposition 2. *The functor -_* is fully faithful and the monad $\Diamond P = (P^*)_*$ is idempotent.*

4 Equivalences from Monoidal Adjunctions

We now want to relate these adjunctions to the monoidal structures presented in Section 2. The first step is to define natural transformations and adjunctions that are compatible with monoidal structures [1].

Definition 4. *Let $(F,\gamma,\gamma_0),(G,\delta,\delta_0) : (\mathbb{C},\otimes,I) \to (\mathbb{D},\oplus,J)$ be monoidal functors. A natural transformation $\tau : F \to G$ is* monoidal *if*

$$\tau_I \circ \gamma_0 = \delta_0, \qquad \tau_{A\otimes B} \circ \gamma_{A,B} = \delta_{A,B} \circ (\tau_A \oplus \tau_B).$$

There is a straightforward dual definition of opmonoidal natural transformations *between opmonoidal functors.*

Definition 5. *An adjunction $L \dashv R : \mathbb{C} \to \mathbb{D}$ with unit η and counit ε, between monoidal categories $(\mathbb{C},\otimes,I)$ and $(\mathbb{D},\oplus,J)$, is a* monoidal adjunction *if L and R are monoidal functors and η and ε are monoidal natural transformations. Similarly, the adjunction is* opmonoidal *if both L and R are opmonoidal and η and ε are opmonoidal natural transformations. There is an intermediate case, sometimes called* colax-lax monoidal adjunction*, which happens when L is opmonoidal with structure (ϕ,ϕ_0), R is monoidal with structure (γ,γ_0) and the following identities hold:*

$$R\phi_0 \circ \eta_I = \gamma_0, \qquad R\gamma_{A,B} \circ \eta_{A\otimes B} = \phi_{LA,LB} \circ (\eta_A \otimes \eta_B)\,.$$

The following theorem provides a way to transport a monoidal structure from a functor to its adjoint and obtain a colax-lax monoidal adjunction. This is a particular case of a phenomenon known as a *doctrinal adjunction* [9].

Proposition 3. *Let $L \dashv R$ be an adjunction between monoidal categories $(C,\otimes,I)$ and $(D,\oplus,J)$.*

If L is an opmonoidal functor, then R has structure of monoidal functor and the adjunction is colax-lax. Moreover, if L is a strong monoidal functor, then η and ε are monoidal natural transformations.

Dually, if R is a monoidal functor, then L has structure of opmonoidal functor and the adjunction is colax-lax. Moreover, if R is a strong monoidal functor, then η and ε are opmonoidal natural transformations.

The following propositions state the monoidal structure of the functors involved in the Kleisli and Cayley adjunctions.

Proposition 4. *Considering* $[\mathbb{F},\mathbb{S}]$ *with Day convolution and* $[\mathbb{F}^{\mathrm{op}}\times\mathbb{F},\mathbb{S}]_{\mathrm{s}}$ *with Bénabou's tensor, then the functor* $\text{-}_!$ *is strong monoidal,* -^* *is monoidal and Cayley's adjunction is a monoidal adjunction.*

Proof. The strong monoidality of Cayley's functor was proven by Pastro and Street [16]. Applying Proposition 3, we obtain the later results. ☐

In the case of the adjunction $\text{-}_* \dashv \text{-}^*$, we obtain a "weaker" result.

Proposition 5. *Considering* $[\mathbb{F},\mathbb{S}]$ *with substitution product and* $[\mathbb{F}^{\mathrm{op}}\times\mathbb{F},\mathbb{S}]_{\mathrm{s}}$ *with Bénabou's tensor, then the functor* -_* *is monoidal,* -^* *is opmonoidal and Kleisli's adjunction is a colax-lax adjunction.*

Proof. The monoidality of Kleisli's functor is well-known, see for instance [19]. Applying Proposition 3, we obtain that -_* is monoidal. ☐

We refer to the monoidal structure of -_* by (ξ,ξ_0). What we meant above by "weaker" is that we do not obtain that -_* is strong monoidal. As we will see later, this prevents some symmetry with the Cayley adjunction. Instead of trying to obtain this condition by a careful analysis involving finiteness and imposing extra conditions [2], we rather just work with monoidality of -_*, and put some extra effort in the next section when we show that *monads are promiscuous*.

