Categorifying Computations into Components via Arrows as Profunctors

Kazuyuki Asada   Ichiro Hasuo

Research Institute for Mathematical Sciences, Kyoto University, Japan
PRESTO Research Promotion Program, Japan Science and Technology Agency
http://www.kurims.kyoto-u.ac.jp/~{asada,ichiro}

Abstract

The notion of *arrow* by Hughes is an axiomatization of the algebraic structure possessed by structured computations in general. We claim that an arrow also serves as a basic *component calculus* for composing state-based systems as components—in fact, it is a *categorified* version of arrow that does so. In this paper, following the second author's previous work with Heunen, Jacobs and Sokolova, we prove that a certain coalgebraic modeling of components—which generalizes Barbosa's—indeed carries such arrow structure. Our coalgebraic modeling of components is parametrized by an arrow A that specifies computational structure exhibited by components; it turns out that it is this arrow structure of A that is lifted and realizes the (categorified) arrow structure on components. The lifting is described using the first author's recent characterization of an arrow as an internal strong monad in **Prof**, the bicategory of small categories and *profunctors*.

Keywords: algebra, arrow, coalgebra, component, computation, profunctor

1 Introduction

1.1 Arrow for Computation

In functional programming, the word *computation* often refers to a procedure which is not necessarily *purely functional*, typically involving some *side-effect* such as I/O, global state, non-termination and non-determinism. The most common way to organize such computations is by means of a *(strong) monad* [21], as is standard in Haskell. However side-effect—that is "structured output"—is not the only cause for the failure of pure functionality. A *comonad* can be used to encapsulate "structured input" [26]; the combination of a monad and a comonad via a distributive law can be used for input and output that are both structured. There are much more additional structure that a functional programmer would like to think of as "computations"; Hughes' notion of *arrow* [13] is a general axiomatization of such.[1]

Let $\mathbb{C}$ be a Cartesian category of types and pure functions, in a functional programming sense. The notion of arrow over $\mathbb{C}$ is an algebraic one: it axiomatizes

[1] The word "arrow" is reserved for Hughes' notion throughout the paper. An "arrow" in a category will be called a morphism or a 1-cell.

This paper is electronically published in
Electronic Notes in Theoretical Computer Science
URL: www.elsevier.nl/locate/entcs

those operators which the set of computations should be equipped with, and those equations which those operators should satisfy. More specifically, an arrow A is

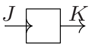
- carried by a family of sets $A(J,K)$ for each $J,K \in \mathbb{C}$;
- equipped with the following three families of operators arr, $\ggg$ and first:

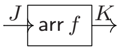
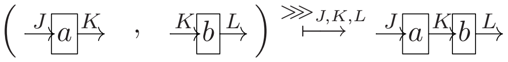
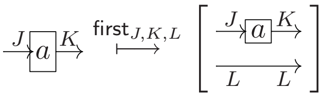
$$\begin{array}{ccll} & \mathsf{arr}\,f \in A(J,K) & & \text{for each morphism } f : J \to K \text{ in } \mathbb{C}, \\ A(J,K) \times A(K,L) & \xrightarrow{\ggg_{J,K,L}} & A(J,L) & \text{for each } J,K,L \in \mathbb{C}, \\ A(J,K) & \xrightarrow{\mathsf{first}_{J,K,L}} & A(J \times L, K \times L) & \text{for each } J,K,L \in \mathbb{C}; \end{array}$$

- that are subject to several equational axioms: among them is

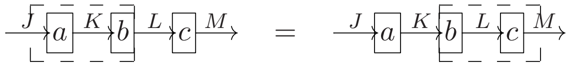
$$\begin{array}{c} (a \ggg_{J,K,L} b) \ggg_{J,L,M} c \quad = \quad a \ggg_{J,K,M} (b \ggg_{K,L,M} c) \\ \text{for each } a \in A(J,K), b \in A(K,L), c \in A(L,M). \end{array} \qquad (\ggg\text{-Assoc})$$

The other axioms are presented later in Def. 3.1.

The intuitions are clear: presenting an A-computation from J to K by a box , the three operators ensure that we can combine computations in the following ways.

- (Embedding of pure functions) 
- (Sequential composition) 
- (Sideline) 

The ($\ggg$-Assoc) axiom above, for example, ensures that the following compositions of three consecutive A-computations are identical.

 (1)

A strong monad T on $\mathbb{C}$ induces an arrow A_T by: $A_T(J,K) = \mathbb{C}(J,TK) = \mathcal{K}\ell(T)(J,K)$. Here $\mathcal{K}\ell(T)$ denotes the Kleisli category (see e.g. [21]). Prior to arrows, the notion of *Freyd category* is devised as another axiomatization of algebraic properties that are expected from "computations" [23, 19]. The latter notion of Freyd category come with a stronger categorical flavor; in [16] it is shown to be equivalent to the notion of arrow.

Remark 1.1 The previous arguments are true as long as we think of an arrow as carried by sets, with $A(J,K)$ being a set. This is our setting. However this is not an entirely satisfactory view in functional programming where one sees A as a type constructor—$A(J,K)$ should rather be an object of $\mathbb{C}$. In this case one can think of several variants of arrow and Freyd category. See [2].

1.2 Arrow as Component Calculus

The goal of the current paper is to settle *components* as *categorification* of computations, via (the algebraic theory of) arrows. Let us elaborate on this slogan.

A *component* here is in the sense of *component calculi*. Components are systems which, combined with one another by means of some component calculus, yield a bigger, more complicated system. This "divide-and-conquer" strategy brings order to design processes of large-scale systems that are otherwise messed up due to the very scale and complexity of the systems to be designed.

We follow the coalgebraic modeling of components in [5]—which is also used in [11]—extending it later to an arrow-based modeling. In [5] a component is modeled as a coalgebra of the following type:

$$c \;:\; X \longrightarrow \bigl(T(X \times K)\bigr)^{J} \quad \text{in } \mathbf{Set}. \tag{2}$$

$\xrightarrow{J}\boxed{c}\xrightarrow{K}$

Here J is the set of possible input to the component; K is that of possible output; X is the set of (internal) states of the component which is a state-based machine; and T is a monad on **Set** that models the computational effect exhibited by the system. Overall, a coalgebraic component is a state-based system with specified input and output ports; it can be drawn as above on the right.

A crucial observation here is as follows. The notion of arrow in §1.1 is to axiomatize algebraic operators on computations as boxes—such as sequential composition $\xrightarrow{J}\boxed{a}\xrightarrow{K}\boxed{b}\xrightarrow{L}$. Then, by regarding such boxes as components rather than as computations, we can employ the axiomatization of arrow as algebraic structure on components—a *component calculus*—with which one can compose components. The calculus is a basic one that allows embedding of pure functions, sequential composition and sideline. In fact in the second author's previous work [11] with Heunen, Jacobs and Sokolova, such algebraic operators on coalgebraic components (2) are defined and shown to satisfy the equational axioms.

1.3 Categorifying Computations into Components

Despite this similarity between computations and components, there is one level climbed up from the former to the latter: from *sets* to *categories*. Let $\mathcal{A}(J, K)$ denote the collection of coalgebraic components like in (2), with input-type J, output-type K and fixed effect T, but with varying state spaces X. Then it is just natural to include morphisms between coalgebras in the overall picture, as behavior-preserving maps (see e.g. [24]) between components. Hence $\mathcal{A}(J, K)$ is now a *category*, specifically that of $\bigl(T(\_ \times K)\bigr)^{J}$-coalgebras. In contrast, with respect to computations there is no general notion of morphism between them, so the collection $A(J, K)$ of A-computations is a *set*.

This step of *categorification* [3] is not just for fun but in fact indispensable when we consider equational axioms. Later on we will concretely define the sequential composition $\xrightarrow{J}\boxed{c}\xrightarrow{K}\boxed{d}\xrightarrow{L}$ of coalgebraic components with matching I/O types; at now we note that the state space of the composite is the product $X \times Y$ of the state space X of c and Y of d. Now let us turn to the axiom

$$(c \ggg d) \ggg e \;=\; c \ggg (d \ggg e)\;\;. \tag{$\ggg$-Assoc}$$

Denoting e's state space by U, the state space of the LHS is $(X \times Y) \times U$ while that of the RHS is $X \times (Y \times U)$. These are, as sets, not identical! Therefore the axiom can be at best satisfied up-to an isomorphism between components as coalgebras (and it is the case, see [11]). We note this phenomenon that the notion of satisfaction of equational axioms gets relaxed—from up-to equality to up-to an isomorphism—is typical with categorification [3].

This additional structure obtained through categorification, namely morphisms between components, has been further exploited in [11]. There it is shown that final coalgebras—the notion that only makes sense in presence of morphisms between coalgebras—form an arrow that is internal to the "arrow" of components, realizing an instance of the *microcosm principle* [4, 12]. An application of such nested algebraic structure (namely of arrows) is a *compositionality result*: the behavior of composed components can be computed from the behavior of each component.

We shall refer to the categorified notion of arrow—carried by components—as *categorical arrow*.

1.4 Lifting of Arrow Structure via Profunctors

To summarize: computations carry algebraic structure of an arrow; components carry a categorified version of it. The contribution of the current paper is to make the relationship between computations and components more direct. This is by developing the following scenario:

- given an arrow A,
- we define the notion of *(arrow-based) A-component* which generalizes Barbosa's modeling (2),
- and we show that these A-components carry categorical arrow structure that is in fact a lifting of the original arrow structure of A.

Therefore: we categorify A-computations to A-components.

A weaker version of this scenario has been already presented in [11]. However the last lifting part was obscured in details of direct calculations. What is novel in this paper is to work in **Prof**, the bicategory of profunctors. In fact, it is one theme of this paper to demonstrate use of calculations in **Prof**.

The starting point for this profunctor approach is [16]. There the $\mathsf{arr}, \ggg$-fragment of arrow (without first) is identified with a monoid in the category $[\mathbb{C}^{\mathrm{op}} \times \mathbb{C}, \mathbf{Set}]$ of bifunctors, where the latter is equipped with suitable monoidal structure. This means—in terms of profunctors that will be described in §2—that an arrow A (without first) is a *monad* in **Prof**, in an internal sense like in [25].

What really made our profunctor approach feasible was a further observation by the first author [1]. There the remaining first operator—whose mathematical nature was buried away in its dinaturality—is identified with a certain 2-cell in **Prof**. In fact, this 2-cell is a *strength* in an internal sense. Therefore an arrow (with its full set of operators, arr, $\ggg$ and first) is a *strong monad* in **Prof**. This observation pleasantly parallels the informal view of arrows as generalization of strong monads.

