Arrows are Strong Monads

Kazuyuki Asada

Research Institute for Mathematical Sciences, Kyoto University,
Kyoto 606-8502, Japan

asada@kurims.kyoto-u.ac.jp

Abstract

Hughes' *arrows* were shown, by Jacobs et al., to be roughly monads in the bicategory $\mathbf{Prof}$ of profunctors (distributors, modules). However in their work as well as others', the categorical nature of the `first` operator was not pursued and its formulation remained rather ad hoc. In this paper, we identify `first` with *strength* for a monad, therefore: *arrows are strong monads in* $\mathbf{Prof}$. Strong monads have been widely used in the semantics of functional programming after Moggi's seminal work, therefore our observation establishes categorical canonicity of the notion of arrow.

Categories and Subject Descriptors F.3.2 [*Logics and Meanings of Programs*]: Semantics of Programming Languages—Denotational semantics

General Terms Languages, Theory

Keywords arrow, strong monad, Freyd category, computational effect, profunctor

1. Introduction

The notion of *arrow* was introduced by Hughes [17, 18] for functional programming languages such as Haskell. Its benefits are claimed to be as follows. First, it provides more natural syntax for Point-free programming; second, it allows programmers to organize a bigger variety of computational effects than the notion of *monad* does.

This paper is about categorical semantics of arrows. There have been roughly two approaches towards that goal. The first one employs *premonoidal categories* with certain additional structure (called *Freyd categories*) [3, 22, 26]. The other is the approach based on *profunctors* (also called *distributors* or *modules*) [4, 6], which we follow.

In the papers [15, 19], Jacobs et al. characterized an arrow as a monoid in a monoidal category $[\mathbb{C}^{\mathrm{op}} \times \mathbb{C}, \mathbf{Set}]$ that is additionally equipped with the so-called `first` operator. The monoidal products in the monoidal category $[\mathbb{C}^{\mathrm{op}} \times \mathbb{C}, \mathbf{Set}]$ are given by certain coends; they are in fact the same as composition of profunctors. Then one readily sees that *an arrow is a monad*—in an *internal* sense—in the bicategory $\mathbf{Prof}$ of profunctors, equipped with a `first` operator.

MSFP 2010 September 25, 2010, Baltimore, Maryland, USA.
Copyright © 2005 ACM [to be supplied]…$10.00

In this way we have got a clean understanding of the categorical nature of arrows—except for the `first` operator. It is our main result that identifies `first` with a *strength* for a monad in $\mathbf{Prof}$. Therefore: *arrows are strong monads in* $\mathbf{Prof}$. The notion of strength for a monad is employed in Moggi's seminal work [24] in order to accommodate "context information" within computational effects—as the `first` operator does for arrows. Our main result (Theorem 14) makes this intuitive analogy a mathematically rigorous one.

Use of profunctors in theoretical computer science is not new: In categorical semantics of programming languages, profunctors—often called modules (or discrete indexed categories)—are sometimes used instead of usual categories [21, 23]. In [7, 16, 25], profunctors are used as models in a domain theory for concurrency. Also potential use of profunctors can be seen in recent work: e.g. [8, 12].

Profunctor can be viewed also as "generalized relation": a profunctor is to a functor what a relation is to a function. Therefore our result is illustrated as:

$$\frac{\text{strong monad}}{\text{arrow}} = \frac{\text{functor}}{\text{profunctor}} = \frac{\text{function}}{\text{relation}} \quad .$$

We define the notions of arrow and of strong monad in $\mathbf{Prof}$ not only over cartesian categories but over any monoidal categories. After identifying arrows as strong monads in $\mathbf{Prof}$, we define the notion of internal strong monad in a general setting, i.e., not only in $\mathbf{Prof}$. From this, we obtain some variants of the notion of arrow.

1.1 Relation to Other Work

`ist` Operator

As we already mentioned, this work follows the profunctor approach that originated in [15, 19]. In these works, they introduced what they called "internal strength", i.e., `ist` operator; then they showed the equivalence between `ist` and `first`. The formulation of `ist` is pretty similar to that of `first`, but has certain technical advantage, which they exploited.

It should be noted that their `ist` operator's definition is quite different from the genuine *internal strength*, i.e., the notion of strength defined for a monad in an internal way in a bicategory. The latter is what we call (internal) *strength* throughout the present paper.

The notion of strength is a monoidal notion: it is defined for any monoidal object as in Section 6, and not only for a cartesian object. On the other hand, the definition of `ist` operator and the proof of the equivalence between `ist` operator and strength (Theorem 16) need the structure of cartesian products. Therefore, `ist` operator is not what should be called strength. However we can prove `ist` operator to be equivalent to strength; the name for `ist` was with fortunate wisdom.

Enriched Freyd Category and Self-enrichment

In [3], Atkey investigated many variants of the notion of arrow. Among them, he gave a categorical definition of arrow and also defined the notion of *enriched Freyd category*, and he showed that they are equivalent.

Atkey's definition of arrow and that given in [15, 19] are different: the former is *(self-)enriched*; the latter is not. This gap between enriched and non-enriched notions was not present for monads (in $\mathbf{Cat}$). This is because in a cartesian closed category—a common setting for functional programming—strong monads are the same thing as enriched monads.

In the present paper, we first give the theorem that arrows are strong monads in $\mathbf{Prof}$ in the non-enriched setting (in Theorem 14), in order to concentrate on other aspects than such self-enrichment problem. Then, in Section 6, we define a "self-enriched" version of strong monad in $\mathbf{Prof}$ in Definition 21, and show that it is equivalent to Atkey's definition of arrow in Theorem 23. One technical advantage of our characterization for arrow is that we can define it over any symmetric monoidal category; while for Atkey's definition, we use structures of cartesian categories (see Definition 22).

Arrows as Relative Monads

In [1], Altenkirch et al. gave a generalized notion of monad, i.e., the notion of *relative monad*. A relative monad is defined over a (base) functor, while a monad is defined over a (base) category. They showed that for a category $\mathbb{C}$, relative monads over the Yoneda embedding $y : \mathbb{C} \longrightarrow \widehat{\mathbb{C}}$ bijectively correspond to $(\mathtt{arr}, \ggg)$-fragments of arrows over $\mathbb{C}$ (in Definition 9), and hence to monads over $\mathbb{C}$ in $\mathbf{Prof}$ (by Theorem 10).

They gave also a *Kleisli construction* and an *Eilenberg-Moore construction* for a relative monad, and showed that the Kleisli construction gives the initial object and the Eilenberg-Moore construction gives the terminal object in some suitable category. On the other hand, also for a monad in $\mathbf{Prof}$, we have the notions of Kleisli object and Eilenberg-Moore object in $\mathbf{Prof}$, as in the paper [27] by Street.

For a relative monad R over the Yoneda embedding y of a category $\mathbb{C}$ and the corresponding monad $\mathcal{A}$ in $\mathbf{Prof}$, the Kleisli category for R is isomorphic to the Kleisli object for $\mathcal{A}$. They are isomorphic to the Kleisli category of the corresponding Freyd category (without premonoidal structures).

One different point between the relative monad approach and our approach in $\mathbf{Prof}$ is that the Eilenberg-Moore category for R is not necessarily equivalent to the Eilenberg-Moore object for $\mathcal{A}$, which is the same as the Kleisli object for $\mathcal{A}$ (because of the duality $\mathbf{Prof} \cong \mathbf{Prof}^{\mathrm{op}}$). (However the "right adjoint" pseudo functor $\widehat{(\text{-})} : \mathbf{Prof} \longrightarrow \mathbf{CAT}$ maps the Eilenberg-Moore object for $\mathcal{A}$ to the Eilenberg-Moore category for R.)

Uustalu, the third author of [1], extended the correspondence between $(\texttt{arr}, \ggg)$-fragments of arrows and relative monads to that between arrows (with first operators) and *strong relative monads* [28].

Terminology

In the paper we use the term *arrow* exclusively for Hughes' notion, and not for arrows between objects in categories. For the latter, we reserve the term *morphism*.

Outline

In Section 2, we briefly recall the notion of arrow. In Section 3, we review the notions of dinatural transformation, end, coend, and profunctor. In Section 4, we recall the results in [19] that a fragment of arrow is a monad in $\mathbf{Prof}$.

In Section 5, we go straight to the main result that arrows are strong monads in $\mathbf{Prof}$. Also we give a direct proof of the equivalence between `ist` operator and strength for a monad in $\mathbf{Prof}$. In Section 6, we commence a more general development in which strong monads in a $\mathbf{Gray}$-monoid are considered. As a benefit of this generalization, we give some variants of the notion of arrow: especially in Section 6.3, we give a "self-enriched" version of strong monad in $\mathbf{Prof}$, which corresponds to Atkey's definition of arrow.

2. Background: the Notion of Arrow

The following review to the notion of arrow is brief. See e.g. [19] for more illustration and examples.

The notion of arrow [17, 18] was introduced as an extension of that of monad [5, 24, 29], and is defined in Haskell as a type constructor class:

$$\begin{aligned}
&\text{class } \texttt{Arrow}\ A \text{ where}\\
&\quad \texttt{arr} :: (s \to t) \to A\,s\,t\\
&\quad (\ggg) :: A\,s\,t \to A\,t\,u \to A\,s\,u\\
&\quad \texttt{first} :: A\,s\,t \to A\,(s,u)\ (t,u)
\end{aligned}$$

where (s,t) is the product type of types s and t.

