Categorifying Computations into Components via Arrows as Profunctors☆

Ichiro Hasuo[a,b,*], Kazuyuki Asada[a,1]

[a] *Research Institute for Mathematical Sciences, Kyoto University*
[b] *PRESTO Research Promotion Program, Japan Science and Technology Agency*

Abstract

The notion of *arrow* by Hughes is an axiomatization of the algebraic structure possessed by structured computations in general. We claim that the same axiomatization of 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 first 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 second 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* [2], as is standard in Haskell. However side-effect—"structured output"—is not the only cause for the failure of pure functionality. A *comonad* can be used to encapsulate "structured input" [3]; the combination of a monad and a comonad via a distributive

☆An earlier version [1] of this paper is presented at the 10th International Workshop on Coalgebraic Methods in Computer Science (CMCS 2010), Paphos, Cyprus, March 2010.

*Corresponding author. Research Institute for Mathematical Sciences, Kyoto University, 606-8502 Japan. Tel: +81 75 753 7251, fax: +81 75 753 7266.

Email addresses: `ichiro@kurims.kyoto-u.ac.jp` (Ichiro Hasuo), `asada@kurims.kyoto-u.ac.jp` (Kazuyuki Asada)

URL: `http://www.kurims.kyoto-u.ac.jp/~ichiro` (Ichiro Hasuo), `http://www.kurims.kyoto-u.ac.jp/~asada` (Kazuyuki Asada)

[1] Partly supported by the Global COE Program *Fostering Top Leaders in Mathematics* at Kyoto University, funded by Japan Society for the Promotion of Science.

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* [4] is a general axiomatization of such.[2]

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 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

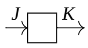
- carried by a family of sets $\{A(J,K)\}_{J,K}$ for each $J, K \in \mathbb{C}$, an element $a \in A(J,K)$ of which is an A-*computation* from J to K;
- equipped with the following three families of operators arr, $\ggg$ and first:

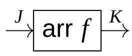
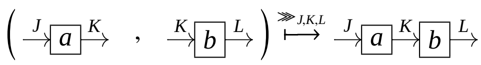
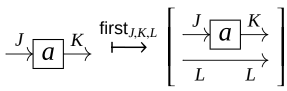
$$\begin{array}{rcll} & \mathsf{arr}\, f \in A(J,K) & & \text{for each morphism } f : J \to K \text{ in } \mathbb{C}, \\ A(J,K) \times A(K,L) & \xrightarrow{\ggg_{J,K,L}} & A(J,L) & \text{for each } J,K,L \in \mathbb{C}, \\ A(J,K) & \xrightarrow{\mathsf{first}_{J,K,L}} & A(J \times L, K \times L) & \text{for each } J,K,L \in \mathbb{C}; \end{array} \tag{1}$$

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

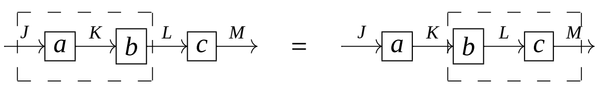
$$\begin{array}{c} (a \ggg_{J,K,L} b) \ggg_{J,L,M} c = a \ggg_{J,K,M} (b \ggg_{K,L,M} c) \\ \text{for each } a \in A(J,K), b \in A(K,L), c \in A(L,M). \end{array} \tag{$\ggg$-Assoc}$$

  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 in the above, for example, ensures that the following compositions of three consecutive A-computations are identical.

 (2)

Arrows generalize monads. In fact, a strong monad T on $\mathbb{C}$ induces an arrow A_T by

$$A_T(J,K) = \mathbb{C}(J, TK) = \mathcal{K}\ell(T)(J,K) \ . \tag{3}$$

[2]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.

Here $\mathcal{K}\ell(T)$ denotes the Kleisli category (see e.g. Moggi [2]). Prior to arrows, the notion of *Freyd category* is devised as another axiomatization of algebraic properties that are expected from "computations" [5, 6]. The latter notion of Freyd category comes with a stronger categorical flavor; in Jacobs et al. [7] it is shown to be equivalent to the notion of arrow.

Remark 1.1. What has been said is true as long as we think of an arrow as carried by sets, i.e. 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 Atkey [8]. The discussion later in §5.1 is also relevant.

1.2. Arrow as Component Calculus

The current paper's goal 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 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 monad-based coalgebraic modeling of components in Barbosa [9]—which is also used in Hasuo et al. [10]—and extend it later to an arrow-based modeling. In [9] a component is modeled as a coalgebra of the following type:

$$c \;:\; X \longrightarrow (T(X \times K))^J \quad \text{in } \mathbf{Sets}. \tag{4}$$

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 **Sets** 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 $\xrightarrow{J}\boxed{c}\xrightarrow{K}$.

A crucial observation here is as follows. The notion of arrow in §1.1 axiomatizes 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 same 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 first author's previous work [10] with Heunen, Jacobs and Sokolova, such algebraic operators on coalgebraic components (4) are defined and shown to satisfy the equational axioms.

1.3. Categorifying Computations into Components

Despite the similarity between computations and components that we have just described, there is one level gap between them: from *sets* to *categories*. Let $\mathcal{A}(J, K)$ denote the collection of coalgebraic components like in (4), 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. Rutten [11]) between components. Hence $\mathcal{A}(J, K)$ is now a *category*, specifically that of $(T(\_ \times K))^J$-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* [12] 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; at this point 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

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

Denoting e's state space by U, the state space of the left-hand side is $(X \times Y) \times U$ while that of the right-hand side 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 [10]). We note that this phenomenon—the notion of satisfaction of equational axioms gets relaxed, from up-to equality to up-to an isomorphism—is typical with categorification [12].

This additional structure obtained through categorification, namely morphisms between components, has been further exploited in [10]. There it is shown that final coalgebras—the notion that makes sense only in presence of morphisms between coalgebras—form an arrow that is internal to the "arrow" of components, realizing an instance of the *microcosm principle* [13, 14]. An application of such nested algebraic structure (namely that 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*. The table below summarizes the overall picture.

	arrow A	categorical arrow $\mathscr{A}$
carrier	$\{A(J,K)\}_{J,K\in\mathbb{C}}$, a family of sets $a \in A(J,K)$: a computation	$\{\mathscr{A}(J,K)\}_{J,K\in\mathbb{C}}$, a family of categories $a \in \mathscr{A}(J,K)$: a component
equations satisfied	up-to equality	up-to isomorphisms
example	$A(J,K) = \mathcal{K}\ell(T)(J,K)$ with T: a monad	$\mathscr{A}(J,K) = \mathbf{Coalg}(T(\_ \times K))^J$ with T: a monad

1.4. Lifting of Arrow Structure via Profunctors

In short: computations carry algebraic structure of an arrow; components carry a categorified version of it. The technical 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 monad-based modeling (4),
- 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 [10]. However the last lifting part of the scenario 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 [7]. There the $\mathsf{arr}, \ggg$-fragment of arrow (without first) is identified with a monoid in the category $[\mathbb{C}^{\mathrm{op}} \times \mathbb{C}, \mathbf{Sets}]$ 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 Street [15].

What really makes our profunctor approach feasible is a further observation by the second author [16]. 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 Kelly [17], which is a useful reference also in the current non-enriched (i.e. **Sets**-enriched) setting. In §3 we follow [16, 7] 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**.

The current version departs from the previous workshop version [1] most notably in §5. The manipulation of 2-cells is now described in a much more structural manner, using a novel bicategory **StProf**. The details that have been omitted in [1] are presented as much as the space allows. We also explicitly settle the problem of size (§5.1); in the previous version [1] we only hinted possible solutions.

2. Categorical Preliminaries

2.1. End and Coend

In the sequel we shall 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.

$$\begin{array}{ccccccc} & \overset{F(f,X)}{\nearrow} & F(X,X) & \xrightarrow{\varphi_X} & G(X,X) & \overset{G(X,f)}{\searrow} & \\ F(X',X) & & & & & & G(X,X') \\ & \underset{F(X',f)}{\searrow} & F(X',X') & \xrightarrow[\varphi_{X'}]{} & G(X',X') & \underset{G(f,X')}{\nearrow} & \end{array} \tag{5}$$

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

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

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

  $$\begin{array}{ccccc} & & F(X', X') & & \\ & \overset{\pi_{X'}}{\nearrow} & & \overset{F(f, X')}{\searrow} & \\ \int_X F(X, X) & & & & F(X, X') \\ & \underset{\pi_X}{\searrow} & & \underset{F(X, f)}{\nearrow} & \\ & & F(X, X) & & \end{array}$$

  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 *coprojections* $\iota_X : F(X, X) \to \int^X F(X, X)$ for each $X \in \mathbb{C}$. Its universality, together with that of an end, can be written as follows.

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

(Co)ends need not exist; they do exist for example when $\mathbb{C}$ is small and $\mathbb{D}$ is (co)complete. See below.

The reader is referred to Mac Lane [18, 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 [18, §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 $F, G : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Sets}$, we obtain a bifunctor

$$[F(+,-), G(-,+)] \;:\; \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbf{Sets} \;, \quad (X, Y) \longmapsto [F(Y, X), G(X, Y)] \;. \tag{6}$$

Here $[S, T]$ denotes the set of functions from S to T, i.e. an exponential in **Sets**. Note the variance: since $[-,+]$ is contravariant in its first argument, the variance of arguments of F is opposed in (6). Taking this functor (6) 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 **Sets** 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 in* **Sets***:*

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

Proof. It is due to the following correspondences.

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

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

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 $F, G : \mathbb{C} \to$* **Sets***. By* $\mathrm{Nat}(F, G)$ *we denote the set of natural transformations $F \Rightarrow G$. We have*

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

The celebrated *Yoneda lemma* reduces the set $\mathrm{Nat}(\mathbb{C}(X, \_), F)$ of natural transformations into FX (see e.g. [18, 19]). Interpreted via Cor. 2.3, it yields:

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

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

The lemma becomes useful in the calculations below: it means an end on the left-hand side "cancels" with a hom-functor occurring in it.

From the end-form, we obtain the following coend-form. Its proof is straightforward but illuminating. It allows us to "cancel" a coend with a hom-functor inside it.

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

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

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

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

Here $(\dagger)$ is because the hom-functor $[\_\, , S\,]$ turns a colimit into a limit [18, §V.4], hence a coend into an end. Obviously the composite isomorphism is natural in S; therefore we have shown that

$$\mathbf{y}(\, \int^{X'} \mathbb{C}(X', X) \times FX'\,) \xrightarrow{\cong} \mathbf{y}(FX) \quad : \; \mathbb{C} \longrightarrow \mathbf{Sets} \;, \tag{7}$$

where $\mathbf{y} : \mathbb{C}^{\mathrm{op}} \to [\mathbb{C}, \mathbf{Sets}]$ is the (contravariant) Yoneda embedding. By the Yoneda lemma the functor $\mathbf{y}$ is full and faithful; therefore it reflects isomorphisms. Hence (7) proves the claim. □

2.3. Profunctor

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

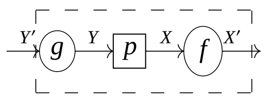
$$\frac{\mathbb{C} \nrightarrow \mathbb{D}, \quad \text{a profunctor}}{\mathbb{D}^{\mathrm{op}} \times \mathbb{C} \longrightarrow \mathbf{Sets}, \quad \text{a functor}}$$

The notion of profunctor is also called *distributor*, *bimodule* or *module*. For more detailed treatment of profunctors see e.g. Benabou [20] and Borceux [21].