1.5 Organization of the Paper

In §2 we will introduce the necessary notions of dinatural transformation, (co)end and profunctor, in a rather leisurely pace. The two forms of the Yoneda lemma—

the end- and coend-forms—are basic there. The materials there are essentially extracted from [17], which is a useful reference also in the current non-enriched (i.e. **Set**-enriched) setting. In §3 we follow [1, 16] and identify an arrow with an internal strong monad in **Prof**, setting **Prof** as our universe of discourse. In §4 we generalize Barbosa's coalgebraic components into arrow-based components. The main result—arrow-based components form a categorical arrow—is stated there. Its actual proof is in the subsequent §5 which is devoted to manipulation of 2-cells in **Prof**.

2 Categorical Preliminaries

2.1 End and Coend

In the sequel we shall very often encounter a functor of the type $F : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$, where a category $\mathbb{C}$ occurs twice with different variance. Given two such $F, G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$, a *dinatural transformation* $\varphi : F \Rightarrow G$ consists of a family of morphisms in $\mathbb{D}$

$$\varphi_X \;:\; F(X, X) \longrightarrow G(X, X) \quad \text{for each } X \in \mathbb{C}$$

which is *dinatural*: for each morphism $f : X \to X'$ the following diagram commutes.

$$\begin{tikzcd}
& F\,(X, X) \arrow[r, "\varphi_X"] & G\,(X, X) \arrow[dr, "G(X,f)"] & \\
F\,(X', X) \arrow[ur, "F(f,X)"] \arrow[dr, "F(X',f)"'] & & & G\,(X, X') \\
& F\,(X', X') \arrow[r, "\varphi_{X'}"'] & G\,(X', X') \arrow[ur, "G(f,X')"'] &
\end{tikzcd} \tag{3}$$

Note the difference from a *natural transformation* $\psi : F \Rightarrow G$. The latter consists of a greater number of morphisms in $\mathbb{D}$: $\psi_{X,Y} : F(X, Y) \to G(X, Y)$ for each $X, Y \in \mathbb{C}$.

Two successive dinatural transformations $\varphi_1 : F_1 \Rightarrow F_2$ and $\varphi_2 : F_2 \Rightarrow F_3$ do not necessarily compose: dinaturality of each do not guarantee dinaturality of the obvious candidate of the composition $(\varphi_2 \circ \varphi_1)_X = (\varphi_2)_X \circ (\varphi_1)_X$. This makes it a tricky business to organize dinatural transformations in a categorical manner. Nevertheless, working with arrows, examples of dinaturality abound.

Dinaturality subsumes naturality: a natural transformation $\psi : F \Rightarrow G : \mathbb{C} \to \mathbb{D}$ can be thought of as a dinatural transformation, by presenting it as $\psi : F \circ \pi_2 \Rightarrow G \circ \pi_2 : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$. Here $\pi_2 : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{C}$ is a projection.

(Co)end is the notion that is obtained by replacing naturality (for (co)cones) by dinaturality, in the definition of (co)limit. Precisely:

Definition 2.1 (End and coend) Let $\mathbb{C}, \mathbb{D}$ be categories and $F : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$ be a functor.

- An *end* of F consists of an object $\int_{X \in \mathbb{C}} F\,(X, X)$ in $\mathbb{D}$ together with *projections*

$$\pi_X : \bigl(\textstyle\int_{X \in \mathbb{C}} F\,(X, X)\bigr) \longrightarrow F\,(X, X) \quad \text{for each } X \in \mathbb{C}$$

such that, for each morphism $f : X \to X'$ in $\mathbb{C}$, the following diagram commutes.

$$\begin{tikzcd} & F(X',X') \arrow[dr, "F(f,X')"] & \\ \int_X F(X,X) \arrow[ur, "\pi_{X'}"] \arrow[dr, "\pi_X"'] & & F(X,X') \\ & F(X,X) \arrow[ur, "F(X,f)"'] & \end{tikzcd}$$

In other words: the family $\{\pi_X\}_{X\in\mathbb{C}}$ forms a dinatural transformation from the constant functor $\Delta(\int_X F(X,X))$ to the functor F. An end is defined to be a universal one among such data: given an object $Y \in \mathbb{D}$ and a dinatural transformation $\varphi : \Delta Y \Rightarrow F$, there is a unique morphism $f : Y \to \int_X F(X,X)$ such that $\pi_X \circ f = \varphi_X$ for each $X \in \mathbb{C}$.

- A *coend* of F is a dual notion of an end. It consists of an object $\int^{X\in\mathbb{C}} F(X,X)$ in $\mathbb{D}$ together with *injections* $\iota_X : F(X,X) \to \int^X F(X,X)$ for each $X \in \mathbb{C}$. Its universality, together with that of an end, can be written as follows.

$$\frac{f : Y \longrightarrow \int_X F(X,X)}{\varphi_X : Y \to F(X,X)\,,\ \text{dinatural in } X} \qquad \frac{f : \int^X F(X,X) \longrightarrow Y}{\varphi_X : F(X,X) \to Y,\ \text{dinatural in } X}$$

See [20, Chap. IX] for more on (co)ends. Described there is the way to transform a functor $F : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbb{D}$ into $F^{\S} : \mathbb{C}^{\S} \to \mathbb{D}$, in such a way that the (co)end of F coincides with the (co)limit of $F^{\S}$. Therefore existence of (co)ends depends on the (co)completeness property of $\mathbb{D}$. In fact (co)end subsumes (co)limit, just as dinaturality subsumes naturality. Therefore a useful notational convention is to denote (co)limits also as (co)ends: for example $\mathrm{Colim}_X FX$ as $\int^X FX$.

Recalling the construction of any limit by a product and an equalizer [20, §V.2], an intuition about an end $\int_X F(X,X)$ is as follows: it is the product $\prod_X F(X,X)$ which is "cut down" so as to satisfy dinaturality. Dually, a coend $\int^X F(X,X)$ is the coproduct $\coprod_X F(X,X)$ quotiented modulo dinaturality.

2.2 Two Forms of the Yoneda Lemma

A typical example of an end arises as a set of (di)natural transformations. Given a small category $\mathbb{C}$ and functors $F, G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Set}$, we obtain a bifunctor

$$[F(+,-), G(-,+)] \ :\ \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbf{Set}\ ,\quad (X,Y) \longmapsto [F(Y,X), G(X,Y)]\ . \tag{4}$$

Here $[S,T]$ denotes the set of functions from S to T, i.e. an exponential in $\mathbf{Set}$. Note the variance: since $[-,+]$ is contravariant in its first argument, the variance of arguments of F is opposed in (4). Taking this functor (4) as F in Def. 2.1, we define an end $\int_X [F(X,X), G(X,X)]$. Such an end does exist when $\mathbb{C}$ is a small category, because $\mathbf{Set}$ has small limits (hence small ends).

Proposition 2.2 *Let us denote the set of dinatural transformations from F to G by* $\mathrm{Dinat}(F,G)$*. We have a canonical isomorphism*

$$\mathrm{Dinat}(F,G) \xrightarrow{\cong} \int_X [F(X,X), G(X,X)]\ .$$

Proof It is due to the following correspondences.

$$\begin{array}{c} 1 \to \int_X \left[F\left(X,X\right), G\left(X,X\right)\right] \\ \hline\hline 1 \to \left[F\left(X,X\right), G\left(X,X\right)\right] \text{ dinatural in } X \\ \hline\hline F\left(X,X\right) \to G\left(X,X\right) \text{ dinatural in } X \end{array} \quad \begin{array}{l} (\dagger) \\ (\ddagger) \end{array}$$

Here (†) is by Def. 2.1; dinaturality is preserved along (‡) because of the naturality of Currying. 2

The composite $\mathrm{Dinat}(F,G) \xrightarrow{\cong} \int_X [F(X,X), G(X,X)] \xrightarrow{\pi_X} [F(X,X), G(X,X)]$ carries a dinatural transformation φ to its X-component φ_X.

Since dinaturality subsumes naturality (§2.1), we have an immediate corollary:

Corollary 2.3 *Let $\mathbb{C}$ be a small category and $F, G : \mathbb{C} \to \mathbf{Set}$. By $\mathrm{Nat}(F,G)$ we denote the set of natural transformations $F \Rightarrow G$. We have*

$$\mathrm{Nat}(F,G) \xrightarrow{\cong} \int_X [FX, GX] \ . \qquad 2$$

The celebrated *Yoneda lemma* reduces the set $\mathrm{Nat}(\mathbb{C}(X,\_), F)$ of natural transformations into FX (see e.g. [20, 6]). Interpreted via Cor. 2.3, it yields:

Lemma 2.4 *(The Yoneda lemma, end-form) Given a small category $\mathbb{C}$ and a functor $F : \mathbb{C} \to \mathbf{Set}$, we have a canonical isomorphism*

$$\int_{X' \in \mathbb{C}} \left[\mathbb{C}\left(X, X'\right), FX'\right] \xrightarrow{\cong} FX \ . \qquad 2$$

The lemma becomes extremely useful in the calculations in the sequel: it means an end on the LHS "cancels" with a hom-functor occurring in it.

From the end-form, we obtain the following coend-form. Its proof is easy but illuminating.

Lemma 2.5 *(The Yoneda lemma, coend-form) Given a small category $\mathbb{C}$ and a functor $F : \mathbb{C} \to \mathbf{Set}$, we have a canonical isomorphism*

$$\int^{X' \in \mathbb{C}} FX' \times \mathbb{C}(X', X) \xrightarrow{\cong} FX \ .$$

Proof We have the following canonical isomorphisms, for each $S \in \mathbf{Set}$.

$$\begin{array}{lll} \left[\, \int^{X'} FX' \times \mathbb{C}(X',X)\,,\, S \,\right] & \overset{\cong}{\Rightarrow} \int_{X'} \left[\, FX' \times \mathbb{C}(X',X)\,,\, S \,\right] & (\dagger) \\ & \overset{\cong}{\Rightarrow} \int_{X'} \left[\, \mathbb{C}(X',X)\,,\, [FX', S] \,\right] & \text{Currying} \\ & \overset{\cong}{\Rightarrow} [FX, S] & \text{the Yoneda lemma, end-form.} \end{array}$$

Here (†) is because the hom-functor $[\_, S]$ turns a colimit into a limit [20, §V.4], hence a coend into an end. Obviously the composite isomorphism is natural in S; therefore we have shown that

$$\mathbf{y}\big(\textstyle\int^{X'} \mathbb{C}(X',X) \times FX'\big) \xrightarrow{\cong} \mathbf{y}(FX) \quad : \ \mathbb{C} \longrightarrow \mathbf{Set} \ , \qquad (5)$$

where $\mathbf{y} : \mathbb{C}^{\mathrm{op}} \to [\mathbb{C}, \mathbf{Set}]$ is the (contravariant) Yoneda embedding. By the Yoneda lemma the functor $\mathbf{y}$ is full and faithful; therefore it reflects isomorphisms. Hence (5) proves the claim. 2

This coend-form allows us to "cancel" a coend with a hom-functor inside it.