An instance of `Arrow` must satisfy the following *arrow laws*:

$$\big(a \ggg b\big) \ggg c \;=\; a \ggg \big(b \ggg c\big) \qquad \text{(assoc)}$$

$$\texttt{arr}\ (g \circ f) \;=\; \texttt{arr}\, f \ggg \texttt{arr}\, g \qquad \text{(comp)}$$

$$\texttt{arr}\,\mathrm{id} \ggg a \;=\; a \;=\; a \ggg \texttt{arr}\,\mathrm{id} \qquad \text{(id)}$$

$$\texttt{first}\,a \ggg \texttt{arr}\ (\mathrm{id}\times f) = \texttt{arr}\ (\mathrm{id}\times f) \ggg \texttt{first}\,a \tag{f-din}$$

$$\texttt{first}\,a \ggg \texttt{arr}\,\pi_1 = \texttt{arr}\,\pi_1 \ggg a \tag{f-π}$$

$$\texttt{first}\,a \ggg \texttt{arr}\,\alpha = \texttt{arr}\,\alpha \ggg \texttt{first}\ (\texttt{first}\,a) \tag{f-α}$$

$$\texttt{first}\ (\texttt{arr}\,f) = \texttt{arr}\ (f\times \mathrm{id}) \tag{f-η}$$

$$\texttt{first}\ \big(a \ggg b\big) = \texttt{first}\,a \ggg \texttt{first}\,b \tag{f-μ}$$

where $\pi_1 :: (s,t) \to s,\ \alpha :: (s,(t,u)) \to ((s,t),u)$.

From `first`, we can define its dual:

$$\texttt{second} :: A\,s\,t \to A\,(u,s)\ (u,t)$$

$$\texttt{second}\,a = \texttt{arr}\,\gamma \ggg \texttt{first}\,a \ggg \texttt{arr}\,\gamma$$

where $\gamma :: (s,t) \to (t,s)$.

Arrows subsume monads: given a monad T, we obtain its *Kleisli arrow* $s \to Tt$. Similarly, a comonad S induces its *coKleisli arrow* $Ss \to t$. And we can further fit these together: with a distributive law between a monad and a comonad, we can construct the *biKleisli arrow* $Ss \to Tt$.

3. Preliminaries

In this section we introduce the notion of profunctor. Before that, yet, we first recall some necessary notions related to profunctors.

3.1 End, Coend and Yoneda Lemmas

Here we recall the notions of dinatural transformation, end and coend; the two forms of Yoneda lemma, i.e., end form and coend form; and Reversing Lemma. For further details, see [20] (especially, Sec. 3.10).

Like homset functors $\mathbb{C}(-,-) : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Set}$ or exponentiation bifunctors on SMCCs, we sometimes come across bifunctors with mixed variance. Dinatural transformations are between such contra-co-variant bifunctors $F, G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$, while natural transformations are between functors $F, G : \mathbb{C} \to \mathbb{D}$.

Definition 1 (Dinatural transformation) For categories $\mathbb{C}, \mathbb{D}$, and functors $F, G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$, a *dinatural transformation* α from F to G is an indexed family of morphisms

$$\alpha_C : F(C, C) \to G(C, C) \qquad (C \in |\mathbb{C}|)$$

which is *dinatural in* C, i.e., for any morphism $f : C \to C'$ the following diagrams commutes:

$$\begin{tikzcd}
 & F(C,C) \arrow[r, "\alpha_C"] & G(C,C) \arrow[dr, "G(C,f)"] & \\
F(C',C) \arrow[ur, "F(f,C)"] \arrow[dr, "F(C',f)"'] & & & G(C,C') \\
 & F(C',C') \arrow[r, "\alpha_{C'}"'] & G(C',C') \arrow[ur, "G(f,C')"'] &
\end{tikzcd}$$

Two successive dinatural transformations do not necessarily compose, in which sense this notion is not that of “morphism” between such bifunctors. Nevertheless dinatural transformations are useful, as we will see many times in the paper.

Note that the notion of dinatural transformation subsume that of natural transformation: for given functors $F, G : \mathbb{C} \to \mathbb{D}$, natural transformations between F, G bijectively correspond to dinatural transformations between $F \circ \pi', G \circ \pi' : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$. Also we will see many examples of dinatural transformations where the domain functors (F above) or the codomain functors (G above) are constant functors, as in the next definitions of end and coend.

End (resp. coend) is a kind of limit (resp. colimit) where dinatural transformations are used instead of natural transformations. These notions play a very important role for profunctor theory.

Definition 2 (End and Coend) For categories $\mathbb{C}, \mathbb{D}$, and a functor $F : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$,

- an *end of* F is an object $\int_{C \in \mathbb{C}} F(C, C)$ in $\mathbb{D}$ with *projection* morphisms

$$\pi_{C'} : \left(\int_{C \in \mathbb{C}} F(C, C) \right) \to F\left(C', C'\right) \qquad \left(C' \in |\mathbb{C}|\right)$$

  which form a dinatural transformation from the $\int_{C \in \mathbb{C}} F(C, C)$-constant functor to the functor F. Then these are required to be universal among such data: i.e., for any object D in $\mathbb{D}$ and any dinatural transformation $(\alpha_{C'} : D \to F(C', C'))_{C'}$,there is a unique morphism $f : D \to \int_{C \in \mathbb{C}} F(C, C)$ such that $\pi_{C'} \circ f = \alpha_{C'}$ for all C'.
- Dually, a *coend of* F is an object $\int^{C \in \mathbb{C}} F(C, C)$ in $\mathbb{D}$ with *injection* morphisms

$$\iota_{C'} : F\left(C', C'\right) \to \int^{C \in \mathbb{C}} F(C, C) \qquad \left(C' \in |\mathbb{C}|\right)$$

  which forms an end of $F^{\mathrm{op}} : (\mathbb{C}^{\mathrm{op}})^{\mathrm{op}} \times \mathbb{C}^{\mathrm{op}} \to \mathbb{D}^{\mathrm{op}}$. $\square$

The universality of an end and a coend can be written respectively as the following bijective correspondences:

$$\begin{array}{c} f : D \to \int_{C \in \mathbb{C}} F(C,C) \\ \hline\hline \left(D \overset{f_C}{\to} F(C,C)\right)_C \text{ dinatural in } C \end{array} \tag{1}$$

$$\begin{array}{c} f : \int^{C \in \mathbb{C}} F(C,C) \to D \\ \hline\hline \left(F(C,C) \overset{f_C}{\to} D\right)_C \text{ dinatural in } C \end{array} \tag{2}$$

The complete and cocomplete category $\mathbf{Set}$ has all ends and coends.

A typical and important example of an end is a set of dinatural transformations:

Lemma 3 *For a small category $\mathbb{C}$ and functors $F, G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Set}$, we have the bifunctor $[F(+,-), G(-,+)] : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Set}$, where $+$ and $-$ indicate respectively variant and covariant arguments, and $[-,+]$ is the exponentiation of $\mathbf{Set}$.*

Then, denoting the set of all dinatural transformations from F to G by $\mathbf{Dinat}(F,G)$, we have a canonical isomorphism

$$\mathbf{Dinat}(F,G) \cong \int_{C \in \mathbb{C}} \left[F(C,C), G(C,C)\right].$$

Proof. Because of the following bijective correspondences:

$$\begin{array}{c} 1 \to \int_{C \in \mathbb{C}} \left[F\big(C,C\big), G(C,C)\right] \\ \hline\hline \left(1 \to \left[F(C,C), G(C,C)\right]\right)_C \text{ dinatural in } C \\ \hline\hline \end{array} \quad \text{(by (1))}$$

$$\left(F\left(C,C\right)\to G\left(C,C\right)\right)_C \text{ dinatural in } C$$

As noted above, natural transformations are subsumed by dinatural transformations. Hence, for functors $F, G : \mathbb{C} \to \mathbf{Set}$, we have also a canonical isomorphism

$$\mathbf{Nat}\left(F,G\right) \cong \int_{C\in\mathbb{C}} \left[FC, GC\right]$$

where the left hand side is the set of natural transformations from F to G. These isomorphisms are useful throughout the paper for calculating (di)natural transformations, especially in the proof of Theorem 14.

By the usual Yoneda lemma, for a functor $F : \mathbb{C} \to \mathbf{Set}$, there is canonically the natural isomorphism $FC \cong \mathbf{Nat}\left(y\left(C\right), F\right)$. Replacing the right hand side with the end isomorphic to it, we have the Yoneda lemma, end-form:

Lemma 4 *(The Yoneda lemma, end-form) For a small category* $\mathbb{C}$ *and a functor* $F : \mathbb{C} \to \mathbf{Set}$, *we have a canonical natural isomorphism*

$$FC \cong \int_{C'\in\mathbb{C}} \left[\mathbb{C}\left(C, C'\right), FC'\right].$$

Above, we use the exponentiation $[-,-]$ in $\mathbf{Set}$. This is called also *cotensor* in enriched category theory [20], and there is its dual notion called *tensor*, which is just cartesian product in the case of $\mathbf{Set}$. With this we can obtain the dual version of the above Yoneda Lemma:

Lemma 5 *(The Yoneda lemma, coend-form) For a small category*

$\mathbb{C}$ *and a functor* $F : \mathbb{C} \to \mathbf{Set}$, *we have a canonical natural isomorphism*

$$\int^{C' \in \mathbb{C}} F(C') \times \mathbb{C}(C', C) \cong FC \,.$$

Note that we can of course apply the above two lemmas also to a contravariant functor $F : \mathbb{C}^{\mathrm{op}} \to \mathbf{Set}$.

We need also the next lemma in the proof of the main theorem. As homset functors reverse colimits into limits in the negative position, they reverse also coends into ends.

Lemma 6 *(Reversing Lemma) For a category* $\mathbb{C}$ *and a functor* $F : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Set}$, *we have a canonical natural isomorphism*

$$\left[\int^{C \in \mathbb{C}} F(C, C), -\right] \cong \int_{C \in \mathbb{C}} [F(C, C), -] \,.$$

3.2 Profunctor

The notion of *profunctor* (also called *distributor* or *module*) forms the very basis of our analysis of arrows. The notion dates back for quite a while, and its relevance to theoretical computer science has been recently recognized [8, 12].

Here we recall the notion of profunctor, the bicategory $\mathbf{Prof}$, the embedding of $\mathbf{Cat}$ into $\mathbf{Prof}$, and tensor product in $\mathbf{Prof}$.