There are two principal 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, $P(Y, X) \in \mathbf{Sets}$.

- 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.



(8)

The last point ("$\mathbb{C}$-$\mathbb{D}$-action") motivates another 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 (8) and the diagrams in §1 for computations/components. It is indeed this similarity that allows us to formalize arrows as certain profunctors (§3).

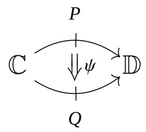
Definition 2.7 (Composition of profunctors). Given two successive profunctors $P : \mathbb{C} \nrightarrow \mathbb{D}$ and $Q : \mathbb{D} \nrightarrow \mathbb{E}$, their *composition* $Q \circ P : \mathbb{C} \nrightarrow \mathbb{E}$ is defined by the following coend. For $U \in \mathbb{E}$ and $X \in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

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

When profunctors are seen as generalized relations, this composition operation corresponds to *relational composition*: $(S \circ R)(x, z)$ if and only if $\exists y.(R(x, y) \wedge S(y, z))$. When seen as generalized modules, it corresponds to *tensor product* of modules over rings. 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 $\big(\xrightarrow{U}\boxed{q}\xrightarrow{Y},\ \xrightarrow{Y}\boxed{p}\xrightarrow{X}\big)$, 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 $\mathbb{D}$.

$$\big(\xrightarrow{U}\boxed{q}\xrightarrow{Y}\bigcirc\!\!\!\!\!\!\;f\ \xrightarrow{Y'},\ \xrightarrow{Y'}\boxed{q}\xrightarrow{X}\big) \simeq \big(\xrightarrow{U}\boxed{q}\xrightarrow{Y},\ \xrightarrow{Y}\bigcirc\!\!\!\!\!\!\;f\ \xrightarrow{Y'}\boxed{p}\xrightarrow{X}\big) \ .$$

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 $P, Q : \mathbb{D}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Sets}$. All these data can be organized in the following "2-categorical" manner.



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 [21]) 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 $\mathrm{Hom} : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Sets}$; 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 the direct image of a functor and tensor products in **Prof**. For the former, [20] is a principal reference; Fiore's notes [22] are not specifically on profunctors but provide useful insights into relevant mathematical concepts.

A function $f : S \to T$ induces the *direct image* relation $f_* : S \nrightarrow T$, defined by: $f_*(s, t)$ if and only if $t = f(s)$. There is an analogous construction from functors to profunctors.

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

$$\text{the } \textit{direct image} \text{ profunctor} \qquad F_* : \mathbb{C} \nrightarrow \mathbb{D} \qquad \text{by } F_*(Y, X) = \mathbb{D}(Y, FX) \ .$$

The mapping $(\_)_*$ also applies to natural transformations in an obvious way; this determines a *pseudo functor* (see e.g. [21]) $(\_)_* : \mathbf{Cat} \to \mathbf{Prof}$ that embeds **Cat** in **Prof**.

Notations 2.10. Throughout the rest of the paper, the direct image F_* of a functor F shall be simply denoted by F. 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 $\times$ in **Cat** lifts **Prof**: given profunctors $F : \mathbb{C} \nrightarrow \mathbb{C}'$ and $G : \mathbb{D} \nrightarrow \mathbb{D}'$, we define

$$F \times G : \mathbb{C} \times \mathbb{D} \nrightarrow \mathbb{C}' \times \mathbb{D}' \quad \text{by} \quad (F \times G)(X', Y', X, Y) = F(X', X) \times G(Y', Y) \ . \tag{9}$$

The symbol $\times$ occurring in the last denotes the Cartesian product in **Sets**. The lifted operator $\times$ 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 2.11. *The operation $\times$ 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 $(Q \circ P) \times (Q' \circ P') \overset{\cong}{\Rightarrow} (Q \times Q') \circ (P \times P')$.*

Proof. This is due to the Fubini theorem for coends. See [18, §IX.8] □

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

3. Arrows as Profunctors

We review the results in [7, 16] 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 [4]). 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). The operators must satisfy the following equational axioms.

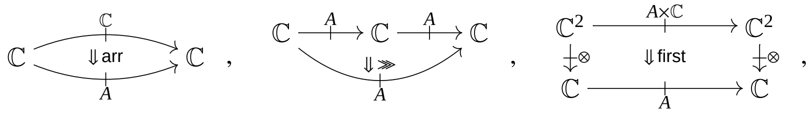
$$\begin{array}{rcll}
(a \ggg b) \ggg c & = & a \ggg (b \ggg c) & (\ggg\text{-Assoc}) \\
\mathsf{arr}\,(g \circ f) & = & \mathsf{arr}\, f \ggg \mathsf{arr}\, g & (\mathsf{arr}\text{-Func1}) \\
\mathsf{arr}\,\mathrm{id}_J \ggg_{J,J,K} a & = a = & a \ggg_{J,K,K} \mathsf{arr}\,\mathrm{id}_K & (\mathsf{arr}\text{-Func2})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

Here some subscripts are suppressed. The morphism $\rho_K : K \otimes I \overset{\cong}{\to} K$ is the right unitality isomorphism; $\alpha_{K,L,M} : (K \otimes L) \otimes M \overset{\cong}{\to} K \otimes (L \otimes M)$ is the associativity isomorphism. The names of the axioms hint their correspondence to the (premonoidal) structure of *Freyd categories* [5, 6].

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

Definition 3.2 (Prof-arrow). 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:



  where all the diagrams are in **Prof**,
- subject to the equalities in Table 1. Recall Notations 2.10; for example the profunctor $\langle \mathbb{C}, I \rangle$ in (first-ρ) is the functor $\langle \mathbb{C}, I \rangle : \mathbb{C} \to \mathbb{C}^2$, $X \mapsto (X, I)$, embedded in **Prof** by taking its direct image.

The notion of **Prof**-arrow is in fact a familiar one: it is a *strong monad* in **Prof**, defined internally in the sense of [15]. This means the following. 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 [23, 2].[3] 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 (first-$\ggg$) interpreted in **Cat** is read as the commutativity of the following diagram.

$$\begin{array}{ccccc} T^2X \otimes Y & \xrightarrow{\mathsf{str}'} & T(TX \otimes Y) & \xrightarrow{T\mathsf{str}'} & T^2(X \otimes Y) \\ {\scriptstyle \mu^T \otimes Y}\downarrow & & & & \downarrow{\scriptstyle \mu^T} \\ TX \otimes Y & & \xrightarrow[\mathsf{str}']{} & & T(X \otimes Y) \end{array}$$

(Internal) strong monads can be defined in any bicategory with suitable monoidal structure. Later in §5 we introduce a bicategory **StProf**; strong monads therein play an important role.

Proposition 3.3 ([16]). *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 [16] for a detailed proof, we present 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).

[3] 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 $\mathsf{str} : X \otimes TY \to T(X \otimes Y)$. These two are equivalent when the base category $\mathbb{C}$ is symmetric monoidal.

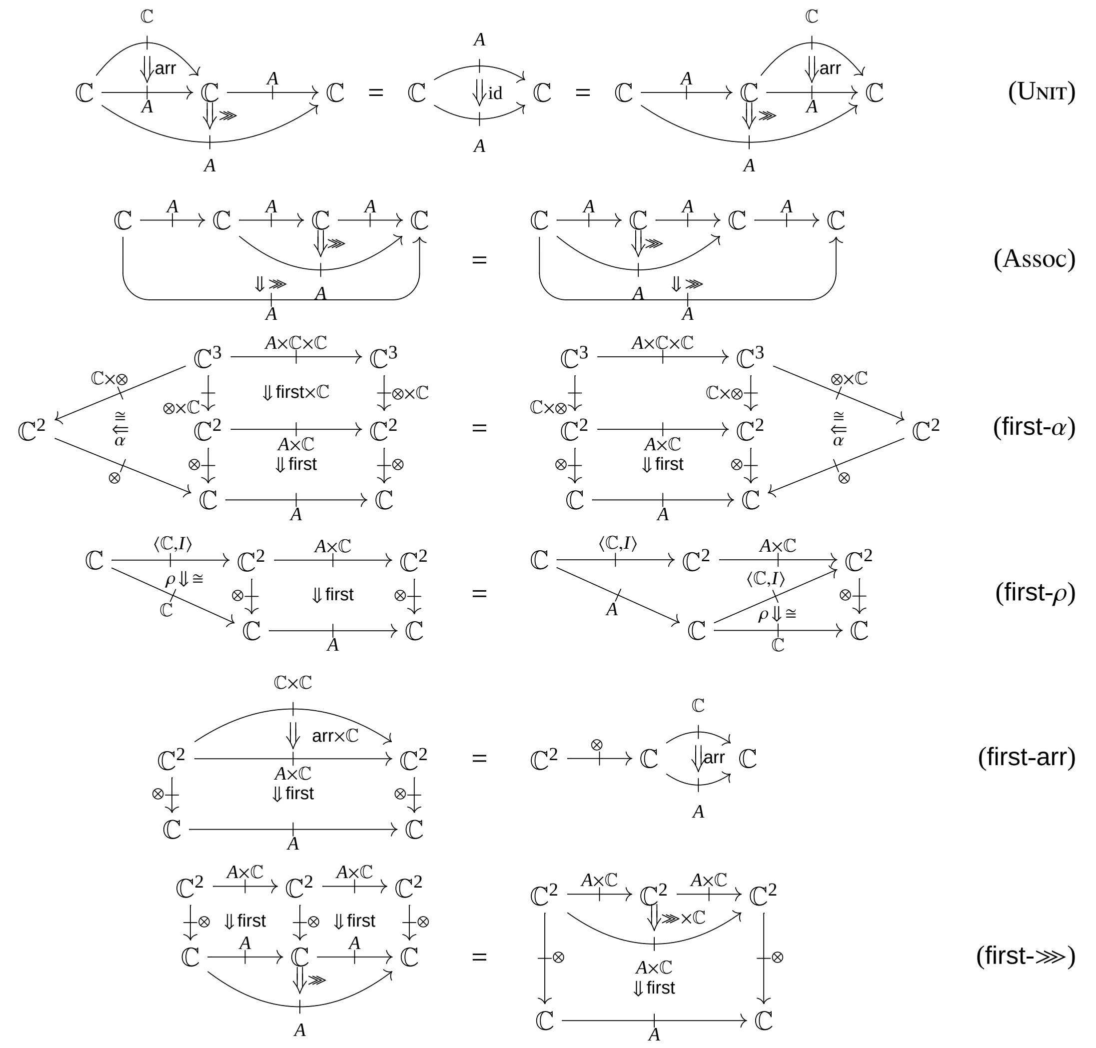


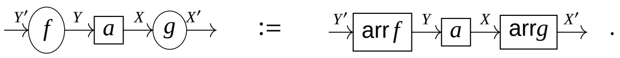
Table 1: Equational axioms for **Prof**-arrow

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 left-hand side below, where $\circ$ denotes composition of profunctors (Def. 2.7).