2.3 Profunctor

Definition 2.6 Let $\mathbb{C}$ and $\mathbb{D}$ be small categories. A *profunctor* P from $\mathbb{C}$ to $\mathbb{D}$ is a functor $P : \mathbb{D}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Set}$. It is denoted by $P : \mathbb{C} \nrightarrow \mathbb{D}$ (see on the right).

$$\begin{array}{c} \mathbb{C} \nrightarrow \mathbb{D} \\ \hline\hline \mathbb{D}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbf{Set} \end{array}$$

The notion of profunctor is also called *distributor*, *bimodule* or *module*. For more detailed treatment of profunctors see e.g. [7,9].

There are principally two ways to understand profunctors. One is as "generalized relations": profunctors are to functors what relations are to functions. The differences between a profunctor $P : \mathbb{C} \nrightarrow \mathbb{D}$ and a relation $R : S \nrightarrow T$ are as follows.

- A relation is two-valued: for each element $s \in S$ and $t \in T$, $R(s,t)$ is either empty (i.e. $(s,t) \notin R$) or filled (i.e. $(s,t) \in R$). In contrast, a profunctor is valued with arbitrary sets, that is, $P(Y,X) \in \mathbf{Set}$.
- The functoriality of a profunctor P induces *action* of morphisms in $\mathbb{C}$ and $\mathbb{D}$. For illustration let us depict an element $p \in P(Y,X)$ by a box $\xrightarrow{Y}\boxed{p}\xrightarrow{X}$. Given two morphisms $g : Y' \to Y$ in $\mathbb{D}$ and $f : X \to X'$ in $\mathbb{C}$, functoriality of P yields an element $P(g,f)(p) \in P(Y',X')$ (note the variance); the latter element is best depicted as follows.
$$\xrightarrow{Y'}\textcircled{g}\xrightarrow{Y}\boxed{p}\xrightarrow{X}\textcircled{f}\xrightarrow{X'} \tag{6}$$

The latter point motivates a different way of looking at profunctors: as generalized *modules* as in the theory of rings. These generalized modules are carried by a family of sets $\{P(Y,X)\}_{X\in\mathbb{C},Y\in\mathbb{D}}$, with left-action of $\mathbb{C}$-arrows and right-action of $\mathbb{D}$-arrows. Also notice the similarity between (6) and the diagrams in §1 for computations/components. It is indeed this similarity that allows us to formalize arrows as certain profunctors (§3).

Definition 2.7 (Composition of profunctors) Given two successive profunctors $P : \mathbb{C} \nrightarrow \mathbb{D}$ and $Q : \mathbb{D} \nrightarrow \mathbb{E}$, their *composition* $Q \circ P : \mathbb{C} \nrightarrow \mathbb{E}$ is defined by the following coend. For $U \in \mathbb{E}$ and $X \in \mathbb{C}$,

$$(Q \circ P)(U,X) = \int^{Y\in\mathbb{D}} Q(U,Y) \times P(Y,X) \quad .$$

For profunctors as generalized relations, this composition operation corresponds to a *relational composition*: $(S \circ R)(x,z)$ if and only if $\exists y.\bigl(R(x,y) \wedge S(y,z)\bigr)$. For profunctors as modules, it corresponds to *tensor product* of modules. In any case, recall from §2.1 that the coend in Def. 2.7 is a coproduct $\coprod_Y Q(U,Y) \times P(Y,X)$—a bunch of pairs $(\xrightarrow{U}\boxed{q}\xrightarrow{Y}, \xrightarrow{Y}\boxed{p}\xrightarrow{X})$, with varying Y—quotiented modulo a certain equivalence $\simeq$. This equivalence $\simeq$ (dictated by dinaturality) intuitively says: the choice of intermediate $Y \in \mathbb{D}$ does not matter. Specifically, the equivalence $\simeq$ is generated by the following relation; here $f : Y \to Y'$ is a morphism in $\mathbb{D}$.

$$\Bigl(\xrightarrow{U}\boxed{q}\xrightarrow{Y}\textcircled{f}\xrightarrow{Y'},\ \xrightarrow{Y'}\boxed{p}\xrightarrow{X}\Bigr)\ \simeq\ \Bigl(\xrightarrow{U}\boxed{q}\xrightarrow{Y},\ \xrightarrow{Y}\textcircled{f}\xrightarrow{Y'}\boxed{p}\xrightarrow{X}\Bigr)\ .$$

$$\begin{tikzcd}
\mathbb{C} \arrow[r, bend left, "P", "|" marking] \arrow[r, bend right, "Q"', "|" marking] \arrow[r, phantom, "\Downarrow\psi"] & \mathbb{D}
\end{tikzcd}$$

An appropriate notion of *morphism* between parallel profunctors $P, Q : \mathbb{C} \nrightarrow \mathbb{D}$ is provided by a natural transformation $\psi : P \Rightarrow Q$, where P and Q are thought of as functors $P, Q : \mathbb{D}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Set}$. All these data can be organized in a "2-categorical" manner as on the right. A problem now is that (horizontal) composition of 1-cells (i.e. profunctors) is not strictly associative: due to Def. 2.7 of composition by coends and products, associativity can be only ensured up-to coherent isomorphisms. The same goes for unitality; therefore profunctors form a *bicategory* (see [9]) instead of a 2-category.

Definition 2.8 (The bicategory **Prof**) The bicategory **Prof** has small categories as 0-cells, profunctors as 1-cells and natural transformations between them as 2-cells. The identity 1-cell $\mathbb{C} \nrightarrow \mathbb{C}$ is given by the hom-functor $\mathbb{C}(-,+) : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Set}$; it is the unit for composition because of the Yoneda lemma, coend-form (Lem. 2.5).

2.4 Some Properties of **Prof**

Here we describe some structural properties of **Prof** that will be exploited later, namely direct/inverse image of functors and a tensor product. For the former [7] is a principal reference; the notes [10] are not specifically on profunctors but provide useful insights into relevant mathematical concepts.

A function $f : S \to T$ induces two relations: the *direct image* $f_* : S \nrightarrow T$ and the *inverse image* $f^* : T \nrightarrow S$, defined by: $f_*(s,t)$ iff $t = f(s)$ iff $f^*(t,s)$. There are analogous constructions from functors to profunctors.

Definition 2.9 Let $F : \mathbb{C} \to \mathbb{D}$ be a functor between small categories. It gives rise to

$$\begin{array}{rll}
\text{the \emph{direct image} profunctor} & F_* : \mathbb{C} \longrightarrow\!\!\!\!\!\!\!\!\shortmid\;\;\;\; \mathbb{D} & \text{by } F_*(Y,X) = \mathbb{D}(Y,FX)\ ; \\
\text{the \emph{inverse image} profunctor} & F^* : \mathbb{D} \longrightarrow\!\!\!\!\!\!\!\!\shortmid\;\;\;\; \mathbb{C} & \text{by } F^*(X,Y) = \mathbb{D}(FX,Y)\ .
\end{array}$$

The mapping $(\_)_*$ also applies to natural transformations in an obvious way; this determines a *pseudo functor* (see e.g. [9]) $(\_)_* : \mathbf{Cat} \to \mathbf{Prof}$ that embeds **Cat** in **Prof**.

The following relationship between direct and inverse images is fundamental.

Lemma 2.10 *Given a functor $F : \mathbb{C} \to \mathbb{D}$, its direct and inverse images form an adjunction $F_* \dashv F^*$ internally in* **Prof**. *That is, there are canonical 2-cells*

$$\begin{tikzcd}
\mathbb{C} \arrow[r, "\mathrm{id}", "|" marking] \arrow[d, "F_*"', "|" marking] & \mathbb{C} \arrow[phantom, dl, "\Downarrow\eta", near start] \\
\mathbb{D} \arrow[ur, "F^*"', "|" marking] &
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \mathbb{C} \arrow[d, "F_*", "|" marking] \\
\mathbb{D} \arrow[ur, "F^*", "|" marking] \arrow[r, "\mathrm{id}"', "|" marking] \arrow[r, phantom, "\Downarrow\varepsilon", shift left=2ex] & \mathbb{D}
\end{tikzcd}$$

that satisfy the "triangular identities" (see e.g. [20]):

$$\begin{tikzcd}
\mathbb{C} \arrow[r, "\mathrm{id}", "\shortmid"{marking}] \arrow[d, "F_*"', "\shortmid"{marking}] \arrow[dr, phantom, "\Downarrow\eta", near start] & \mathbb{C} \arrow[d, "F_*"', "\shortmid"{marking}] \\
\mathbb{D} \arrow[ru, "\shortmid"{marking}] \arrow[r, "\mathrm{id}"', "\shortmid"{marking}] \arrow[ur, phantom, "\Downarrow\varepsilon", near end, shift right=3] & \mathbb{D}
\end{tikzcd} = \mathrm{id} \quad ; \qquad
\begin{tikzcd}
 & \mathbb{C} \arrow[r, "\mathrm{id}", "\shortmid"{marking}] \arrow[d, "F_*", "\shortmid"{marking}] \arrow[dr, phantom, "\Downarrow\eta", near start] & \mathbb{C} \\
\mathbb{D} \arrow[ru, "F^*", "\shortmid"{marking}] \arrow[r, "\mathrm{id}"', "\shortmid"{marking}] \arrow[ur, phantom, "\Downarrow\varepsilon", near end, shift right=3] & \mathbb{D} \arrow[ru, "F^*"', "\shortmid"{marking}] &
\end{tikzcd} = \mathrm{id} \quad .$$

Note that "internal" adjunction, when it is in the 2-category **Cat** of categories and functors, coincides with the usual notion of adjunction.

Proof The 2-cells η and ε are natural transformations of the following types.

$$\begin{aligned}
\eta : \qquad & \mathbb{C}(-,+) \Longrightarrow \textstyle\int^{Y} \mathbb{D}(F(-),Y) \times \mathbb{D}(Y,F(+)) && \text{by Def. 2.7} \\
& \xrightarrow{\cong} \mathbb{D}(F(-),F(+)) && \text{by Lem. 2.5} \\
\varepsilon : \quad & \textstyle\int^{X} \mathbb{D}(-,FX) \times \mathbb{D}(FX,+) \Longrightarrow \mathbb{D}(-,+) && \text{by Def. 2.7}
\end{aligned}$$

The former is given by the functor F's action on morphisms; the latter is by composition of morphisms in $\mathbb{D}$. Checking the equalities is easy using Lem. 2.5. 2

Notations 2.11 Throughout the rest of the paper, the direct image F_* of a functor F shall be simply denoted by F. We shall not omit $(\_)^*$ for inverse images. The identity profunctor $\mathrm{id} : \mathbb{C} \nrightarrow \mathbb{C}$—that is the hom-functor—will be often denoted by $\mathbb{C} : \mathbb{C} \nrightarrow \mathbb{C}$.