As mentioned in the Introduction, a profunctor is to a functor what a relation is to a function. This analogy is used repeatedly for illustration. For further details and illustrations for profunctors, see [4, 6].

Definition 7 (Profunctor) Let $\mathbb{C}$ and $\mathbb{D}$ be small categories. A *profunctor* from $\mathbb{C}$ to $\mathbb{D}$ is a functor $\mathbb{D}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbf{Set}$. We denote such a profunctor by $\mathcal{F} : \mathbb{C} \;\mathrel{\mkern3mu\vcenter{\hbox{$\scriptscriptstyle+$}}\mkern-12mu{\to}}\; \mathbb{D}$, i.e.,

$$\frac{\mathcal{F} : \mathbb{C} \;\mathrel{\mkern3mu\vcenter{\hbox{$\scriptscriptstyle+$}}\mkern-12mu{\to}}\; \mathbb{D}, \quad \text{a profunctor}}{\mathcal{F} : \mathbb{D}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbf{Set}, \quad \text{a functor}} \tag{3}$$

For successive profunctors $\mathcal{F} : \mathbb{C} \;\mathrel{\mkern3mu\vcenter{\hbox{$\scriptscriptstyle+$}}\mkern-12mu{\to}}\; \mathbb{D}$ and $\mathcal{G} : \mathbb{D} \;\mathrel{\mkern3mu\vcenter{\hbox{$\scriptscriptstyle+$}}\mkern-12mu{\to}}\; \mathbb{E}$, their *composition* $\mathcal{G} \circ \mathcal{F} : \mathbb{C} \;\mathrel{\mkern3mu\vcenter{\hbox{$\scriptscriptstyle+$}}\mkern-12mu{\to}}\; \mathbb{E}$ is defined—under the correspondence (3)—as the following functor:

$$\begin{array}{ccc} \mathbb{E}^{\mathrm{op}} \times \mathbb{C} & \xrightarrow{\mathcal{G} \circ \mathcal{F}} & \mathbf{Set} \\ (E, C) & \longmapsto & \int^{D \in \mathbb{D}} \mathcal{G}(E, D) \times \mathcal{F}(D, C) \end{array} \tag{4}$$

Its action on morphisms is the obvious one.

For a category $\mathbb{C}$, we define the *identity* profunctor

$$\mathrm{Id}^{+}_{\mathbb{C}} : \mathbb{C} \;\mathrel{\mkern3mu\vcenter{\hbox{$\scriptscriptstyle+$}}\mkern-12mu{\to}}\; \mathbb{C} \quad \text{by} \quad \mathbb{C}(-,-) : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbf{Set}\,. \tag{5}$$

Given two parallel profunctors $\mathcal{F}, \mathcal{F}' : \mathbb{C} \;\mathrel{\mkern3mu\vcenter{\hbox{$\scriptscriptstyle+$}}\mkern-12mu{\to}}\; \mathbb{D}$, a *2-cell between profunctors* $\sigma : \mathcal{F} \Longrightarrow \mathcal{F}'$ is just a natural transformation $\sigma : \mathcal{F} \Longrightarrow \mathcal{F}'$ where $\mathcal{F}$ and $\mathcal{F}'$ are regarded as the functors from $\mathbb{D}^{\mathrm{op}} \times \mathbb{C}$ to $\mathbf{Set}$. $\square$

Now, let us illustrate the definition above using the analogy with relations. A relation R between sets S and T is a subset of the product set $S \times T$, in other words, a function from $S \times T$ to the two points set $\{0, 1\}$. Thus, a profunctor—$\mathcal{F} : \mathbb{D}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbf{Set}$—is a $\mathbf{Set}$-many valued relation, while a relation is two valued. Next let us recall how the relational composition $S \circ R$ is defined:

$$(S \circ R)(c,e) \quad \overset{\text{def}}{\Longleftrightarrow} \quad \exists d.\ R(c,d) \wedge S(d,e)$$

Then there are obvious similarity between composition for profunctors (4) and that for relations: coend $\int^D$ corresponds to $\exists d.$, and $\times$ to $\wedge$. Also, 2-cells σ between profunctors $\mathcal{F}$ and $\mathcal{F}'$, i.e., natural transformations $\left(\sigma_{D,C} : \mathcal{F}(D,C) \longrightarrow \mathcal{F}'(D,C)\right)_{D,C}$ correspond to the inclusion order between relations:

$$R \leq R' \overset{\text{def}}{\Longleftrightarrow} \forall c.d. \left(R(c,d) \Rightarrow R'(c,d)\right) \quad .$$

This analogy between profunctor and relation will be helpful to understand many notions for profunctor.

The notion of 2-category is now used in many different contexts; in a 2-category, there is a notion of 2-cell, which is "morphism between morphisms." A typical example is the 2-category $\mathbf{Cat}$ of categories, functors, and natural transformations.

One might imagine that the notions of category, profunctor between categories, and 2-cell between profunctors form a 2-category. However, the composition for profunctors is not strictly associative, because of the coends and products in the definition. Likewise, identity profunctors are not strictly unital. Nevertheless, they hold up to canonical iso-2-cells; e.g., the iso-2-cells for the unitality can be gotten by the Yoneda lemma, coend-form. Such notion which is similar to that of 2-category, but whose associativity and unitality of the composition of 1-cells are required just up to iso-2-cells, is called *bicategory* [6].

We denote by $\mathbf{Prof}$ the bicategory consisting of small categories as its 0-cells (objects), profunctors as its 1-cells (morphisms), and 2-cells between profunctor as its 2-cells.

Next we introduce an embedding of $\mathbf{Cat}$ into $\mathbf{Prof}$. This is identity on 0-cells (small categories), maps a functor $F : \mathbb{C} \longrightarrow \mathbb{D}$ to the following profunctor $F_* : \mathbb{C} \nrightarrow \mathbb{D}$ called *direct image of* F,

$$F_* (-, +) \overset{\text{def}}{=} \mathbb{D} (-, F+) \ : \ \mathbb{D}^{\text{op}} \times \mathbb{C} \longrightarrow \mathbf{Set} \qquad (6)$$

and maps a natural transformation $\sigma : F \Longrightarrow F' : \mathbb{C} \longrightarrow \mathbb{D}$ to the natural transformation $\mathbb{D} (\text{id}, \sigma) : F_* \Longrightarrow F'_* : \mathbb{C} \nrightarrow \mathbb{D}$.

This embedding forms a *pseudofunctor* [6] from the 2-category $\mathbf{Cat}$ to the bicategory $\mathbf{Prof}$, i.e. preserve composition and identities for 1-cells up to iso-2-cells. This embedding corresponds—via the foregoing analogy—to the embedding of the category $\mathbf{Set}$ into the category $\mathbf{Rel}$ of sets and relations, which maps a function to its graph relation.

As cartesian products in $\mathbf{Set}$ lift to tensor products in $\mathbf{Rel}$, cartesian products in $\mathbf{Cat}$ lift to *tensor products* in $\mathbf{Prof}$. For profunctors $\mathcal{F} : \mathbb{C} \nrightarrow \mathbb{C}'$ and $\mathcal{G} : \mathbb{D} \nrightarrow \mathbb{D}'$, we define $\mathcal{F} \times \mathcal{G} : \mathbb{C} \times \mathbb{D} \nrightarrow \mathbb{C}' \times \mathbb{D}'$ by the following:

$$(\mathcal{F} \times \mathcal{G}) \left(C', D', C, D\right) \overset{\text{def}}{=} \mathcal{F} \left(C', C\right) \times \mathcal{G} \left(D', D\right) \qquad (7)$$

It is obvious that the operator $\times$ acts also on natural transformations between profunctors.

4. Monad in Prof

In this short section, we review the result by Jacobs et al. [15, 19] that monads in $\mathbf{Prof}$ are equivalent to the (`arr`,$\ggg$)-fragment of arrows. This forms also a preliminary step toward our main result, Theorem 14.

4.1 Monads in $\mathbf{Prof}$ as Bases of Arrows

Many notions in category theory consist of functors and natural transformations, subject to some commutative diagrams. For example, an *adjunction* is a tuple (F, G, η, ϵ) of two functors and two natural transformations, subject to the two triangular laws [6, Theorem 3.1.5]. Once one takes a 2-categorical view on this—functors and natural transformations are 1-cells and 2-cells in $\mathbf{Cat}$—it is straightforward to define an *adjunction* in an arbitrary 2-category, or in a bicategory, in an "internal" way [27]. The notion of (internal) monad is also one of such.

Definition 8 (Monad) Let $\mathscr{P}$ be a bicategory, and $\mathbb{C}$ be a 0-cell in $\mathscr{P}$. A *monad over* $\mathbb{C}$ *in* $\mathscr{P}$ is a triple $(\mathcal{A}, \eta, \mu)$ of an endo-1-cell $\mathcal{A} : \mathbb{C} \longrightarrow \mathbb{C}$ in $\mathscr{P}$ and 2-cells $\eta : \mathrm{Id}_{\mathbb{C}} \Longrightarrow \mathcal{A}$ and $\mu : \mathcal{A}^2 \Longrightarrow \mathcal{A}$ in $\mathscr{P}$ satisfying the usual commutative diagrams of associativity and unitality. $\square$

It should be noted that this internal definition coincides with the usual one if $\mathscr{P} = \mathbf{Cat}$.

Definition 9 ((`arr`,$\ggg$)-Fragment of Arrow) For a category $\mathbb{C}$, an *(`arr`,$\ggg$)-fragment of arrow* over $\mathbb{C}$ consists of a mapping $\mathrm{Ar} : |\mathbb{C}| \times |\mathbb{C}| \longrightarrow |\mathbf{Set}|$ and two families of mappings:

$$\begin{aligned} \mathtt{arr}_{AB} &: \mathbb{C}(A, B) \longrightarrow \mathrm{Ar}(A, B) \\ \ggg_{ABC} &: \mathrm{Ar}(A, B) \times \mathrm{Ar}(B, C) \longrightarrow \mathrm{Ar}(A, C) \end{aligned}$$

These must satisfy Axioms (assoc), (comp), and (id) in Section 2, where we use variables f, g for morphisms in $\mathbb{C}$, and use a, b, c for elements in $(\mathrm{Ar}(A, B))_{A,B}$. $\square$

The map Ar above is a mapping to $|\mathbf{Set}|$ rather than $|\mathbb{C}|$. This is unnatural if we use the above definition of arrows to model arrows as type constructors in Section 2. This involves a subtle size issue, and we solve it in Section 6.3, till then we shall separate such a size issue. Here and in the next section, we focus on the correspondence between arrows and strong monads in $\mathbf{Prof}$.