The diagram of adjunctions in the previous section can now be unfolded. The functors in the upper part are opmonoidal and the functors in the lower part are monoidal.

$$\begin{tikzcd}
([\mathbb{F},\mathbb{S}],\star) \arrow[r, bend left, "\text{-}_!"] \arrow[r, phantom, "\perp"] & ([\mathbb{F}^{\mathrm{op}}\times\mathbb{F},\mathbb{S}]_{\mathrm{s}},\otimes) \arrow[l, bend left, "\text{-}^*"] \arrow[r, bend left, "\text{-}^*"] \arrow[r, phantom, "\perp"] & ([\mathbb{F},\mathbb{S}],\circ) \arrow[l, bend left, "\text{-}_*"]
\end{tikzcd}$$

We now give a categorical characterisation of monoids which are also algebras (coalgebras) for an idempotent monad (comonad). This definition provides a basis for presenting the right hand side of Equations (1) and (2).

Definition 6. *Given an idempotent monad (comonad)* $T:\mathbb{C}\to\mathbb{C}$ *on a monoidal category* $(\mathbb{C},\otimes,I)$*, a* T-monoid *is a quadruple* (M,m,e,α) *where*

- $(M, m: M\otimes M\to M, e: I\to M)$ *is a monoid in* $\mathbb{C}$,
- $(M,\alpha: TM\to M)$ *is a* T*-algebra* ($(M,\alpha: M\to TM)$ *is a* T*-coalgebra).*

Remark 5. *Notice that there is no explicit coherence condition between the monoid structure and the* T*-algebra structure. However, the idempotency of the monad provides coherence between the two structures. For example, if* T *is a strong monad, then a* T*-monoid in our sense is automatically a* T*-monoid in the sense of Fiore and Saville [5].*

We can form a category of T-monoids by considering morphisms between T-monoids which are morphisms as monoids and morphisms as T-algebras between the underlying objects. However, as the monad (comonad) is idempotent, the requirement of being a T-algebra morphism becomes trivial. In fact, the T-algebra structure of a T-monoid is like a property: an object X can have at most one T-algebra structure, and in case it does, it is η_X^{-1} (ε_X^{-1}). Despite that, we define a category of T-monoids, as it will be useful to have a concrete name for it.

Definition 7. *Given an idempotent monad (comonad) T on $\mathbb{C}$, the* category of T-monoids*,* $\mathrm{Mon}(T)$*, consists of*

- *Objects: T-monoids.*
- *Morphisms: T-monoid homomorphisms, i.e. morphisms in $\mathbb{C}$ that are both monoid morphisms and T-algebra (coalgebra) morphisms at the same time.*

From the comment above, it is clear that $\mathrm{Mon}(T)$ is equivalent to the full subcategory of $\mathrm{Mon}(\mathbb{C})$ such that the underlying object of the monoid has a T-algebra structure.

We are now ready to revisit the results of Lindley et al., and obtain their categorical analogues. We do so by postulating the equalities in Equations (1) and (2) as equivalences between categories.

4.1 Idioms are Oblivious

As already described, Lindley et al. [11] characterise idioms as the arrows in which the isomorphism

$$\begin{tikzcd} A \rightsquigarrow B \arrow[r, bend left, "delay"] \arrow[r, phantom, "\cong"] & 1 \rightsquigarrow (A \to B) \arrow[l, bend left, "force"] \end{tikzcd}$$

holds. The right to left morphism is available in any arrow as

$$\begin{array}{l} \mathit{force} :: (1 \rightsquigarrow (a \to b)) \to (a \rightsquigarrow b) \\ \mathit{force} = \lambda f.\ \mathit{arr}\ (\lambda x.\ ((),x)) \ggg \mathit{first}\, f \ggg \mathit{arr}\ (\lambda (f,a).\, f\ a) \end{array}$$