The Cartesian product operator × in **Cat** lifts **Prof**; given profunctors $F : \mathbb{C} \nrightarrow \mathbb{C}'$ and $G : \mathbb{D} \nrightarrow \mathbb{D}'$, we define

$$F \times G : \mathbb{C} \times \mathbb{D} \nrightarrow \mathbb{C}' \times \mathbb{D}' \quad \text{by} \quad (F \times G)(X',Y',X,Y) = F(X',X) \times G(Y',Y) \ . \tag{7}$$

The symbol × occurring in the last denotes the Cartesian product in **Set**. The lifted operator × in **Prof** makes it a "monoidal bicategory," a notion whose precise definition involves delicate handling of coherence. We shall not do that in this paper. Nevertheless, we will need the following property.

Lemma 2.12 *The operation* × *on* **Prof** *is bifunctorial: that is, given four profunctors* $\mathbb{C} \overset{P}{\nrightarrow} \mathbb{D} \overset{Q}{\nrightarrow} \mathbb{E}$ *and* $\mathbb{C}' \overset{P'}{\nrightarrow} \mathbb{D}' \overset{Q'}{\nrightarrow} \mathbb{E}'$ *we have* $(Q \circ P) \times (Q' \circ P') \xrightarrow{\cong} (Q \times Q') \circ (P \times P')$.

Proof This is due to the Fubini theorem for coends. See [20, §IX.8] 2

It is obvious that the operator × acts also on 2-cells (that are natural transformations).

3 Arrows as Profunctors

We review the results in [16, 1] that identify Hughes' notion of arrow with a profunctor with additional algebraic structure.

First we present the precise definition of arrow. Usually it is defined over a Cartesian category $\mathbb{C}$. However, since it is rather the monoidal structure of $\mathbb{C}$ that is essential, we shall work with a monoidal category.

Definition 3.1 (Arrow [13]) Given a monoidal category $\mathbb{C} = (\mathbb{C}, \otimes, I)$, an *arrow* over $\mathbb{C}$ consists of carrier sets $\{A(J,K)\}_{J,K\in\mathbb{C}}$ and operators arr, $\ggg$ and first as described in §1.1. The operators must satisfy the following equational axioms.

$$
\begin{aligned}
(a \ggg b) \ggg c &= a \ggg (b \ggg c) && (\ggg\text{-Assoc})\\
\mathsf{arr}\,(g \circ f) &= \mathsf{arr}\, f \ggg \mathsf{arr}\, g && (\mathsf{arr}\text{-Func1})\\
\mathsf{arr}\,\mathrm{id}_J \ggg_{J,J,K} a = a &= a \ggg_{J,K,K} \mathsf{arr}\,\mathrm{id}_K && (\mathsf{arr}\text{-Func2})\\
\mathsf{first}_{J,K,I}\, a \ggg \mathsf{arr}\,\rho_K &= \mathsf{arr}\,\rho_K \ggg a && (\rho\text{-Nat})\\
\mathsf{first}_{J,K,L}\, a \ggg \mathsf{arr}(\mathrm{id}_K \otimes f) &= \mathsf{arr}(\mathrm{id}_J \otimes f) \ggg \mathsf{first}_{J,K,M}\, a && (\mathsf{arr}\text{-Centr})\\
(\mathsf{first}_{J,K,L\otimes M}\, a) \ggg (\mathsf{arr}\,\alpha_{K,L,M}) &= (\mathsf{arr}\,\alpha_{J,L,M}) \ggg \mathsf{first}(\mathsf{first}\, a) && (\alpha\text{-Nat})\\
\mathsf{first}_{J,K,L}(\mathsf{arr}\, f) &= \mathsf{arr}(f \otimes \mathrm{id}_L) && (\mathsf{arr}\text{-Premon})\\
\mathsf{first}_{J,L,M}(a \ggg b) &= (\mathsf{first}_{J,K,M}\, a) \ggg (\mathsf{first}_{K,L,M}\, b) && (\mathsf{first}\text{-Func})
\end{aligned}
$$

Here some subscripts are suppressed. The morphism $\rho_K : K \otimes I \xrightarrow{\cong} K$ is the right unitor isomorphism; α denotes an associator isomorphism. The names of the axioms hint their correspondence to the (premonoidal) structure of *Freyd categories* [23,19].

Next we introduce the corresponding construct in **Prof**, which we shall tentatively call a **Prof**-*arrow*.

Definition 3.2 Let $\mathbb{C} = (\mathbb{C}, \otimes, I)$ be a small monoidal category. A **Prof**-*arrow* over $\mathbb{C}$ is:

- a profunctor $A : \mathbb{C} \nrightarrow \mathbb{C}$,
- equipped with natural transformations $\mathsf{arr}, \ggg, \mathsf{first}$ of the following types:

$$
\begin{tikzcd}
\mathbb{C} \arrow[r, bend left, "\mathbb{C}"] \arrow[r, bend right, "A"'] \arrow[r, phantom, "\Downarrow \mathsf{arr}"] & \mathbb{C}
\end{tikzcd}
\;,\qquad
\begin{tikzcd}
\mathbb{C} \arrow[r, "A"] \arrow[rr, bend right, "A"'] & \mathbb{C} \arrow[r, "A"] \arrow[d, phantom, "\Downarrow \ggg"] & \mathbb{C} \\
& {} &
\end{tikzcd}
\;,\qquad
\begin{tikzcd}
\mathbb{C}^2 \arrow[r, "A\times\mathbb{C}"] \arrow[d, "\otimes"'] \arrow[dr, phantom, "\Downarrow \mathsf{first}"] & \mathbb{C}^2 \arrow[d, "\otimes"] \\
\mathbb{C} \arrow[r, "A"'] & \mathbb{C}
\end{tikzcd}
\;,
$$

  where all the diagrams are in **Prof**,
- subject to the equalities in Table 1. Recall Notations 2.11; for example the profunctor $\langle \mathbb{C}, I\rangle$ in $(\mathsf{first}\text{-}\rho)$ is the functor $\langle \mathbb{C}, I\rangle : X \mapsto (X, I)$, embedded in **Prof** by taking its direct image.

The notion of **Prof**-arrow is in fact a familiar one: it is an internal *strong monad* in **Prof**. Indeed, when one draws the same 2-cells in **Cat** instead of in **Prof**—replacing A by T, arr by η^T, $\ggg$ by μ^T and first by str'—the definition coincides with that of strong monad [18, 21].[2] More specifically, the first two axioms in Table 1 are for the monad laws; and the remaining axioms asserts compatibility of strength with monoidal and monad structure. For example, the axiom $(\mathsf{first}\text{-}\ggg)$

[2] The corresponding strength operator str' is of the type $\mathsf{str}' : TX \otimes Y \to T(X \otimes Y)$, which is slightly different from the usual strength operator that is $\mathsf{str} : X \otimes TY \to T(X \otimes Y)$.

interpreted in **Cat** is read as the commutativity of the following diagram.

$$\begin{tikzcd}
T^2X \otimes Y \arrow[r, "\mathsf{str}'"] \arrow[d, "\mu^T \otimes Y"'] & T(TX \otimes Y) \arrow[r, "T\mathsf{str}'"] & T^2(X \otimes Y) \arrow[d, "\mu^T"] \\
TX \otimes Y \arrow[rr, "\mathsf{str}'"'] & & T(X \otimes Y)
\end{tikzcd}$$

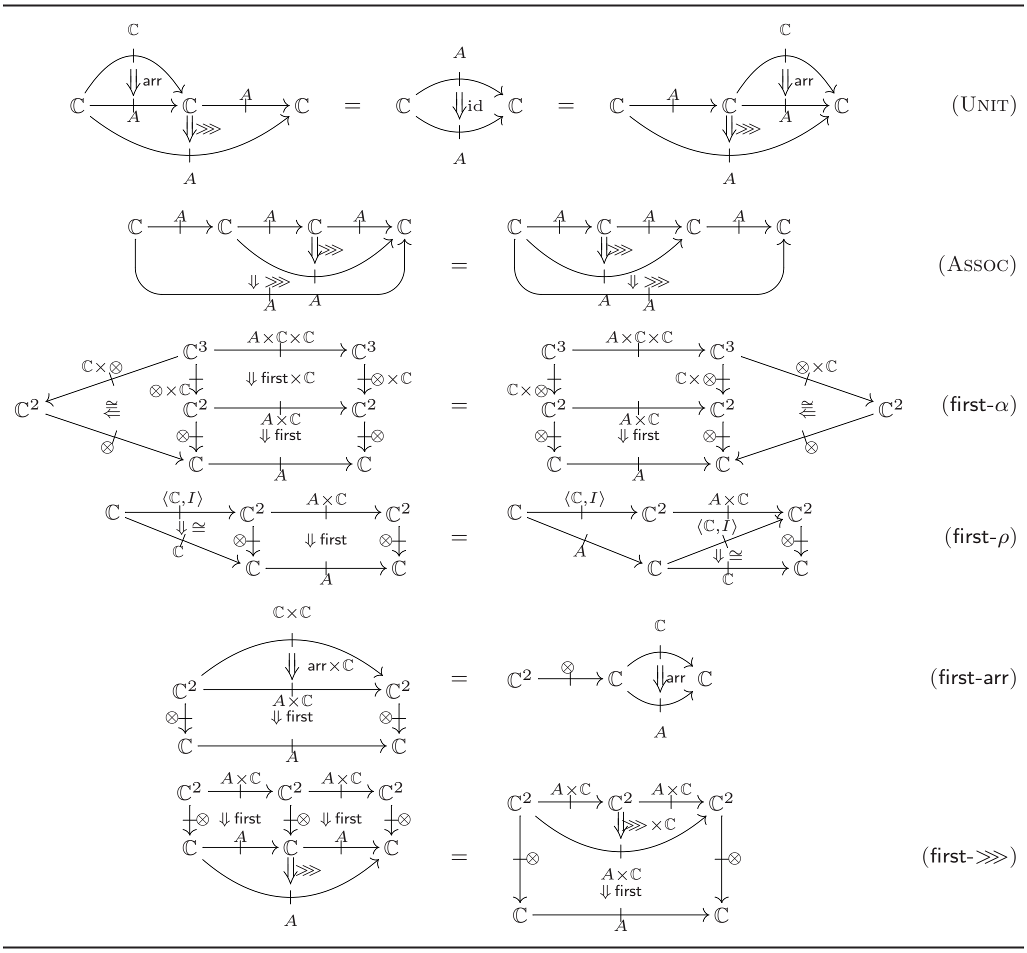


Table 1
Equational axioms for **Prof**-arrow

Proposition 3.3 *[1] For a monoidal category $\mathbb{C}$ that is small, the notion of arrow (Def. 3.1) and that of* **Prof***-arrow (Def. 3.2) are equivalent.*

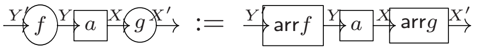
Proof While the reader is referred to [1] for a detailed proof, we shall illustrate a few highlights in the correspondence between the two notions. We shall write arr', $\ggg'$ and first' (with primes) for the three operators of a **Prof**-arrow (Def. 3.2), to distinguish them from the corresponding operators of an arrow (Def. 3.1).