$$
\begin{aligned}
&\mathrm{Nat}(\ (\otimes \circ (A \times \mathbb{C}))(-, +_1, +_2)\ ,\ (A \circ \otimes)(-, +_1, +_2)\ ) \\
&\cong \int_{X,K,Y \in \mathbb{C}} [\ (\otimes \circ (A \times \mathbb{C}))(X, K, Y)\ ,\ (A \circ \otimes)(X, K, Y)\ ] && \text{by Cor. 2.3} \\
&\cong \int_{X,K,Y} [\ \int^{J,L} \mathbb{C}(X, J \otimes L) \times A(J, K) \times \mathbb{C}(L, Y)\ ,\ \int^{U} A(X, U) \times \mathbb{C}(U, K \otimes Y)\ ] \\
& && \text{by Def. 2.7, Def. 2.9 and (9)} \\
&\cong \int_{X,K,Y,J,L} [\ \mathbb{C}(X, J \otimes L) \times A(J, K) \times \mathbb{C}(L, Y)\ ,\ \int^{U} A(X, U) \times \mathbb{C}(U, K \otimes Y)\ ] \\
& && \text{since a hom-functor } [-, S] \text{ turns a coend into an end} \\
&\cong \int_{X,K,Y,J,L} [\ \mathbb{C}(X, J \otimes L),\ [\ A(J, K),\ [\ \mathbb{C}(L, Y)\ ,\ \int^{U} A(X, U) \times \mathbb{C}(U, K \otimes Y)\ ]]] && \text{by Currying} \\
&\cong \int_{J,K,L} [\ A(J, K),\ A(J \otimes L, K \otimes L)\ ] && \text{by canceling } X, Y \text{ by Lem. 2.4 and } U \text{ by Lem. 2.5} \\
&\cong \mathrm{Nat}_{J,K}\mathrm{Dinat}_{L}(\ A(J, K),\ A(J \otimes L, K \otimes L)\ ) && \text{by Prop. 2.2 and Cor. 2.3.}
\end{aligned}
$$

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 opposite 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 $\mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to$ **Sets**. 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:



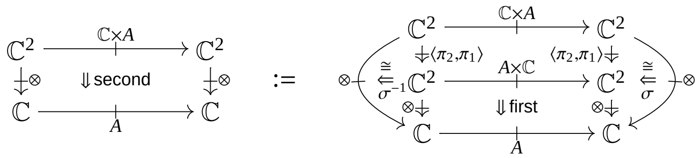
Each of the arrow operators yields its corresponding **Prof**-arrow operator; the latter's (di)naturality is derived from the arrow axioms. So are the equational axioms for a **Prof**-arrow. □

Prop. 3.3 offers a novel mathematical understanding of the notion of arrow. It endows the seemingly complicated original axiomatization (Def. 3.1) with categorical canonicity. Its treatment of first as a strength also seems simpler than that in Freyd categories: the latter involves technicalities like premonoidal categories and central morphisms. It is this simplicity that is exploited in the rest of the paper.

Remark 3.4. A size issue about Prop. 3.3 should be noted. While the original definition of arrow (Def. 3.1) makes sense for any base category $\mathbb{C}$ without size restriction, its characterization in **Prof** (Def. 3.2) requires $\mathbb{C}$ be small. Without the restriction the composition $A \circ A$—the domain of the 2-cell $\ggg$ in **Prof**—is not necessarily well-defined: $A \circ A$ is defined via coends in **Sets** (Def. 2.7) and **Sets** is only small-complete. This becomes a real obstacle later where we consider an arrow A over $\mathbb{C} =$ **Sets** which is not small. We shall fix this problem by "upgrading" the size of profunctors; see §5.1.

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

Definition 3.5. 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**.



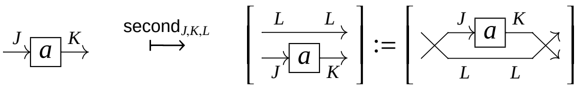
(10)

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 2.10). The 2-cell σ is the symmetry isomorphism $\sigma_{X,Y} : X \otimes Y \overset{\cong}{\Rightarrow} Y \otimes X$.

Notations 3.6. 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 right-hand side in (10).

The other class is due to the properties of the operation $\times$ on **Prof**, typically Lem. 2.11. Such iso 2-cells will be denoted by empty polygons, like the top one on the right-hand side in (10).

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



Lemma 3.7. *Regarding the* second *operator, the equalities in Table 2 hold.*

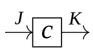
Proof. Use (first-α), (first-arr), (first-ρ), (first-$\ggg$) in Table 1 and the coherence for an SMC $\mathbb{C}$. □

4. Arrow-Based Components

4.1. Main Contribution

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 over **Sets**, and $J, K \in$ **Sets**. An *(arrow-based) A-component* with input-type J, output-type K and computational structure A is a coalgebra for the functor $A(J, \_ \times K) :$ **Sets** $\to$ **Sets**. That is,



as
$$\begin{array}{c} A(J, X \times K) \\ \uparrow c \\ X \end{array} \quad .$$

Here an arrow A is in the sense of Def. 3.1. In Def. 3.1 the base category $\mathbb{C}$ of an arrow need not be small; thus we choose (**Sets**, $\times$, 1) as $\mathbb{C}$. Our modeling specializes to Barbosa's (4) when we take as A a monad-based arrow A_T in (3). Our modeling not only generalizes Barbosa's but also brings conceptual clarity to the subsequent technical development.

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

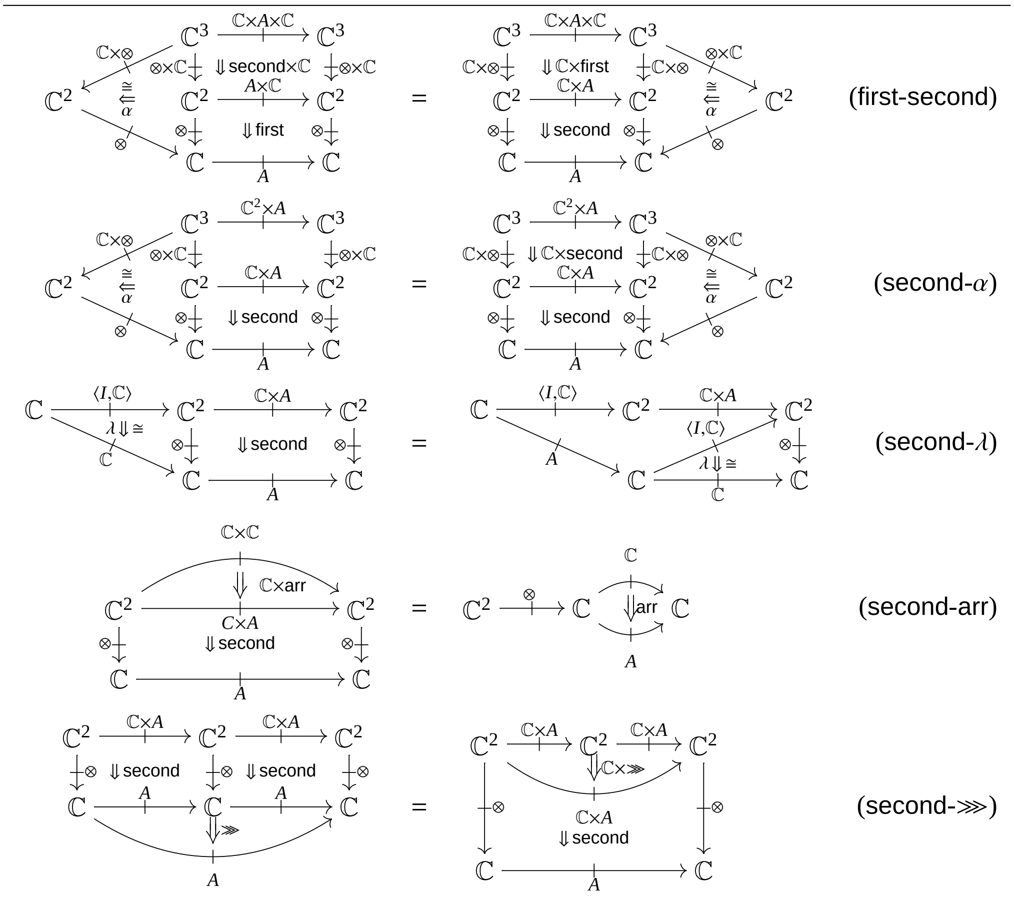


Table 2: Equalities that hold for the second operator

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

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

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

  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.

$$\begin{array}{ccc} \mathcal{A}(J,K) \times \mathcal{A}(K,L) \times \mathcal{A}(L,M) & \xrightarrow{\ggg_{J,K,L} \times \mathrm{id}} & \mathcal{A}(J,L) \times \mathcal{A}(L,M) \\ {\scriptstyle \mathrm{id} \times \ggg_{K,L,M}} \downarrow & \Downarrow \cong & \downarrow {\scriptstyle \ggg_{J,L,M}} \\ \mathcal{A}(J,K) \times \mathcal{A}(K,M) & \xrightarrow[\ggg_{J,K,M}]{} & \mathcal{A}(J,M) \end{array} \tag{11}$$

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 [10], 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 [10] do.

The notion of categorical arrow in Def. 4.2 could be formalized on any monoidal category $\mathbb{C}$ other than **Sets**. We do not need such additional generality.

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

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

One use of the theorem is as follows. We can appeal to the formalization [14, 10] of the *microcosm principle* [13] to obtain the following *compositionality result*.

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

1. *The family $\{Z_{J,K}\}_{J,K}$ of sets carries canonical arrow structure. This gives us e.g. "composition of behaviors"*

$$Z_{J,K} \times Z_{K,L} \stackrel{\ggg^Z}{\longrightarrow} Z_{J,L} \quad .$$

2. *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.*

$$
\begin{array}{ccc}
A(J,(X\times Y)\times L) & \dashrightarrow & A(J, Z_{J,L}\times L) \\
\uparrow{\scriptstyle c\ggg d} & & \cong\uparrow{\scriptstyle final} \\
X\times Y & \overset{\mathsf{beh}_{c\ggg d}}{\dashrightarrow} & Z_{J,L} \\
& \searrow_{\mathsf{beh}_c\times\mathsf{beh}_d}\ (*) & \uparrow{\scriptstyle \ggg^Z} \\
& & Z_{J,K}\times Z_{K,L}
\end{array}
$$

Here $c \ggg d$ *is "composition of components" using the categorical arrow structure in Thm. 4.4;* $\ggg^Z$ *is "composition of behaviors" derived above in the item 1; and* $\mathsf{beh}_{c\ggg d}$ *is the behavior map for the composed components induced by coinduction (the square on the top).* □

Similar microcosm arguments are employed in [24] for deriving traced monoidal structure of the category T-**Res** of T-*resumptions*. This is done all at once for a variety of computational effects, modeled by a monad T. T-resumptions are identified with T-*strategies* (see [25]); by applying the Int-construction [26] we obtain the category Int(T-**Res**) of T-*games*.

4.2. Lax Arrow Functor

In [14, 10] 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 in proving our main theorem (Thm. 4.4). Restricting the general definitions and results in [14, 10] to the current setting, we obtain the following.