Theorem 10 *[15, 19] For a small category $\mathbb{C}$, the notion of monad over $\mathbb{C}$ in $\mathbf{Prof}$ is equivalent to that of (`arr`,$\ggg$)-fragment of arrow over $\mathbb{C}$.* $\square$

Proof. (Sketch) Given a monad $(\mathcal{A}, \eta, \mu)$ in $\mathbf{Prof}$ over a small category $\mathbb{C}$, $\mathcal{A}$ is a functor from $\mathbb{C}^{\mathrm{op}} \times \mathbb{C}$ to $\mathbf{Set}$; this corresponds to the mapping Ar of a fragment of arrow, where the functoriality of $\mathcal{A}$ is recovered with `arr` and $\ggg$. The natural transformations η has components $\eta_{A,B} : \mathbb{C}\,(A, B) \longrightarrow \mathcal{A}\,(A, B)$ since $\mathbb{C}(-, -)$ is an identity 1-cell in $\mathbf{Prof}$. This corresponds to `arr`. Finally, μ is, by (4), a natural transformation with components

$$\mu_{A,C} : \Big(\int^{B} \mathcal{A}\,(A, B) \times \mathcal{A}\,(B, C)\Big) \longrightarrow \mathcal{A}\,(A, C) \ ,$$

which corresponds to $\ggg$ by the universality (2) of the coend. $\square$

In the remaining sections, we use `arr`, $\ggg$ and η, μ interchangeably, especially in Definition 12.

5. Arrows as Strong Monads in Prof

In this section, we present our main result (Theorem 14) that arrows are strong monads in $\mathbf{Prof}$. We give this theorem not only for cartesian categories, but for any monoidal categories.

In order to show the main result, we first need to define the notion of strength in $\mathbf{Prof}$. Then we generalize the definition of `first` operator, from that for cartesian categories to that for monoidal categories. Finally we show that these notions of strength and `first` are equivalent.

5.1 Strength in $\mathbf{Prof}$

First let us review strength in $\mathbf{Cat}$. A strength for a monad over a monoidal category $(\mathbb{C}, \otimes, \mathrm{I})$ is a natural transformation $\mathtt{str}_{A,B} : TA{\otimes}B \longrightarrow T\,(A{\otimes}B)$ satisfying certain axioms. This 2-cell in $\mathbf{Cat}$ can be also described as the following:

$$\begin{tikzcd}
\mathbb{C} \times \mathbb{C} \arrow[r, "T\times \mathrm{Id}_{\mathbb{C}}"] \arrow[d, "\otimes"'] \arrow[dr, phantom, "\Downarrow \mathtt{str}"] & \mathbb{C} \times \mathbb{C} \arrow[d, "\otimes"] \\
\mathbb{C} \arrow[r, "T"'] & \mathbb{C}
\end{tikzcd}$$

In a similar way, we define the notion of strength in $\mathbf{Prof}$.

Definition 11 (Strength) Let $(\mathbb{C}, \otimes, \mathrm{I}, \alpha, \lambda, \rho)$ be a monoidal category, and $(\mathcal{A}, \eta, \mu)$ be a monad over $\mathbb{C}$ in $\mathbf{Prof}$.

A *strength* for the monad $\mathcal{A}$ in $\mathbf{Prof}$ is a 2-cell `str` in $\mathbf{Prof}$ in the following diagram such that it satisfies the usual four axioms with α, ρ, η, μ (for these axioms, see Definition 20).

$$\begin{tikzcd}
\mathbb{C} \times \mathbb{C} \arrow[r, "\mathcal{A}\times \mathrm{Id}^{+}_{\mathbb{C}}", "\shortmid" marking] \arrow[d, "\otimes_*"', "\shortmid" marking] \arrow[dr, phantom, "\Downarrow \mathtt{str}"] & \mathbb{C} \times \mathbb{C} \arrow[d, "\otimes_*", "\shortmid" marking] \\
{} & {}
\end{tikzcd}$$

$$\begin{tikzcd} \mathbb{C} \arrow[r, "\mathcal{A}"', "|" marking] & \mathbb{C} \end{tikzcd}$$

We call a monad in $\mathbf{Prof}$ equipped with a strength in $\mathbf{Prof}$ a *strong monad in* $\mathbf{Prof}$. $\square$

This is a key notion in the paper. It should be noted that since the two vertical 1-cells in the above diagram are in $\mathbf{Prof}$, we use the embedding (6) of $\mathbf{Cat}$ into $\mathbf{Prof}$, so that the functor $\otimes : \mathbb{C} \times \mathbb{C} \longrightarrow \mathbb{C}$ is replaced with its direct image $\otimes_*$.

5.2 `first` Operator for Monoidal Categories

To give the main theorem for any monoidal categories, here we generalize the definition of `first` operators from that for cartesian categories to that for monoidal categories. (This modification is used in the proof of the main result. It is not an indirect way but a natural way even for cartesian categories; a cartesian category seen in $\mathbf{Prof}$ is not a "cartesian object" but a monoidal object (see Section 6) in $\mathbf{Prof}$.)

In order to give a categorical definition of `first` operators for monads in $\mathbf{Prof}$ over monoidal categories, we need to modify Axiom (f-π) (described in Section 2), since a monoidal category does not necessarily have projections π. For this, we use $\rho : A \otimes \mathrm{I} \longrightarrow A$ instead of π:

Definition 12 [`first` Operators for Monoidal Categories] For a monoidal category $(\mathbb{C}, \otimes, \mathrm{I}, \alpha, \lambda, \rho)$ and a monad $(\mathcal{A}, \eta, \mu)$ over $\mathbb{C}$ in $\mathbf{Prof}$, a `first` *operator* is a family of morphisms $\mathtt{first}_{A,B,C} : \mathcal{A}(A, B) \longrightarrow \mathcal{A}(A \otimes C, B \otimes C)$ natural in $A, B \in \mathbb{C}$, dinatural in $C \in \mathbb{C}$, and satisfying Axioms (f-α), (f-η), and (f-μ) in Section 2 and (f-ρ):

$$\texttt{first}\, a \ggg \texttt{arr}\, \rho \;=\; \texttt{arr}\, \rho \ggg a \tag{f-ρ}$$

Note that the naturality in A and B above are redundant: they are derived from Axioms (f-η) and (f-μ). Also note that Axiom (f-din) in Section 2 is equivalent to the dinaturality above.

The propriety of the above definition is justified by the next proposition:

Proposition 13 *In the situation of Definition 12, if the monoidal category $\mathbb{C}$ is a cartesian category, then Axiom* (f-ρ) *is equivalent to Axiom* (f-π). $\square$

Proof. The direction from (f-π) to (f-ρ) is trivial. For the converse, Axiom (f-π) is gotten from (f-ρ) as the following:

$$\begin{tikzcd}
& \mathcal{A}(A\times C, B\times C) \arrow[rr, "\mathcal{A}(A\times C, B\times !_C)"] \arrow[dr, phantom, "\text{(dinaturality)}"] & & \mathcal{A}(A\times C, B\times 1) \arrow[dd, "\mathcal{A}(A\times C, \rho)"] \\
\mathcal{A}(A, B) \arrow[ur, "\texttt{first}"] \arrow[rr, "\texttt{first}"] \arrow[dr, "\mathcal{A}(\rho, B)"'] & & \mathcal{A}(A\times 1, B\times 1) \arrow[ur, "\mathcal{A}(A\times !_C, B\times 1)"] \arrow[dl, "\mathcal{A}(A\times 1, \rho)"] \arrow[dr, phantom, "\text{(functoriality)}"] & \\
& \mathcal{A}(A\times 1, B) \arrow[rr, "\mathcal{A}(A\times !_C, B)"'] \arrow[ur, phantom, "\text{(f-}\rho\text{)}"] & & \mathcal{A}(A\times C, B)
\end{tikzcd}$$

In the above, $!_C : C \longrightarrow 1$ is the unique map from C to the terminal, hence $\rho \circ A \times !_C : A \times C \longrightarrow A \times 1 \longrightarrow A$ is equal to the projection $\pi_1 : A \times C \longrightarrow A$. The upper path of the above diagram is the left hand side of (f-π), and the lower path is the right hand side. $\square$

5.3 Arrows as Strong Monads in $\mathbf{Prof}$

Now we show the main result.