Let us first observe that a 2-cell first' in **Prof** gives rise to the first operator in Def. 3.1. The former is an element of the LHS below, where $\ggg$ denotes composition of profunctors (Def. 2.7).

$$
\begin{aligned}
&\mathrm{Nat}\big(\,(\otimes\circ(A\times\mathbb{C}))(-,+_1,+_2)\,,\ (A\circ\otimes)(-,+_1,+_2)\,\big)\\
&\cong \textstyle\int_{X,K,Y\in\mathbb{C}}\big[\,(\otimes\circ(A\times\mathbb{C}))(X,K,Y)\,,\ (A\circ\otimes)(X,K,Y)\,\big] && \text{by Cor. 2.3}\\
&\cong \textstyle\int_{X,K,Y}\big[\,\int^{J,L}\mathbb{C}(X,J\otimes L)\times A(J,K)\times\mathbb{C}(L,Y)\,,\ \int^{U}A(X,U)\times\mathbb{C}(U,K\otimes Y)\,\big]\\
&&& \text{by Def. 2.7, Def. 2.9 and (7)}\\
&\cong \textstyle\int_{X,K,Y,J,L}\big[\,\mathbb{C}(X,J\otimes L)\times A(J,K)\times\mathbb{C}(L,Y)\,,\ \int^{U}A(X,U)\times\mathbb{C}(U,K\otimes Y)\,\big]\\
&&& \text{since a hom-functor } [-,S] \text{ turns a coend into an end}\\
&\cong \textstyle\int_{X,K,Y,J,L}\big[\,\mathbb{C}(X,J\otimes L),\ \big[\,A(J,K),\ \big[\,\mathbb{C}(L,Y)\,,\ \int^{U}A(X,U)\times\mathbb{C}(U,K\otimes Y)\,\big]\big]\big]\\
&&& \text{by Currying}\\
&\cong \textstyle\int_{J,K,L}\big[\,A(J,K),\ A(J\otimes L,K\otimes L)\,\big]\\
&&& \text{by canceling } X,Y \text{ by Lem. 2.4 and } U \text{ by Lem. 2.5}\\
&\cong \mathrm{Nat}_{J,K}\mathrm{Dinat}_{L}\big(\,A(J,K),\ A(J\otimes L,K\otimes L)\,\big) && \text{by Prop. 2.2 and Cor. 2.3.}
\end{aligned}
$$

Therefore a 2-cell first' in **Prof** gives rise to a family of functions $A(J,K)\to A(J\otimes L,K\otimes L)$ that is natural in J,K and dinatural in L. This is precisely the type of the first operator in Def. 3.1. The equational axioms of an arrow are indeed satisfied due to those of a **Prof**-arrow. We note that the axiom (arr-Centr) is satisfied not because of any specific axiom of a **Prof**-arrow, but because of the dinaturality of first' as a 2-cell in **Prof**.

For the reverse direction where an arrow induces a **Prof**-arrow, we have to equip the carrier $\{A(J,K)\}_{J,K}$ of an arrow with action of morphisms in $\mathbb{C}$, rendering A into a functor $\mathbb{C}^{\mathrm{op}}\times\mathbb{C}\to\mathbf{Set}$. This is done with the help of arrow operators. Specifically,

$A(g,f)(a) := \mathsf{arr}\,f \ggg a \ggg \mathsf{arr}\,g$, that is  .

Each of the arrow operators yield its corresponding **Prof**-arrow operator; the latter's (di)naturality is derived from the arrow axioms. So are the equational axioms for a **Prof**-arrow. 2

Prop. 3.3 offers a novel mathematical understanding of the notion of arrow. Its axiomatization seems to have stronger justifications than the original one (Def. 3.1) does. It also seems simpler than the treatment of first in Freyd categories which involves technicalities like premonoidal categories and central morphisms. It is this simplicity that is exploited in the rest of the paper.

When the base monoidal category $\mathbb{C}$ is symmetric—which is our setting in the sequel—we can obtain another sideline operator second.

Definition 3.4 Let A be an arrow over a small symmetric monoidal category (SMC) $\mathbb{C}$. We define an extra operator second as the following 2-cell in **Prof**.

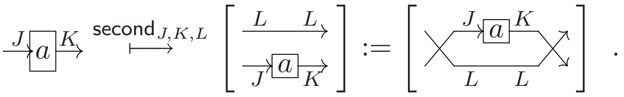
$$
\begin{tikzcd}
\mathbb{C}^2 \arrow[r, "\mathbb{C}\times A", "|" marking] \arrow[d, "\otimes", "|" marking] \arrow[dr, phantom, "\Downarrow\mathsf{second}"] & \mathbb{C}^2 \arrow[d, "\otimes", "|" marking] \\
\mathbb{C} \arrow[r, "A"', "|" marking] & \mathbb{C}
\end{tikzcd}
\quad := \quad
\begin{tikzcd}
\mathbb{C}^2 \arrow[r, "\mathbb{C}\times A", "|" marking] \arrow[d, "\langle\pi_2,\pi_1\rangle", "|" marking] \arrow[dd, bend right=60, "\otimes"', "|" marking] & \mathbb{C}^2 \arrow[d, "\langle\pi_2,\pi_1\rangle"', "|" marking] \arrow[dd, bend left=60, "\otimes", "|" marking] \\
\mathbb{C}^2 \arrow[r, "A\times\mathbb{C}", "|" marking] \arrow[d, "\otimes"', "|" marking] \arrow[dr, phantom, "\Downarrow\mathsf{first}"] \arrow[l, phantom, "\overset{\cong}{\Leftarrow}"] & \mathbb{C}^2 \arrow[d, "\otimes", "|" marking] \arrow[r, phantom, "\overset{\cong}{\Leftarrow}"] \\
\mathbb{C} \arrow[r, "A"', "|" marking] & \mathbb{C}
\end{tikzcd}
\tag{8}
$$

Here the profunctor $\langle \pi_2, \pi_1 \rangle$ is the direct image of the functor $\langle \pi_2, \pi_1 \rangle : \mathbb{C}^2 \to \mathbb{C}^2$, mapping (X, Y) to (Y, X) (cf. Notations 2.11).

Notations 3.5 In the above diagrams as well as elsewhere, there appear two different classes of iso 2-cells in **Prof**. One class is due to the unitality/associativity/symmetry of $\otimes$ on a monoidal base category $\mathbb{C}$; they are iso 2-cells in **Cat** embedded in **Prof** via direct image (§2.4). Such iso 2-cells shall be filled explicitly with the $\cong$ sign, like the two on the RHS in (8).

The other class is due to the properties of the operation $\times$ on **Prof**, typically Lem. 2.12. Such iso 2-cells will be denoted by empty polygons, like the one on the RHS in (8).

Some calculations like in the proof of Prop. 3.3 reveal that this new operator realizes a class of functions $A(J, K) \xrightarrow{\mathsf{second}_{J,K,L}} A(L \times J, L \times K)$, that is graphically



Lemma 3.6 *Between the* first *and* second *operators, the following equality holds.*

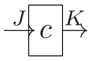
$$\begin{tikzcd}
 & \mathbb{C}^3 \arrow[ld, "\mathbb{C}\times\otimes"'] \arrow[r, "\mathbb{C}\times A\times\mathbb{C}"] \arrow[d, "\otimes\times\mathbb{C}"'] \arrow[rd, phantom, "\Downarrow \mathsf{second}\times\mathbb{C}"] & \mathbb{C}^3 \arrow[d, "\otimes\times\mathbb{C}"] \arrow[rd, "\mathbb{C}\times\otimes"] & & & \mathbb{C}^3 \arrow[r, "\mathbb{C}\times A\times\mathbb{C}"] \arrow[d, "\mathbb{C}\times\otimes"'] \arrow[rd, phantom, "\Downarrow \mathbb{C}\times\mathsf{first}"] & \mathbb{C}^3 \arrow[d, "\mathbb{C}\times\otimes"] \\
\mathbb{C}^2 \arrow[rd, "\otimes"'] \arrow[r, phantom, "\overset{\cong}{\Leftarrow}"] & \mathbb{C}^2 \arrow[r, "A\times\mathbb{C}"] \arrow[d, "\otimes"'] \arrow[rd, phantom, "\Downarrow \mathsf{first}"] & \mathbb{C}^2 \arrow[d, "\otimes"] \arrow[r, phantom, "\overset{\cong}{\Leftarrow}"] & \mathbb{C}^2 \arrow[ld, "\otimes"] & = & \mathbb{C}^2 \arrow[r, "\mathbb{C}\times A"] \arrow[d, "\otimes"'] \arrow[rd, phantom, "\Downarrow \mathsf{second}"] & \mathbb{C}^2 \arrow[d, "\otimes"] \\
 & \mathbb{C} \arrow[r, "A"'] & \mathbb{C} & & & \mathbb{C} \arrow[r, "A"'] & \mathbb{C}
\end{tikzcd}$$

Proof Use the equality (first-α) and the coherence for an SMC $\mathbb{C}$. 2

4 Arrow-Based Components

In this section we develop the scenario in §1.4 in technical terms. First we introduce an arrow-based coalgebraic modeling of components.

Definition 4.1 (A-component) Let A be an arrow on **Set**, and $J, K \in$ **Set**. An *(arrow-based) A-component* with input-type J, output-type K and *computational structure* A is a coalgebra for the functor $A(J, \_ \times K) : \mathbf{Set} \to \mathbf{Set}$. That is,



as

$$\begin{tikzcd}
A(J, X \times K) \\
X \arrow[u, "c"']
\end{tikzcd} .$$

Here an arrow A is in the sense of Def. 3.1. There the base $\mathbb{C}$ of an arrow need not be small; thus we choose $(\mathbf{Set}, \times, 1)$ as $\mathbb{C}$. Our modeling specializes to Barbosa's (2) when we take as A a monad-based arrow A_T (§1.1). Our modeling not only generalizes Barbosa's one but also brings conceptual clarity to the subsequent arguments.

Our goal is to lift the arrow structure of A to the categorical arrow structure of A-components. Let us make this goal precise.

