Promonads and String Diagrams for Effectful Categories

Mario Román
Tallinn University of Technology
mroman@ttu.ee

Premonoidal and Freyd categories are both generalized by non-cartesian Freyd categories: effectful categories. We construct string diagrams for effectful categories in terms of the string diagrams for a monoidal category with a freely added object. We show that effectful categories are pseudomonoids in a monoidal bicategory of promonads with a suitable tensor product.

1 Introduction

Category theory has two sucessful applications that are rarely combined: monoidal string diagrams [21] and functional programming semantics [28]. We use string diagrams to talk about quantum transformations [1], relational queries [6], and even computability [31]; at the same time, proof nets and the geometry of interaction [11, 5] have been widely applied in computer science [2, 16]. On the other hand, we traditionally use monads and comonads, Kleisli categories and premonoidal categories to explain effectful functional programming [17, 18, 28, 34, 44]. Even if we traditionally employ Freyd categories with a cartesian base [32], we can also consider non-cartesian Freyd categories [40], which we call *effectful categories*.

Contributions. These applications are well-known. However, some foundational results in the intersection between string diagrams, premonoidal categories and effectful categories are missing in the literature. This manuscript contributes two such results.

- We introduce string diagrams for effectful categories. Jeffrey [20] was the first to preformally employ string diagrams of premonoidal categories. His technique consists in introducing an extra wire – which we call the *runtime* – that prevents some morphisms from interchanging. We promote this preformal technique into a result about the construction of free premonoidal, Freyd and effectful categories: the free premonoidal category can be constructed in terms of the free monoidal category with an extra wire.

  Our slogan, which constitutes the statement of Theorem 2.14, is

  "Premonoidal categories are Monoidal categories with a Runtime."

- We prove that effectful categories are promonad pseudomonoids. Promonads are the profunctorial counterpart of monads; they are used to encode effects in functional programming (where they are given extra properties and called *arrows* [17]). We claim that, in the same way that monoidal categories are pseudomonoids in the bicategory of categories [42], premonoidal effectful categories are pseudomonoids in a monoidal bicategory of promonads. This result justifies the role of effectful categories as a foundational object.

1.1 Synopsis

Sections 2.1 and 2.2 contain mostly preliminary material on premonoidal, Freyd and effectful categories. Our first original contribution is in Section 2.3; we prove that premonoidal categories are monoidal

Submitted to:
ACT 2022

categories with runtime (Theorem 2.14). Section 3 makes explicit the well-known theory of profunctors, promonads and identity-on-objects functors. In Section 4, we introduce the pure tensor of promonads. We use it in Section 5 to prove our second main contribution (Theorem 5.3).

2 Premonoidal and Effectful Categories

2.1 Premonoidal categories

Premonoidal categories are monoidal categories without the *interchange law*, $(f \otimes \mathrm{id}) ⨟ (\mathrm{id} \otimes g) \neq (\mathrm{id} \otimes g) ⨟ (f \otimes \mathrm{id})$. This means that we cannot tensor any two arbitrary morphisms, $(f \otimes g)$, without explicitly stating which one is to be composed first, $(f \otimes \mathrm{id}) ⨟ (\mathrm{id} \otimes g)$ or $(\mathrm{id} \otimes g) ⨟ (f \otimes \mathrm{id})$, and the two compositions are not equivalent (Figure 1).

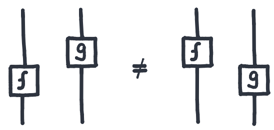

Figure 1: The interchange law does not hold in a premonoidal category.

In technical terms, the tensor of a premonoidal category $(\otimes) \colon \mathbb{C} \times \mathbb{C} \to \mathbb{C}$ is not a functor, but only what is called a *sesquifunctor*: independently functorial on each variable. Tensoring with any identity is itself a functor $(\bullet \otimes \mathrm{id}) \colon \mathbb{C} \to \mathbb{C}$, but there is no functor $(\bullet \otimes \bullet) \colon \mathbb{C} \times \mathbb{C} \to \mathbb{C}$.

A good motivation for dropping the interchange law can be found when describing transformations that affect some global state. These effectful processes should not interchange in general, because the order in which we modify the global state is meaningful. For instance, in the Kleisli category of the *writer monad*, $(\Sigma^* \times \bullet) \colon \mathsf{Set} \to \mathsf{Set}$ for some alphabet $\Sigma \in \mathsf{Set}$, we can consider the function $\mathsf{print} \colon \Sigma^* \to \Sigma^* \times 1$. The order in which we "print" does matter (Figure 2).

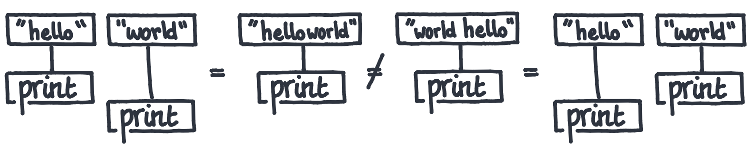

Figure 2: Writing does not interchange.

Not surprisingly, the paradigmatic examples of premonoidal categories are the Kleisli categories of Set-based monads $T \colon \mathsf{Set} \to \mathsf{Set}$ (more generally, of strong monads), which fail to be monoidal unless the monad itself is commutative [13, 33, 34, 14]. Intuitively, the morphisms are "effectful", and these effects do not always commute.

However, we may still want to allow some morphisms to interchange. For instance, apart from asking the same associators and unitors of monoidal categories to exist, we ask them to be *central*: that means that they interchange with any other morphism. This notion of centrality forces us to write the definition of premonoidal category in two different steps: first, we introduce the minimal setting in which centrality can be considered (*binoidal* categories [34]) and then we use that setting to bootstrap the full definition of premonoidal category with central coherence morphisms.

Definition 2.1 (Binoidal category). A *binoidal category* is a category $\mathbb{C}$ endowed with an object $I \in \mathbb{C}$ and an object $A \otimes B$ for each $A \in \mathbb{C}$ and $B \in \mathbb{C}$. There are functors $(A \otimes \bullet) \colon \mathbb{C} \to \mathbb{C}$, and $(\bullet \otimes B) \colon \mathbb{C} \to \mathbb{C}$ that coincide on $(A \otimes B)$, even if $(\bullet \otimes \bullet)$ is not itself a functor.

Again, this means that we can tensor with identities (whiskering), functorially; but we cannot tensor two arbitrary morphisms: the interchange law stops being true in general. The *centre*, $\mathscr{Z}(\mathbb{C})$, is the wide subcategory of morphisms that do satisfy the interchange law with any other morphism. That is, $f\colon A \to B$ is *central* if, for each $g\colon A' \to B'$,

$$(f \otimes \mathrm{id}_{A'}) ⨾ (\mathrm{id}_B \otimes g) = (\mathrm{id}_A \otimes g) ⨾ (f \otimes \mathrm{id}_{B'}), \text{ and } (\mathrm{id}_{A'} \otimes f) ⨾ (g \otimes \mathrm{id}_B) = (g \otimes \mathrm{id}_A) ⨾ (\mathrm{id}_{B'} \otimes f).$$

Definition 2.2. A *premonoidal category* is a binoidal category $(\mathbb{C}, \otimes, I)$ together with the following coherence isomorphisms $\alpha_{A,B,C}\colon A \otimes (B \otimes C) \to (A \otimes B) \otimes C$, $\rho_A\colon A \otimes I \to A$ and $\lambda_A\colon I \otimes A \to A$ which are central, natural *separately at each given component*, and satisfy the pentagon and triangle equations.

A premonoidal category is *strict* when these coherence morphisms are identities. A premonoidal category is moreover *symmetric* when it is endowed with a coherence isomorphism $\sigma_{A,B}\colon A \otimes B \to B \otimes A$ that is central and natural at each given component, and satisfies the symmetry condition and hexagon equations.

Remark 2.3. The coherence theorem of monoidal categories still holds for premonoidal categories: every premonoidal is equivalent to a strict one. We will construct the free strict premonoidal category using string diagrams. However, the usual string diagrams for monoidal categories need to be restricted: in premonoidal categories, we cannot consider two morphisms in parallel unless any of the two is *central*.

2.2 Effectful and Freyd categories

Premonoidal categories immediately present a problem: what are the strong premonoidal functors? If we want them to compose, they should preserve centrality of the coherence morphisms (so that the central coherence morphisms of $F ⨾ G$ are these of F after applying G), but naively asking them to preserve all central morphisms rules out important examples [40]. The solution is to explicitly choose some central morphisms that represent "pure" computations. These do not need to form the whole centre: it could be that some morphisms considered *effectful* just "happen" to fall in the centre of the category, while we do not ask our functors to preserve them. This is the well-studied notion of a *non-cartesian Freyd category*, which we shorten to *effectful monoidal category* or *effectful category*.[1]

Effectful categories are premonoidal categories endowed with a chosen family of central morphisms. These central morphisms are called pure morphisms, constrasting with the general, non-central, morphisms that fall outside this family, which we call effectful.

Definition 2.4. An *effectful category* is an identity-on-objects functor $\mathbb{V} \to \mathbb{C}$ from a monoidal category $\mathbb{V}$ (the pure morphisms, or "values") to a premonoidal category $\mathbb{C}$ (the effectful morphisms, or "computations"), that strictly preserves all of the premonoidal structure and whose image is central. It is *strict* when both are. A *Freyd category* [24] is an effectful category where the pure morphisms form a cartesian monoidal category.

Effectful categories solve the problem of defining premonoidal functors: a functor between effectful categories needs to preserve only the pure morphisms. We are not losing expressivity: premonoidal categories are effectful with their centre, $\mathscr{Z}(\mathbb{C}) \to \mathbb{C}$. From now on, we study effectful categories.

[1]The name "Freyd category" sometimes assumes cartesianity of the pure morphisms, but it is also used for the general case. Choosing to call "effectful categories" to the general case and reserving the name "Freyd categories" for the cartesian ones avoids this clash of nomenclature. There exists also the more fine-grained notion of "Cartesian effect category" [10], which generalizes Freyd categories and may further justify calling "effectful category" to the general case.

Definition 2.5 (Effectful functor)**.** Let $\mathbb{V} \to \mathbb{C}$ and $\mathbb{W} \to \mathbb{D}$ be effectful categories. An *effectful functor* is a quadruple $(F, F_0, \varepsilon, \mu)$ consisting of a functor $F\colon \mathbb{C} \to \mathbb{D}$ and a functor $F_0\colon \mathbb{V} \to \mathbb{W}$ making the square commute, and two natural and pure isomorphisms $\varepsilon\colon J \cong F(I)$ and $\mu\colon F(A \otimes B) \cong F(A) \otimes F(B)$ such that they make F_0 a monoidal functor. It is *strict* if these are identities.

When drawing string diagrams in an effectful category, we shall use two different colours to declare if we are depicting either a value or a computation (Figure 3).

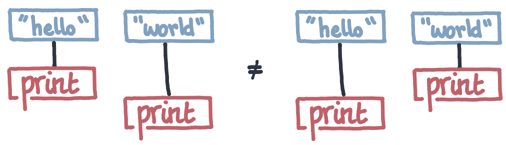

Figure 3: "Hello world" is not "world hello".

Here, the values "hello" and "world" satisfy the interchange law as in an ordinary monoidal category. However, the effectful computation "print" does not need to satisfy the interchange law. String diagrams like these can be found in the work of Alan Jeffrey [20]. Jeffrey presents a clever mechanism to graphically depict the failure of interchange: all effectful morphisms need to have a control wire as an input and output. This control wire needs to be passed around to all the computations in order, and it prevents them from interchanging.

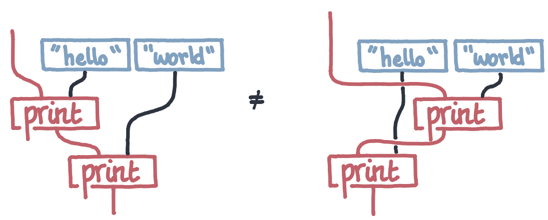

Figure 4: An extra wire prevents interchange.

A common interpretation of monoidal categories is as theories of resources. We can interpret premonoidal categories as monoidal categories with an extra resource – the "runtime" – that needs to be passed to all computations. The next section promotes Jeffrey's observation into a theorem.