Theorem 14 *For a monoidal small category* $(\mathbb{C}, \otimes, \mathrm{I}, \alpha, \lambda, \rho)$ *and a monad* $(\mathcal{A}, \eta, \mu)$ *on* $\mathbb{C}$ *in* $\mathbf{Prof}$*, there is a bijective correspondence between* `first` *operators and strengths for* $(\mathcal{A}, \eta, \mu)$ *in* $\mathbf{Prof}$.
Proof. The following calculation of ends and coends shows the correspondence between strength (in LHS) and `first` (in RHS):

$$\mathbf{Nat}\Big((\otimes_*) \circ \big(\mathcal{A} \times \mathrm{Id}^+_{\mathbb{C}}\big)\,,\ \mathcal{A} \circ (\otimes_*)\Big)$$

$$\cong \qquad\qquad \text{(by Lemma 3)}$$

$$\int_{A,B,C} \Big[\Big(\otimes_* \circ \big(\mathcal{A} \times \mathrm{Id}^+_{\mathbb{C}}\big)\Big)\big(C, (A,B)\big),\ \Big(\mathcal{A} \circ \otimes_*\Big)\big(C, (A,B)\big)\Big]$$

$$= \qquad\qquad \text{(by Definition 7 - (4))}$$

$$\int_{A,B,C} \Bigg[\int^{A',B'} \Big(\otimes_*\big(C, (A',B')\big)\Big) \times \Big(\big(\mathcal{A} \times \mathrm{Id}^+_{\mathbb{C}}\big)\big((A',B'), (A,B)\big)\Big)\,,$$
$$\int^{C'} \mathcal{A}\big(C, C'\big) \times \Big(\otimes_*\big(C', (A,B)\big)\Big) \ \Bigg]$$

$$= \quad \text{(by the definitions of } (-)_* \text{ (6), tensor } \times \text{ (7), and } \mathrm{Id}^+_{\mathbb{C}} \text{ (5))}$$

$$\int_{A,B,C} \left[\int^{A',B'} \mathbb{C}\left(C, A' \otimes B'\right) \times \mathcal{A}\left(A', A\right) \times \mathbb{C}\left(B', B\right), \int^{C'} \mathcal{A}\left(C, C'\right) \times \mathbb{C}\left(C', A \otimes B\right) \right]$$

$$\cong \qquad \text{(by Lemma 5 on } B', C' \text{)}$$

$$\int_{A,B,C} \left[\int^{A'} \mathbb{C}\left(C, A' \otimes B\right) \times \mathcal{A}\left(A', A\right), \quad \mathcal{A}\left(C, A \otimes B\right) \right]$$

$$\cong \qquad \text{(by Lemma 6, then Currying)}$$

$$\int_{A,B,C,A'} \left[\mathbb{C}\left(C, A' \otimes B\right), \left[\mathcal{A}\left(A', A\right), \mathcal{A}\left(C, A \otimes B\right) \right] \right]$$

$$\cong \qquad \text{(by Lemma 4 on } C \text{)}$$

$$\int_{A,B,A'} \left[\mathcal{A}\left(A', A\right), \mathcal{A}\left(A' \otimes B, A \otimes B\right) \right]$$

$$\cong \qquad \text{(by Lemma 3)}$$

$$\mathbf{Nat}_{A',A} \mathbf{Dinat}_B \Big(\mathcal{A}\left(A', A\right), \mathcal{A}\left(A' \otimes B, A \otimes B\right) \Big)$$

The remaining is to prove the equivalences between the axioms for strength and those for `first`: the four axioms for strength with α, ρ, η, and μ correspond respectively to (f-α), (f-ρ), (f-η), and (f-μ). The proofs of these equivalences are tedious but straightforward. $\square$

In this way we can characterize simply the notion of arrow only with basic structures of the bicategory $\mathbf{Prof}$. This simplicity and rich structures of $\mathbf{Prof}$ make it easy to calculate, in $\mathbf{Prof}$, properties and structures on arrows, as in the paper [2]. Besides, this characterization provides a better justification of the axioms of arrows, and provides us clean understanding of the notion of arrow.

5.4 On `ist` Operators

On a cartesian category as a base of an arrow, there is an alternative to a `first` operator, i.e., an `ist` operator [15, 19].

Applying the theorem in the previous subsection to cartesian small categories, we see that strengths are equivalent to `first` operators, hence also to `ist` operators. However, we give a direct proof of correspondences between strengths in $\mathbf{Prof}$ and `ist` operators. The proof shows how the notion of `ist` operator depends on cartesian products.

Definition 15 (`ist` Operator) For a cartesian category $\mathbb{C}$ and a monad $(\mathcal{A}, \eta, \mu)$ over $\mathbb{C}$ in $\mathbf{Prof}$, an *ist operator* is a family of morphisms $\mathtt{ist}_{A,B} : \mathcal{A}(A, B) \longrightarrow \mathcal{A}(A, B \times A)$ natural in $B \in \mathbb{C}$, dinatural in $A \in \mathbb{C}$, and satisfying the following axioms:

$$\begin{aligned}
\mathtt{ist}\,(\mathtt{arr}\, f) &= \mathtt{arr}\,\langle f, \mathrm{id}\rangle \\
\mathtt{ist}\, a \ggg \mathtt{arr}\, \pi_1 &= a \\
\mathtt{ist}\,\big(a \ggg b\big) &= \\
&\quad \mathtt{ist}\, a \ggg \mathtt{ist}\,\big(\mathtt{arr}\, \pi_1 \ggg b\big) \ggg \mathtt{arr}\,(\mathrm{id} \times \pi_2) \\
\mathtt{ist}\,(\mathtt{ist}\, a) &= \mathtt{ist}\, a \ggg \mathtt{arr}\,\langle \mathrm{id}, \pi_2\rangle
\end{aligned}$$

Theorem 16 *For a cartesian small category $\mathbb{C}$, and a monad $(\mathcal{A}, \eta, \mu)$ on $\mathbb{C}$ in $\mathbf{Prof}$, there is a bijective correspondence between* `ist` *operators and strengths for $(\mathcal{A}, \eta, \mu)$ in $\mathbf{Prof}$.* $\square$

Proof. Here we concentrate only on the correspondence between (di)natural transformations of `ist` operators (in RHS) and strengths (in LHS):

$$\mathbf{Nat}\Big((\times_*) \circ \left(\mathcal{A} \times \mathrm{Id}_{\mathbb{C}}^{+}\right),\ \mathcal{A} \circ (\times_*)\Big)$$

$\cong$ (by Lemma 3 and definitions, as in the proof of Theorem 14)

$$\int_{A,B,C} \left[\int^{A',B'} \mathbb{C}\left(C, A'{\times}B'\right) \times \mathcal{A}\left(A', A\right) \times \mathbb{C}\left(B', B\right), \right.$$
$$\left. \int^{C'} \mathcal{A}\left(C, C'\right) \times \mathbb{C}\left(C', A{\times}B\right) \right]$$

$\cong$ (by the adjointness of the cartesian product $A'{\times}B'$)

$$\int_{A,B,C} \left[\int^{A',B'} \mathbb{C}\left(C, A'\right) \times \mathbb{C}\left(C, B'\right) \times \mathcal{A}\left(A', A\right) \times \mathbb{C}\left(B', B\right), \right.$$
$$\left. \int^{C'} \mathcal{A}\left(C, C'\right) \times \mathbb{C}\left(C', A{\times}B\right) \right]$$

$\cong$ (by Lemma 5 on A', B', C')

$$\int_{A,B,C} \left[\mathcal{A}(C, A) \times \mathbb{C}\left(C, B\right), \mathcal{A}(C, A{\times}B)\right]$$

$\cong$ (by Currying)

$$\int_{A,B,C} \Big[\mathbb{C}\,(C,B)\,,\big[\mathcal{A}\,(C,A)\,,\mathcal{A}\,(C,A{\times}B)\big]\Big]$$

$\cong$ (by Lemma 4 on B)

$$\int_{A,C} \big[\mathcal{A}\,(C,A)\,,\mathcal{A}\,(C,A{\times}C)\big]$$

$\cong$ (by Lemma 3)

$$\mathbf{Dinat}_{A,C}\Big(\mathcal{A}\,(C,A)\,,\ \mathcal{A}\,(C,A{\times}C)\Big)$$

The steps are almost the same as in the proof of Theorem 14, except that we used adjointness of cartesian products for the second isomorphism. □

6. Generalizing Arrow

In this section, we generalize the definition of strong monad by enlarging a class of ambient bicategories in which internal strong monad is defined. So far in the paper, such an ambient bicategory is $\mathbf{Cat}$ or $\mathbf{Prof}$. We see first which kind of bicategories can be used for such purpose; then we define the notion of strength in there.

By this generalization we obtain some variants of the notion of arrow. Especially in the last, we give a definition of “self-enriched” version of arrow, which is shown to be equivalent to Atkey’s definition of arrow.

6.1 Gray Monoid

Monads in $\mathbf{Cat}$ are defined over any category, while strong monads in $\mathbf{Cat}$ are defined only over monoidal categories. Hence when we define strong monads in bicategories, we first have to define *monoidal objects* in bicategories.

Much like we need monoidal categories as ambient categories in which we define monoid objects, we need *monoidal bicategories* [13] as ambient bicategories in which we define monoidal objects. (These phenomena of occurrences of similar kinds of inner and outer structures are called "the microcosm principles", advocated by Baez and Dolan [14].)

We can in fact use monoidal bicategories to define monoidal objects. However, the structural isomorphisms of a monoidal bicategory—up to which the composition of 1-cells and also the monoidal product are associative and unital—are cumbersome and make the essence blurred. Therefore we here use the notion of $\mathbf{Gray}$ monoid, which is, roughly, "strictified monoidal bicategory".