Definition 4.2 (Categorical arrow) A *categorical arrow* consists of

- a family $\{\mathcal{A}(J,K)\}_{J,K}$ of *carrier* categories indexed by $J,K \in \mathbf{Set}$;
- (interpretation of) arrow operators arr, $\ggg$ and first (cf. Def. 3.1), namely functors

$$\begin{array}{cccl}
\mathbf{1} & \xrightarrow{\mathsf{arr}\,f} & \mathcal{A}(J,K) & \text{for each function } f : J \to K \text{ in } \mathbf{Set}, \\
\mathcal{A}(J,K)\times\mathcal{A}(K,L) & \xrightarrow{\ggg_{J,K,L}} & \mathcal{A}(J,L) & \text{for each } J,K,L \in \mathbf{Set}, \\
\mathcal{A}(J,K) & \xrightarrow{\mathsf{first}_{J,K,L}} & \mathcal{A}(J\times L, K\times L) & \text{for each } J,K,L \in \mathbf{Set}.
\end{array}$$

Here the category $\mathbf{1}$ is the one-object and one-arrow (i.e. terminal) category; and

- the operators are subject to the arrow axioms in Def. 3.1, up-to isomorphisms. For example, as to the axiom ($\ggg$-Assoc), the following diagram must commute up-to an isomorphism.

$$\begin{tikzcd}
\mathcal{A}(J,K)\times\mathcal{A}(K,L)\times\mathcal{A}(L,M) \arrow[r, "\ggg_{J,K,L}\times\mathrm{id}"] \arrow[d, "\mathrm{id}\times\ggg_{K,L,M}"'] \arrow[dr, phantom, "\Downarrow\cong"] & \mathcal{A}(J,L)\times\mathcal{A}(L,M) \arrow[d, "\ggg_{J,L,M}"] \\
\mathcal{A}(J,K)\times\mathcal{A}(K,M) \arrow[r, "\ggg_{J,K,M}"'] & \mathcal{A}(J,M)
\end{tikzcd} \tag{9}$$

The graphical understanding of a categorical arrow is the same as that of an arrow; see §1.1. In §1.3 we described why it is natural and necessary to require the axioms be satisfied only up-to isomorphisms.

Remark 4.3 Satisfaction up-to isomorphisms raises a *coherence* issue. The precise coherence condition for categorical arrows is described in [11], in a more general form of coherence for categorical models of FP-theories. Although we shall not further discuss the coherence issue, the calculations later in §5 provide us a much better grip on it than the direct calculations in [11] do.

The notion of categorical arrow in Def. 4.2 could be formalized on any monoidal category $\mathbb{C}$ other than $\mathbf{Set}$, although we do not need such additional generality.

The main contribution of this paper is the following result as well as its proof presented using the rest of the paper.

Theorem 4.4 *(Main contribution) Let A be an arrow on* $\mathbf{Set}$*. The categories* $\{\,\mathbf{Coalg}(A(J,\_\times K))\,\}_{J,K}$ *of A-components carry a categorical arrow.*

On top of it, we can appeal to the formalization [12, 11] of the *microcosm principle* [4] to obtain the following *compositionality result.*

Corollary 4.5 *In the setting of Thm. 4.4, assume further that for each $J,K \in$* $\mathbf{Set}$ *the functor $A(J,\_\times K)$ has a final coalgebra $\zeta_{J,K} : Z_{J,K} \xrightarrow{\cong} A(J, Z_{J,K}\times K)$.*

(i) *The family $\{Z_{J,K}\}_{J,K}$ is canonically an arrow.*

(ii) *Behaviors by coinduction are compositional with respect to arrow operators. For example, with respect to the operator $\ggg$, this means the following. Given two A-components $c : X \to A(J, X\times K)$ and $d : Y \to A(K, Y\times L)$ with* *matching I/O types, the triangle (∗) below commutes.*

$$\begin{tikzcd}
A(J,(X\times Y)\times L) \arrow[rr, dashed] & & A(J, Z_{J,L}\times L) \\
X\times Y \arrow[u, "c\ggg d"'] \arrow[rr, dashed, "\mathsf{beh}_{c\ggg d}"] \arrow[drr, "\mathsf{beh}_c\times\mathsf{beh}_d"'] & (*) & Z_{J,L} \arrow[u, "\mathit{final}"', "\cong"] \\
& & Z_{J,K}\times Z_{K,L} \arrow[u, "\ggg^Z"']
\end{tikzcd}$$

Here $c \ggg d$ is "composition of components" using the categorical arrow structure in Thm. 4.4; $\ggg^Z$ is "composition of behaviors" derived in (i); and $\mathsf{beh}_{c\ggg d}$ *is the behavior map for the composed components induced by coinduction (the square on the top).* 2

In [12, 11] it is shown that algebraic structure carried by the categories of coalgebras—like the one in Thm 4.4—can be obtained by:

- the same structure on the base categories, and
- the *lax compatibility* of the signature functors with the relevant algebraic structure.

In this case the algebraic structure on the base categories lifts to the categories of coalgebras. We shall follow this path. Restricting the general definitions and results in [12, 11] to the current setting, we obtain the following.

Definition 4.6 Let $\{F_{J,K} : \mathbf{Set} \to \mathbf{Set}\}_{J,K}$ be a family of endofunctors, indexed by $J, K \in \mathbf{Set}$. It is said to be a *lax arrow functor* if:

- it is equipped with the following natural transformations

$$\begin{aligned}
&F_{\mathsf{arr}\,f} \;:\; 1 \longrightarrow F_{J,K}(1) && \text{for each } f : J \to K \text{ in } \mathbf{Set};\\
&F_{\ggg_{J,K,L}} \;:\; F_{J,K}(X)\times F_{K,L}(Y) \longrightarrow F_{J,L}(X\times Y) && \text{natural in } X, Y, \text{ for each } J,K,L\in\mathbf{Set};\\
&F_{\mathsf{first}_{J,K,L}} \;:\; F_{J,K}(X) \longrightarrow F_{J\times L,K\times L}(X) && \text{natural in } X, \text{ for each } J,K,L\in\mathbf{Set};
\end{aligned}$$

- that are subject to the equations in Table 2, that are parallel to those in Def. 3.1. The diagrams there are all in **Set**; obvious subscripts are suppressed.

A lax arrow functor therefore looks like an arrow (think of $F_{J,K}(X)$ in place of $A(J,K)$), but it carries an extra parameter (like X, Y or $X\times Y$) around.

Proposition 4.7 *If $\{F_{J,K}\}_{J,K}$ is a lax arrow functor, then $\{\mathbf{Coalg}(F_{J,K})\}_{J,K}$ is canonically a categorical arrow.*

Proof This follows from a general result like [11, Thm. 4.6]. Here we shall briefly illustrate what the categorical arrow $\{\mathbf{Coalg}(F_{J,K})\}_{J,K}$ looks like, by describing the sequential composition operator $\ggg : \mathbf{Coalg}(F_{J,K})\times\mathbf{Coalg}(F_{K,L}) \to \mathbf{Coalg}(F_{J,L})$. Using the natural transformation $F_{\ggg}$ in Def. 4.6 it is defined as follows.

$$\begin{pmatrix} F_{J,K}X & & F_{K,L}Y \\ \uparrow c & , & \uparrow d \\ X & & Y \end{pmatrix} \overset{\ggg}{\longmapsto} \begin{array}{c} F_{J,L}(X\times Y) \\ \uparrow F_{\ggg} \\ F_{J,K}X\times F_{K,L}Y \\ \uparrow c\times d \\ X\times Y \end{array}$$

$$\boxed{(\ggg\text{-}\textsc{Assoc})}$$

$$\begin{tikzcd}
F_{J,K}X \times F_{K,L}Y \times F_{L,M}U \arrow[r, "\mathrm{id}\times F_{\ggg}"] \arrow[d, "F_{\ggg}\times\mathrm{id}"'] & F_{J,K}X \times F_{K,M}(Y \times U) \arrow[dd, "F_{\ggg}"] \\
F_{J,L}(X \times Y) \times F_{L,M}U \arrow[d, "F_{\ggg}"'] & \\
F_{J,M}((X \times Y) \times U) \arrow[r, "\cong"] & F_{J,M}(X \times (Y \times U))
\end{tikzcd}$$

$$\boxed{(\mathsf{arr}\text{-}\textsc{Func1})}$$

$$\begin{tikzcd}
1 \arrow[d, "\langle F_{\mathsf{arr}\,f}, F_{\mathsf{arr}\,g}\rangle"'] \arrow[ddr, "F_{\mathsf{arr}(g\circ f)}"] & \\
F_{J,K}1 \times F_{K,L}1 \arrow[d, "F_{\ggg}"'] & \\
F_{J,L}(1 \times 1) \arrow[r, "\cong"] & F_{J,L}1
\end{tikzcd}$$

$$\boxed{(\mathsf{arr}\text{-}\textsc{Func2})}$$

$$\begin{tikzcd}
F_{J,K}X \arrow[r, "\langle \mathrm{id}, F_{\mathsf{arr}\,\mathrm{id}_K}\rangle"] \arrow[d, "\langle F_{\mathsf{arr}\,\mathrm{id}_J}, \mathrm{id}\rangle"'] \arrow[ddr, "\mathrm{id}" description] & F_{J,K}X \times F_{K,K}1 \arrow[d, "F_{\ggg}"] \\
F_{J,J}1 \times F_{J,K}X \arrow[d, "F_{\ggg}"'] & F_{J,K}(X \times 1) \arrow[d, "\cong"'] \\
F_{J,L}(1 \times X) \arrow[r, "\cong"] & F_{J,K}X
\end{tikzcd}$$

$$\boxed{(\rho\text{-}\textsc{Nat})}$$

$$\begin{tikzcd}
F_{J,K}X \arrow[r, "\langle F_{\mathsf{arr}\,\pi_1}, \mathrm{id}\rangle"] \arrow[d, "F_{\mathsf{first}}"'] & F_{J\times 1,J}1 \times F_{J,K}X \arrow[d, "F_{\ggg}"] \\
F_{J\times 1,K\times 1}X \arrow[d, "\langle \mathrm{id}, F_{\mathsf{arr}\,\pi_1}\rangle"'] & F_{J\times 1,K}(1 \times X) \arrow[ddd, "\cong"'] \\
F_{J\times 1,K\times 1}X \times F_{K\times 1,K}1 \arrow[d, "F_{\ggg}"'] & \\
F_{J\times 1,K}(X \times 1) \arrow[r, "\cong"] & F_{J\times 1,K}X
\end{tikzcd}$$