From the arrow laws, it can be checked that *force*, as defined by Lindley et al., is exactly the morphism given by the counit

$$\varepsilon^{!}_{(P,\mathrm{str}^P)} : P^{*}{}_{!}(X,Y) = P(1, X \to Y) \longrightarrow P(X,Y)$$

of the comonad $\Box$. Therefore, the isomorphism described in Equation (1) can be expressed as a coalgebra for the idempotent comonad $\Box$, i.e. an inverse for $\varepsilon^{!}$. The other part of the equation, the arrow, might be seen as a monoid in $[\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]_{\mathrm{s}}$ as we described in Section 2, thus obtaining that the equivalent presentation of the equation:

$$\text{Monoid in } [\mathbb{F},\mathbb{S}]_{\star} = \Box\text{-monoid in } [\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]_{\mathrm{s}}$$

To show this equivalence, we must be able to move monoids from one category to another. Doing this through a monoidal functor is covered by the following classical theorem (see for instance [1]):

Theorem 4. *A monoidal functor $(F,\phi) : (\mathbb{C},\otimes,I) \to (\mathbb{D},\oplus,J)$ can be lifted to a functor*

$$\begin{array}{rcl} F^{\bullet} & : & \mathrm{Mon}(\mathbb{C}) \longrightarrow \mathrm{Mon}(\mathbb{D}) \\ & & (M,m,e) \mapsto (FM, Fm \circ \phi, Fe \circ \phi_0) \end{array}$$

A categorical explanation of the equation is given by the following theorem, for which we give a direct proof using the theorem above.

Theorem 5. *Let $L \dashv R : (\mathbb{C},\otimes,I) \to (\mathbb{D},\oplus,J)$ be an adjunction in which η is an isomorphism, and L is strong monoidal. Then,* $\mathrm{Mon}(\mathbb{C})$ *is equivalent to* $\mathrm{Mon}(LR)$*.*

Proof. We give a direct proof of the equivalence. Using Proposition 3, we obtain a monoidal structure on R, and moreover, that the adjunction $L \dashv R$ is monoidal. As L and R are monoidal functors, they lift monoids, and give the following functors which form the equivalence.

$$\begin{aligned} L^\bullet \quad : \quad & \mathrm{Mon}\,(\mathbb{C}) \longrightarrow \mathrm{Mon}\,(LR) \\ & (M,m,e) \mapsto (LM, Lm \circ \phi, Le \circ \phi_0, L\eta) \end{aligned}$$

$$\begin{aligned} R^\bullet \quad : \quad & \mathrm{Mon}\,(LR) \longrightarrow \mathrm{Mon}\,(\mathbb{C}) \\ & (M,m,e,\alpha) \mapsto (RM, Rm \circ \psi, Re \circ \psi_0) \end{aligned}$$

In addition, we need to provide natural isomorphisms $\beta : \mathrm{Id}_{\mathrm{Mon}(LR)} \cong L^\bullet \circ R^\bullet$ and $\gamma : \mathrm{Id}_{\mathrm{Mon}(\mathbb{C})} \cong R^\bullet \circ L^\bullet$. For γ we simply propose η, while for β we use the LR-coalgebra.

$$\begin{aligned} &\gamma_{(M,m,e)} : (M,m,e) \longrightarrow R^\bullet L^\bullet (M,m,e) && \beta_{(M,m,e,\alpha)} : (M,m,e,\alpha) \longrightarrow L^\bullet R^\bullet (M,m,e,\alpha) \\ &\gamma_{(M,m,e)} = \eta && \beta_{(M,m,e,\alpha)} = \alpha \end{aligned}$$

That β is well-defined comes from $\alpha = \varepsilon^{-1}$, ε being a monoidal natural transformation and (M,α) being a coalgebra for LR. □