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

- it is equipped with the following natural transformations:

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

- that are subject to the equations in Table 3, that are parallel to those in Def. 3.1. The diagrams there are all in **Sets**; 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.*

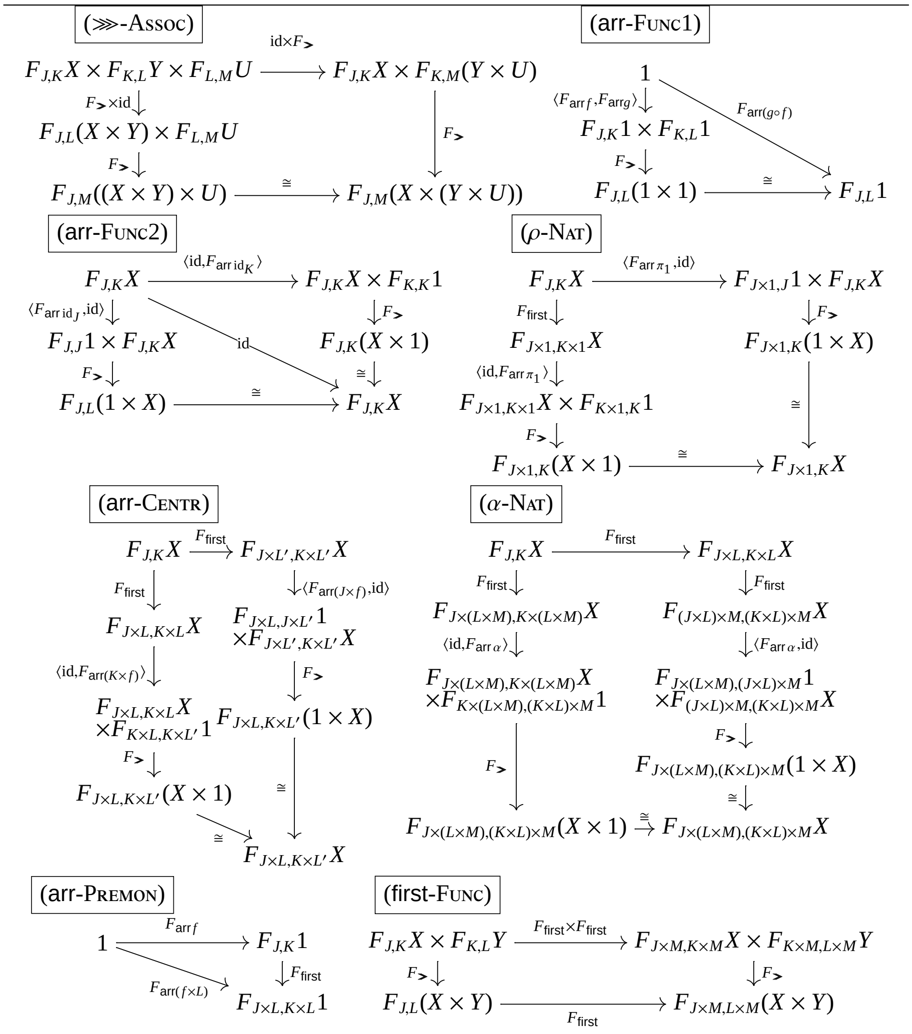


Table 3: Equational axioms for lax arrow functors

Proof. This follows from a general result like [10, Thm. 4.6]. Here we briefly illustrate what the categorical arrow structure of $\{\mathbf{Coalg}(F_{J,K})\}_{J,K}$ looks like, by describing the sequential composition $\ggg$: $\mathbf{Coalg}(F_{J,K}) \times \mathbf{Coalg}(F_{K,L}) \longrightarrow \mathbf{Coalg}(F_{J,L})$. Using $F_{\ggg}$ in Def. 4.6 it is defined as follows.

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

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. □

This proposition reduces Thm. 4.4 to the fact 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

5.1. The Size Issue

There is one technical problem—of a bookkeeping kind—lying in front of us: the size issue. It has been briefly discussed in Rem. 3.4. 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 **Sets** not being small, the arrow A in Def. 4.1 cannot be a 1-cell in **Prof**. At the same time we need the arrow A to be based on **Sets** so that $A(J, \_ \times K)$ is an endofunctor **Sets** $\to$ **Sets**. There are two possible ways round.

- We upgrade the size of profunctors. We use the category **Ens** of sets and classes of certain sizes, so that **Ens** has **Sets**-indexed colimits/coends. A profunctor $P : \mathbb{C} \nrightarrow \mathbb{D}$ is then defined to be a bifunctor $\mathbb{D}^{op} \times \mathbb{C} \to$ **Ens**. This upgrade is purely for the sake of abstract arguments: we will still require the arrow A to be "**Sets**-valued" (Def. 5.3).
- We replace **Sets** by some small cocomplete category defined internally in a suitable topos [27]. In other words, we develop our theory on top of a certain type theory which is modeled by such a topos.

We take the first path.

Definition 5.1 (The category Ens). We fix **Ens** to be the category of (small) sets and (large) classes whose sizes are within a suitable limit. We assume the following properties of **Ens**:

- **Ens** has colimits of **Sets**-sized diagrams. In particular, it has **Sets**-indexed coends.
- **Ens** is Cartesian closed.

Using such **Ens**, we override the previous definitions. This upgrade is in effect throughout the rest of the paper.

Definition 5.2.
- Let $\mathbb{C}$ and $\mathbb{D}$ be categories. A *profunctor* $P : \mathbb{C} \nrightarrow \mathbb{D}$ is a bifunctor $P : \mathbb{D}^{op} \times \mathbb{C} \to$ **Ens**.
- The bicategory **Prof** is such that:

- a 0-cell is a locally small category $\mathbb{C}$ whose collection of objects is not bigger than that of **Sets**, that is, $|\,\mathrm{Obj}(\mathbb{C})| \leq |\,\mathrm{Obj}(\mathbf{Sets})|$;
- a 1-cell $P : \mathbb{C} \nrightarrow \mathbb{D}$ is a profunctor (**Ens**-valued, as defined above); and
- a 2-cell is a natural transformation, much like in the previous definition of **Prof**.

An identity 1-cell $\mathbb{C} \nrightarrow \mathbb{C}$ is given by the bifunctor $\mathbb{C}^{\mathrm{op}} \times \mathbb{C} \overset{\mathrm{Hom}}{\rightarrow} \mathbf{Sets} \hookrightarrow \mathbf{Ens}$; note that a 0-cell $\mathbb{C} \in \mathbf{Prof}$ is locally small. Composition of 1-cells (cf. Def. 2.7)

$$(Q \circ P)(U, X) = \int^{Y \in \mathbb{D}} Q(U, Y) \times P(Y, X) \qquad \text{given } \mathbb{C} \overset{P}{\nrightarrow} \mathbb{D} \overset{Q}{\nrightarrow} \mathbb{E}$$

is now well-defined for a non-small $\mathbb{D} \in \mathbf{Prof}$, due to the extended cocompleteness property of **Ens**.

We note that the size upgrade does not affect validity of the Yoneda lemma.

In Prop. 3.3 an arrow over *small* $\mathbb{C}$ is characterized as a certain profunctor. Under the current size upgrade, the category $\mathbb{C} = \mathbf{Sets}$ also falls within the range.

Definition 5.3 (Prof-arrow). Let $\mathbb{C}$ be a monoidal category which belongs to **Prof**: that is, $\mathbb{C}$ is locally small and has at most as many objects as **Sets** does. A **Prof**-*arrow* over $\mathbb{C}$ is an internal strong monad $A : \mathbb{C} \nrightarrow \mathbb{C}$ in **Prof** (cf. Def. 5.3), which is **Sets**-*valued*: the bifunctor $A : \mathbb{C}^{\mathrm{op}} \times \mathbb{C} \to \mathbf{Ens}$ must factor as follows.

$$\begin{array}{ccc} \mathbb{C}^{\mathrm{op}} \times \mathbb{C} & \xrightarrow{\quad A \quad} & \mathbf{Ens} \\ & \searrow & \uparrow \\ & & \mathbf{Sets} \end{array}$$

Proposition 5.4. *Let $\mathbb{C}$ be a monoidal category subject to the size restriction in Def. 5.3. The notion of arrow over $\mathbb{C}$ (Def. 3.1) is equivalent to that of* **Prof***-arrow (Def. 5.3).* □

5.2. Lifting an Arrow to a Categorical Arrow

Our main result (Thm. 4.4) is about lifting

- an arrow A of computations
- to a categorical arrow $\{\, \mathbf{Coalg}(A(J, \_ \times K)\,)\}_{J,K}$ of components.

The following lemma proves it, when combined with Prop. 4.7.

Lemma 5.5. *Let A be an arrow over* **Sets**. *The family $\{A(J, \_ \times K)\}_{J,K}$ of endofunctors is a lax arrow functor (Def. 4.6).*

We aim at proving the lemma, in such a way that the arrow structure of A is reflected in the structure of the endofunctors $\{A(J, \_ \times K)\}_{J,K}$ as directly as possible. We take the following two steps.

- (Def. 5.6, Lem. 5.7) We introduce a new bicategory **StProf** of *stateful profunctors*; and then show that an internal strong monad in **Prof** (identified with an arrow A by Prop. 5.4) induces the same structure in **StProf**, in a canonical manner.
- (Lem. 5.9) We then show that an internal strong monad in **StProf** canonically induces a lax arrow functor.

This separation of steps offers a more structured view of the calculations in the earlier version [1] of the paper. For instance, the equalities in [1, Table 3] can now be systematically understood as the axioms for an internal strong monad in **StProf**, translated into 2-cells in **Prof**. For the record we shall present as many technical details as the space allows. The details may seem overwhelming; nevertheless, as is often the case with 2-categorical/bicategorical arguments, the underlying intuition is simple.

Definition 5.6 (The bicategory StProf). The bicategory **StProf** is defined as follows.

- A 0-cell of **StProf** is the same as that of **Prof**: it is a locally small category of a suitable size (Def. 5.2).