$$\boxed{(\mathsf{arr}\text{-}\textsc{Centr})}$$

$$\begin{tikzcd}
F_{J,K}X \arrow[r, "F_{\mathsf{first}}"] \arrow[d, "F_{\mathsf{first}}"'] & F_{J\times L',K\times L'}X \arrow[d, "\langle F_{\mathsf{arr}(J\times f)}, \mathrm{id}\rangle"] \\
F_{J\times L,K\times L}X \arrow[d, "\langle \mathrm{id}, F_{\mathsf{arr}(K\times f)}\rangle"'] & F_{J\times L,J\times L'}1 \times F_{J\times L',K\times L'}X \arrow[d, "F_{\ggg}"] \\
F_{J\times L,K\times L}X \times F_{K\times L,K\times L'}1 \arrow[d, "F_{\ggg}"'] & F_{J\times L,K\times L'}(1 \times X) \arrow[d, "\cong"'] \\
F_{J\times L,K\times L'}(X \times 1) \arrow[r, "\cong"] & F_{J\times L,K\times L'}X
\end{tikzcd}$$

$$\boxed{(\alpha\text{-}\textsc{Nat})}$$

$$\begin{tikzcd}
F_{J,K}X \arrow[r, "F_{\mathsf{first}}"] \arrow[d, "F_{\mathsf{first}}"'] & F_{J\times L,K\times L}X \arrow[d, "F_{\mathsf{first}}"] \\
F_{J\times(L\times M),K\times(L\times M)}X \arrow[d, "\langle \mathrm{id}, F_{\mathsf{arr}\,\alpha}\rangle"'] & F_{(J\times L)\times M,(K\times L)\times M}X \arrow[d, "\langle F_{\mathsf{arr}\,\alpha}, \mathrm{id}\rangle"] \\
\begin{pmatrix} F_{J\times(L\times M),K\times(L\times M)}X \\ \times F_{K\times(L\times M),(K\times L)\times M}1 \end{pmatrix} \arrow[d, "F_{\ggg}"'] & \begin{pmatrix} F_{J\times(L\times M),(J\times L)\times M}1 \\ \times F_{(J\times L)\times M,(K\times L)\times M}X \end{pmatrix} \arrow[d, "\cong"'] \\
F_{J\times(L\times M),(K\times L)\times M}(X \times 1) \arrow[dr, "\cong"] & F_{J\times(L\times M),(K\times L)\times M}(1 \times X) \arrow[d, "\cong"'] \\
& F_{J\times(L\times M),(K\times L)\times M}X
\end{tikzcd}$$

$$\boxed{(\mathsf{arr}\text{-}\textsc{Premon})}$$

$$\begin{tikzcd}
1 \arrow[r, "F_{\mathsf{arr}\,f}"] \arrow[dr, "F_{\mathsf{arr}(f\times L)}"'] & F_{J,K}1 \arrow[d, "F_{\mathsf{first}}"] \\
& F_{J\times L,K\times L}1
\end{tikzcd}$$

$$\boxed{(\mathsf{first}\text{-}\textsc{Func})}$$

$$\begin{tikzcd}
F_{J,K}X \times F_{K,L}Y \arrow[r, "F_{\mathsf{first}}\times F_{\mathsf{first}}"] \arrow[d, "F_{\ggg}"'] & F_{J\times M,K\times M}X \times F_{K\times M,L\times M}Y \arrow[d, "F_{\ggg}"] \\
F_{J,L}(X \times Y) \arrow[r, "F_{\mathsf{first}}"'] & F_{J\times M,L\times M}(X \times Y)
\end{tikzcd}$$

Table 2
Equational axioms for lax arrow functors

The definitions are similar for the other arrow operators. The arrow axioms are satisfied due to the corresponding equational condition on the lax arrow functor. 2

This proposition reduces our goal (Thm. 4.4) to showing that the family $\{A(J, \_ \times K)\}_{J,K}$ is a lax arrow functor. This is what will be shown in the next section, through manipulation of 2-cells in **Prof**.

5 Calculations in Prof

There is one technical issue in front of us: the size issue. The 0-cells of **Prof** are *small* categories; the smallness restriction is necessary for composition of profunctors to be well-defined (Def. 2.7). However, with **Set** being not small, the arrow A in Def. 4.1 cannot be a 1-cell in **Prof**. The arrow A needs to be based on **Set** so that $A(J, \_ \times K)$ is an endofunctor **Set** → **Set**.

In this paper we shall get round of the problem by pretending that **Set** is small. There are two possible justifications.

- We can resort to the category **Ens** of classes when it is needed—such as when we take composition of profunctors via a coend. This means upgrading all the sizes that appear in the definition of **Prof**: its 0-cells are locally small categories; its 1-cells $P : \mathbb{C} \nrightarrow \mathbb{D}$ are bifunctors $\mathbb{D}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Ens}$. In this case, in Def. 4.1, we would restrict the arrow A to be *small*, in the sense that its image $A(J, K)$ restricts to **Set**.

$$\begin{tikzcd}
\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \arrow[r, "A"] \arrow[dr, dashed] & \mathbf{Ens} \\
& \mathbf{Set} \arrow[u, hook]
\end{tikzcd}$$

- We replace **Set** by some small cocomplete category defined internally in a suitable topos [14]. In other words, we develop our theory on top of a certain type theory which is modeled by such a topos.

In any case, we would like to isolate the size issue as much as possible. Therefore in the sequel we first establish those technical results which hold for any small symmetric monoidal category $(\mathbb{C}, \otimes, I)$. These results are proved by manipulating 2-cells in **Prof**. After that we instantiate $(\mathbb{C}, \otimes, I)$ by $(\mathbf{Set}, \times, 1)$—pretending that **Set** is small.

Definition 5.1 Let $(\mathbb{C}, \otimes, I)$ be a small SMC, and A be an arrow on it. There arise three 2-cells in **Prof**—which we denote by F^A_{arr}, $F^A_{\ggg}$ and F^A_{first}—of the following types.

$$\begin{tikzcd}
& \mathbb{C} \arrow[dl, "(I\otimes\_)"', "\shortmid"{marking}] \arrow[dr, "\mathbb{C}", "\shortmid"{marking}] \arrow[d, phantom, "\Downarrow F^A_{\mathsf{arr}}"] & \\
\mathbb{C} \arrow[rr, "A"', "\shortmid"{marking}] & {} & \mathbb{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{C}^3 \arrow[r, "\mathbb{C}\times\otimes", "\shortmid"{marking}] \arrow[d, "\otimes\times\mathbb{C}"', "\shortmid"{marking}] & \mathbb{C}^2 \arrow[r, "\mathbb{C}\times A", "\shortmid"{marking}] \arrow[d, phantom, "\Downarrow F^A_{\ggg}"] & \mathbb{C}^2 \arrow[d, "\otimes", "\shortmid"{marking}] & \\
\mathbb{C}^2 \arrow[r, "\otimes"', "\shortmid"{marking}] & \mathbb{C} \arrow[rr, bend right, "A"', "\shortmid"{marking}] & \mathbb{C} \arrow[r, "A"', "\shortmid"{marking}] & \mathbb{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{C}^3 \arrow[r, "\otimes\times\mathbb{C}", "\shortmid"{marking}] \arrow[d, "\mathbb{C}\times\otimes"', "\shortmid"{marking}] & \mathbb{C}^2 \arrow[r, "A\times\mathbb{C}", "\shortmid"{marking}] \arrow[d, phantom, "\Downarrow F^A_{\mathsf{first}}"] & \mathbb{C}^2 \arrow[d, "\otimes", "\shortmid"{marking}] \\
\mathbb{C}^2 \arrow[r, "\otimes"', "\shortmid"{marking}] & \mathbb{C} \arrow[r, "A"', "\shortmid"{marking}] & \mathbb{C}
\end{tikzcd}$$

Explicitly, these 2-cells are given by the following composites.

$$\begin{tikzcd}
& \mathbb{C} \arrow[dl, "I\otimes\_"', "\shortmid"{marking}] \arrow[dr, "\mathbb{C}", "\shortmid"{marking}] \arrow[d, phantom, "\Downarrow\cong"] & \\
\mathbb{C} \arrow[rr, "\mathbb{C}", "\shortmid"{marking}] \arrow[rr, bend right=50, "A"', "\shortmid"{marking}] \arrow[rr, phantom, bend right=25, "\Downarrow \mathsf{arr}"] & {} & \mathbb{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{C}^3 \arrow[r, "\mathbb{C}\times\otimes", "\shortmid"{marking}] \arrow[d, "\otimes\times\mathbb{C}"', "\shortmid"{marking}] \arrow[dr, phantom, "\Downarrow\cong"] & \mathbb{C}^2 \arrow[r, "\mathbb{C}\times A", "\shortmid"{marking}] \arrow[d, "\otimes", "\shortmid"{marking}] \arrow[dr, phantom, "\Downarrow \mathsf{second}"] & \mathbb{C}^2 \arrow[d, "\otimes", "\shortmid"{marking}] & \\
\mathbb{C}^2 \arrow[r, "\otimes"', "\shortmid"{marking}] & \mathbb{C} \arrow[r, "A"', "\shortmid"{marking}] \arrow[rr, bend right=40, "A"', "\shortmid"{marking}] \arrow[rr, phantom, bend right=20, "\Downarrow\ggg"] & \mathbb{C} \arrow[r, "A"', "\shortmid"{marking}] & \mathbb{C}
\end{tikzcd}
\qquad
\begin{tikzcd}
\mathbb{C}^3 \arrow[r, "\otimes\times\mathbb{C}", "\shortmid"{marking}] \arrow[d, "\mathbb{C}\times\otimes"', "\shortmid"{marking}] \arrow[dr, phantom, "\Downarrow\cong"] & \mathbb{C}^2 \arrow[r, "A\times\mathbb{C}", "\shortmid"{marking}] \arrow[d, "\otimes", "\shortmid"{marking}] \arrow[dr, phantom, "\Downarrow \mathsf{first}"] & \mathbb{C}^2 \arrow[d, "\otimes", "\shortmid"{marking}] \\
\mathbb{C}^2 \arrow[r, "\otimes"', "\shortmid"{marking}] & \mathbb{C} \arrow[r, "A"', "\shortmid"{marking}] & \mathbb{C}
\end{tikzcd}$$

Here the 1-cell $I \otimes \_$ on the left is the direct image of the functor $X \mapsto I \otimes X$ (Notations 2.11); recall that I denotes the monoidal unit. Also recall Notations 3.5. The 2-cells $\mathsf{arr}, \ggg, \mathsf{first}, \mathsf{second}$ are due to the arrow structure of A (Def. 3.2, 3.4).