When we apply this theorem to the case $\mathbb{C} = [\mathbb{F},\mathbb{S}]$, $\mathbb{D} = [\mathbb{F}^{\mathrm{op}} \times \mathbb{F},\mathbb{S}]_{\mathrm{s}}$, $L = \text{-}_!$ and $R = \text{-}^*$, we obtain the semantic account of Equation (1):

$$\mathrm{Mon}\,([\mathbb{F},\mathbb{S}]) \text{ and } \mathrm{Mon}\,(\Box) \text{ are equivalent categories.}$$

We close the discussion on idioms by making the following observation. The monoidal structure given by the Day convolution on $[\mathbb{F},\mathbb{S}]$ might be recovered from the one in $[\mathbb{F}^{\mathrm{op}} \times \mathbb{F},\mathbb{S}]_{\mathrm{s}}$. As the comonad $\Box$ is monoidal (being the composition of two monoidal functors), we could lift the monoidal structure from $[\mathbb{F}^{\mathrm{op}} \times \mathbb{F},\mathbb{S}]_{\mathrm{s}}$ to the category of $\Box$-coalgebras [15], which, as we noted before, is equivalent to the category of finitary functors.

4.2 Monads are Promiscuous

In the case of monads, Lindley et al. [11] characterise them as the arrows in which the isomorphism

$$\begin{tikzcd} A \rightsquigarrow B \arrow[r, bend left, "eval"] \arrow[r, phantom, "\cong"] & A \to (1 \rightsquigarrow B) \arrow[l, bend left, "lave"] \end{tikzcd}$$

holds. Instead of postulating an inverse to a morphism from left to right, they follow Hughes [7] and ask for a mapping

$$app :: (a \rightsquigarrow b, a) \rightsquigarrow b$$

which has to satisfy a set of three laws

$$\begin{aligned} first\ (arr\ (\lambda x.\ arr\ (\lambda y.\ (x,y)))) \ggg app &= arr\ id \\ first\ (arr\ (g \ggg)) \ggg app &= second\ g \ggg app \\ first\ (arr\ (\ggg h)) \ggg app &= app \ggg h \end{aligned}$$

These are the laws originally proposed by Hughes. Notice that such a mapping *app* might not be expressed in our framework. Naively, it would be an element of $P(P(A,B)\times A,B)$, which is not well-defined as $P(A,B)$ is not necessarily a finite set.

It can be shown, using the arrow laws, that giving a mapping *app* which satisfies the three axioms above is equivalent to giving an inverse to the mapping

$$\begin{aligned}&\mathit{eval} :: (a \rightsquigarrow b) \to (a \to (1 \rightsquigarrow b))\\ &\mathit{eval} = \lambda c.\ \lambda a.\ \mathit{arr}\ (\lambda ().\ a) \ggg c\end{aligned}$$

This map is already present in the translation of Lindley et al. in the proof of the equivalence, and corresponds to the morphism given by the unit

$$\eta^*_{(P,\mathrm{str}^P)} : P(X,Y) \longrightarrow P^*{}_*(X,Y) = X \to P(1,Y)$$

of the monad $\Diamond$. Therefore, the isomorphism described in Equation (2) can be expressed as a algebra for the idempotent monad $\Diamond$, i.e. an inverse for η^*. The arrow part of the equation might be seen as a monoid in $[\mathbb{F}^{\mathrm{op}}\times\mathbb{F},\mathbb{S}]_{\mathrm{s}}$, thus obtaining that the equivalent presentation:

$$\text{Monoid in } [\mathbb{F},\mathbb{S}]_\circ = \Diamond\text{-monoid in } [\mathbb{F}^{\mathrm{op}}\times\mathbb{F},\mathbb{S}]_{\mathrm{s}}$$

To prove the equivalence induced by Equation (2), we might want to use a kind-of-dual theorem to Theorem 5. However, notice that this time the left adjoint functor is not strong monoidal, but just opmonoidal, which means that we cannot easily map monoids to monoids. We need to ask for additional conditions to ensure that the left adjoint lifts monoids. This case is covered by the following lemma, which is due to Porst and Street [18].