2.3 Premonoidals are monoidals with runtime

String diagrams rely on the fact that the morphisms of the monoidal category freely generated over a polygraph of generators are string diagrams on these generators, quotiented by topological deformations [22]. We justify string diagrams for premonoidal categories by proving that the freely generated effectful category over a pair of polygraphs (for pure and effectful generators, respectively) can be constructed as the freely generated monoidal category over a particular polygraph that includes an extra wire.

Definition 2.6. A *polygraph* $\mathcal{G}$ (analogue of a *multigraph* [38]) is given by a set of objects, $\mathcal{G}_{\text{obj}}$, and a set of arrows $\mathcal{G}(A_0, \dots, A_n; B_0, \dots, B_m)$ for any two sequences of objects $A_0, \dots, A_n$ and $B_0, \dots, B_m$. A morphism of polygraphs $f\colon \mathcal{G} \to \mathcal{H}$ is a function between their object sets, $f_{\text{obj}}\colon \mathcal{G}_{\text{obj}} \to \mathcal{H}_{\text{obj}}$, and a function between their corresponding morphism sets,

$$f_{A_0,\dots,A_n;B_0,\dots,B_n}\colon \mathcal{G}(A_0, \dots, A_n; B_0, \dots, B_m) \to \mathcal{H}(f_{\text{obj}}(A_0), \dots, f_{\text{obj}}(A_n); f_{\text{obj}}(B_0), \dots, f_{\text{obj}}(B_m)).$$

A *polygraph couple* is a pair of polygraphs $(\mathscr{V},\mathscr{G})$ sharing the same objects, $\mathscr{V}_{obj} = \mathscr{G}_{obj}$. A morphism of polygraph couples $(u, f): (\mathscr{V},\mathscr{G}) \to (\mathscr{W},\mathscr{H})$ is a pair of morphisms of polygraphs, $u: \mathscr{V} \to \mathscr{W}$ and $f: \mathscr{G} \to \mathscr{H}$, such that they coincide on objects, $f_{obj} = u_{obj}$.

Remark 2.7. There exists an adjunction between polygraphs and strict monoidal categories. Any monoidal category $\mathbb{C}$ can be seen as a polygraph $\mathscr{U}_{\mathbb{C}}$ where the edges $\mathscr{U}_{\mathbb{C}}(A_0,\dots,A_n;B_0,\dots,B_m)$ are the morphisms $\mathbb{C}(A_0 \otimes \dots \otimes A_n, B_0 \otimes \dots \otimes B_m)$, and we forget about composition and tensoring. Given a polygraph $\mathscr{G}$, the free strict monoidal category $\mathsf{Mon}(\mathscr{G})$ is the strict monoidal category that has as morphisms the string diagrams over the generators of the polygraph.

We will construct a similar adjunction between polygraph couples and effectful categories. Let us start by formally adding the runtime to a free monoidal category.

Definition 2.8 (Runtime monoidal category)**.** Let $(\mathscr{V},\mathscr{G})$ be a polygraph couple. Its *runtime monoidal category*, $\mathsf{Mon}_{\mathsf{Run}}(\mathscr{V},\mathscr{G})$, is the monoidal category freely generated from adding an extra object – the runtime, R – to the input and output of every effectful generator in $\mathscr{G}$ (but not to those in $\mathscr{V}$), and letting that extra object be braided with respect to every other object of the category.

In other words, it is the monoidal category freely generated by the following polygraph, $\mathsf{Run}(\mathscr{V},\mathscr{G})$, (Figure 5), assuming $A_0,\dots,A_n$ and $B_0,\dots,B_m$ are distinct from R

- $\mathsf{Run}(\mathscr{V},\mathscr{G})_{obj} = \mathscr{G}_{obj} + \{R\} = \mathscr{V}_{obj} + \{R\}$,
- $\mathsf{Run}(\mathscr{V},\mathscr{G})(R, A_0,\dots,A_n; R, B_0,\dots,B_n) = \mathscr{G}(A_0,\dots,A_n; B_0,\dots,B_n)$,
- $\mathsf{Run}(\mathscr{V},\mathscr{G})(A_0,\dots,A_n; B_0,\dots,B_n) = \mathscr{V}(A_0,\dots,A_n; B_0,\dots,B_n)$,
- $\mathsf{Run}(\mathscr{V},\mathscr{G})(R, A_0; A_0, R) = \mathsf{Run}(\mathscr{V},\mathscr{G})(A_0, R; R, A_0) = \{\sigma\}$,

with $\mathsf{Run}(\mathscr{V},\mathscr{G})$ empty in any other case, and quotiented by the braiding axioms for R (Figure 6).

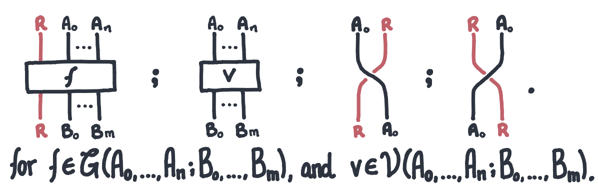

Figure 5: Generators for the runtime monoidal category.

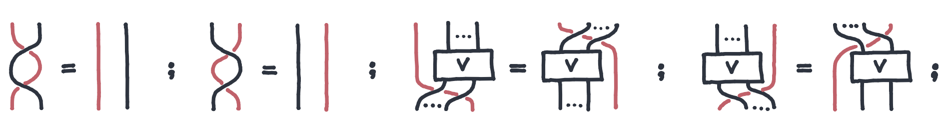

Figure 6: Axioms for the runtime monoidal category.

Somehow, we are asking the runtime R to be in the Drinfeld centre [9] of the monoidal category. The extra wire that R provides is only used to prevent interchange, and so it does not really matter where it is placed in the input and the output. We can choose to always place it on the left, for instance – and indeed we will be able to do so – but a better solution is to just consider objects "up to some runtime braidings". This is formalized by the notion of *braid clique*.

Definition 2.9 (Braid clique)**.** Given any list of objects $A_0,\dots,A_n$ in $\mathscr{V}_{obj} = \mathscr{G}_{obj}$, we construct a *clique* [43, 39] in the category $\mathsf{Mon}_{\mathsf{Run}}(\mathscr{V},\mathscr{G})$: we consider the objects, $A_0 \otimes \dots \otimes R_{(i)} \otimes \dots \otimes A_n$, created by inserting the runtime R in all of the possible $0 \leqslant i \leqslant n+1$ positions; and we consider the family of commuting isomorphisms constructed by braiding the runtime,

$$\sigma_{i,j} \colon A_0 \otimes \ldots \otimes R_{(i)} \otimes \ldots \otimes A_n \to A_0 \otimes \ldots \otimes R_{(j)} \otimes \ldots \otimes A_n.$$

We call this the *braid clique*, $\mathrm{Braid}_R(A_0, \ldots, A_n)$, on that list.

Definition 2.10. A *braid clique morphism*, $f \colon \mathrm{Braid}_R(A_0, \ldots, A_n) \to \mathrm{Braid}_R(B_0, \ldots, B_m)$ is a family of morphisms in the runtime monoidal category, $\mathrm{Mon}_{\mathrm{Run}}(\mathscr{V}, \mathscr{G})$, from each of the objects of first clique to each of the objects of the second clique,

$$f_{ik} \colon A_0 \otimes \ldots \otimes R_{(i)} \otimes \ldots \otimes A_n \to B_0 \otimes \ldots \otimes R_{(k)} \otimes \ldots \otimes B_m,$$

that moreover commutes with all braiding isomorphisms, $f_{ij} \fatsemi \sigma_{jk} = \sigma_{il} \fatsemi f$.

A braid clique morphism $f \colon \mathrm{Braid}_R(A_0, \ldots, A_n) \to \mathrm{Braid}_R(B_0, \ldots, B_m)$ is fully determined by *any* of its components, by pre/post-composing it with braidings. In particular, a braid clique morphism is always fully determined by its leftmost component $f_{00} \colon R \otimes A_0 \otimes \ldots \otimes A_n \to R \otimes B_0 \otimes \ldots \otimes B_m$.

Lemma 2.11. *Let $(\mathscr{V}, \mathscr{G})$ be a polygraph couple. There exists a premonoidal category, $\mathrm{Eff}(\mathscr{V}, \mathscr{G})$, that has objects the braid cliques, $\mathrm{Braid}_R(A_0, \ldots, A_n)$, in $\mathrm{Mon}_{\mathrm{Run}}(\mathscr{V}, \mathscr{G})$, and as morphisms the braid clique morphisms between them. See Appendix, Lemma A.1.*

Lemma 2.12. *Let $(\mathscr{V}, \mathscr{G})$ be a polygraph couple. There exists an identity-on-objects functor $\mathrm{Mon}(\mathscr{V}) \to \mathrm{Eff}(\mathscr{V}, \mathscr{G})$ that strictly preserves the premonoidal structure and whose image is central. See Appendix, Lemma A.2.*

Lemma 2.13. *Let $(\mathscr{V}, \mathscr{G})$ be a polygraph couple and consider the effectful category determined by $\mathrm{Mon}(\mathscr{V}) \to \mathrm{Eff}(\mathscr{V}, \mathscr{G})$. Let $\mathbb{V} \to \mathbb{C}$ be a strict effectful category endowed with a polygraph couple morphism $F \colon (\mathscr{V}, \mathscr{G}) \to \mathscr{U}(\mathbb{V}, \mathbb{C})$. There exists a unique strict effectful functor from $(\mathrm{Mon}(\mathscr{V}) \to \mathrm{Eff}(\mathscr{V}, \mathscr{G}))$ to $(\mathbb{V} \to \mathbb{C})$ commuting with F as a polygraph couple morphism. See Appendix, Lemma A.3.*

Theorem 2.14 (Runtime as a resource)**.** *The free strict effectful category over a polygraph couple $(\mathscr{V}, \mathscr{G})$ is $\mathrm{Mon}(\mathscr{V}) \to \mathrm{Eff}(\mathscr{V}, \mathscr{G})$. Its morphisms $A \to B$ are in bijection with the morphisms $R \otimes A \to R \otimes B$ of the runtime monoidal category,*

$$\mathrm{Eff}(\mathscr{V}, \mathscr{G})(A, B) \cong \mathrm{Mon}_{\mathrm{Run}}(\mathscr{V}, \mathscr{G})(R \otimes A, R \otimes B).$$

Proof. We must first show that $\mathrm{Mon}(\mathscr{V}) \to \mathrm{Eff}(\mathscr{V}, \mathscr{G})$ is an effectful category. The first step is to see that $\mathrm{Eff}(\mathscr{V}, \mathscr{G})$ forms a premonoidal category (Lemma 2.11). We also know that $\mathrm{Mon}(\mathscr{V})$ is a monoidal category: in fact, a strict, freely generated one. There exists an identity on objects functor, $\mathrm{Mon}(\mathscr{V}) \to \mathrm{Eff}(\mathscr{V}, \mathscr{G})$, that strictly preserves the premonoidal structure and centrality (Lemma 2.12).

Let us now show that it is the free one over the polygraph couple $(\mathscr{V}, \mathscr{G})$. Let $\mathbb{V} \to \mathbb{C}$ be an effectful category, with an polygraph couple map $F \colon (\mathscr{V}, \mathscr{G}) \to \mathscr{U}(\mathbb{V}, \mathbb{C})$. We can construct a unique effectful functor from $(\mathrm{Mon}(\mathscr{V}) \to \mathrm{Eff}(\mathscr{V}, \mathscr{G}))$ to $(\mathbb{V} \to \mathbb{C})$ giving its universal property (Lemma 2.13). □

Corollary 2.15 (String diagrams for effectful categories)**.** *We can use string diagrams for effectful categories, quotiented under the same isotopy as for monoidal categories, provided that we do represent the runtime as an extra wire that needs to be the input and output of every effectful morphism.*

3 Profunctors and Promonads

We have elaborated on string diagrams for effectful categories. Let us now show that effectful categories are fundamental objects. The profunctorial counterpart of a monad is a promonad. Promonads have been widely used for functional programming semantics, although usually with an extra assumption of strength and under the name of "arrows" [15, 17, 18]. Promonads over a category endow it with some new, "effectful", morphisms; while the base morphisms of the category are called the "pure" morphisms. This terminology will coincide when regarding effectful categories as promonads.