Definition 17 ($\mathbf{Gray}$ monoid) A $\mathbf{Gray}$ *monoid* $\mathscr{P}$ is a 2-category with the following structures:

- a 0-cell $\mathbb{I}$,
- for each 0-cell $\mathbb{C}$ in $\mathscr{P}$, two 2-functors $\mathscr{L}_{\mathbb{C}}, \mathscr{R}_{\mathbb{C}} : \mathscr{P} \longrightarrow \mathscr{P}$ satisfying the following conditions

$$\mathscr{L}_{\mathbb{C}}(\mathbb{D}) = \mathscr{R}_{\mathbb{D}}(\mathbb{C}) \quad \text{(and define } \mathbb{C} \boxtimes \mathbb{D} \overset{\text{def}}{=} \mathscr{L}_{\mathbb{C}}(\mathbb{D})\text{)},$$
$$\mathscr{L}_{\mathbb{I}} = \mathscr{R}_{\mathbb{I}} = \mathrm{Id}_{\mathscr{P}},$$
$$\mathscr{L}_{\mathbb{C}\boxtimes\mathbb{D}} = \mathscr{L}_{\mathbb{C}}\mathscr{L}_{\mathbb{D}}, \qquad \mathscr{R}_{\mathbb{C}\boxtimes\mathbb{D}} = \mathscr{R}_{\mathbb{D}}\mathscr{R}_{\mathbb{C}}, \qquad \mathscr{R}_{\mathbb{D}}\mathscr{L}_{\mathbb{C}} = \mathscr{L}_{\mathbb{C}}\mathscr{R}_{\mathbb{D}},$$

  for all 0-cells $\mathbb{C}$, $\mathbb{D}$, and

- for each 1-cells $\mathcal{F} : \mathbb{C} \longrightarrow \mathbb{C}', \mathcal{G} : \mathbb{D} \longrightarrow \mathbb{D}'$, an invertible 2-cell

$$\begin{tikzcd}
\mathbb{C} \boxtimes \mathbb{D} \arrow[r, "\mathscr{L}_{\mathbb{C}}(\mathcal{G})"] \arrow[d, "\mathscr{R}_{\mathbb{D}}(\mathcal{F})"'] \arrow[dr, phantom, "\overset{\mathsf{c}_{\mathcal{F},\mathcal{G}}}{\Rightarrow}"] & \mathbb{C} \boxtimes \mathbb{D}' \arrow[d, "\mathscr{R}_{\mathbb{D}'}(\mathcal{F})"] \\
\mathbb{C}' \boxtimes \mathbb{D} \arrow[r, "\mathscr{L}_{\mathbb{C}'}(\mathcal{G})"'] & \mathbb{C}' \boxtimes \mathbb{D}'
\end{tikzcd}$$

satisfying certain coherence axioms such as $\mathsf{c}_{\mathrm{Id}_{\mathbb{C}},\mathrm{Id}_{\mathbb{D}}} = \mathrm{id}_{\mathbb{C}\boxtimes\mathbb{D}}$, see [11] for details. □

In what follows, we denote $\mathscr{L}_{\mathbb{C}}$ by $\mathbb{C}\boxtimes(-)$ and $\mathscr{R}_{\mathbb{C}}$ by $(-)\boxtimes\mathbb{C}$.

Note that for 1-cells $\mathcal{F} : \mathbb{C} \longrightarrow \mathbb{C}'$ and $\mathcal{G} : \mathbb{D} \longrightarrow \mathbb{D}'$, $\mathbb{C}\boxtimes\mathcal{G}; \mathcal{F}\boxtimes\mathbb{D}'$ and $\mathcal{F}\boxtimes\mathbb{D}; \mathbb{C}'\boxtimes\mathcal{G}$ are not necessarily the same, so we cannot denote them simply by $\mathcal{F}\boxtimes\mathcal{G}$. However they are isomorphic with the iso-2-cell $\mathsf{c}_{\mathcal{F},\mathcal{G}}$, and we can always interchange them.

In the definition of $\mathbf{Gray}$-monoid, all structural isomorphisms of monoidal bicategory are replaced with identities, except for c. Even associativity and unitality—both of the composition of 1-cells and of monoidal product—are strict. By this simplicity, we can keep definitions of monoidal object and strength accessible.

All $\mathbf{Gray}$ monoids form monoidal bicategories, and conversely by the strictification theorem [13], all monoidal bicategories are monoidally equivalent to some $\mathbf{Gray}$ monoids. (It is not true that all monoidal bicategories are monoidally equivalent to some *monoidal 2-categories* [13].) Hence we identify monoidal bicategory with $\mathbf{Gray}$ monoid.

The 2-category $\mathbf{Cat}$ forms a monoidal 2-category with cartesian products. The bicategory $\mathbf{Prof}$ forms a monoidal bicategory with the tensor product $\times$, as the category $\mathbf{Rel}$ forms a monoidal category with the tensor product $\times$.

We can generalize $\mathbf{Prof}$ with enriched category theory:

Example 18 Let $\mathbb{V}$ be a cocomplete SMCC. Then, there is a notion of $\mathbb{V}$-profunctor, also called $\mathbb{V}$-module [20].

For $\mathbb{V}$-categories $\mathbb{C}$ and $\mathbb{D}$, a *$\mathbb{V}$-profunctor* $\mathcal{F}$ from $\mathbb{C}$ to $\mathbb{D}$ is a $\mathbb{V}$-functor $\mathcal{F} : \mathbb{D}^{\mathrm{op}} \boxtimes \mathbb{C} \longrightarrow \mathbb{V}$, and for parallel $\mathbb{V}$-profunctors $\mathcal{F}$ and $\mathcal{F}'$, a *2-cell between $\mathbb{V}$-profunctors* $\mathcal{F}$ and $\mathcal{F}'$ is a $\mathbb{V}$-natural transformation from $\mathcal{F}$ to $\mathcal{F}'$.

Then, as the monoidal bicategory $\mathbf{Prof}$, we can define a monoidal bicategory $\mathbb{V}$-$\mathbf{Prof}$ of small $\mathbb{V}$-categories, $\mathbb{V}$-profunctors, and 2-cells between $\mathbb{V}$-profunctors. The monoidal product in $\mathbb{V}$-$\mathbf{Prof}$ is the same as the tensor product in $\mathbb{V}$-$\mathbf{Cat}$, which we denote by $\boxtimes$. In the case that $\mathbb{V} = \mathbf{Set}$, $\mathbf{Set}$-$\mathbf{Prof}$ is $\mathbf{Prof}$. $\Box$

This $\mathbb{V}$-$\mathbf{Prof}$ plays an important role in Section 6.3.

6.2 Monoidal Object and Strong Monad

First we recall the notion of monoidal object in a $\mathbf{Gray}$ monoid [11].

Definition 19 (Monoidal Object) Let $(\mathscr{P}, \boxtimes, \mathbb{I})$ be a $\mathbf{Gray}$ monoid.

A *monoidal object* (or *pseudomonoid*) *in* $(\mathscr{P}, \boxtimes, \mathbb{I})$ is a 0-cell $\mathbb{C}$ in $\mathscr{P}$ together with 1-cells $\otimes : \mathbb{C} \boxtimes \mathbb{C} \longrightarrow \mathbb{C}$ and $\mathrm{I} : \mathbb{I} \longrightarrow \mathbb{C}$ in $\mathscr{P}$ and invertible 2-cells α, λ, and ρ below such that they satisfy coherence axioms quite similar to those for monoidal categories. (See [11], for further detail.)

$$\begin{tikzcd}
\mathbb{C}^3 \arrow[r, "\otimes\boxtimes\mathbb{C}"] \arrow[d, "\mathbb{C}\boxtimes\otimes"'] \arrow[dr, phantom, "\Downarrow\alpha"] & \mathbb{C}^2 \arrow[d, "\otimes"] \\
\mathbb{C}^2 \arrow[r, "\otimes"'] & \mathbb{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{C} \arrow[r, "\mathrm{I}\boxtimes\mathbb{C}"] \arrow[dr, equal] & \mathbb{C}^2 \arrow[d, "\otimes"'] \arrow[dl, phantom, "\Downarrow\lambda"] \arrow[dr, phantom, "\Downarrow\rho"] & \mathbb{C} \arrow[l, "\mathbb{C}\boxtimes\mathrm{I}"'] \arrow[dl, equal] \\
& \mathbb{C} &
\end{tikzcd}$$

Monoidal objects in $\mathbf{Cat}$ are monoidal categories.

Monoidal objects in $\mathbf{Prof}$ are called *promonoidal categories* [9, 10]. In Section 5.1, we considered only the promonoidal categories induced from monoidal categories via the direct image embedding. (The direct image embedding is monoidal pseudofunctor, hence maps monoidal objects in $\mathbf{Cat}$ to those in $\mathbf{Prof}$.)

Now we define strong monads in $\mathbf{Gray}$-monoids:

Definition 20 (Strong Monad in $\mathbf{Gray}$-Monoid) Let $(\mathscr{P}, \boxtimes, \mathbb{I})$ be a $\mathbf{Gray}$-monoid, $(\mathbb{C}, \otimes, \mathrm{I}, \alpha, \lambda, \rho)$ be a monoidal object in $(\mathscr{P}, \boxtimes, \mathbb{I})$, and $(\mathcal{A}, \eta, \mu)$ be a monad over $\mathbb{C}$ in $\mathscr{P}$.

A *strength* for the monad $\mathcal{A}$ in $(\mathscr{P}, \boxtimes, \mathbb{I})$ is a 2-cell $\mathtt{str}$ in $\mathscr{P}$ in the following diagram such that it satisfies the axioms (s-α), (s-ρ), (s-η), and (s-μ) below.