Lemma 1. *Let $(L,\varphi,\varphi_0) : (\mathbb{C},\otimes,I) \to (\mathbb{D},\oplus,J)$ be an opmonoidal functor and (C,m,e) a monoid in $\mathbb{C}$. If*

- $\varphi_0 : LI \to J$,
- $\varphi_{C,C} : L(C\otimes C) \to LC \oplus LC$,
- $\varphi_{C\otimes C,C} : L((C\otimes C)\otimes C) \to L(C\otimes C)\oplus LC$

are invertible, then $(LC, Lm\circ(\varphi_{C,C})^{-1}, Le\circ(\varphi_0)^{-1})$ is a monoid in $\mathbb{D}$.

Using this lemma, we can prove the following theorem, which underlies the basic structure of Equation (2).

Theorem 6. *Let $L\dashv R : (\mathbb{C},\otimes,I) \to (\mathbb{D},\oplus,J)$ be an adjunction in which ε is an isomorphism, (R,γ,γ_0) is a monoidal functor and the morphisms*

$$L\gamma_0 : LI \to LRJ, \qquad\qquad L(\gamma\circ(\eta\otimes\mathrm{id})) : L(C\otimes RD) \to LR(LC\oplus D)$$

are invertible. Then, $\mathrm{Mon}\,(\mathbb{D})$ *is equivalent to* $\mathrm{Mon}\,(RL)$.

Proof. Again, we give a concrete description of the equivalence. As R is monoidal, it preserves monoids, so we use Theorem 4 to lift it to monoids.

$$\begin{aligned}R^\bullet \quad : \quad &\mathrm{Mon}\,(\mathbb{D}) \longrightarrow \mathrm{Mon}\,(RL)\\ &(M,m,e) \mapsto (RM, Rm\circ\gamma, Re\circ\gamma_0, R\varepsilon)\end{aligned}$$

The inverse function this time is a bit more complex, as L is not monoidal, and therefore does not automatically preserve monoids. We use Lemma 1 to transport monoids. For a M-monoid (M,m,e,α), we can construct the required inverses as

- $(\varphi_0)^{-1} = (L\gamma_0)^{-1} \circ \varepsilon^{-1}$,
- $(\varphi_{C,C})^{-1} = L(\mathrm{id} \otimes \alpha) \circ (L(\gamma \circ (\eta \otimes \mathrm{id})))^{-1} \circ \varepsilon^{-1}$,
- $(\varphi_{C\otimes C,C})^{-1} = L(\mathrm{id} \otimes \alpha) \circ (L(\gamma \circ (\eta \otimes \mathrm{id})))^{-1} \circ \varepsilon^{-1}$.

We can apply Lemma 1, and obtain a functor $L^\bullet$:

$$\begin{aligned} L^\bullet \quad : \quad & \mathrm{Mon}\,(RL) \longrightarrow \mathrm{Mon}\,(\mathbb{D}) \\ & (M, m, e, \alpha) \mapsto (LM, Lm \circ (\varphi_{C,C})^{-1}, Le \circ (\varphi_0)^{-1}) \end{aligned}$$

Notice that each $(\varphi_{C,C})^{-1}$ depends on an α. To prove the equivalence, we need to provide natural isomorphisms $\gamma : \mathrm{Id}_{\mathrm{Mon}(\mathbb{D})} \cong L^\bullet \circ R^\bullet$ and $\beta : R^\bullet \circ L^\bullet \cong \mathrm{Id}_{\mathrm{Mon}(RL)}$. We propose the following.

$$\begin{aligned} &\beta_{(M,m,e)} : L^\bullet R^\bullet(M, m, e) \longrightarrow (M, m, e) && \gamma_{(M,m,e,\alpha)} : R^\bullet L^\bullet(M, m, e, \alpha) \longrightarrow (M, m, e, \alpha) \\ &\beta_{(M,m,e)} = \varepsilon && \gamma_{(M,m,e,\alpha)} = \alpha \end{aligned}$$

These are invertible, as from assumptions ε is invertible and α has η as its inverse. □

The following proposition provides the missing hypotheses to use the theorem above in the case we are interested.