In this section, we introduce profunctors and promonads. In the following sections, we show that effectful categories are to promonads what monoidal categories are to categories: they are the pseudomonoids of a suitably constructed monoidal bicategory of promonads. In order to obtain this result, we introduce the pure tensor of promonads in Section 4. The pure tensor of promonads combines the effects of two promonads over different categories into a single one. In some sense, it does so in the universal way that turns "purity" into "centrality" (Theorem 4.2).

3.1 Profunctors: an algebra of processes

Profunctors $P\colon \mathbb{A}^{op} \times \mathbb{B} \to \mathsf{Set}$ [3, 7, 4] can be thought as indexing *families of processes* $P(A,B)$ by the types of an input channel A and an output channel B [23].

The category $\mathbb{A}$ has as morphisms the pure transformations $f\colon A' \to A$ that we can apply to the input of a process $p \in P(A,B)$ to obtain a new process, which we call $(f > p) \in P(A',B)$. Analogously, the category $\mathbb{B}$ has as morphisms the pure transformations $g\colon B \to B'$ that we can apply to the output of a process $p \in P(A,B)$ to obtain a new process, which we call $(p < g) \in P(A,B')$. The profunctor axioms encode the compositionality of these transformations.

Definition 3.1. A *profunctor* $(P,>,<)$ between two categories $\mathbb{A}$ and $\mathbb{B}$ is a family of sets $P(A,B)$ indexed by objects of $\mathbb{A}$ and $\mathbb{B}$, and endowed with jointly functorial left and right actions of the morphisms of $\mathbb{A}$ and $\mathbb{B}$, respectively. Explicitly, types of these actions are $(>)\colon \hom(A',A) \times P(A',B) \to P(A,B)$, and $(<)\colon \hom(B,B') \times P(A,B) \to P(A,B')$. They must satisfy

- compatibility, $(f > p) < g = f > (p < g)$,
- preserve identities, $\mathrm{id} > p = p$, and $p < \mathrm{id} = p$,
- and composition, $(p < f) < g = p < (f \mathbin{⨾} g)$ and $f > (g > p) = (f \mathbin{⨾} g) > p$.

More succinctly, a *profunctor* $P\colon \mathbb{A} \nrightarrow \mathbb{B}$ is a functor $P\colon \mathbb{A}^{op} \times \mathbb{B} \to \mathsf{Set}$. When presented as a family of sets with a pair of actions, profunctors are sometimes called *bimodules*.

A profunctor homomorphism $\alpha\colon P \to Q$ transforms processes of type $P(A,B)$ into processes of type $Q(A,B)$. The homomorphism affects only the effectful processes, and not the pure transformations we could apply in $\mathbb{A}$ and $\mathbb{B}$. This means that $\alpha(f > p) = f > \alpha(p)$ and that $\alpha(p < g) = \alpha(p) < g$.

Definition 3.2 (Profunctor homomorphism)**.** A *profunctor homomorphism* from the profunctor $P\colon \mathbb{A} \nrightarrow \mathbb{B}$ to the profunctor $Q\colon \mathbb{A} \nrightarrow \mathbb{B}$ is a family of functions $\alpha_{A,B}\colon P(A,B) \to Q(A,B)$ preserving the left and right actions, $\alpha(f > p < g) = f > \alpha(p) < g$. Equivalently, it is a natural transformation $\alpha\colon P \to Q$ between the two functors $\mathbb{A}^{op} \times \mathbb{B} \to \mathsf{Set}$.

How to compose two families of processes? Assume we have a process $p \in P(A,B_1)$ and a process $q \in Q(B_2,C)$. Moreover, assume we have a transformation $f\colon B_1 \to B_2$ translating from the output of the second to the input of the first. In this situation, we can plug together the processes: $p \in P(A,B_1)$ writes to an output of type B_1, which is translated by f to an input of type B_2, then used by $q \in Q(B_2,C)$.

There are two slightly different ways of describing this process, depending on whether we consider the translation to be part of the first or the second process. We could translate just after finishing the first process, $(p < f, q)$; or translate just before starting the second process, $(p, f > q)$.

These are two different pairs of processes, with different types. However, if we take the process interpretation seriously, it does not really matter when to apply the translation. These two descriptions represent the same process. They are *dinaturally equivalent* [23, 25].

Definition 3.3 (Dinatural equivalence)**.** Let $P \colon \mathbb{A} \nrightarrow \mathbb{B}$ and $Q \colon \mathbb{B} \nrightarrow \mathbb{C}$ be two profunctors. Consider the set of matching pairs of processes, with a given input A and output C,

$$R_{P,Q}(A,C) = \sum_{B \in \mathbb{B}} P(A,B) \times Q(B,C).$$

Dinatural equivalence $(\sim)$, on the set $R_{P,Q}(A,C)$ is the smallest equivalence relation satisfying $(p < g, q) \sim (p, g > q)$. The set of matching processes $R_{P,Q}(A,C)$ quotiented by dinaturality $(\sim)$ is written as $(P \diamond Q)(A,C)$. It is a particular form of colimit over the category $\mathbb{B}$, called a *coend*, usually denoted by an integral sign.

$$(P \diamond Q)(A,C) = R_{P,Q}(A,C)/(\sim) = \int^{B \in \mathbb{B}} P(A,B) \times Q(B,C).$$

Definition 3.4 (Profunctor composition)**.** The composition of two profunctors $P \colon \mathbb{A} \nrightarrow \mathbb{B}$ and $Q \colon \mathbb{B} \nrightarrow \mathbb{C}$ is the profunctor $(P \diamond Q) \colon \mathbb{A} \nrightarrow \mathbb{C}$ has as processes the matching pairs of processes in P and Q quotiented by dinaturality on $\mathbb{B}$,

$$(p, g < q) \sim (p > g, q).$$

Its actions are the left and right actions of p and q, respectively, $f > (p,q) < g = (f > p, q < g)$.

The identity profunctor $\mathbb{A} \colon \mathbb{A} \nrightarrow \mathbb{A}$ has as processes the morphisms of the category $\mathbb{A}$, it is given by the hom-sets. Its actions are pre and post-composition, $f > h < g = f \mathbin{⨾} h \mathbin{⨾} g$.

Profunctors are better understood as providing a double categorical structure to the category of categories. A double category $\mathbb{D}$ contains 0-cells (or "objects"), two different types of 1-cells (the "arrows" and the "proarrows"), and cells [37]. Arrows compose in an strictly associative and unital way, while proarrows come equipped with natural isomorphisms representing associativity and unitality. We employ the graphical calculus of double categories [29], with arrows going left to right and proarrows going top to bottom.

Definition 3.5. The double category of categories, **CAT**, has as objects the small categories $\mathbb{A}, \mathbb{B}, \dots$, as arrows the functors between them, $F \colon \mathbb{A} \to \mathbb{A}'$, as proarrows the profunctors between them, $P \colon \mathbb{A} \nrightarrow \mathbb{B}$, and as cells, the natural transformations, $\alpha_{A,B} \colon P(A,B) \to Q(FA, GB)$.

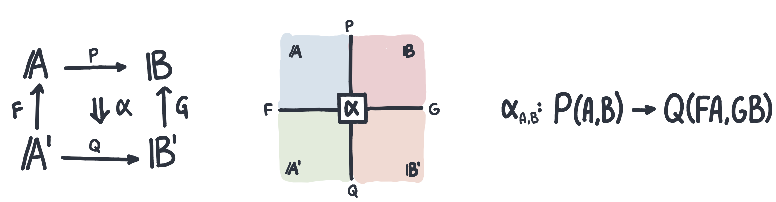

Figure 7: Cell in the double category of categories.

Every functor has a companion and a conjoint profunctors: their representable and corepresentable profunctors [12]. This structure makes **CAT** into the paradigmatic example of a proarrow equipment (or *framed bicategory* [37]).

3.2 Promonads: new morphisms for an old category

Promonads are to profunctors what monads are to functors.[2] It may be then surprising to see that so little attention has been devoted to them, relative to their functorial counterparts. The main source of examples and focus of attention has been the semantics of programming languages [17, 30, 18]. Strong monads are commonly used to give categorical semantics of effectful programs [28], and the so-called *arrows* (or *strong promonads*) strictly generalize them.

Part of the reason behind the relative unimportance given to promonads elsewhere may stem from the fact that promonads over a category can be shown in an elementary way to be equivalent to identity-on-objects functors from that category [25]. The explicit proof is, however, difficult to find in the literature, and so we include it here (Theorem 3.9).

Under this interpretation, promonads are new morphisms for an old category. We can reinterpret the old morphisms into the new ones in a functorial way. The paradigmatic example is again that of Kleisli or cokleisli categories of strong monads and comonads. This structure is richer than it may sound, and we will explore it further during the rest of this text.

Definition 3.6 (Monoids and promonoids)**.** A *monoid* in a double category is an arrow $T\colon \mathbb{A} \to \mathbb{A}$ together with cells $m \in \mathrm{hom}(M \otimes M; 1, 1; M)$ and $e \in \mathrm{cell}(1; 1, 1; M)$, called multiplication and unit, satisfying unitality and associativity. A *promonoid* in a double category is a proarrow $M\colon \mathbb{A} \nrightarrow \mathbb{A}$ together with cells $m \in \mathrm{cell}(1; M \otimes M, M, 1)$ and $e \in \mathrm{cell}(1; 1, M; 1)$, called promultiplication and prounit, satisfying unitality and associativity.

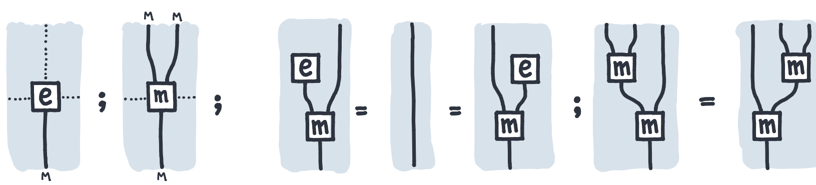

Figure 8: Data and axioms of a promonoid in a double category.

Dually, we can define *comonoids* and *procomonoids*.

A monad is a monoid in the category of categories, functors and profunctors **Cat**. In the same way, a promonad is a promonoid in **Cat**.

Definition 3.7. A *promonad* $(P, \star, ^\circ)$ over a category $\mathbb{C}$ is a profunctor $P\colon \mathbb{C} \nrightarrow \mathbb{C}$ together with natural transformations representing inclusion $(^\circ)_{X,Y}\colon \mathbb{C}(X,Y) \to P(X,Y)$ and multiplication $(\star)_{X,Y}\colon P(X,Y) \times P(Y,Z) \to P(X,Z)$, and such that

i. the right action is premultiplication, $f^\circ \star p = f > p$;
ii. the left action is posmultiplication, $p \star f^\circ = p < f$;
iii. multiplication is dinatural, $p \star (f > q) = (p < f) \star q$;
iv. and multiplication is associative, $(p_1 \star p_2) \star p_3 = p_1 \star (p_2 \star p_3)$.

Equivalently, promonads are promonoids in the double category of categories, where the dinatural multiplication represents a transformation from the composition of the profunctor P with itself.

Lemma 3.8 (Kleisli category of a promonad)**.** *Every promonad* $(P, \star, ^\circ)$ *induces a category with the same objects as its base category, but with hom-sets given by* $P(\bullet, \bullet)$*, composition given by* $(\star)$ *and identities given by* (id°)*. This is called its* Kleisli category, $\mathrm{kleisli}(P)$*. Moreover, there exists an identity-on-objects functor* $\mathbb{C} \to \mathrm{kleisli}(P)$*, defined on morphisms by the unit of the promonad. See Appendix, Lemma B.1.*

[2] To quip, a promonad is just a monoid on the category of endoprofunctors.

The converse is also true: every category $\mathbb{C}$ with an identity-on-objects functor from some base category $\mathbb{V}$ arises as the Kleisli category of a promonad.

Theorem 3.9. *Promonads over a category $\mathbb{C}$ correspond to identity-on-objects functors from the category $\mathbb{C}$. Given any identity-on-objects functor $i \colon \mathbb{C} \to \mathbb{D}$ there exists a unique promonad over $\mathbb{C}$ having $\mathbb{D}$ as its Kleisli category: the promonad given by the profunctor $\hom_{\mathbb{D}}(i(\bullet), i(\bullet))$. See Appendix, Theorem B.2.*

3.3 Homomorphisms and transformations of promonads

We have characterized promonads as identity-on-objects functors. We now characterize the homomorphisms and transformations of promonads as suitable pairs of functors and natural transformations.