- A 1-cell $\mathbb{C} \xrightarrow{(n,P)} \mathbb{D}$ of **StProf** is a pair of a natural number $n \in \mathbb{N}$ and a profunctor $\mathbf{Sets}^n \times \mathbb{C} \overset{P}{\longrightarrow\!\!\!\!\!\!\mapsto} \mathbb{D}$. That is,

$$\frac{\dfrac{\mathbb{C} \xrightarrow{(n,P)} \mathbb{D} \quad \text{in } \mathbf{StProf}}{\mathbf{Sets}^n \times \mathbb{C} \;\mathrel{\mapsto}\; \mathbb{D} \quad \text{in } \mathbf{Prof}}}{\mathbb{D}^{\mathrm{op}} \times \mathbf{Sets}^n \times \mathbb{C} \longrightarrow \mathbf{Ens} \;, \quad \text{a functor}}$$

- A 2-cell $\mathbb{C} \underset{(m,Q)}{\overset{(n,P)}{\Downarrow (f,\varphi)}} \mathbb{D}$ in **StProf** is a pair (f, φ) of a function[4] $f : n \to m$ and a 2-cell

$$\begin{array}{ccc} \mathbf{Sets}^n \times \mathbb{C} & \overset{P}{\longrightarrow\!\!\!\!\!\!\mapsto} & \mathbb{D} \\ {\scriptstyle \prod_f \times \mathbb{C}} \big\downarrow & \Downarrow \varphi & \nearrow_{Q} \\ \mathbf{Sets}^m \times \mathbb{C} & & \end{array} \qquad \text{in } \mathbf{Prof}.$$

  Here the functor $\prod_f : \mathbf{Sets}^n \to \mathbf{Sets}^m$ is defined by

$$\prod_f \;:\; (X_1, \ldots, X_n) \quad \longmapsto \quad \Big(\prod_{i \in f^{-1}(1)} X_i \,,\, \ldots \,,\, \prod_{i \in f^{-1}(m)} X_i \Big) \;;$$

  where, to be precise, the set $\prod_{i \in f^{-1}(j)} X_i$ is defined to be $X_{i_1} \times (\cdots \times (X_{i_{k-1}} \times X_{i_k}) \cdots)$, with $i_1 < \ldots < i_k$ is the increasing enumeration of the set $f^{-1}(j) = \{i_1, \ldots, i_k\}$. In other words, the component φ above is a natural transformation

$$P(D, X_1, \ldots, X_n, C) \xrightarrow{\varphi_{D,X_1,\ldots,X_n,C}} Q(D, \prod_{i \in f^{-1}(1)} X_i, \ldots, \prod_{i \in f^{-1}(m)} X_i, C) \;, \quad \text{natural in } D, X_1, \ldots, X_n, C.$$

- Composition of 1-cells: given successive $\mathbb{C} \xrightarrow{(n,P)} \mathbb{D} \xrightarrow{(m,Q)} \mathbb{E}$, its composition is defined to be $\mathbb{C} \xrightarrow{(m+n, Q \circledcirc P)} \mathbb{E}$, where the profunctor $Q \circledcirc P : \mathbf{Sets}^{m+n} \times \mathbb{C} \mathrel{\mapsto} \mathbb{E}$ is the following composite

$$Q \circledcirc P \quad := \quad \big(\mathbf{Sets}^{m+n} \times \mathbb{C} \overset{\mathbf{Sets}^m \times P}{\longrightarrow\!\!\!\!\!\!\mapsto} \mathbf{Sets}^m \times \mathbb{D} \overset{Q}{\longrightarrow\!\!\!\!\!\!\mapsto} \mathbb{E} \big) \;, \tag{12}$$

  that is,

$$(Q \circledcirc P)(E, X_1, \ldots, X_m, X'_1, \ldots, X'_n, C) = \int^{D} Q(E, X_1, \ldots, X_m, D) \times P(D, X'_1, \ldots, X'_n, C) \;.$$

[4]Here the natural number n is identified with the n-element set $\{1, 2, \ldots, n\}$.

- Identity 1-cells: given $\mathbb{C} \in \mathbf{StProf}$, the identity 1-cell on $\mathbb{C}$ is defined to be $(0, \mathrm{Hom}) : \mathbb{C} \mathrel{\circ\!\!\to} \mathbb{C}$, using the hom-functor.

- Horizontal composition of 2-cells: given 2-cells $\mathbb{C} \underset{(n',P')}{\overset{(n,P)}{\Downarrow (f,\varphi)}} \mathbb{D} \underset{(m',Q')}{\overset{(m,Q)}{\Downarrow (g,\psi)}} \mathbb{E}$ in **StProf**, their horizontal composition is defined to be $\mathbb{C} \underset{(m'+n',Q' \circledcirc P')}{\overset{(m+n,Q \circledcirc P)}{\Downarrow (g+f,\psi \circledcirc \varphi)}} \mathbb{E}$. Here the function $g + f : m + n \to m' + n'$ is the obvious one:

$$(g + f)(i) = \begin{cases} g(i) & \text{if } i \le m, \\ m' + f(i - m) & \text{if } i > m; \end{cases} \tag{13}$$

and the natural transformation $\psi \circledcirc \varphi$ is defined as follows.

$$(\psi \circledcirc \varphi)_{E,\vec{X},\vec{Y},C} := \int^{D \in \mathbb{D}} \psi_{E,\vec{X},D} \times \varphi_{D,\vec{Y},C} \quad :$$

$$\int^{D} Q(E, X_1, \ldots, X_m, D) \times P(D, Y_1, \ldots, Y_n, C)$$

$$\longrightarrow \int^{D} Q'(E, \prod_{i \in g^{-1}(1)} X_i, \ldots, \prod_{i \in g^{-1}(m')} X_i, D) \times P'(D, \prod_{j \in f^{-1}(1)} Y_j, \ldots, \prod_{j \in f^{-1}(n')} Y_j, C) \ .$$

That is, using a diagram in **Prof**,

$$\psi \circledcirc \varphi \ := \ \begin{array}{ccccc} & \mathbf{Sets}^{m+n} \times \mathbb{C} & \xrightarrow{\mathbf{Sets}^{m} \times P} & \mathbf{Sets}^{m} \times \mathbb{D} & \xrightarrow[\Downarrow \psi]{Q} \ \mathbb{E} \\ \textstyle\prod_{g+f} \times \mathbb{C} & {\scriptstyle \prod_g \times \mathbf{Sets}^n \times \mathbb{C}} \downarrow & (*) & \downarrow {\scriptstyle \prod_g \times \mathbb{D}} & \nearrow {\scriptstyle Q'} \\ (\dagger) & \mathbf{Sets}^{m'+n} \times \mathbb{C} & \xrightarrow{\mathbf{Sets}^{m'} \times P} & \mathbf{Sets}^{m'} \times \mathbb{D} & \\ & {\scriptstyle \mathbf{Sets}^{m'} \times \prod_f \times \mathbb{C}} \downarrow & \Downarrow \mathbf{Sets}^{m'} \times \varphi & \nearrow {\scriptstyle \mathbf{Sets}^{m'} \times P'} & \\ & \mathbf{Sets}^{m'+n'} \times \mathbb{C} & & & \end{array} \tag{14}$$

where $(*)$ commutes up-to an isomorphism because of the bifunctoriality of $\times$ (Lem. 2.11); and $(\dagger)$ commutes due to the definition of $\prod_{g+f}$.

- Vertical composition of 2-cells: given 2-cells $\mathbb{C} \ \begin{array}{c} {\scriptstyle (n,P)} \\ {\scriptstyle (m,Q)} \ \Downarrow (f,\varphi) \\ \Downarrow (g,\psi) \\ {\scriptstyle (k,R)} \end{array} \ \mathbb{D}$ in **StProf**, their vertical composition is defined to be $\mathbb{C} \underset{(k,R)}{\overset{(n,P)}{\Downarrow (g \circ f,\psi \odot \varphi)}} \mathbb{D}$. Here the natural transformation $\psi \odot \varphi$ is defined by the following composite.

$$\begin{array}{ccc} P(D, X_1, \ldots, X_n, C) & \xrightarrow{\varphi_{D,X_1,\ldots,X_n,C}} & Q(D, \prod_{i \in f^{-1}(1)} X_i, \ldots, \prod_{i \in f^{-1}(m)} X_i, C) \\ {\scriptstyle (\psi \odot \varphi)_{D,\vec{X},C}} \downarrow & & \downarrow {\scriptstyle \psi_{D,\prod X_i,\ldots,\prod X_i,C}} \\ R(D, \prod_{i \in (g \circ f)^{-1}(1)} X_i, \ldots, \prod_{i \in (g \circ f)^{-1}(k)} X_i, C) & \underset{\cong}{\longleftarrow} & R(D, \prod_{j \in g^{-1}(1)} \prod_{i \in f^{-1}(j)} X_i, \ldots, \prod_{j \in g^{-1}(k)} \prod_{i \in f^{-1}(j)} X_i, C) \end{array}$$

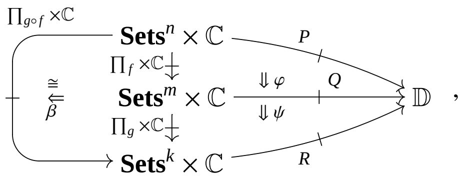
The isomorphism on the bottom row arises from the canonical isomorphisms

$$\prod_{j\in g^{-1}(1)}\prod_{i\in f^{-1}(j)} X_i \xrightarrow{\cong} \prod_{i\in(g\circ f)^{-1}(1)} X_i \ , \quad \ldots , \quad \prod_{j\in g^{-1}(k)}\prod_{i\in f^{-1}(j)} X_i \xrightarrow{\cong} \prod_{i\in(g\circ f)^{-1}(k)} X_i \qquad (15)$$

in a symmetric monoidal category (**Sets**, $\times$, 1); note that we have $(g\circ f)^{-1}(l) = \coprod_{j\in g_{-1}(l)} f^{-1}(j)$.

Using a diagram in **Prof**, the above definition amounts to

$\psi \odot \varphi \quad :=$



(16)

where the isomorphism β is the canonical ones like in (15), bundled up together. We shall refer to such β as a *normalizing isomorphism*.

It is straightforward to verify that the above data together form a bicategory.

As we did for **Prof** (see (9)), we also extend the operation $\times$ of taking product categories to a "tensor" in **StProf**. Given $\mathbb{C} \overset{(n,F)}{\multimapdotboth} \mathbb{D}$ and $\mathbb{C}' \overset{(n',F')}{\multimapdotboth} \mathbb{D}'$, we define the 1-cell $\mathbb{C}\times\mathbb{C}' \overset{(n,F)\times(n',F')}{\multimapdotboth} \mathbb{D}\times\mathbb{D}'$ to be the pair $(n+n', F\times F') : \mathbb{C}\times\mathbb{C}' \multimapdotboth \mathbb{D}\times\mathbb{D}'$, where $F\times F'$ is the profunctor defined in (9).

Lemma 5.7. *Let A be an arrow over* **Sets***. The 1-cell*

$$(1,\overline{A}) \ : \ \mathbf{Sets} \multimapdotboth \mathbf{Sets} \quad \textit{in } \mathbf{StProf},$$

where a profunctor $\overline{A}$ is defined by

$$\overline{A} \ : \ \mathbf{Sets}^{\mathrm{op}}\times\mathbf{Sets}\times\mathbf{Sets} \longrightarrow \mathbf{Sets} \hookrightarrow \mathbf{Ens} \ , \quad (J,X,K) \longmapsto A(J, X\times K) \ ,$$

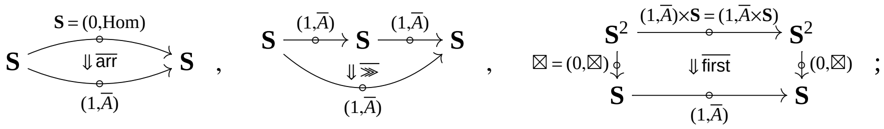
is canonically an internal strong monad in **StProf**.

Notations 5.8. In what follows we often denote the category **Sets** by **S**, for the sole purpose of saving space. The operation $\times$: **Sets** $\times$ **Sets** $\to$ **Sets** of taking products of sets is often denoted by $\boxtimes$ instead, to distinguish it from product of categories and the "tensors" on **Prof** and **StProf**.

Recall that we denote a functor $F:\mathbb{C}\to\mathbb{D}$, embedded in **Prof** by taking its direct image F_*, also by F (Notations 2.10). We shall extend this convention to **StProf**. Namely, given a functor $F:\mathbb{C}\to\mathbb{D}$, we denote its embedding $(0,F_*):\mathbb{C}\multimapdotboth\mathbb{D}$ also by F. Recall also that we often denote the identity 1-cell $\mathrm{id}_{\mathbb{C}}:\mathbb{C}\nrightarrow\mathbb{C}$ in **Prof** by $\mathbb{C}$ (Notations 2.10); we shall use this convention for **StProf**, too.