$$\begin{tikzcd}
\mathbb{C}^2 \arrow[r, "\mathcal{A}\boxtimes\mathbb{C}"] \arrow[d, "\otimes"'] \arrow[dr, phantom, "\Downarrow\mathtt{str}"] & \mathbb{C}^2 \arrow[d, "\otimes"] \\
\mathbb{C} \arrow[r, "\mathcal{A}"'] & \mathbb{C}
\end{tikzcd}$$

$$\begin{tikzcd}
\mathbb{C}^3 \arrow[r, "\mathcal{A}\boxtimes\mathbb{C}^2"] \arrow[dr, "\mathbb{C}\boxtimes\otimes"'] \arrow[dd, "\otimes\boxtimes\mathbb{C}"'] & \mathbb{C}^3 \arrow[r, "\mathbb{C}\boxtimes\otimes"] \arrow[d, phantom, "\Downarrow\mathtt{c}"] & \mathbb{C}^2 \arrow[dd, "\otimes"] \\
 & \mathbb{C}^2 \arrow[ur, "\mathcal{A}\boxtimes\mathbb{C}"'] \arrow[d, "\otimes"'] \arrow[dl, phantom, "\Downarrow\alpha^{-1}"] \arrow[dr, phantom, "\Downarrow\mathtt{str}"] & \\
\mathbb{C}^2 \arrow[r, "\otimes"'] & \mathbb{C} \arrow[r, "\mathcal{A}"'] & \mathbb{C}
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
\mathbb{C}^3 \arrow[r, "\mathcal{A}\boxtimes\mathbb{C}^2"] \arrow[dd, "\otimes\boxtimes\mathbb{C}"'] \arrow[dr, phantom, "\Downarrow\mathtt{str}\boxtimes\mathbb{C}"] & \mathbb{C}^3 \arrow[r, "\mathbb{C}\boxtimes\otimes"] \arrow[d, "\otimes\boxtimes\mathbb{C}"] \arrow[dr, phantom, "\Downarrow\alpha^{-1}"] & \mathbb{C}^2 \arrow[dd, "\otimes"] \\
 & \mathbb{C}^2 \arrow[dr, "\otimes"] \arrow[d, phantom, "\Downarrow\mathtt{str}"] & \\
\mathbb{C}^2 \arrow[r, "\otimes"'] \arrow[ur, "\mathcal{A}\boxtimes\mathbb{C}"] & \mathbb{C} \arrow[r, "\mathcal{A}"'] & \mathbb{C}
\end{tikzcd}
\qquad (\text{s-}\alpha)$$

$$\begin{tikzcd}
\mathbb{C} \arrow[r, "\mathbb{C}\boxtimes\mathrm{I}"] \arrow[dr, equal] & \mathbb{C}^2 \arrow[r, "\mathcal{A}\boxtimes\mathbb{C}"] \arrow[d, "\otimes"] \arrow[dl, phantom, "\Downarrow\rho"] \arrow[dr, phantom, "\Downarrow\mathtt{str}"] & \mathbb{C}^2 \arrow[d, "\otimes"] \\
 & \mathbb{C} \arrow[r, "\mathcal{A}"'] & \mathbb{C}
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
\mathbb{C} \arrow[r, "\mathbb{C}\boxtimes\mathrm{I}"] \arrow[dr, "\mathcal{A}"'] & \mathbb{C}^2 \arrow[r, "\mathcal{A}\boxtimes\mathbb{C}"] \arrow[d, phantom, "\Downarrow\mathtt{c}^{-1}"] & \mathbb{C}^2 \arrow[d, "\otimes"] \arrow[dl, phantom, "\Downarrow\rho"] \\
 & \mathbb{C} \arrow[ur, "\mathbb{C}\boxtimes\mathrm{I}"] \arrow[r, equal] & \mathbb{C}
\end{tikzcd}
\qquad (\text{s-}\rho)$$

$$\begin{tikzcd}
 & \mathbb{C}^2 \arrow[dr, "\otimes"] \arrow[d, phantom, "\Downarrow\mathtt{str}"] & \\
\mathbb{C}^2 \arrow[ur, bend left=60, equal, "\Downarrow\eta\boxtimes\mathbb{C}"'] \arrow[ur, "\mathcal{A}\boxtimes\mathbb{C}"'] \arrow[r, "\otimes"'] & \mathbb{C} \arrow[r, "\mathcal{A}"'] & \mathbb{C}
\end{tikzcd}
\quad = \quad
\begin{tikzcd}
\mathbb{C}^2 \arrow[r, "\otimes"] & \mathbb{C} \arrow[r, bend left=60, equal, "\Downarrow\eta"'] \arrow[r, "\mathcal{A}"'] & \mathbb{C}
\end{tikzcd}
\qquad (\text{s-}\eta)$$

$$\begin{tikzcd}
\mathbb{C}^2 \arrow[r, "\mathcal{A}\boxtimes\mathbb{C}"] \arrow[dd, "\otimes"'] & \mathbb{C}^2 \arrow[r, "\mathcal{A}\boxtimes\mathbb{C}"] \arrow[d, "\otimes"] \arrow[ld, phantom, "\Downarrow\mathtt{str}"] \arrow[rd, phantom, "\Downarrow\mathtt{str}"] & \mathbb{C}^2 \arrow[dd, "\otimes"] \\
& \mathbb{C} \arrow[rd, "\mathcal{A}"] \arrow[d, phantom, "\Downarrow\mu"] & \\
\mathbb{C} \arrow[ru, "\mathcal{A}"] \arrow[rr, "\mathcal{A}"'] & {} & \mathbb{C}
\end{tikzcd}
=
\begin{tikzcd}
\mathbb{C}^2 \arrow[r, "\mathcal{A}\boxtimes\mathbb{C}"] \arrow[rr, bend right, "\mathcal{A}\boxtimes\mathbb{C}"'] \arrow[d, "\otimes"'] & \mathbb{C}^2 \arrow[r, "\mathcal{A}\boxtimes\mathbb{C}"] \arrow[d, phantom, "\Downarrow\mu\boxtimes\mathbb{C}"] & \mathbb{C}^2 \arrow[d, "\otimes"] \\
\mathbb{C} \arrow[rr, "\mathcal{A}"'] & \Downarrow\mathtt{str} & \mathbb{C}
\end{tikzcd}
\tag{s-μ}$$

We call a monad in $\mathscr{P}$ equipped with a strength in $(\mathscr{P}, \boxtimes, \mathbb{I})$ a *strong monad* in $(\mathscr{P}, \boxtimes, \mathbb{I})$. $\square$

By this definition with $\mathscr{P} = \mathbf{Prof}$, we can in fact define arrows not only over monoidal categories as in Section 5 but also over any promonoidal categories; on the other hand, we have no similar theorem to Theorem 14, because `first`-operators are defined only for monoidal categories.

6.3 Self-enrichment of Arrow

In the last place, we consider a solution for the following problem pointed out in the paper [3]: For a CCC $\mathbb{C}$, let us identify types and terms in Haskell with objects and morphisms in $\mathbb{C}$, respectively. An arrow $\mathcal{A}$ in Haskell is a type constructor, hence maps two types to a type; i.e., $\mathcal{A}$ is not a functor $\mathbb{C}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbf{Set}$ but a functor $\mathbb{C}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbb{C}$.

In order to replace $\mathbf{Set}$ with $\mathbb{C}$, a naive way is to consider strong monads in $\mathbb{C}$-$\mathbf{Prof}$ in Example 18. However then $\mathbb{C}$ is required to be small cocomplete and small at the same time, which implies that $\mathbb{C}$ must be a preorder. The use of the synthetic domain theory, which admits non-trivial internal small cocomplete categories, is not natural just for this purpose; for it unnecessarily restricts models.

The solution that we give for this is use of cocompletion and to distinguish "small" ones from "large" ones.

Now let $\mathbb{V}$ be an SMC, $\mathbb{V}'$ be a sufficiently cocomplete SMCC, and $J : \mathbb{V} \longrightarrow \mathbb{V}'$ be a symmetric strong monoidal fully faithful functor, whose leading example is the Yoneda embedding $y : \mathbb{V} \longrightarrow \widehat{\mathbb{V}}$. Then $\mathbb{V}$ can be a $\mathbb{V}'$-category with hom-objects $\hom_{\mathbb{V}}(A, B) \overset{\mathrm{def}}{=} JA \multimap_{\mathbb{V}'} JB$. The underlying category of this $\mathbb{V}'$-category $\mathbb{V}$ is isomorphic to the category $\mathbb{V}$ itself; also J can be $\mathbb{V}'$-functor whose underlying functor is again J.

For a $\mathbb{V}'$-profunctor $\mathcal{F} : \mathbb{C} \looparrowright \mathbb{D}$, we call $\mathcal{F}$ $\mathbb{V}$-*small* if the $\mathbb{V}'$-functor $\mathcal{F} : \mathbb{D}^{\mathrm{op}} \boxtimes \mathbb{C} \longrightarrow \mathbb{V}'$ is factorized along the $\mathbb{V}'$-functor $J : \mathbb{V} \longrightarrow \mathbb{V}'$:

$$\begin{tikzcd}
\mathbb{D}^{\mathrm{op}} \boxtimes \mathbb{C} \arrow[r, "\mathcal{F}"] \arrow[dr, "\mathcal{F}^{\circ}"'] & \mathbb{V}' \\
& \mathbb{V} \arrow[u, "J"']
\end{tikzcd}$$

Such factors of $\mathcal{F}$ are unique up to natural $\mathbb{V}'$-isomorphism, and we denote the $\mathbb{V}'$-functor from $\mathbb{D}^{\mathrm{op}} \boxtimes \mathbb{C}$ to $\mathbb{V}$ by $\mathcal{F}^{\circ}$. (Note that when J is the Yoneda embedding, which is injective, $\mathcal{F}^{\circ}$ is uniquely determined.)

Definition 21 For an SMC $\mathbb{V}$ and an embedding $J : \mathbb{V} \longrightarrow \mathbb{V}'$ as above, a *small strong monad over* $\mathbb{V}$ *in* $\mathbb{V}'$-$\mathbf{Prof}$ *(with respect to J)* is a strong monad over $\mathbb{V}$ in $\mathbb{V}'$-$\mathbf{Prof}$ whose underlying $\mathbb{V}'$-endoprofunctor is $\mathbb{V}$-small. $\square$

We show that the above definition gives a generalization of Atkey's definition of arrow, from that over a cartesian category to that over a monoidal category.

In the definition below, we use the coKleisli category of the comonad $D \times (-)$ for each object D in a cartesian category $\mathbb{C}$. Note that if a base category is cartesian, so is its coKleisli category.