Definition 3.10 (Promonoid homomorphism)**.** Let $(\mathbb{A}, M, m, e)$ and $(\mathbb{B}, N, n, u)$ be promonoids in a double category. A promonoid homomorphism is an arrow $T \colon \mathbb{A} \to \mathbb{B}$ together with a cell $t \in \mathrm{cell}(F; M, N; F)$ that preserves the promonoid promultiplication and prounit.

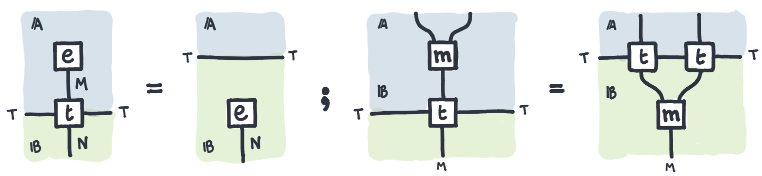

Figure 9: Axioms for a promonoid homomorphism.

Definition 3.11 (Promonad homomorphism)**.** Let $(\mathbb{A}, P, \star, ^{\circ})$ and $(\mathbb{B}, Q, \star, ^{\circ})$ be two promonads, possibly over two different categories. A promonad homomorphism (F_0, F) is a functor between the underlying categories $F_0 \colon \mathbb{A} \to \mathbb{B}$ and a natural transformation $F_{X,Y} \colon P(X,Y) \to Q(FX, FY)$ preserving composition and inclusions. That is, $F(p_1 \star p_2) = F(p_1) \star F(p_2)$, and $F(f^{\circ}) = F_0(f)^{\circ}$.

Proposition 3.12. *A promonad homomorphism between two promonads understood as identity-on-objects functors, $\mathbb{V} \to \mathbb{C}$ and $\mathbb{W} \to \mathbb{D}$, is equivalently a pair of functors (F_0, F) that commute strictly with the two identity-on-objects functors on objects $F_0(X) = F(X)$ and morphisms $F_0(f)^{\circ} = F(f^{\circ})$. See Appendix, Proposition B.3.*

Definition 3.13 (Promonoid modification)**.** Let $(\mathbb{A}, M, m, e)$ and $(\mathbb{B}, N, n, u)$ be promonoids in a double category, and let $t \in \mathrm{cell}(F; M, N; F)$ and $r \in \mathrm{cell}(G; M, N; G)$ be promonoid homomorphisms. A *promonoid modification* is a cell $\alpha \in \mathrm{cell}(F; 1, 1; G)$ such that its precomposition with t is its postcomposition r.

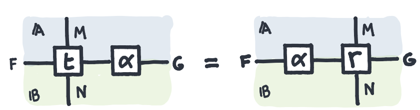

Figure 10: Axiom for a promonoid transformation.

Definition 3.14. A *promonad modification* between two promonad homomorphisms (F_0, F) and (G_0, G) between the same promonads $(\mathbb{A}, P, \star, ^{\circ})$ and $(\mathbb{B}, Q, \star, ^{\circ})$ is a natural transformation $\alpha_X \colon F_0(X) \to G_0(X)$ such that $\alpha_X > G(p) = F(p) < \alpha_Y$ for each $p \in P(X,Y)$.

Proposition 3.15. *A promonad modification between two promonad homomorphisms understood as commutative squares of identity-on-objects functors $F_0(f)^{\circ} = F(f^{\circ})$ and $G_0(f)^{\circ} = G(f^{\circ})$ is a natural* *transformation* $\alpha\colon F_0 \Rightarrow G_0$ *that can be lifted via the identity-on-objects functor to a natural transformation* $\alpha^\circ\colon F \Rightarrow G$*. In other words, a pure natural transformation.*

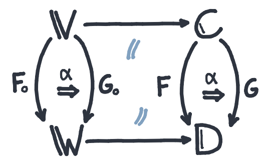

Figure 11: Promonad modifications are cylinder transformations.

Summarizing this section, we have shown a correspondence between promonads, their homomorphisms and modifications, and identity-on-objects functors, squares and cylinder transformations of squares. The double category structure allows us to talk about homomorphisms and modifications, which would be more difficult to address in a bicategory structure.

Promonad	Identity-on-objects functor	Theorem 3.9
Promonad homomorphism	Commuting square	Proposition 3.12
Promonad modification	Cylinder transformation	Proposition 3.15

4 Pure Tensor of Promonads

This section introduces the *pure tensor of promonads*. The pure tensor of promonads combines the effects of two promonads, possibly over different categories, into the effects of a single promonad over the product category. Effects do not generally interchange. However, this does not mean that no morphisms should interchange in the pure tensor of promonads: in our interpretation of a promonad $\mathbb{V} \to \mathbb{C}$, the morphisms coming from the inclusion are *pure*, they produce no effects; pure morphisms with no effects should always interchange with effectful morphisms, even if effectful morphisms do not interchange among themselves.

A practical way to encode and to remember all of the these restrictions is to use monoidal string diagrams. This is another application of the idea of runtime: we introduce an extra wire so that all the rules of interchange become ordinary interchange laws in a monoidal category. That is, we insist again that effectful morphisms are just pure morphisms using a shared resource – the runtime. When we compute the pure tensor of two promonads, the runtime needs to be shared between the impure morphisms of both promonads.

4.1 Pure tensor, via runtime

Definition 4.1 (Pure tensor)**.** Let $\mathbb{C}\colon \mathbb{V} \nrightarrow \mathbb{V}$ and $\mathbb{D}\colon \mathbb{W} \nrightarrow \mathbb{W}$ be two promonads. Their *pure tensor*, $\mathbb{C} * \mathbb{D}\colon \mathbb{V} \times \mathbb{W} \to \mathbb{V} \times \mathbb{W}$, is a promonad over $\mathbb{V} \times \mathbb{W}$ where elements of $\mathbb{C} * \mathbb{D}(X,Y;X',Y')$, the morphisms $X \otimes R \otimes Y \to X' \otimes R \otimes Y'$ in the freely presented monoidal category generated by the elements of Figure 12 and quotiented by the axioms of Figure 13.

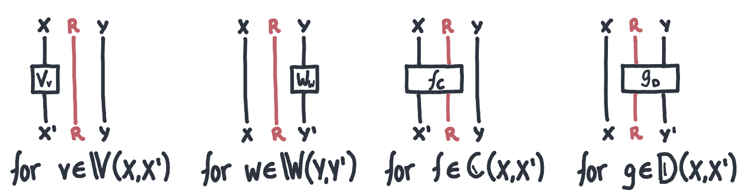

Figure 12: Generators for the elements of the pure tensor of promonads.

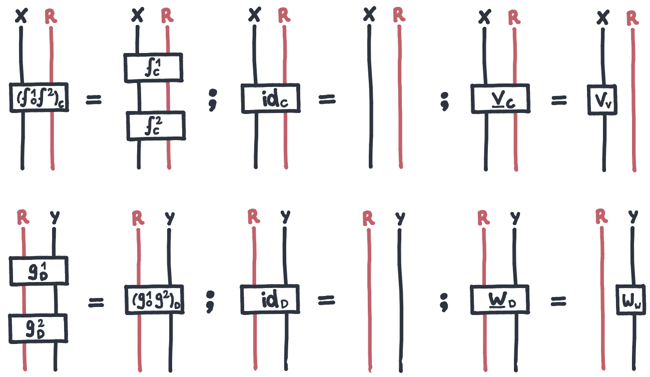

Figure 13: Axioms for the elements of the pure tensor of promonads.

Multiplication is defined by composition in the monoidal category, and the unit is defined by the inclusion of pairs, as depicted in Figure 14.

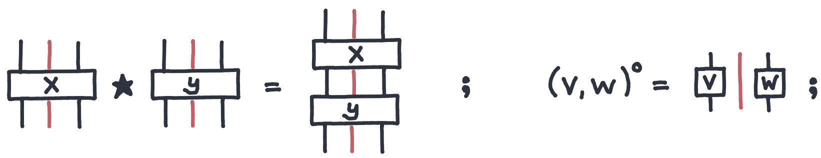

Figure 14: The pure tensor promonad.

In other words, the elements of the pure tensor are the morphisms the category presented by the graph that has as objects the pairs of objects (X,Y) with $X \in \mathbb{V}_{\text{obj}}$ and $Y \in \mathbb{W}_{\text{obj}}$, formally written as $X \otimes R \otimes Y$; and the morphisms generated by

- an edge $f_{\mathbb{C}} \colon X \otimes R \otimes Y \to X' \otimes R \otimes Y$ for each arrow $f \in \mathbb{C}(X,X')$ and each object $Y \in \mathbb{W}$;
- an edge $g_{\mathbb{D}} \colon X \otimes R \otimes Y \to X \otimes R \otimes Y'$ for each arrow $g \in \mathbb{D}(Y,Y')$ and each object $X \in \mathbb{V}$;
- an edge $v_{\mathbb{V}} \colon X \otimes R \otimes Y \to X' \otimes R \otimes Y$ for each arrow $v \in \mathbb{V}(X,X')$ and each object $Y \in \mathbb{W}$;
- and an edge $w_{\mathbb{W}} \colon X \otimes R \otimes Y \to X \otimes R \otimes Y'$ for each arrow $w \in \mathbb{W}(Y,Y')$ and each object $X \in \mathbb{V}$;

quotiented by centrality of pure morphisms: $f_{\mathbb{C}} \mathbin{⨟} w_{\mathbb{W}} = w_{\mathbb{W}} \mathbin{⨟} f_{\mathbb{C}}$ and $g_{\mathbb{D}} \mathbin{⨟} v_{\mathbb{V}} = v_{\mathbb{V}} \mathbin{⨟} g_{\mathbb{D}}$; by compositions and identities of one promonad: $f_{\mathbb{C}} \mathbin{⨟} f'_{\mathbb{C}} = (f \star f')_{\mathbb{C}}$ and $\text{id}_{\mathbb{C}} = \text{id}$; by compositions and identities of the other promonad: $g_{\mathbb{D}} \mathbin{⨟} g'_{\mathbb{D}} = (g \star g')_{\mathbb{D}}$ and $\text{id}_{\mathbb{D}} = \text{id}$; and by the coincidence of pure morphisms and their effectful representatives: $v_{\mathbb{V}} = v^{\circ}_{\mathbb{C}}$ and $w_{\mathbb{W}} = w^{\circ}_{\mathbb{D}}$.

Crucially in this definition, $f_{\mathbb{C}}$ and $g_{\mathbb{D}}$ do not interchange: they are sharing the runtime, and that prevents the application of the interchange law. The pure tensor of promonads, $\mathbb{C} * \mathbb{D}$, takes its name from the fact that, if we interpret the promonads $\mathbb{V} \to \mathbb{C}$ and $\mathbb{W} \to \mathbb{D}$ as declaring the morphisms in $\mathbb{V}$ and $\mathbb{W}$ as pure, then the pure morphisms of the composition interchange with all effectful morphisms. The spirit is similar to the *free product of groups with commuting subgroups* [27].

4.2 Universal property of the pure tensor

There are multiple canonical ways in which one could combine the effects of two promonads, $\mathbb{C} \colon \mathbb{V} \nrightarrow \mathbb{V}$ and $\mathbb{D} \colon \mathbb{W} \nrightarrow \mathbb{W}$, into a single promonad, such as taking the product of both, $\mathbb{C} \times \mathbb{D} \colon \mathbb{V} \times \mathbb{W} \nrightarrow \mathbb{V} \times \mathbb{W}$. Let us show that the pure tensor has a universal property: it is the universal one in which we can include impure morphisms from each promonads, interchanging with pure morphisms from the other promonad, so that purity is preserved.