Proof. (Of Lem. 5.7) What we need to do is

- to equip the 1-cell $(1,\overline{A})$ with the following "operator" 2-cells, all of them in **StProf**:



- and to prove that these 2-cells satisfy the same equational axioms as in Table 1.

A 2-cell of the type of $\overline{\mathsf{arr}}$ above is the same thing as a 2-cell

$$\begin{array}{ccc} \mathbf{1}\times\mathbf{S} & \overset{\mathbf{S}}{\longrightarrow\!\!\!\!\!\!\!+\;\;} & \mathbf{S} \\ {\scriptstyle 1\times\mathbf{S}}\downarrow & \Downarrow & \nearrow {\scriptstyle \overline{A}=A(-,+_1\boxtimes+_2)} \\ \mathbf{S}\times\mathbf{S} & & \end{array} \quad , \quad \text{hence} \quad \begin{array}{ccc} \mathbf{1}\times\mathbf{S} & \overset{\pi_2}{\longrightarrow} & \mathbf{S} \\ {\scriptstyle 1\times\mathbf{S}}\downarrow & \Downarrow & \uparrow{\scriptstyle A} \\ \mathbf{S}\times\mathbf{S} & \underset{\boxtimes}{\longrightarrow} & \mathbf{S} \end{array} \; ;$$

here we used the equality

$$\overline{A} = (\,\mathbf{S}^2 \overset{\boxtimes}{\longrightarrow} \mathbf{S} \overset{A}{\longrightarrow} \mathbf{S}\,) \tag{17}$$

that follows from the Yoneda lemma (Lem. 2.5).

We construct a 2-cell of the last type as follows, using A's arrow structure (specifically its arr operator). The iso 2-cell λ therein is the left unitality $\lambda_X : 1 \boxtimes X \overset{\cong}{\Rightarrow} X$, embedded in **Prof**.

$$\overline{\mathsf{arr}} \;:=\; \begin{array}{ccc} \mathbf{S} & \overset{\mathbf{S}}{\longrightarrow} & \mathbf{S} \\ {\scriptstyle \langle 1,\mathbf{S}\rangle}\downarrow & \cong\Downarrow\lambda^{-1} & {\scriptstyle \mathbf{S}}\big(\overset{\mathsf{arr}}{\Rightarrow}\big){\scriptstyle A} \\ \mathbf{S}^2 & \underset{\boxtimes}{\longrightarrow} & \mathbf{S} \end{array} \tag{18}$$

Similarly, a 2-cell of the type of $\overline{\ggg}$ is identified with $\begin{array}{ccc} \mathbf{S}^2\times\mathbf{S} & \overset{\overline{A}\circledcirc\overline{A}}{\longrightarrow} & \mathbf{S} \\ {\scriptstyle \boxtimes\times\mathbf{S}}\downarrow & \Downarrow & \nearrow{\scriptstyle \overline{A}} \\ \mathbf{S}\times\mathbf{S} & & \end{array}$; using the definition (12) of $\circledcirc$ and also (17), this is further identified with the 2-cell $\begin{array}{ccc} \mathbf{S}^3 \overset{\mathbf{S}\times\boxtimes}{\longrightarrow} \mathbf{S}^2 \overset{\mathbf{S}\times A}{\longrightarrow} \mathbf{S}^2 \overset{\boxtimes}{\longrightarrow} \mathbf{S} \overset{A}{\longrightarrow} \mathbf{S} \\ {\scriptstyle \boxtimes\times\mathbf{S}}\downarrow \qquad \Downarrow \qquad \uparrow{\scriptstyle A} \\ \mathbf{S}^2 \underset{\boxtimes}{\longrightarrow} \mathbf{S} \end{array}$ in **Prof**. Such a 2-cell can be constructed as follows.

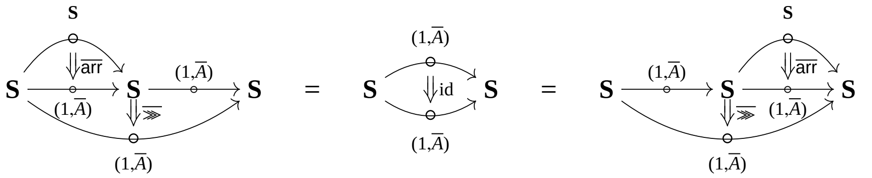
$$\overline{\ggg} \;:=\; \begin{array}{ccccccc} \mathbf{S}^3 & \overset{\mathbf{S}\times\boxtimes}{\longrightarrow} & \mathbf{S}^2 & \overset{\mathbf{S}\times A}{\longrightarrow} & \mathbf{S}^2 & \overset{\boxtimes}{\longrightarrow} & \mathbf{S} \\ {\scriptstyle \boxtimes\times\mathbf{S}}\downarrow & \cong\Downarrow\alpha^{-1} & \searrow{\scriptstyle \boxtimes} & \Downarrow\mathsf{second} & \nearrow{\scriptstyle A} & \Downarrow\ggg & \downarrow{\scriptstyle A} \\ \mathbf{S}^2 & & \underset{\boxtimes}{\longrightarrow} & & \mathbf{S} & \underset{A}{\longrightarrow} & \mathbf{S} \end{array} \tag{19}$$

Here α is the associativity isomorphism $(X\boxtimes Y)\boxtimes Z \overset{\cong}{\Rightarrow} X\boxtimes(Y\boxtimes Z)$; second is a derived operator of the arrow A (Def. 3.5); and $\ggg$ is an arrow operator of A.

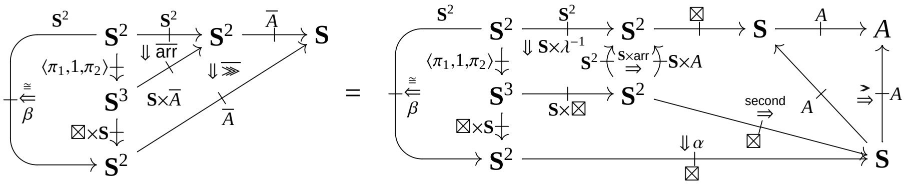
A 2-cell of the type of $\overline{\mathsf{first}}$ is identified with $\begin{array}{ccc} \mathbf{S}\times\mathbf{S}^2 & \overset{\boxtimes\circledcirc(\overline{A}\times\mathbf{S})}{\longrightarrow} & \mathbf{S}\times\mathbf{S}^2 \\ {\scriptstyle \mathbf{S}\times\mathbf{S}^2}\downarrow & \Downarrow & \nearrow{\scriptstyle \overline{A}\circledcirc\boxtimes} \\ \mathbf{S}\times\mathbf{S} & & \end{array}$ in **Prof**, by the definition of **StProf**. Again expanding $\circledcirc$ and $\overline{A}$ using (12) and (17), it is identified with a 2-cell $\begin{array}{ccccc} \mathbf{S}^3 & \overset{\boxtimes\times\mathbf{S}}{\longrightarrow} & \mathbf{S}^2 & \overset{A\times\mathbf{S}}{\longrightarrow} & \mathbf{S}^2 \\ {\scriptstyle \mathbf{S}\times\boxtimes}\downarrow & & \Downarrow & & \downarrow{\scriptstyle\boxtimes} \\ \mathbf{S}^2 & \underset{\boxtimes}{\longrightarrow} & \mathbf{S} & \underset{A}{\longrightarrow} & \mathbf{S} \end{array}$ in **Prof**. We define

$$\overline{\mathsf{first}} \;:=\; \begin{array}{ccccc} \mathbf{S}^3 & \overset{\boxtimes\times\mathbf{S}}{\longrightarrow} & \mathbf{S}^2 & \overset{A\times\mathbf{S}}{\longrightarrow} & \mathbf{S}^2 \\ {\scriptstyle \mathbf{S}\times\boxtimes}\downarrow & \cong\Downarrow\alpha & \downarrow{\scriptstyle\boxtimes} & \Downarrow\mathsf{first} & \downarrow{\scriptstyle\boxtimes} \\ \mathbf{S}^2 & \underset{\boxtimes}{\longrightarrow} & \mathbf{S} & \underset{A}{\longrightarrow} & \mathbf{S} \end{array} \; . \tag{20}$$

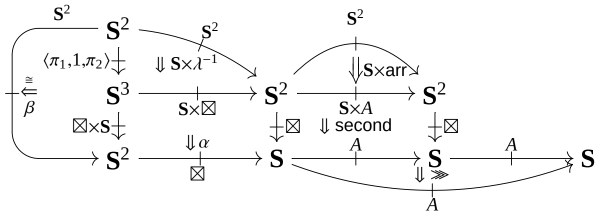
It remains to verify the equational axioms. Take the axiom (UNIT):

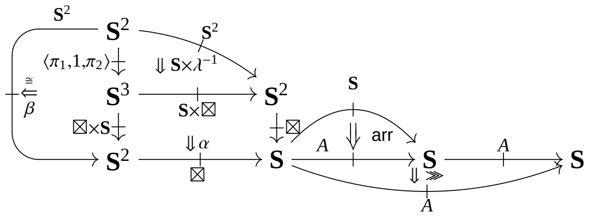
 (21)

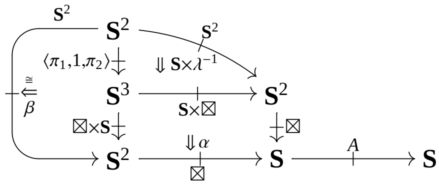
Using (14) and (16), the composed 2-cell on the left-hand side is the same thing as the following 2-cell in **Prof**. Recall that β is the normalizing isomorphism in (16).



by def. of $\overline{\mathsf{arr}}, \overline{\ggg}$

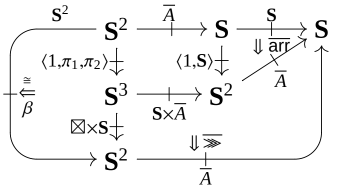
=  by reorganizing 2-cells

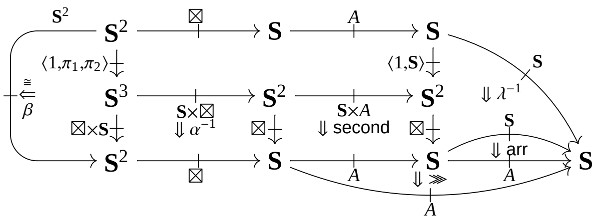
=  by (second-arr) in Lem. 3.7

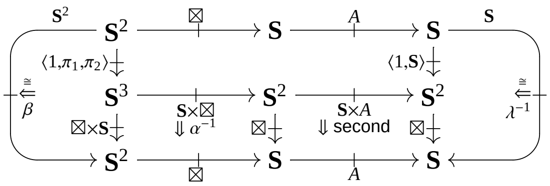
=  by (UNIT) for A

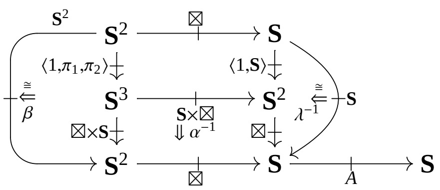
= $\mathrm{id}_{A\circ\boxtimes}$ by coherence, note that α, β, λ are all canonical isomorphisms in an SMC.