Proposition 6. *The morphisms*

$$\xi_0{}^* : \mathrm{Hom}^* \to (i_*)^*, \qquad (\xi \circ (\eta^* \otimes \mathrm{id}))^* : (P \otimes F_*)^* \to ((P^* \oplus F)_*)^*$$

are invertible.

Proof. That these morphisms are invertible can be better seen by unfolding the functor definitions on a finite set Z:

$$\xi_0{}^*{}_Z : \mathrm{Hom}(1, Z) \to (1 \to Z)$$

$$(\xi \circ (\eta^* \otimes \mathrm{id}))^*{}_Z : \int^W P(1, W) \times (iW \to FZ) \to \left(1 \to \int^W P(1, W) \times (iW \to FZ)\right)$$

As the morphisms are only using the component 1, they are invertible (see remark on Subsection 2.3). □

Finally, we obtain the categorical counterpart of Equation (2) by applying this theorem to $\mathbb{C} = [\mathbb{F}^{\mathrm{op}} \times \mathbb{F}, \mathbb{S}]_{\mathrm{s}}$, $\mathbb{D} = [\mathbb{F}, \mathbb{S}]$, $L =$ -*, $R =$ -$_*$:

$$\mathrm{Mon}\,([\mathbb{F}, \mathbb{S}]) \text{ and } \mathrm{Mon}\,(\Diamond) \text{ are equivalent categories.}$$

Note that in this case we cannot make a similar observation to the one at the end of the previous subsection. The monad $\Diamond$ is not monoidal, and therefore, we cannot (easily) lift the monoidal structure from profunctors to functors. This is, partly, the symmetry that gets broken from -$_*$ not being strong monoidal.

5 Conclusions and Further Work

The relationship between different interfaces for computational effects has been studied from different perspectives. The purely syntactic approach has been covered by Lindley et al. [11]. A more programmatic account is covered by Haskell's libraries, where the different interfaces are connected by deriving type-class instances. On the other hand, the semantic point of view for the connection was not very developed. Generally, the semantic interpretation for notions of computation focusses only on one of the interfaces, disregarding the connections between them. In this paper we have taken a first step towards the connection from the perspective of *notions of computation as monoids* [19].

As already remarked in the introduction, a direction of future work is to replace the category $\mathbb{F}$ by a general category $\mathbb{C}$ with an injection $i : \mathbb{C} \to \mathbb{S}$, subject to conditions to be determined, and probably similar to those used in *relative monads* [2]. A related problem is to understand these constructions in terms of Freyd categories and their variations [10, 3], as they give a semantics for arrows that do not suffer from size issues.

Acknowledgments. *The author is thankful to Soichiro Fujii, Pierre-Louis Curien, Marcelo Fiore and Ignacio López Franco for discussions on related topics, as well as to Hans-E. Porst and Ross Street for quick clarification on their results and pointing to related bibliography. The author is also grateful to the anonymous reviewers for their comments and suggestions.*

References

[1] Marcelo Aguiar & Swapneel Mahajan (2010): *Monoidal Functors, Species and Hopf Algebras*. CRM monograph series, American Mathematical Society, doi:10.1090/crmm/029.

[2] Thorsten Altenkirch, James Chapman & Tarmo Uustalu (2015): *Monads need not be endofunctors*. *Logical Methods in Computer Science* Volume 11, Issue 1, doi:10.2168/LMCS-11(1:3)2015.

[3] Robert Atkey (2011): *What is a Categorical Model of Arrows?* *Electronic Notes in Theoretical Computer Science* 229(5), pp. 19 – 37, doi:10.1016/j.entcs.2011.02.014. Proceedings of the Second Workshop on Mathematically Structured Functional Programming (MSFP 2008).