Theorem 4.2. *Let* $\mathbb{C}\colon \mathbb{V} \nrightarrow \mathbb{V}$ *and* $\mathbb{D}\colon \mathbb{W} \nrightarrow \mathbb{W}$ *be two promonads and let* $\mathbb{C} * \mathbb{D}\colon \mathbb{V}\times\mathbb{W} \to \mathbb{V}\times\mathbb{W}$ *be their pure tensor. There exist a pair of promonad homomorphisms* $L\colon \mathbb{C}\times\mathbb{W} \to \mathbb{C}*\mathbb{D}$ *and* $R\colon \mathbb{V}\times\mathbb{D} \to \mathbb{C}*\mathbb{D}$*. These are universal in the sense that, for every pair of promonad homomorphisms,* $A\colon \mathbb{C}\times\mathbb{W}\to\mathbb{E}$ *and* $B\colon \mathbb{V}\times\mathbb{D}\to\mathbb{E}$*, there exists a unique promonad homomorphism* $(A\vee B)\colon \mathbb{C}*\mathbb{D}\to\mathbb{E}$ *that commutes strictly with them,* $(A\vee B) \mathbin{⨾} L = A$ *and* $(A\vee B)\mathbin{⨾} R = B$*. See Appendix, Theorem B.4.*

5 Effectful Categories are Pseudomonoids

We will now use the pure tensor of promonads to justify effectful categories as the promonadic counterpart of monoidal categories: effectful categories are pseudomonoids in the monoidal bicategory of promonads with the pure tensor. Pseudomonoids [42, 45] are the categorification of monoids. They are still formed by a 0-cell representing the carrier of the monoid and a pair of 1-cells representing multiplication and units. However, we weaken the requirement for associativity and unitality to the existence of invertible 2-cells, called the *associator* and *unitor*.

In the same way that monoids live in monoidal categories, pseudomonoids live in monoidal bicategories. A monoidal bicategory $\mathbb{A}$ is a bicategory in which we can tensor objects with a pseudofunctor $(\boxtimes)\colon \mathbb{A}\times\mathbb{A}\to\mathbb{A}$ and we have a tensor unit $I\colon 1\to\mathbb{A}$, these are associative and unital up to equivalence, and satisfy certain coherence equations up to invertible modification [36].

5.1 Pseudomonoids

Definition 5.1. In a monoidal bicategory, a *pseudomonoid* over a 0-cell M is a pair of 1-cells, $M\boxtimes M\to M$ and $I\to M$, together with the following triple of invertible 2-cells representing associativity and unitality (Figure 15), and satisfying the pentagon and triangle equations (see Appendix, Figure 24). A homomorphism of pseudomonoids is given by a 1-cell between their underlying 0-cells and the following invertible 2-cells, representing preservation of the multiplication and the unit (Figure 15), and satisfying compatibility with associativity and unitality (see Appendix, Figure 28).

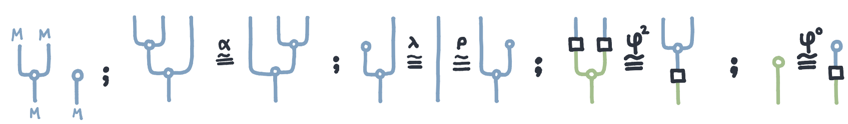

Figure 15: Data for a pseudomonoid and pseudomonoid homomorphism.

A pseudomonoid is *strict* when the associators and unitors are identity cells. Note that, in strict 2-categories (sometimes called 2-categories, in contrast to bicategories), this is the same as a monoid in the monoidal category that we obtain by ignoring the 2-cells.

Remark 5.2. A pseudomonoid in the monoidal bicategory of categories with the cartesian product of categories, $(\mathbf{Cat},\times)$ is a monoidal category. A strict pseudomonoid in the same monoidal bicategory is a strict monoidal category.

A strict pseudomonoid in the monoidal bicategory of categories with the funny tensor product of categories $(\mathbf{Cat},\square)$ is a strict premonoidal category. However, it is not immediately clear how to recover premonoidal categories as pseudomonoids. A naive attempt will fail: $(\mathbf{Cat},\square)$ is usually made into a monoidal bicategory with non-necessarily-natural transformations, but we do want our coherence morphisms to be natural, so we must ask at least naturality. This will not be enough: taking natural transformations as 2-cells will give us premonoidal categories where the associators and unitors do not need to be *central*. Centrality is what requires a more careful approach.

5.2 Effectful categories are promonad pseudomonoids

Promonads form a monoidal category with the pure tensor product and moreover a strict monoidal bicategory with promonad modifications. Effectful categories are the pseudomonoids in this category.

Theorem 5.3. *An effectful category (or monoidal Freyd category) is a pseudomonoid on the monoidal 2-category of promonads with promonad homomorphism, promonad transformations and the pure tensor of promonads. A pseudomonoid homomorphism between effectful categories is an effectful functor.*

As a consequence, preomonoidal categories with their centre *are pseudomonoids. See Appendix, Theorem D.1.*

6 Conclusions

Premonoidal categories are monoidal categories with runtime, and we can still monoidal string diagrams and unrestricted topological deformations to reason about premonoidal categories. Instead of dealing directly with premonoidal categories, we employ the better behaved notion of non-cartesian Freyd categories, effectful categories. There exists a more fine-grained notion of “Cartesian effect category” [10], which generalizes Freyd categories and justifies calling “effectful category” to the general case.

Promonads have been arguably under-appreciated, possibly because of their characterization as “just” identity-on-objects functors. However, speaking of promonads as the proarrow counterpart of monads makes many aspects of the theory of monads clearer: every monad and every comonad induce a promonad (their Kleisli category) via the proarrow equipment, monad morphisms lift to promonad morphisms, distributive laws of monads induce a way of composing morphisms from different kleisli categories [8]. Justifying effectful categories in terms of promonads highlights their importance as the monadic counterpart of monoidal categories.

Ultimately, this is a first step towards our more ambitious project of presenting the categorical structure of programming languages in a purely diagrammatic way, revisiting Alan Jeffrey’s work [20, 19, 35]. The internal language of premonoidal categories and effectful categories is given by the *arrow do-notation* [30]; at the same time, we have shown that it is given by suitable string diagrams. This correspondence allows us to translate between programs and string diagrams (Figure 16).

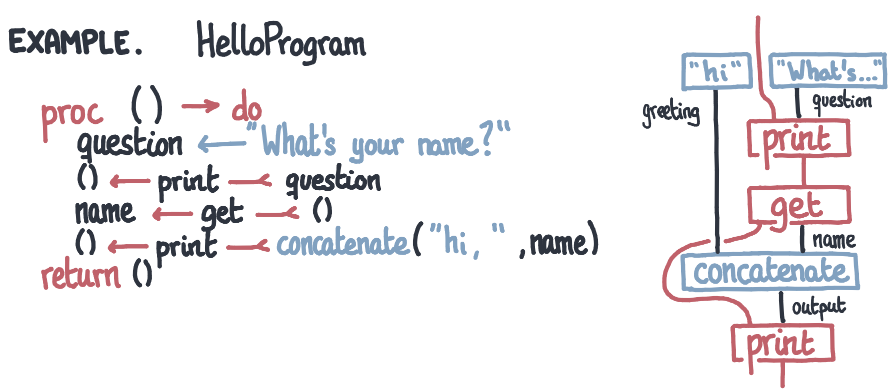

Figure 16: Premonoidal program in arrow do-notation and string diagrams.

7 Acknowledgements

The author thanks conversations with Edward Morehouse, Matt Earnshaw, Niels Voorneveld, Elena Di Lavore. The author thanks the anonymous reviewers at ACT'22 for their careful reading and their many suggestions. Mario Román was supported by the European Union through the ESF funded Estonian IT Academy research measure (2014-2020.4.05.19-0001). This work was also supported by the Estonian Research Council grant PRG1210.

References

[1] Samson Abramsky & Bob Coecke (2009): *Categorical quantum mechanics*. Handbook of quantum logic and quantum structures 2, pp. 261–325. arXiv:0808.1023.

[2] Samson Abramsky, Esfandiar Haghverdi & Philip J. Scott (2002): *Geometry of Interaction and Linear Combinatory Algebras*. Math. Struct. Comput. Sci. 12(5), pp. 625–665, doi:10.1017/S0960129502003730. Available at `https://doi.org/10.1017/S0960129502003730`.

[3] Jean Bénabou (1967): *Introduction to bicategories*. In: Reports of the midwest category seminar, Springer, pp. 1–77.

[4] Jean Bénabou (2000): *Distributors at work*. Lecture notes written by Thomas Streicher 11.

[5] Richard F Blute, J Robin B Cockett, Robert AG Seely & Todd H Trimble (1996): *Natural deduction and coherence for weakly distributive categories*. Journal of Pure and Applied Algebra 113(3), pp. 229–296.

[6] Filippo Bonchi, Jens Seeber & Pawel Sobocinski (2018): *Graphical Conjunctive Queries*. In Dan R. Ghica & Achim Jung, editors: 27th EACSL Annual Conference on Computer Science Logic, CSL 2018, September 4-7, 2018, Birmingham, UK, LIPIcs 119, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 13:1–13:23, doi:10.4230/LIPIcs.CSL.2018.13. Available at `https://doi.org/10.4230/LIPIcs.CSL.2018.13`.

[7] Francis Borceux (1994): *Handbook of categorical algebra: volume 1, Basic category theory*. 1, Cambridge University Press.

[8] Eugenia Cheng (2021): *Distributive Laws for Lawvere Theories (Invited Talk)*. In Fabio Gadducci & Alexandra Silva, editors: 9th Conference on Algebra and Coalgebra in Computer Science, CALCO 2021, August 31 to September 3, 2021, Salzburg, Austria, LIPIcs 211, Schloss Dagstuhl - Leibniz-Zentrum für Informatik, pp. 1:1–1:1, doi:10.4230/LIPIcs.CALCO.2021.1. Available at `https://doi.org/10.4230/LIPIcs.CALCO.2021.1`.

[9] Vladimir Drinfeld, Shlomo Gelaki, Dmitri Nikshych & Victor Ostrik (2010): *On braided fusion categories I*. Selecta Mathematica 16(1), pp. 1–119.

[10] Jean-Guillaume Dumas, Dominique Duval & Jean-Claude Reynaud (2011): *Cartesian effect categories are Freyd-categories*. Journal of Symbolic Computation 46(3), pp. 272–293.

[11] Jean-Yves Girard (1989): *Geometry of interaction 1: Interpretation of System F*. In: Studies in Logic and the Foundations of Mathematics, 127, Elsevier, pp. 221–260.

[12] Marco Grandis & Robert Paré (1999): *Limits in double categories*. Cahiers de topologie et géométrie différentielle catégoriques 40(3), pp. 162–220.

[13] René Guitart (1980): *Tenseurs et machines*. Cahiers de topologie et géométrie différentielle catégoriques 21(1), pp. 5–62.

[14] Jules Hedges (2019): *Folklore: Monoidal kleisli categories*. Available at `https://julesh.com/2019/04/18/folklore-monoidal-kleisli-categories/`.

[15] Chris Heunen & Bart Jacobs (2006): *Arrows, like Monads, are Monoids*. In Stephen D. Brookes & Michael W. Mislove, editors: Proceedings of the 22nd Annual Conference on Mathematical Foundations of Programming Semantics, MFPS 2006, Genova, Italy, May 23-27, 2006, Electronic Notes in Theoretical Computer Science 158, Elsevier, pp. 219–236, doi:10.1016/j.entcs.2006.04.012. Available at `https://doi.org/10.1016/j.entcs.2006.04.012`.

[16] Naohiko Hoshino, Koko Muroya & Ichiro Hasuo (2014): *Memoryful geometry of interaction: from coalgebraic components to algebraic effects*. In Thomas A. Henzinger & Dale Miller, editors: Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014, ACM, pp. 52:1–52:10, doi:10.1145/2603088.2603124. Available at `https://doi.org/10.1145/2603088.2603124`.

[17] John Hughes (2000): *Generalising monads to arrows*. Science of Computer Programming 37(1-3), pp. 67–111, doi:10.1016/S0167-6423(99)00023-4. Available at `https://doi.org/10.1016/S0167-6423(99)00023-4`.

[18] Bart Jacobs, Chris Heunen & Ichiro Hasuo (2009): *Categorical semantics for arrows*. J. Funct. Program. 19(3-4), pp. 403–438, doi:10.1017/S0956796809007308. Available at `https://doi.org/10.1017/S0956796809007308`.