Similarly, the composed 2-cell on the right-hand side of (21) is the following 2-cell in **Prof**.



$=$  by def. of $\overline{\mathsf{arr}}$, $\overline{\ggg}$

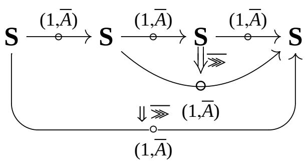
$=$  by (UNIT) for A

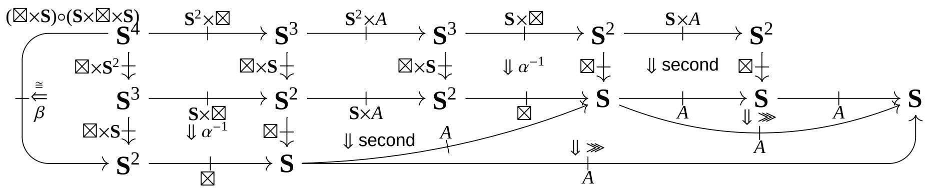
$=$  by (second-λ) in Lem. 3.7

$= \quad \mathrm{id}_{A \circ \boxtimes}$ by coherence for $(\mathbf{S}, \boxtimes, 1)$.

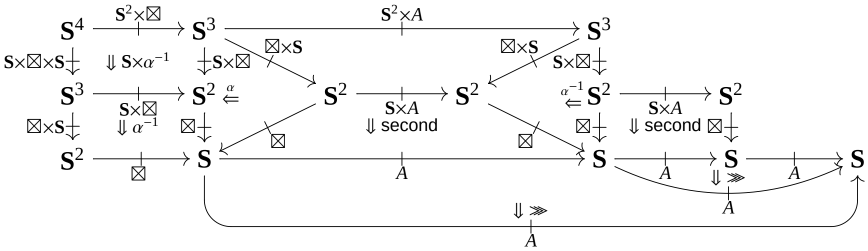
Similar straightforward calculations verify the other axioms for $(1, \overline{A})$ in **StProf**. In its course we use (14), (16), the axioms for A (Table 1) and the equalities in Table 2. We present the proofs for the axioms (Assoc) and (first-$\ggg$). Recall that 1-cells denoted by $\twoheadrightarrow$ means the diagram is in **StProf**; 1-cells denoted by $\nrightarrow$ means it is in **Prof**.

For the axiom (Assoc),



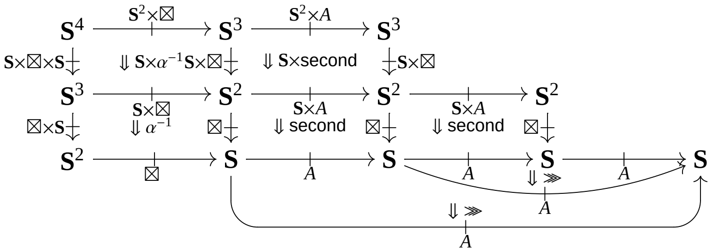
$=$ 

=



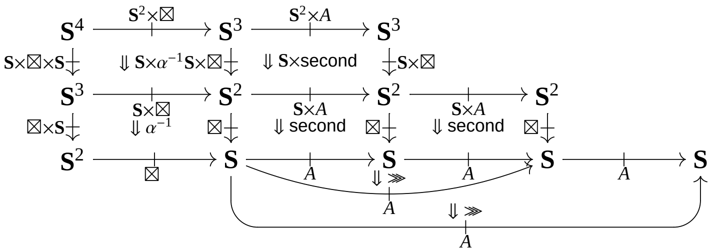
by coherence for $(\mathbf{S}, \boxtimes, 1)$

=



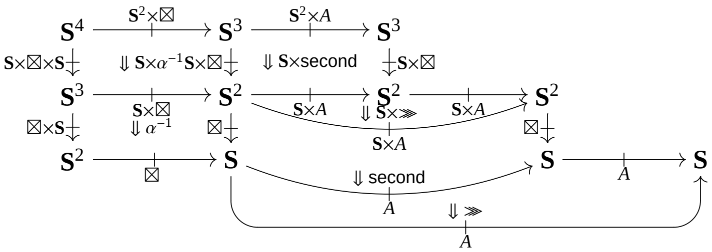
by (second-α)

=



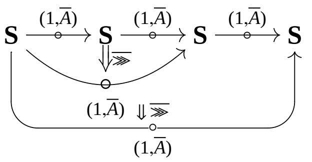
by (Assoc)

=



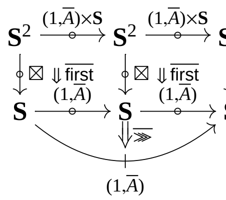
by (second-⋙)

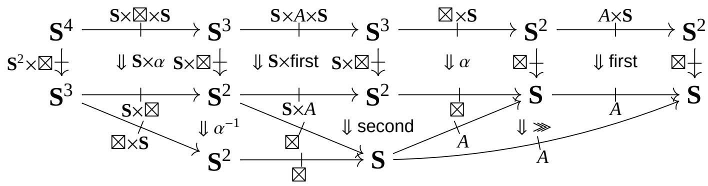
=

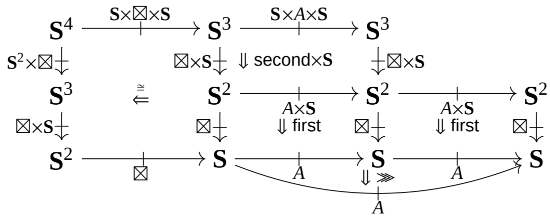


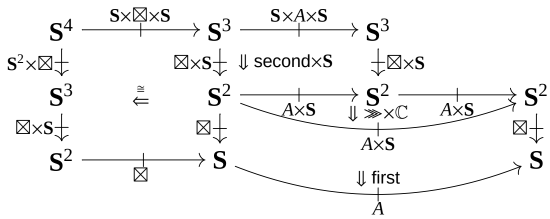
.

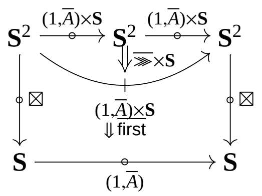
For the axiom (first-⋙),



= 

=  by (first-second) in Lem. 3.7

=  by (first-⋙)

=  .

This concludes the proof. □

Lemma 5.9. *Let* $\mathbf{S} \overset{(1,P)}{\longrightarrow} \mathbf{S}$ *be an internal strong monad in* **StProf**, *equipped with "operator" 2-cells* $\overline{\mathsf{arr}}$, $\overline{⋙}$ *and* $\overline{\mathsf{first}}$. *Assume that* P *is* **Sets**-valued*: that is,*

> *for each* $J, X, K \in$ **Sets**, *the collection* $P(J, X, K) \in$ **Ens** *is small and hence belongs to the category* **Sets**.

Then the family $\{F_{J,K}\}_{J,K \in \mathbf{Sets}}$ *of functors, defined by*

$$F_{J,K} \;:=\; P(J, \_\,, K) \quad : \quad \mathbf{Sets} \longrightarrow \mathbf{Sets}$$

is canonically a lax arrow functor (Def. 4.6).

Proof. What we have to do is to define three "operators" $F_{\mathsf{arr} f}$, $F_{\ggg}$ and F_{first} and show that they satisfy the equalities in Table 3.

Given a function $f : J \to K$ in **Sets**, to define $F_{\mathsf{arr} f} : 1 \longrightarrow F_{J,K}1 = P(J,1,K)$ we use the "operator" $\mathbf{S} \underset{(1,P)}{\overset{(0,\mathbf{S})}{\Downarrow \overline{\mathsf{arr}}}} \mathbf{S}$. By Def. 5.6, the latter 2-cell in **StProf** is identified with a natural transformation $\overline{\mathsf{arr}}_{J,K} : \mathbf{S}(J,K) \longrightarrow P(J,1,K)$, natural in J,K. We set

$$F_{\mathsf{arr} f} \;:=\; (\overline{\mathsf{arr}}_{J,K})(f) \;. \tag{22}$$

Similarly by Def. 5.6, the operator $\overline{\ggg}$ is identified with a natural transformation

$$\overline{\ggg}_{J,X,Y,L} \;:\; \int^{K\in\mathbf{S}} P(J,X,K)\times P(K,Y,L) \longrightarrow P(J,X\times Y,L) \;, \quad \text{natural in } J,X,Y,L.$$

We set $(F_{\ggg_{J,K,L}})_{X,Y}$ to be the composite

$$(F_{\ggg_{J,K,L}})_{X,Y} \;:=\; \Big(P(J,X,K)\times P(K,Y,L) \xrightarrow{\iota_K} \int^{K\in\mathbf{S}} P(J,X,K)\times P(K,Y,L) \xrightarrow{\overline{\ggg}_{J,X,Y,L}} P(J,X\times Y,L) \Big) \;. \tag{23}$$

Here ι_K denotes a coprojection into a coend.

To define F_{first}, note the following bijective correspondence

$$\Phi_{J,X,K,L,Y,M} \;:\; [\,P(J,X,K), P(L,Y,M)\,] \xrightarrow{\cong} \mathrm{Nat}(\,[\_\,,L]\times P(J,X,K)\,,\; P(\_\,,Y,M)\,) \;; \tag{24}$$

where $[-,+]$ denotes the function space, i.e. the hom-set functor for $\mathbf{S}$. We denote the correspondence by Φ. The correspondence is derived from the Yoneda lemma (Lem. 2.4, see also Cor. 2.3) as well as from the adjunction $S\times\_ \dashv [S,\_]$; due to the naturality of the both ingredients, the correspondence Φ in (24) is obviously natural in J,X,K,L,Y,M.

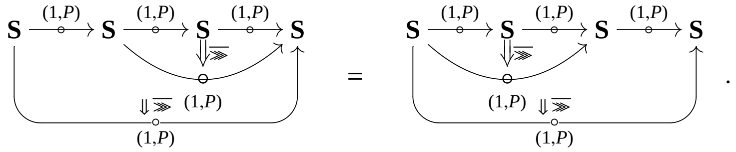
By Def. 5.6, the operator $\begin{array}{ccc} \mathbf{S}^2 & \xrightarrow{(1,P)\times\mathbf{S}\,=\,(1,P\times\mathbf{S})} & \mathbf{S}^2 \\ {\scriptstyle \boxtimes = (0,\boxtimes)}\downarrow & \Downarrow \overline{\mathsf{first}} & \downarrow{\scriptstyle (0,\boxtimes)} \\ \mathbf{S} & \xrightarrow[(1,P)]{} & \mathbf{S} \end{array}$ is a natural transformation

$$\overline{\mathsf{first}}_{U,X,K,L} \;:\; \int^{J,V\in\mathbf{S}} [U, J\times V]\times P(J,X,K)\times[V,L] \longrightarrow \int^{W\in\mathbf{S}} P(U,X,W)\times[W,K\times L] \;;$$

using this and (24), we define F_{first} as follows.

$$(F_{\mathsf{first}_{J,K,L}})_X \;:=\; \Phi^{-1}_{J,X,K,J\times L,X,K\times L} \left[\begin{array}{l} [\_\,,J\times L]\times P(J,X,K) \\ \xrightarrow[\text{Yoneda}]{\cong} \int^V [\_\,,J\times V]\times P(J,X,K)\times[V,L] \\ \xrightarrow{\iota_J} \int^{J,V} [\_\,,J\times V]\times P(J,X,K)\times[V,L] \\ \xrightarrow{\overline{\mathsf{first}}_{\_,X,K,L}} \int^W P(\_\,,X,W)\times[W,K\times L] \\ \xrightarrow[\text{Yoneda}]{\cong} P(\_\,,X,K\times L) \end{array} \right] \;. \tag{25}$$

It remains to verify the equalities in Table 3. For ($\ggg$-Assoc), we use the axiom (Assoc) (see Table 1) for the strong monad $\mathbf{S} \xrightarrow[\circ]{(1,P)} \mathbf{S}$. Namely,



On each side of the equation is a natural transformation between functors of the type $\mathbf{S}^{\mathrm{op}} \times \mathbf{S}^3 \times \mathbf{S} \longrightarrow \mathbf{Ens}$. We take the J, X, Y, U, M-component and obtain the following equality.