Definition 22 [3] An *arrow* on a cartesian category $\mathbb{C}$ consists of a mapping of objects $\mathrm{Ar} : |\mathbb{C}| \times |\mathbb{C}| \longrightarrow |\mathbb{C}|$ and three transformations all natural in D,

$$\begin{aligned}
\mathtt{arr}_{DAB} &: \mathbb{C}\,(D \times A, B) \longrightarrow \mathbb{C}\,(D, \mathrm{Ar}\,(A, B)) \\
\ggg_{DABC} &: \mathbb{C}\,(D, \mathrm{Ar}\,(A, B)) \times \mathbb{C}\,(D, \mathrm{Ar}\,(B, C)) \\
&\qquad\qquad\qquad\qquad \longrightarrow \mathbb{C}\,(D, \mathrm{Ar}\,(A, C)) \\
\mathtt{first}_{DABC} &: \mathbb{C}\,(D, \mathrm{Ar}\,(A, B)) \longrightarrow \mathbb{C}\,(D, \mathrm{Ar}\,(A \times C, B \times C))
\end{aligned}$$

These transformations must satisfy the eight laws (assoc) – (f-μ) in Section 2 where the equations are interpreted as equations between $\mathbb{C}$ morphisms generated by the above transformations. In those equations, we use variables f, g for morphisms in $\mathbb{C}\,(D \times A, B)$ and a, b, c for morphisms in $\mathbb{C}\,(D, \mathrm{Ar}\,(A, B))$; and the composition $\circ$, identity id, product $\times$ and projection π_1 are all those in the coKleisli category noted above. □

Theorem 23 *For a cartesian category $\mathbb{C}$, the notion of arrow over $\mathbb{C}$ defined in Definition 22 is equivalent to that of small strong monad over $\mathbb{C}$ in $\widehat{\mathbb{C}}$-$\mathbf{Prof}$ with respect to the Yoneda embedding $y : \mathbb{C} \longrightarrow \widehat{\mathbb{C}}$.* □

Proof. Basically, this proof is almost the same as Proof 10 and 14.

Since a small strong monad $\mathcal{A}$ is decomposed as $y \circ \mathcal{A}^{\circ}$ by definition, it is obvious that this corresponds to Ar of an arrow of Definition 22:

$$\mathcal{A}^{\circ}(A, B) \overset{\text{def}}{=} \mathrm{Ar}\,(A, B) \tag{8}$$

The unit η of a small strong monad $\mathcal{A}$ over $\mathbb{C}$ is a natural $\widehat{\mathbb{C}}$-transformation from the identity $\widehat{\mathbb{C}}$-profunctor $\mathrm{Id}^{+}_{\mathbb{C}}$ to $\mathcal{A}$, i.e., a family of morphisms

$$\eta_{A,B} : \left(yA {\to_{\widehat{\mathbb{C}}}} yB\right) \longrightarrow y\,(\mathcal{A}^{\circ}(A, B))$$

in $\widehat{\mathbb{C}}$ natural in $A, B \in \mathbb{C}$. Here the presheaf $yA {\to_{\widehat{\mathbb{C}}}} yB$ is isomorphic to the presheaf $\mathbb{C}\,(- \times A, B)$, by Yoneda lemma. On the other hand, by the correspondence (8), $y\,(\mathcal{A}^{\circ}(A, B)) = \mathbb{C}\,(-, \mathrm{Ar}\,(A, B))$. As this, η corresponds to `arr`.

A correspondence between μ and $\ggg$ is readily obtained as in Proof 10.

Finally, we can easily prove that `str` corresponds to `first`, with the full-faithfulness of y, and as in Proof 14, where we use $\widehat{\mathbb{C}}$-enriched version of Lemma 3, 4, 5, and 6 (see [20, Sec. 3.10]).□

Example 24 Besides Yoneda embeddings $\mathbb{C} \longrightarrow \widehat{\mathbb{C}}$, we can also use the inclusion $\mathbf{Set} \hookrightarrow \mathbf{Ens}$, the codomain of which is the category of classes. Then we have the notion of small strong monad over $\mathbf{Set}$ in $\mathbf{Ens}$-$\mathbf{Prof}$: $\mathcal{A} : \mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \longrightarrow \mathbf{Set}$.

In the paper [2], such notion is used, to obtain some *endo*functors over $\mathbf{Set}$—rather than functors from $\mathbf{Set}$ to $\mathbf{Ens}$—for which we can consider coalgebras. □

Acknowledgments

I would like to thank Ichiro Hasuo for encouraging me to write the paper, helpful comments for early draft, and discussions, and also thank Paul-André Melliès for teaching me profunctors, and higher dimensional category theory. Also I am grateful to Shin-ya Katsumata, Bart Jacobs, Pastro Craig, and Naohiko Hoshino for useful comments and discussions.

This work is supported by the GCOE Program, JSPS; and also by the PRESTO Research Promotion Program, JST (principal investigator: Ichiro Hasuo).

References

[1] T. Altenkirch, J. Chapman, and T. Uustalu. Monads need not be endofunctors. In C.-H. L. Ong, editor, *FOSSACS*, volume 6014 of *Lecture Notes in Computer Science*, pages 297–311. Springer, 2010. ISBN 978-3-642-12031-2.

[2] K. Asada and I. Hasuo. Categorifying computations into components via arrows as profunctors. To appear in Proc. International Workshop on Coalgebraic Methods in Computer Science (CMCS 2010)., 2010.

[3] R. Atkey. What is a categorical model of arrows? *Electr. Notes Theor. Comput. Sci*, To appear.

[4] J. Bénabou. Distributors at work. Lecture notes by Thomas Streicher, 2000.

[19] B. Jacobs, C. Heunen, and I. Hasuo. Categorical semantics for arrows. *Journal of Functional Programming*, 19(3-4):403–438, 2009. ISSN 0956-7968.

[20] G. M. Kelly. *Basic Concepts of Enriched Category Theory*. Number 64 in LMS. Cambridge Univ. Press, 1982.

[21] P. B. Levy. *Call-By-Push-Value: A Functional/Imperative Synthesis*,

volume 2 of *Semantics Structures in Computation*. Springer, 2004. ISBN 1-4020-1730-8.

[22] P. B. Levy, A. J. Power, and H. Thielecke. Modelling environments in call-by-value programming languages. *Inf. & Comp.*, 185(2):182–210, 2003. ISSN 0890-5401.

[23] E. Moggi. Computational lambda-calculus and monads. In *LICS*, pages 14–23. IEEE Computer Society, 1989.

[24] E. Moggi. Notions of computation and monads. *Information and Computation*, 93(1):55–92, 1991.

[25] M. Nygaard and G. Winskel. Domain theory for concurrency. *Theor. Comput. Sci.*, 316(1):153–190, 2004.

[26] J. Power and E. Robinson. Premonoidal categories and notions of computation. *Math. Struct. in Comp. Sci.*, 7(5):453–468, 1997.

[27] R. Street. The formal theory of monads. *Journal of Pure and Applied Algebra*, 2:149–168, 1972.

[28] T. Uustalu. Strong relative monads. Short contribution in International Workshop on Coalgebraic Methods in Computer Science (CMCS 2010)., 2010.

[29] P. Wadler. Monads for functional programming. In *Lecture Notes In Computer Science, Vol. 925; Advanced Functional Programming, First International Spring School on Advanced Functional Programming Techniques-Tutorial Text*, pages 24–52, London, UK, 1995. Springer-Verlag. ISBN 3-540-59451-5.

`www.mathematik.tu-darmstadt.de/~streicher/FIBR/DiWo.pdf.gz`.

[5] N. Benton, J. Hughes, and E. Moggi. Monads and effects. In G. Barthe, P. Dybjer, L. Pinto, and J. Saraiva, editors, *Advanced Lectures from Int. Summer School on Applied Semantics, APPSEM 2000 (Caminha, Portugal, 9–15 Sept. 2000)*, volume 2395 of *Lecture Notes in Computer Science*, pages 42–122. Springer-Verlag, Berlin, 2002.

[6] F. Borceux. *Handbook of Categorical Algebra 1: Basic Category Theory*, volume 50 of *Encyclopedia of Mathematics and Its Applications*. Cambridge University Press, Cambridge, 1994.

[7] G. Cattani, M. Fiore, and G. Winskel. A theory of recursive domains with applications to concurrency. *In Proc. of LICS '98*, 0:214–225, 1998. ISSN 1043-6871.

[8] P.-L. Curien. Operads, clones, and distributive laws. preprint, `www.pps.jussieu.fr/~curien`, 2008.

[9] B. Day. On closed categories of functors. In *Reports of the Midwest Category Seminar, IV*, Lecture Notes in Mathematics, Vol. 137, pages 1–38. Springer, Berlin, 1970.

[10] B. Day. On closed categories of functors. II. In *Category Seminar (Proc. Sem., Sydney, 1972/1973)*, pages 20–54. Lecture Notes in Math., Vol. 420. Springer, Berlin, 1974.

[11] B. Day and R. Street. Monoidal bicategories and Hopf algebroids. *Adv. Math.*, 129(1):99–157, 1997. ISSN 0001-8708.

[12] M. Fiore, N. Gambino, M. Hyland, and G. Winskel. The cartesian closed bicategory of generalised species of structures. *Journ. of London Math. Soc.*, 77:203–220, 2008.

[13] R. Gordon, A. J. Power, and R. Street. Coherence for tricategories. *Mem. Amer. Math. Soc.*, 117(558):vi+81, 1995. ISSN 0065-9266.

[14] I. Hasuo, B. Jacobs, and A. Sokolova. The microcosm principle and concurrency in coalgebra. In *Foundations of Software Science and Computation Structures*, volume 4962 of *Lect. Notes Comp. Sci.*, pages 246–260. Springer-Verlag, 2008.

[15] C. Heunen and B. Jacobs. Arrows, like monads, are monoids. *Electr. Notes Theor. Comput. Sci*, 158:219–236, 2006. URL `http://dx.doi.org/10.1016/j.entcs.2006.04.012`.

[16] T. T. Hildebrandt, P. Panangaden, and G. Winskel. A relational model of non-deterministic dataflow. In D. Sangiorgi and R. de Simone, editors, *CONCUR*, volume 1466 of *Lecture Notes in Computer Science*, pages 613–628. Springer, 1998. ISBN 3-540-64896-8.

[17] J. Hughes. Generalising monads to arrows. *Sci. Comput. Program.*, 37(1-3):67–111, 2000.

[18] J. Hughes. Programming with arrows. In V. Vene and T. Uustalu, editors, *Advanced Functional Programming*, volume 3622 of *Lecture Notes in Computer Science*, pages 73–129. Springer, 2004. ISBN 3-540-28540-7.