[4] Geraldine Brady & Todd H. Trimble (2000): *A categorical interpretation of C.S. Peirce's propositional logic Alpha*. *Journal of Pure and Applied Algebra* 149(3), pp. 213 – 239, doi:10.1016/S0022-4049(98)00179-0.

[5] Marcelo Fiore & Philip Saville (2017): *List Objects with Algebraic Structure*. In Dale Miller, editor: *2nd International Conference on Formal Structures for Computation and Deduction (FSCD 2017)*, *Leibniz International Proceedings in Informatics (LIPIcs)* 84, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, pp. 16:1–16:18, doi:10.4230/LIPIcs.FSCD.2017.16.

[6] P. Gabriel & M. Zisman (1967): *Calculus of fractions and homotopy theory*. Ergebnisse der Mathematik und ihrer Grenzgebiete, Springer-Verlag, doi:10.1007/978-3-642-85844-4.

[7] John Hughes (2000): *Generalising monads to arrows*. *Science of Computer Programming* 37(1), pp. 67 – 111, doi:10.1016/S0167-6423(99)00023-4.

[8] Bart Jacobs, Chris Heunen & Ichiro Hasuo (2009): *Categorical Semantics for Arrows*. *J. Funct. Program.* 19(3-4), pp. 403–438, doi:10.1017/S0956796809007308.

[9] G. M. Kelly (1974): *Doctrinal adjunction*. In Gregory M. Kelly, editor: *Category Seminar*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 257–280, doi:10.1007/BFb0063105.

[10] Paul Blain Levy, John Power & Hayo Thielecke (2003): *Modelling environments in call-by-value programming languages*. *Information and Computation* 185(2), pp. 182 – 210, doi:10.1016/S0890-5401(03)00088-9.

[11] Sam Lindley, Philip Wadler & Jeremy Yallop (2011): *Idioms Are Oblivious, Arrows Are Meticulous, Monads Are Promiscuous*. Electronic Notes Theoretical Computer Science 229(5), pp. 97–117, doi:10.1016/j.entcs.2011.02.018.

[12] F. Loregian (2015): *This is the (co)end, my only (co)friend*. ArXiv e-prints. Available at https://arxiv.org/abs/1501.02503.

[13] Saunders Mac Lane (1971): *Categories for the Working Mathematician*. Graduate Texts in Mathematics 5, Springer-Verlag, doi:10.1007/978-1-4612-9839-7. Second edition, 1998.

[14] Conor McBride & Ross Paterson (2008): *Applicative Programming with Effects*. J. Funct. Program. 18(1), pp. 1–13, doi:10.1017/S0956796807006326.

[15] I. Moerdijk (2002): *Monads on tensor categories*. Journal of Pure and Applied Algebra 168(2), pp. 189 – 208, doi:10.1016/S0022-4049(01)00096-2. Category Theory 1999: selected papers, conference held in Coimbra in honour of the 90th birthday of Saunders Mac Lane.

[16] Craig Pastro & Ross Street (2008): *Doubles for monoidal categories*. Theory and Applications of Categories 21, pp. 61–75. Available at http://www.tac.mta.ca/tac/volumes/21/4/21-04abs.html.

[17] Ruben P. Pieters, Tom Schrijvers & Exequiel Rivas (2017): *Handlers for Non-Monadic Computations (Extended Version)*. Technical Report. Available at http://www.cs.kuleuven.be/publicaties/rapporten/cw/CW713.abs.html. Presented at IFL2017.

[18] Hans-E. Porst & Ross Street (2016): *Generalizations of the Sweedler Dual*. Applied Categorical Structures 24(5), pp. 619–647, doi:10.1007/s10485-016-9450-2.

[19] Exequiel Rivas & Mauro Jaskelioff (2017): *Notions of computation as monoids*. Journal of Functional Programming 27, p. e21, doi:10.1017/S0956796817000132.