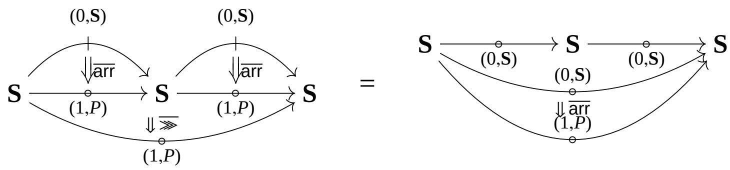
$$\left[\begin{array}{l} \qquad\qquad \int^{K,L} P(J,X,K) \times P(K,Y,L) \times P(L,U,M) \\ \xrightarrow{\int^L \overline{\ggg}_{J,X,Y,L} \times P(L,U,M)} \int^L P(J, X \times Y, L) \times P(L,U,M) \\ \xrightarrow{\overline{\ggg}_{J,(X\times Y),U,M}} P(J, (X \times Y) \times U, M) \\ \xrightarrow[\beta]{\cong} P(J, X \times (Y \times U), M) \end{array}\right]$$

$$= \left[\begin{array}{l} \qquad\qquad \int^{K,L} P(J,X,K) \times P(K,Y,L) \times P(L,U,M) \\ \xrightarrow{\int^K P(J,X,K) \times \overline{\ggg}_{K,Y,U,M}} \int^L P(J,X,K) \times P(K, Y \times U, M) \\ \xrightarrow{\overline{\ggg}_{J,X,Y\times U,M}} P(J, X \times (Y \times U), M) \end{array}\right]$$

By pre-composing the coprojection $\iota_{K,L}$ to each side and using the definition (23), we obtain the commutativity of the following diagram, which is what we aimed to prove.

$$\begin{array}{ccc} P(J,X,K) \times P(K,Y,L) \times P(L,U,M) & \xrightarrow{\mathrm{id}\times F_{>}} & P(J,X,K) \times P(K, Y \times U, M) \\ \downarrow{\scriptstyle F_{>}\times \mathrm{id}} & & \\ P(J, X \times Y, L) \times P(L,U,M) & & \downarrow{\scriptstyle F_{>}} \\ \downarrow{\scriptstyle F_{>}} & & \\ P(J, (X \times Y) \times U, M) & \xrightarrow[\beta]{\cong} & P(J, X \times (Y \times U), M) \end{array}$$

For (arr-Func1), we use the following equality obtained by using (Unit) in Table 1 twice:



By taking the J, L-component of each side and pre-composing the coprojection ι_K, we obtain the diagram of (arr-Func1). The axiom (arr-Func2) can be verified in the same manner.

In verifying the other axioms (i.e. those which involve F_{first}), the correspondence Φ in (24) is crucial. We will also be using the following fact.

Sublemma 5.10. *Let $f : K \to K'$ and $g : J' \to J$ be functions, i.e. morphisms in* **Sets**. *We have*

$$\left[\begin{array}{l} \quad\quad\quad P(J,X,K) \\ \xrightarrow{\langle F_{\mathsf{arr}\,g},\mathrm{id},F_{\mathsf{arr}\,f}\rangle} P(J',1,J)\times P(J,X,K)\times P(K,1,K') \\ \xrightarrow{F_{\gg}\times P(K,1,K')} P(J',1\times X,K)\times P(K,1,K') \\ \xrightarrow{F_{\gg}} P(J',(1\times X)\times 1,K') \\ \xrightarrow{\cong} P(J',X,K') \end{array}\right] = \left[\begin{array}{l} \quad\quad P(J,X,K) \\ \xrightarrow{P(g,X,f)} P(J',X,K') \end{array}\right] ;$$

that is, the composite on the left can be reduced to the functoriality of P.

Proof. (Of the sublemma) The proof is similar to the above verification of the axiom (arr-Func1). Therein it is crucial that the Yoneda correspondence (cf. Lem. 2.5)

$$\int^{J} [J',J]\times P(J,X,K) \xrightarrow{\cong} P(J',X,K)$$

is concretely given by the functoriality of P, carrying an element (g,p) of the left-hand side to $P(g,X,K)(p)$. □

Let us turn to (ρ-Nat) in Table 3. Sublem. 5.10 reduces the axiom to the commutativity of the following diagram.

$$\begin{array}{ccc} P(J,X,K) & \xrightarrow[\cong]{P(\rho_J,X,K)} & \\ {\scriptstyle (F_{\mathsf{first}_{J,K,1}})_X}\downarrow & & \\ P(J\times 1,X,K\times 1) & \xrightarrow[P(J\times 1,X,\rho_K)]{\cong} & P(J\times 1,X,K) \end{array}$$

Since Φ in (24) is bijective, it suffices to show $\Phi(P(J\times 1,X,\rho_K)\circ(F_{\mathsf{first}_{J,K,1}})_X) = \Phi(P(\rho_J,X,K))$.

$$\begin{array}{lll} & \Phi(\, P(J\times 1,X,\rho_K)\circ(F_{\mathsf{first}_{J,K,1}})_X\,) & \\ = & P(\_\,,X,\rho_K)\circ\Phi((F_{\mathsf{first}_{J,K,1}})_X) & \text{by naturality of } \Phi \\ = & \left[\begin{array}{l} \quad\quad [\_\,,J\times 1]\times P(J,X,K) \\ \xRightarrow[\text{Yoneda}]{\cong} \int^{V}[\_\,,J\times V]\times P(J,X,K)\times[V,1] \\ \xRightarrow{\iota_J} \int^{J,V}[\_\,,J\times V]\times P(J,X,K)\times[V,1] \\ \xRightarrow{\overline{\mathsf{first}}_{\_,X,K,1}} \int^{W} P(\_\,,X,W)\times[W,K\times 1] \\ \xRightarrow[\text{Yoneda}]{\cong} P(\_\,,X,K\times 1) \\ \xRightarrow[P(\_,X,\rho_K)]{} P(\_\,,X,K) \end{array}\right] & \text{by def. of } F_{\mathsf{first}}\ (25) \\ = & \left[\begin{array}{l} \quad\quad [\_\,,J\times 1]\times P(J,X,K) \\ \xRightarrow{[\_,\rho_J]\times P(J,X,K)} [\_\,,J]\times P(J,X,K) \\ \xRightarrow[\text{Yoneda}]{\cong} P(\_\,,X,K) \end{array}\right] & \text{by } (\mathsf{first}\text{-}\rho) \text{ in Table 1} \\ = & \Phi(P(\rho_J,X,K))\ . & \end{array}$$

The other axioms are verified in a similar manner. This concludes the proof of Lem. 5.9. □

Lem. 5.7 and 5.9 proves Lem. 5.5, which in combination with Prop. 4.7 proves our main result, Thm. 4.4.

6. Conclusions and Future Work

Inspired by the common graphical understanding (boxes connected by typed wires), we have elaborated on a connection between computations and components; more specifically, algebraic structure possessed by these. The algebraic structure of computations has been axiomatized by the notion of arrow—by Hughes [4]—which is equivalent to that of Freyd category [5, 6]. We have demonstrated that the arrow structure is also carried by components. Its operators (arr, $\ggg$ and first) serve as *connectors* between components, hence as a basic *component calculus*. The latter "component-arrow" turns out to be a *categorified* [12] notion of arrow, whose satisfaction of axioms only up-to isomorphisms is exemplary.

Our technical contribution is as follows. Arrow-based *A-components*—described as coalgebras, with A representing the machines' computation effect—carry canonical (categorified) arrow structure, which is in fact a lifting of the arrow structure of A itself. The "lifting" is best presented in **Prof**, the bicategory of categories and profunctors. There we rely on the second author's observation [16] that an arrow A is the same thing as an internal strong monad in **Prof**. When compared to the previous workshop version [1], the current version presents the lifting process in a more structural manner, using a novel bicategory **StProf**.

The notion of categorical arrow, as a component calculus, is very basic. In fact for the notion of arrow (composing computations) some extensions have been proposed. Notable among them is an extension with a feedback/loop operator [28, 29]. Its categorified version—that is, the corresponding (extended) component calculus—has been studied in [24]. However, unlike the current work, the calculations in [24] are all direct and do not happen in **Prof**. Much like the characterization in [16], the current authors have formulated an arrow with loop as a monad in **Prof** with suitable additional structure. Unfortunately we have not yet found its good use.

Acknowledgments. Thanks are due to Paul-André Melliès for advocating use of profunctors; to Marcelo Fiore, Bart Jacobs, Bartek Klin and Paul Blain Levy for helpful discussions; and to the reviewers of the earlier version [1] for useful comments.

[1] K. Asada, I. Hasuo, Categorifying computations into components via arrows as profunctors, in: Coalgebraic Methods in Computer Science (CMCS 2010), Elect. Notes in Theor. Comp. Sci. To appear.

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

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

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

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

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

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

[8] R. Atkey, What is a categorical model of arrows?, in: V. Capretta, C. McBride (Eds.), Mathematically Structured Functional Programming.

[9] L. Barbosa, Components as Coalgebras, Ph.D. thesis, Univ. Minho, 2001.

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

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

[12] J. C. Baez, J. Dolan, Categorification, Contemp. Math. 230 (1998) 1–36.

[13] J. C. Baez, J. Dolan, Higher dimensional algebra III: *n*-categories and the algebra of opetopes, Adv. Math 135 (1998) 145–206.

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

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

[16] K. Asada, Arrows are strong monads, in: Mathematically Structured Functional Programming (MSFP 2010). To appear.

[17] G. M. Kelly, 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] S. Mac Lane, Categories for the Working Mathematician, Springer, Berlin, 2nd edition, 1998.

[19] M. Barr, C. Wells, Toposes, Triples and Theories, Springer, Berlin, 1985. Available online.

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

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

[22] M. Fiore, Rough notes on presheaves, 2001. Available online.

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

[24] I. Hasuo, B. Jacobs, Component traces, 2010. Preprint.

[25] S. Abramsky, E. Haghverdi, P. Scott, Geometry of interaction and linear combinatory algebras, Math. Struct. in Comp. Sci. 12 (2002) 625–665.

[26] A. Joyal, R. Street, D. Verity, Traced monoidal categories, Math. Proc. Cambridge Phil. Soc. 119(3) (1996) 425–446.

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

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

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