The motivation for this definition is clear from the names of the 2-cells. Indeed, through some calculations in **Prof** and application of the Yoneda lemma, one easily sees that the three 2-cells $F^A_{\mathsf{arr}}, F^A_{\ggg}, F^A_{\mathsf{first}}$ are the same thing as (di)natural transformations

$$F^A_{\mathsf{arr}} \;:\; \mathbb{C}(J, K) \longrightarrow A(J, I \otimes K) \;\;, \qquad \text{natural in } J, K;$$

$$F^A_{\ggg\,J,K,L} \;:\; A(J, X \otimes K) \times A(K, Y \otimes L) \longrightarrow A(J, (X \otimes Y) \otimes L) \;\;,$$
$$\text{natural in } J, L, X, Y, \text{ dinatural in } K,$$

$$F^A_{\mathsf{first}\,J,K,L} \;:\; A(J, X \otimes K) \longrightarrow A(J \otimes L, X \otimes (K \otimes L)) \;\;,$$
$$\text{natural in } J, K, X, \text{ dinatural in } L,$$

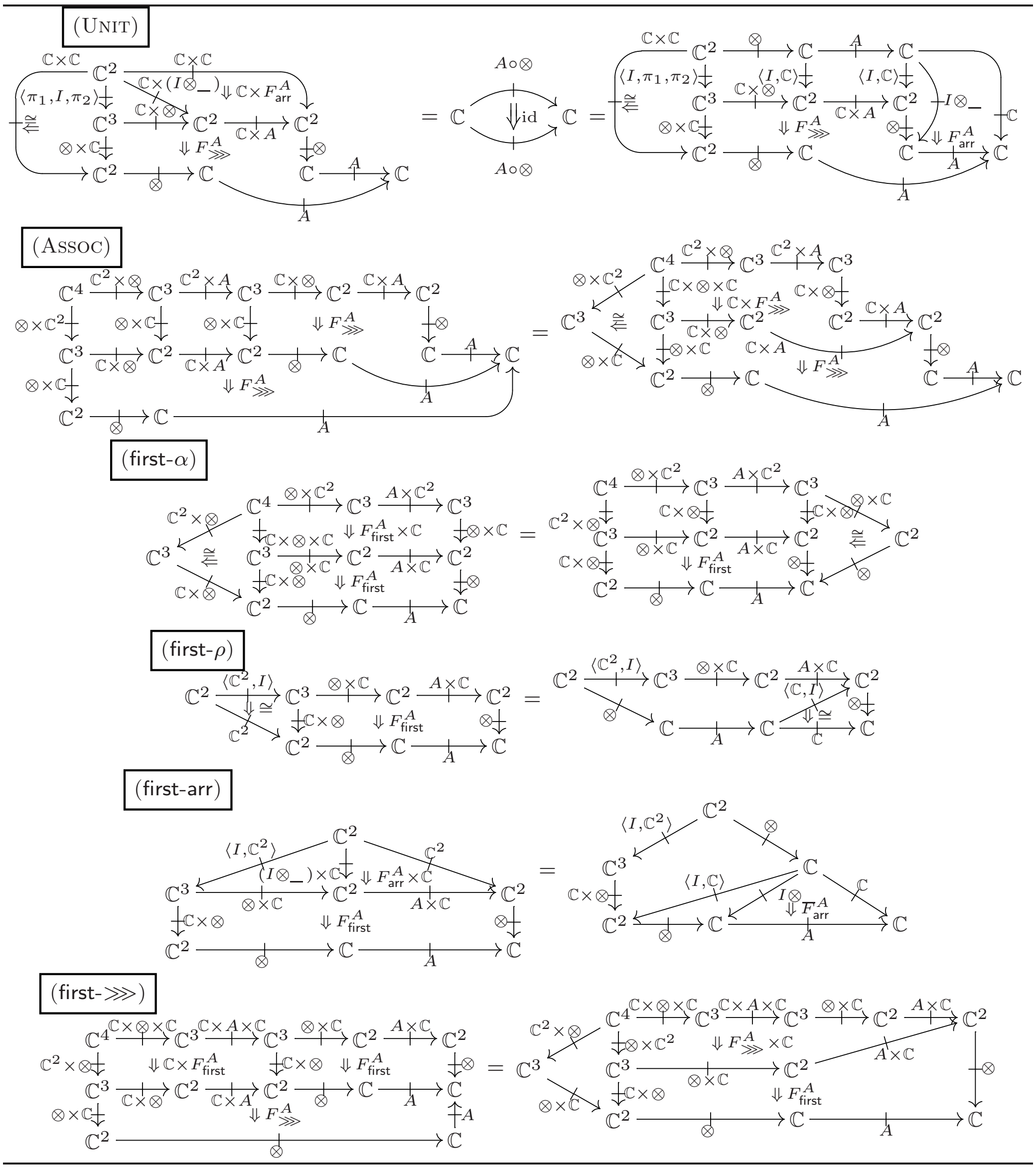


Table 3
Equalities that hold for $F^A_{\mathsf{arr}}, F^A_{\ggg}, F^A_{\mathsf{first}}$

respectively. These (di)natural transformations bear clear similarity to the ones in Def. 4.6 when $F_{J,K}$ is instantiated with $A(J, \_ \otimes K)$.

Let us now turn to equations.

Lemma 5.2 *Let A be an arrow over a small SMC $\mathbb{C}$. The three 2-cells $F^A_{\mathsf{arr}}, F^A_{\ggg}$ and F^A_{first} in Def. 5.1 satisfy the equalities in Table 3; they are parallel to the equalities in Def. 3.2.*

Proof First expand the definitions of $F^A_{\mathsf{arr}}, F^A_{\ggg}$ and F^A_{first}, and then use the equational axioms in Def. 3.2. One also needs Lem. 3.6. 2

The equalities in Table 3 might look complicated. However, coming up with them is rather routine work looking at Def. 5.1 and Def. 3.2.

We now instantiate $(\mathbb{C}, \otimes, I)$ with $(\mathbf{Set}, \times, 1)$, pretending **Set** to be small.

Lemma 5.3 *Let A be an arrow $\mathbf{Set}^{\mathrm{op}} \times \mathbf{Set} \to \mathbf{Set}$. The family $\{A(J, \_ \times K)\}_{J,K}$ of endofunctors is a lax arrow functor.*

Proof The three 2-cells in Def. 5.1 provide the three natural transformations required in Def. 4.6. The equations asserted in Def. 4.6 follow from those in Lem. 5.2. Checking all this is (laborious) routine work. 2

Combining Prop. 4.7 and Lem. 5.3, our main result Thm. 4.4 is proved.

Remark 5.4 A characterization of categorical arrows in the spirit of Prop. 3.3 can possibly yield a even more direct proof of Thm. 4.4. Unfortunately at now we lack necessary infrastructure such as a lifting result like Prop. 4.7. We are currently investigating possible formalization using fibered spans (see e.g. [15]).

In **Prof** the *trace* operator for an arrow (`loop` in [22], see also [8]) can be formalized in a similar way to other operators like $\ggg$. Its description as well as possible application to components will presented in another venue.

Acknowledgments

Thanks are due to Paul-André Melliès for advocating use of profunctors; and to Marcelo Fiore, Bart Jacobs and Bartek Klin for helpful discussions.

References

[1] Asada, K., *Arrows are strong monads* (2009), preprint, www.kurims.kyoto-u.ac.jp/~asada/papers/arrStrMnd.pdf.

[2] Atkey, R., *What is a categorical model of arrows?*, in: V. Capretta and C. McBride, editors, *Mathematically Structured Functional Programming*, 2008.

[3] Baez, J. C. and J. Dolan, *Categorification*, Contemp. Math. **230** (1998), pp. 1–36.

[4] Baez, J. C. and J. Dolan, *Higher dimensional algebra III: n-categories and the algebra of opetopes*, Adv. Math **135** (1998), pp. 145–206.
URL citeseer.ist.psu.edu/article/baez97higherdimensional.html

[5] Barbosa, L., "Components as Coalgebras," Ph.D. thesis, Univ. Minho (2001).

[6] Barr, M. and C. Wells, "Toposes, Triples and Theories," Springer, Berlin, 1985, available online.

[7] Bénabou, J., *Distributors at work*, Lecture notes taken by T. Streicher (2000), www.mathematik.tu-darmstadt.de/~streicher/FIBR/DiWo.pdf.gz.

[8] Benton, N. and M. Hyland, *Traced premonoidal categories*, Theoretical Informatics and Applications **37** (2003), pp. 273–299.

[9] Borceux, F., "Handbook of Categorical Algebra," Encyclopedia of Mathematics **50, 51 and 52**, Cambridge Univ. Press, 1994.

[10] Fiore, M., *Rough notes on presheaves* (2001), available online.

[11] Hasuo, I., C. Heunen, B. Jacobs and A. Sokolova, *Coalgebraic components in a many-sorted microcosm*, in: A. Kurz, M. Lenisa and A. Tarlecki, editors, *CALCO*, Lect. Notes Comp. Sci. **5728** (2009), pp. 64–80.

[12] Hasuo, I., B. Jacobs and A. Sokolova, *The microcosm principle and concurrency in coalgebra*, in: *Foundations of Software Science and Computation Structures*, Lect. Notes Comp. Sci. **4962** (2008), pp. 246–260.

[13] Hughes, J., *Generalising monads to arrows.*, Science of Comput. Progr. **37** (2000), pp. 67–111.

[14] Hyland, J. M. E., *A small complete category*, Ann. Pure & Appl. Logic **40** (1988), pp. 135–165.

[15] Jacobs, B., "Categorical Logic and Type Theory," North Holland, Amsterdam, 1999.

[16] Jacobs, B., C. Heunen and I. Hasuo, *Categorical semantics for arrows*, J. Funct. Progr. **19** (2009), pp. 403–438.

[17] Kelly, G. M., "Basic Concepts of Enriched Category Theory," Number 64 in LMS, Cambridge Univ. Press, 1982, available online:
`http://www.tac.mta.ca/tac/reprints/articles/10/tr10abs.html`.

[18] Kock, A., *Monads on symmetric monoidal closed categories*, Arch. Math. **XXI** (1970), pp. 1–10.

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

[20] Mac Lane, S., "Categories for the Working Mathematician," Springer, Berlin, 1998, 2nd edition.

[21] Moggi, E., *Notions of computation and monads*, Inf. & Comp. **93(1)** (1991), pp. 55–92.

[22] Paterson, R., *A new notation for arrows*, in: *ICFP*, 2001, pp. 229–240.

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

[24] Rutten, J. J. M. M., *Universal coalgebra: a theory of systems*, Theor. Comp. Sci. **249** (2000), pp. 3–80.

[25] Street, R., *The formal theory of monads*, Journ. of Pure & Appl. Algebra **2** (1972), pp. 149–169.

[26] Uustalu, T. and V. Vene, *Comonadic notions of computation*, Elect. Notes in Theor. Comp. Sci. **203** (2008), pp. 263–284.