[19] Alan Jeffrey (1997): *Premonoidal categories and a graphical view of programs*. Preprint at ResearchGate. Available at `https://www.researchgate.net/profile/Alan-Jeffrey/publication/228639836_Premonoidal_categories_and_a_graphical_view_of_programs/links/00b495182cd648a874000000/Premonoidal-categories-and-a-graphical-view-of-programs.pdf`.

[20] Alan Jeffrey (1997): *Premonoidal categories and flow graphs*. Electron. Notes Theor. Comput. Sci. 10, p. 51, doi:10.1016/S1571-0661(05)80688-7. Available at `https://doi.org/10.1016/S1571-0661(05)80688-7`.

[21] André Joyal & Ross Street (1991): *The geometry of tensor calculus, I*. Advances in mathematics 88(1), pp. 55–112.

[22] André Joyal & Ross Street (1991): *The geometry of tensor calculus, I*. Advances in mathematics 88(1), pp. 55–112.

[23] Elena Di Lavore, Giovanni de Felice & Mario Román (2022): *Monoidal Streams for Dataflow Programming*. CoRR abs/2202.02061. arXiv:2202.02061.

[24] Paul Blain Levy (2004): *Call-By-Push-Value: A Functional/Imperative Synthesis*. Semantics Structures in Computation 2, Springer.

[25] Fosco Loregian (2021): *(Co)end Calculus*. London Mathematical Society Lecture Note Series, Cambridge University Press, doi:10.1017/9781108778657.

[26] Saunders Mac Lane (1978): *Categories for the Working Mathematician*. Graduate Texts in Mathematics, Springer New York, doi:10.1007/978-1-4757-4721-8.

[27] Wilhelm Magnus, Abraham Karrass & Donald Solitar (2004): *Combinatorial group theory: Presentations of groups in terms of generators and relations*. Courier Corporation.

[28] Eugenio Moggi (1991): *Notions of Computation and Monads*. Inf. Comput. 93(1), pp. 55–92, doi:10.1016/0890-5401(91)90052-4. Available at `https://doi.org/10.1016/0890-5401(91)90052-4`.

[29] David Jaz Myers (2016): *String Diagrams For Double Categories and Equipments*. arXiv: Category Theory.

[30] Ross Paterson (2001): *A New Notation for Arrows*. In Benjamin C. Pierce, editor: Proceedings of the Sixth ACM SIGPLAN International Conference on Functional Programming (ICFP '01), Firenze (Florence), Italy, September 3-5, 2001, ACM, pp. 229–240, doi:10.1145/507635.507664. Available at `https://doi.org/10.1145/507635.507664`.

[31] Dusko Pavlovic (2013): *Monoidal computer I: Basic computability by string diagrams*. Inf. Comput. 226, pp. 94–116, doi:10.1016/j.ic.2013.03.007. Available at `https://doi.org/10.1016/j.ic.2013.03.007`.

[32] John Power (2002): *Premonoidal categories as categories with algebraic structure*. Theor. Comput. Sci. 278(1-2), pp. 303–321, doi:10.1016/S0304-3975(00)00340-6. Available at `https://doi.org/10.1016/S0304-3975(00)00340-6`.

[33] John Power & Edmund Robinson (1997): *Premonoidal Categories and Notions of Computation*. Math. Struct. Comput. Sci. 7(5), pp. 453–468, doi:10.1017/S0960129597002375. Available at `https://doi.org/10.1017/S0960129597002375`.

[34] John Power & Hayo Thielecke (1999): *Closed Freyd- and kappa-categories*. In Jirí Wiedermann, Peter van Emde Boas & Mogens Nielsen, editors: Automata, Languages and Programming, 26th International Colloquium, ICALP'99, Prague, Czech Republic, July 11-15, 1999, Proceedings, Lecture Notes in Computer

Science 1644, Springer, pp. 625–634, doi:10.1007/3-540-48523-6_59. Available at https://doi.org/10.1007/3-540-48523-6_59.

[35] Mario Román (2022): *Notes on Jeffrey's A Graphical View of Programs*. Available at https://www.ioc.ee/~mroman/data/talks/premonoidalgraphicalview.pdf.

[36] Christopher J. Schommer-Pries (2011): *The Classification of Two-Dimensional Extended Topological Field Theories*. arXiv:1112.1000.

[37] Michael Shulman (2008): *Framed Bicategories and Monoidal Fibrations*. Theory and Applications of Categories 20(18), pp. 650–738.

[38] Michael Shulman (2016): *Categorical logic from a categorical point of view*. Available on the web. Available at https://mikeshulman.github.io/catlog/catlog.pdf.

[39] Michael Shulman (2018): *The 2-Chu-Dialectica construction and the polycategory of multivariable adjunctions*. arXiv preprint arXiv:1806.06082.

[40] Sam Staton & Paul Blain Levy (2013): *Universal properties of impure programming languages*. In Roberto Giacobazzi & Radhia Cousot, editors: The 40th Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '13, Rome, Italy - January 23 - 25, 2013, ACM, pp. 179–192, doi:10.1145/2429069.2429091. Available at https://doi.org/10.1145/2429069.2429091.

[41] Ross Street (1996): *Categorical structures*. Handbook of algebra 1, pp. 529–577.

[42] Ross Street & Brian Day (1997): *Monoidal bicategories and Hopf algebroids*. Adv. Math 129, pp. 99–157.

[43] Todd Trimble (2010): *Coherence Theorem for Monoidal Categories (nLab entry), Section 3. Discussion*. https://ncatlab.org/nlab/show/coherence+theorem+for+monoidal+categories, Last accessed on 2022-05-10.

[44] Tarmo Uustalu & Varmo Vene (2008): *Comonadic Notions of Computation*. In Jiří Adámek & Clemens Kupke, editors: Proceedings of the Ninth Workshop on Coalgebraic Methods in Computer Science, CMCS 2008, Budapest, Hungary, April 4-6, 2008, Electronic Notes in Theoretical Computer Science 203, Elsevier, pp. 263–284, doi:10.1016/j.entcs.2008.05.029. Available at https://doi.org/10.1016/j.entcs.2008.05.029.

[45] Dominic Verdon (2017): *Coherence for braided and symmetric pseudomonoids*. CoRR abs/1705.09354. arXiv:1705.09354.

[46] Mark Weber (2013): *Free products of higher operad algebras*. Theory and applications of categories 28(2), pp. 24–65.

A Effectful string diagrams

During the following two lemmas, we will choose to always deal with the leftmost component of the braid clique morphism. Given any clique $\mathrm{Braid}_R(A_0,\dots,A_n)$ we call $A = A_0 \otimes \dots \otimes A_n$ to its tensoring; clique morphisms $\mathrm{Braid}_R(A_0,\dots,A_n) \to \mathrm{Braid}_R(B_0,\dots,B_m)$ are represented by morphisms $R \otimes A \to R \otimes B$.

Lemma A.1. *Let* $(\mathcal{V},\mathcal{G})$ *be a polygraph couple. There exists a premonoidal category,* $\mathsf{Eff}(\mathcal{V},\mathcal{G})$*, that has as objects the braid cliques,* $\mathrm{Braid}_R(A_0,\dots,A_n)$*, in* $\mathsf{Mon}_{\mathsf{Run}}(\mathcal{V},\mathcal{G})$*, and as morphisms the braid clique morphisms between them.*

Proof. Let us first give $\mathsf{Eff}(\mathcal{V},\mathcal{G})$ category structure. The identity on $\mathrm{Braid}_R(A_0,\dots,A_n)$ is the identity on $R \otimes A$. The composition of a morphism $R \otimes A \to R \otimes B$ with a morphism $R \otimes B \to R \otimes C$ is their plain composition in $\mathsf{Mon}_{\mathsf{Run}}(\mathcal{V},\mathcal{G})$.

Let us now check that it is moreover a premonoidal category. Tensoring of cliques is given by concatenation of lists, which coincides with the tensor in $\mathsf{Mon}_{\mathsf{Run}}(\mathcal{V},\mathcal{G})$. However, it is interesting to note that the tensor of morphisms cannot be defined in this way: a morphism $R \otimes A \to R \otimes B$ cannot be tensored with a morphism $R \otimes A' \to R \otimes B'$ to obtain a morphism $R \otimes A \otimes A' \to R \otimes B \otimes B'$.

Whiskering of a morphism $f \colon R \otimes A \to R \otimes B$ is defined with braidings in the left case, $R \otimes C \otimes A \to R \otimes C \otimes B$, and by plain whiskering in the right case, $R \otimes A \otimes C \to R \otimes B \otimes C$, as depicted in Figure 17.

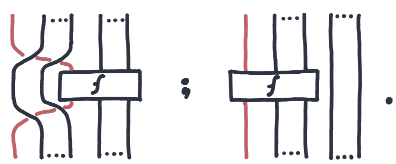

Figure 17: Whiskering in the runtime premonoidal category.

Finally, the associators and unitors are identities, which are always natural and central. □

Lemma A.2. *Let* $(\mathcal{V},\mathcal{G})$ *be a polygraph couple. There exists an identity-on-objects functor* $\mathsf{Mon}(\mathcal{V}) \to \mathsf{Eff}(\mathcal{V},\mathcal{G})$ *that strictly preserves the premonoidal structure and whose image is central. This determines an effectful category.*

Proof. A morphism $v \in \mathsf{Mon}(\mathcal{V})(A,B)$ induces a morphism $(\mathrm{id}_R \otimes v) \in \mathsf{Mon}_{\mathsf{Run}}(\mathcal{V},\mathcal{G})(R \otimes A, R \otimes B)$, which can be read as a morphism of cliques $(\mathrm{id}_R \otimes v) \in \mathsf{Eff}(\mathcal{V},\mathcal{G})(A,B)$. This is tensoring with an identity, which is indeed functorial.

Let us now show that this functor strictly preserves the premonoidal structure. The fact that it preserves right whiskerings is immediate. The fact that it preserves left whiskerings follows from the axioms of symmetry (Figure 18, left). Associators and unitors are identities, which are preserved by tensoring with an identity.

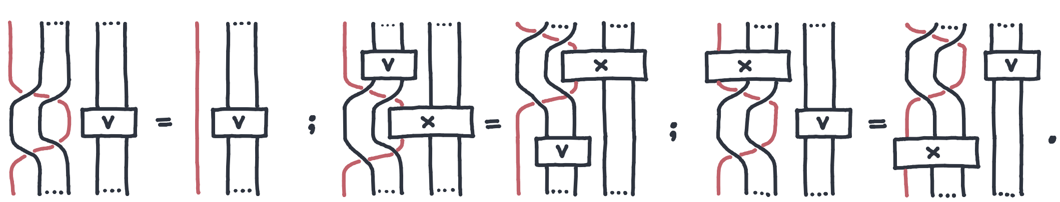

Figure 18: Preservation of whiskerings, and centrality.

Finally, we can check by string diagrams that the image of this functor is central, interchanging with any given $x\colon R\otimes C\to R\otimes D$ (Figure 18, center and right). □

Lemma A.3 (Freeness). *Let $(\mathcal{V},\mathcal{G})$ be a polygraph couple and consider the effectful category determined by $\mathsf{Mon}(\mathcal{V})\to\mathsf{Eff}(\mathcal{V},\mathcal{G})$. Let $\mathbb{V}\to\mathbb{C}$ be a strict effectful category, with a polygraph couple morphism $F\colon(\mathcal{V},\mathcal{G})\to\mathcal{U}(\mathbb{V},\mathbb{C})$. There exists a unique effectful functor from $(\mathsf{Mon}(\mathcal{V})\to\mathsf{Eff}(\mathcal{V},\mathcal{G}))$ to $(\mathbb{V}\to\mathbb{C})$ commuting with F as a polygraph couple morphism.*

Proof. By freeness, there already exists a unique strict monoidal functor $H_0\colon\mathsf{Mon}(\mathcal{V})\to\mathbb{V}$ that sends any object $A\in\mathcal{V}_{\text{obj}}$ to $F_{obj}(A)$. We will show there is a unique way to extend this functor together with the hypergraph assignment $\mathcal{G}\to\mathbb{C}$ into a functor $H\colon\mathsf{Eff}(\mathcal{V},\mathcal{G})\to\mathbb{C}$. Giving such a functor amounts to give some mapping of morphisms containing the runtime R in some position in their input and output,

$$f\colon A_0\otimes\ldots\otimes R\otimes\ldots\otimes A_n\to B_0\otimes\ldots\otimes R\otimes\ldots\otimes B_m$$

to morphisms $H(f)\colon FA_0\otimes\ldots\otimes FA_n\to FB_0\otimes\ldots\otimes FB_n$ in $\mathbb{C}$, in a way that preserves composition, whiskerings, inclusions from $\mathsf{Mon}(\mathcal{V})$, and that is invariant to composition with braidings. In order to define this mapping, we will perform structural induction over the monoidal terms of the runtime monoidal category of the form $\mathsf{Mon}_{\mathsf{Run}}(\mathcal{V},\mathcal{G})(A_0\otimes\ldots\otimes R^{(i)}\otimes\ldots\otimes A_n, R\otimes B_0\otimes\ldots\otimes R^{(j)}\otimes\ldots\otimes B_m)$ and show that it is the only mapping with these properties (Figure 19).

Monoidal terms in a strict, freely presented, monoidal category are formed by identities (id), composition ($\fatsemi$), tensoring ($\otimes$), and some generators (in this case, in Figure 5). Monoidal terms are subject to *(i)* functoriality of the tensor, $\mathrm{id}\otimes\mathrm{id}=\mathrm{id}$ and $(f\fatsemi g)\otimes(h\fatsemi k)=(f\otimes h)\fatsemi(g\otimes k)$; *(ii)* associativity and unitality of the tensor, $f\otimes\mathrm{id}_I=f$ and $f\otimes(g\otimes h)=(f\otimes g)\otimes h$; *(iii)* the usual unitality, $f\fatsemi\mathrm{id}=f$ and $\mathrm{id}\fatsemi f=f$ and associativity $f\fatsemi(g\fatsemi h)=(f\fatsemi g)\fatsemi h$; *(iv)* the axioms of our presentation (in this case, in Figure 6).

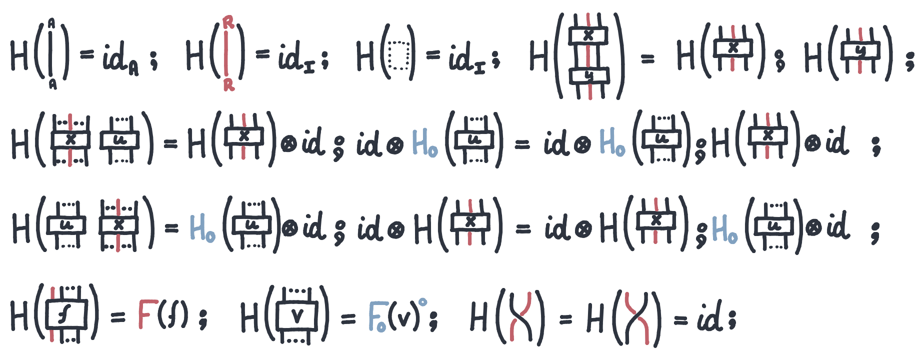

Figure 19: Assignment on morphisms, defined by structural induction on terms.

- If the term is an identity, it can be *(i)* an identity on an object $A\in(\mathcal{V},\mathcal{G})_{\text{obj}}$, in which case it must be mapped to the same identity by functoriality, $H(\mathrm{id}_A)=\mathrm{id}_A$; *(ii)* an identity on the runtime, in which case it must be mapped to the identity on the unit object, $H(\mathrm{id}_R)=\mathrm{id}_I$; or *(iii)* an identity on the unit object, in which case it must be mapped to the identity on the unit, $H(\mathrm{id}_I)=\mathrm{id}_I$.
- If the term is a composition, $(f\fatsemi g)\colon A_0\otimes\ldots\otimes R\otimes\ldots\otimes A_n\to C_0\otimes\ldots\otimes R\otimes\ldots\otimes C_k$, it must be along a boundary of the form $B_0\otimes\ldots\otimes R\otimes\ldots\otimes B_m$: this is because every generator leaves the number of runtimes, R, invariant. Thus, each one of the components determines itself a braid clique morphism. We must preserve composition of braid clique morphisms, so we must map $H(f\fatsemi g)=H(f)\fatsemi H(g)$.

- If the term is a tensor of two terms, $(x \otimes u)\colon A_0 \otimes \ldots \otimes R \otimes \ldots \otimes A_n \to B_0 \otimes \ldots \otimes R \otimes \ldots \otimes B_m$, then only one of them was a term taking R as input and output (without loss of generality, assume it to be the first one) and the other was not: again, by construction, there are no morphisms taking one R as input and producing none, or viceversa. We split this morphism into $x\colon A_0 \otimes \ldots \otimes R \otimes \ldots \otimes A_{i-1} \to B_0 \otimes \ldots \otimes R \otimes \ldots \otimes B_{j-1}$ and $u\colon A_i \otimes \ldots \otimes A_n \to B_j \otimes \ldots \otimes B_m$.

  Again by structural induction, this time over terms $u\colon A_i \otimes \ldots \otimes A_n \to B_j \otimes \ldots \otimes B_m$, we know that the morphism must be either a generator in $\mathscr{V}(A_i,\ldots,A_n;B_j,\ldots,B_n)$ or a composition and tensoring of them. That is, u is a morphism in the image of $\mathsf{Mon}(\mathscr{V})$, and it must be mapped according to the functor $H_0\colon \mathsf{Mon}(\mathscr{V}) \to \mathbb{V}$.

  By induction hypothesis, we know how to map the morphism $x\colon A_0 \otimes \ldots \otimes R \otimes \ldots \otimes A_{i-1} \to B_0 \otimes \ldots \otimes R \otimes \ldots \otimes B_{j-1}$. This means that, given any tensoring $x \otimes u$, we must map it to $H(x \otimes u) = (H(x) \otimes \mathrm{id}) ⨾ (\mathrm{id} \otimes H_0(u)) = (\mathrm{id} \otimes H_0(u)) ⨾ (H(x) \otimes \mathrm{id})$, where $H_0(u)$ is central.
- If the string diagram consists of a single generator, $f\colon R \otimes A \to R \otimes B$, it can only come from a generator $f \in \mathsf{Run}(\mathscr{V},\mathscr{G})(R,A_0,\ldots,A_n;R,B_0,\ldots,B_m) = \mathscr{G}(A_0,\ldots,A_n;B_0,\ldots,B_m)$, which must be mapped to $H(f) = F(f) \in \mathbb{C}(A_0 \otimes \ldots \otimes A_n, B_0 \otimes \ldots \otimes B_m)$. If the string diagram consists of a single braiding, it must be mapped to the identity, because the want the assignment to be invariant to braidings.

Now, we need to prove that this assignment is well-defined with respect to the axioms of these monoidal terms. Our reasoning follows Figure 20.

- The tensor is functorial. We know that $H(\mathrm{id} \otimes \mathrm{id}) = H(\mathrm{id})$, both are identities and that can be formally proven by induction on the number of wires. Now, for the interchange law, consider a quartet of morphisms that can be composed or tensored first and such that, without loss of generality, we assume the runtime to be on the left side. Then, we can use centrality to argue that

$$\begin{aligned} H((x \otimes u) ⨾ (y \otimes v)) &= (H(x) \otimes \mathrm{id}) ⨾ (\mathrm{id} \otimes H_0(u)) ⨾ (H(y) \otimes \mathrm{id}) ⨾ (\mathrm{id} \otimes H_0(v)) \\ &= ((H(x) ⨾ H(y)) \otimes \mathrm{id}) ⨾ (\mathrm{id} \otimes (H_0(u) ⨾ H_0(v))) \\ &= H((x ⨾ y) \otimes (u ⨾ v)). \end{aligned}$$

- The tensor is monoidal. We know that $H(x \otimes \mathrm{id}_I) = (H(x) \otimes \mathrm{id}_I) ⨾ (\mathrm{id} \otimes \mathrm{id}_I) = H(x)$. Now, for associativity, consider a triple of morphisms that can be tensored in two ways and such that, without loss of generality, we assume the runtime to be on the left side. Then, we can use centrality to argue that

$$\begin{aligned} H((x \otimes u) \otimes v) &= (((H(x) \otimes \mathrm{id}) ⨾ (\mathrm{id} \otimes H_0(u))) \otimes \mathrm{id}) ⨾ \mathrm{id} \otimes H_0(v) \\ &= (H(x) \otimes \mathrm{id}) ⨾ (\mathrm{id} \otimes H_0(u) \otimes H_0(v)) \\ &= H(x \otimes (u \otimes v)) \end{aligned}$$

- The terms form a category. And indeed, it is true by construction that $H(x ⨾ (y ⨾ z)) = H((x ⨾ y) ⨾ z)$ and also that $H(x ⨾ \mathrm{id}) = H(x)$ because H preserves composition.
- The runtime category enforces some axioms. The composition of two braidings is mapped to the identity by the fact that H preserves composition and sends both to the identity. Both sides of the braid naturality over a morphism v are mapped to $H_0(v)$; with the multiple braidings being mapped again to the identity.

Thus, H is well-defined and it defines the only possible assignment and the only possible strict pre-monoidal functor. □

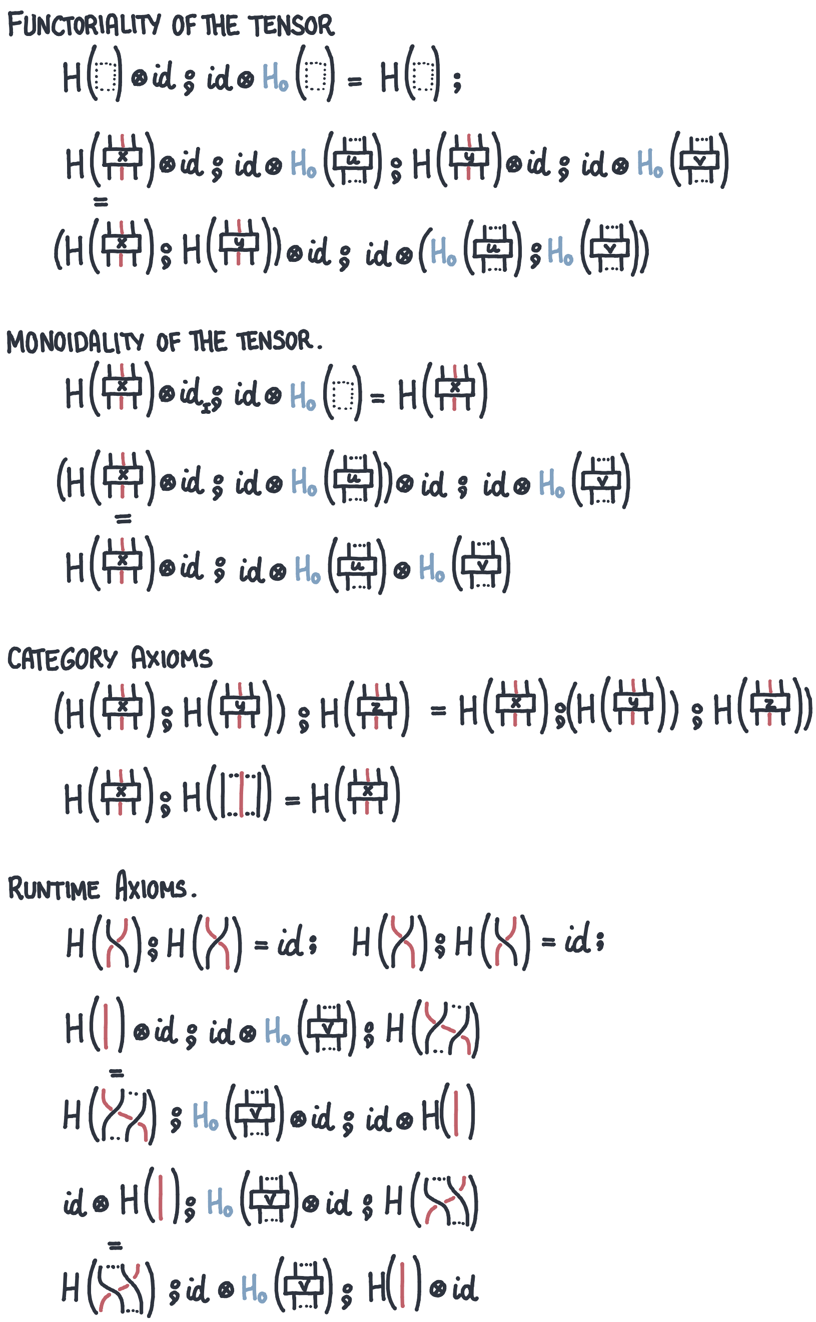

Figure 20: The assignment is well defined.

B Promonads

Lemma B.1 (Kleisli category of a promonad)**.** *Every promonad* $(P, \star, ^\circ)$ *induces a category with the same objects as its base category, but with hom-sets given by* $P(\bullet, \bullet)$*, composition given by* $(\star)$ *and identities given by* (id°)*. This is called its* Kleisli category, kleisli(P)*. Moreover, there exists an identity-on-objects functor* $\mathbb{C} \to \mathrm{kleisli}(P)$*, defined on morphisms by the unit of the promonad.*

Proof. Indeed, let us show that the composition defined by $(\star)$ is unital and associative. Given any $p \in P(A,B)$, we have that identities are neutral with respect to composition on the right, $p \star \mathrm{id}_B^\circ = p < \mathrm{id}_B = p$, and on the left $\mathrm{id}_A^\circ \star u = \mathrm{id}_B > u = u$. Composition is associative by the definition of promonad (Definition 3.7, iii).

Let us now check that the unit of the promonad $(^\circ)$ defines an identity-on-objects functor, which is to say that the assignment on morphisms is functorial. By construction, (id°) is the identity on kleisli(P). Let us show now that the unit of the promonad also preserves composition,

$$f^\circ \star g^\circ = (\mathrm{id}^\circ \star f^\circ) \star g^\circ = \mathrm{id}^\circ < f < g = \mathrm{id}^\circ < (f \mathbin{;} g) = \mathrm{id}^\circ \star (f \mathbin{;} g)^\circ = (f \star g)^\circ. \qquad \square$$

Theorem B.2. *Promonads over a category* $\mathbb{C}$ *correspond to identity-on-objects functors from the category* $\mathbb{C}$*. Given any identity-on-objects functor* $i \colon \mathbb{C} \to \mathbb{D}$ *there exists a unique promonad over* $\mathbb{C}$ *having* $\mathbb{D}$ *as its Kleisli category: the promonad given by the profunctor* $\hom_{\mathbb{D}}(i(\bullet), i(\bullet))$*.*

Proof. Note that the hom-sets of a category $\hom_{\mathbb{D}}(i(\bullet), i(\bullet))$ form a profunctor with actions

$$(p < f) = p \mathbin{;} i(f), \text{ and } (g > p) = i(g) \mathbin{;} p.$$

We define the unit to be the assignment of the functor on morphisms. That is, $f^\circ = i(f)$. We define multiplication of the promonad to be composition in $\mathbb{D}$. Let us now check the axioms of a promonad: premultiplication $f^\circ \star p = i(f) \mathbin{;} p = f > p$, postmultiplication $p \star g^\circ = p \mathbin{;} i(g) = p < g$, dinaturality $p \star (f > q) = p \mathbin{;} i(f) \mathbin{;} q = (p < f) \star q$, and associativity $(p_1 \star p_2) \star p_3 = (p_1 \mathbin{;} p_2 \mathbin{;} p_3) = p_1 \star (p_2 \star p_3)$. We can conclude (with Lemma 3.8) that promonads coincide with identity-on-objects functors. $\square$

Proposition B.3. *A promonad homomorphism between two promonads understood as identity-on-objects functors,* $\mathbb{V} \to \mathbb{C}$ *and* $\mathbb{W} \to \mathbb{D}$*, is equivalently a pair of functors* (F_0, F) *that commute strictly with the two identity-on-objects functors on objects* $F_0(X) = F(X)$ *and morphisms* $F_0(f)^\circ = F(f^\circ)$*.*

Proof. Given a promonad homomorphism (F_0, F), we will construct the pair of functors. One of them is already F_0. For the second one, we observe that $F(p_1 \star p_2) = F(p_1) \star F(p_2)$, and $F(\mathrm{id}) = F_0(\mathrm{id})^\circ = \mathrm{id}$, making F itself into the morphism assignment of a functor. Moreover, this functor, with the same assignment on objects as F_0, makes the square commute.

Given any strictly commutative square of functors with $F_0(f)^\circ = F(f^\circ)$, we can see that, by functoriality, F induces a natural transformation determining a promonad homomorphism.

$$F(f > p < g) = F(f^\circ \star p \star g^\circ) = F_0(f)^\circ \star F(p) \star F_0(g)^\circ = F_0(f) > F(p) < F_0(g).$$

Again by functoriality, and by the commutativity of the square, we can see that it must also satisfy the promonad homomorphism axioms. $\square$

Theorem B.4. *Let $\mathbb{C}\colon \mathbb{V} \nrightarrow \mathbb{V}$ and $\mathbb{D}\colon \mathbb{W} \nrightarrow \mathbb{W}$ be two promonads and let $\mathbb{C} * \mathbb{D}\colon \mathbb{V}\times\mathbb{W} \to \mathbb{V}\times\mathbb{W}$ be their pure tensor. There exist a pair of promonad homomorphisms $L\colon \mathbb{C}\times\mathbb{W} \to \mathbb{C}*\mathbb{D}$ and $R\colon \mathbb{V}\times\mathbb{D} \to \mathbb{C}*\mathbb{D}$. These are universal in the sense that, for every pair of promonad homomorphisms, $A\colon \mathbb{C}\times\mathbb{W}\to\mathbb{E}$ and $B\colon \mathbb{V}\times\mathbb{D}\to\mathbb{E}$, there exists a unique promonad homomorphism $(A\vee B)\colon \mathbb{C}*\mathbb{D}\to\mathbb{E}$ that commutes strictly with them, $(A\vee B)\,⨾\,L = A$ and $(A\vee B)\,⨾\,R = B$.*

Proof. We start by constructing $L\colon \mathbb{C}\times\mathbb{W}\to\mathbb{C}*\mathbb{D}$ and $R\colon \mathbb{V}\times\mathbb{D}\to\mathbb{C}*\mathbb{D}$. These are defined by $L(f,w) = f_{\mathbb{C}}\,⨾\,w_{\mathbb{W}} = w_{\mathbb{W}}\,⨾\,f_{\mathbb{C}}$ and $R(g,v) = g_{\mathbb{D}}\,⨾\,v_{\mathbb{V}} = v_{\mathbb{V}}\,⨾\,g_{\mathbb{D}}$. Let us check that L is promonad homomorphism, R follows a similar reasoning.

- $L((v,w') > (f,w)) = L(v > f, w'\,⨾\,w) = v\,⨾\,f\,⨾\,w'\,⨾\,w = v\,⨾\,w'\,⨾\,f\,⨾\,w = (v,w') > L(f,w)$,
- $L((f,w) < (v,w')) = L(f < v, w\,⨾\,w') = f\,⨾\,v\,⨾\,w\,⨾\,w' = f\,⨾\,w\,⨾\,v\,⨾\,w' = L(f,w) < (v,w')$,
- $L((f,w) \star (f',w')) = L(f\star f', w\,⨾\,w') = f\,⨾\,f'\,⨾\,w\,⨾\,w' = f\,⨾\,w\,⨾\,f'\,⨾\,w' = L(f,w)\star L(f',w')$,
- $L(v^{\circ},w) = v\,⨾\,w = L_0(v,w)^{\circ}$.

We will now construct the promonad homomorphism as $(A\vee B)(f_{\mathbb{C}}) = A(\mathrm{id},f)$ and $(A\vee B)(g_{\mathbb{D}}) = B(g,\mathrm{id})$. This definition is forced by commutation with l and r and automatically defines the promonad homomorphism on all the generators of $\mathbb{C}*\mathbb{D}$. We can see it is well-defined, with the most interesting case being proving that it preserves the interchange of morphisms: indeed, $(A\vee B)(f_{\mathbb{C}}\,⨾\,w_{\mathbb{W}}) = A(f,\mathrm{id})\,⨾\,B(\mathrm{id},w) = A(f,\mathrm{id})\,⨾\,((\mathrm{id},w) > B(\mathrm{id},\mathrm{id})) = A(f,\mathrm{id}) < (\mathrm{id},w) = A(f,w) = (A\vee B)(w_{\mathbb{W}}\,⨾\,f_{\mathbb{C}})$, and the case with (v,g) is analogous. □

C Cliques

Definition C.1 (Clique)**.** In a category $\mathbb{C}$, a *clique* (X,θ), is a family of objects, X_i, indexed by a set $i\in I$, and a family of isomorphisms $\theta_{i,j}\colon X_i \to X_j$ such that $\theta_{ij}\,⨾\,\theta_{jk} = \theta_{ik}$ and $\theta_{ii} = \mathrm{id}$.

Definition C.2 (Clique morphism)**.** A morphism between two cliques in the same category, $f\colon (X,\theta)\to(Y,\psi)$, is a family of morphisms $f_{ij}\colon X_i\to Y_j$ making every possible square commute, which means that $\theta_{ij}\,⨾\,f_{jl} = f_{ik}\,⨾\,\psi_{kl}$.

Proposition C.3. *A clique morphism $f\colon (X,\theta)\to(Y,\psi)$ is completely determined by its value between any two indices, $f_{ij}\colon X_i\to Y_j$.*

Proof. By the definition, $f_{kl} = \theta_{ki}\,⨾\,f_{ij}\,⨾\,\psi_{lk}^{-1}$, where we use that the clique is made up of isomorphisms. □

D Pseudomonoids

Theorem D.1. *An effectful category (or monoidal Freyd category) is a pseudomonoid on the monoidal 2-category of promonads with promonad homomorphism, promonad transformations and the pure tensor of promonads. A pseudomonoid homomorphism between effectful categories is an effectful functor.*

Proof sketch. Consider the data for a pseudomonoid $(\mathbb{V}\to\mathbb{C},\otimes,I,\alpha,\lambda,\rho)$ of promonads with the pure tensor. The promonad $\mathbb{V}\to\mathbb{C}$ gives us the pair of categories that will form the effectful category. We have a pair of promonad homomorphisms, $(\otimes)\colon (\mathbb{V}\times\mathbb{V}\to\mathbb{C}*\mathbb{C})\to(\mathbb{V}\to\mathbb{C})$ and $I\colon (1\to 1)\to(\mathbb{V}\to\mathbb{C})$ that make $\mathbb{V}$ into a monoidal category and $\mathbb{C}$ into a premonoidal category with the same unit and tensor.

Finally, the promonad transformations are natural transformations. This means that the associator and unitor 2-cells are natural transformations describing the associators and unitors of the monoidal category $\mathbb{V}$. These are the same associators and unitors of the premonoidal category $\mathbb{C}$. Crucially, because they are in $\mathbb{V}$, they are central with respect to every morphism in $\mathbb{C}$. □

E Background material

E.1 Monoidal categories

This section of the appendix has been repurposed from a similar summary of monoidal categories [23], but it contains only standard material on monoidal categories that we choose to repeat to fix conventions.

Definition E.1 ([26]). A **monoidal category**, $(\mathbb{C}, \otimes, I, \alpha, \lambda, \rho)$, is a category $\mathbb{C}$ equipped with a functor $\otimes \colon \mathbb{C} \times \mathbb{C} \to \mathbb{C}$, a unit $I \in \mathbb{C}$, and three natural isomorphisms: the associator $\alpha_{A,B,C} \colon (A \otimes B) \otimes C \cong A \otimes (B \otimes C)$, the left unitor $\lambda_A \colon I \otimes A \cong A$ and the right unitor $\rho_A \colon A \otimes I \cong A$; such that $\alpha_{A,I,B} ; (\mathrm{id}_A \otimes \lambda_B) = \rho_A \otimes \mathrm{id}_B$ and $(\alpha_{A,B,C} \otimes \mathrm{id}) ; \alpha_{A,B \otimes C,D} ; (\mathrm{id}_A \otimes \alpha_{B,C,D}) = \alpha_{A \otimes B,C,D} ; \alpha_{A,B,C \otimes D}$. A monoidal category is *strict* if α, λ and ρ are identities.

Definition E.2 (Monoidal functor, [26]). Let

$$(\mathbb{C}, \otimes, I, \alpha^{\mathbb{C}}, \lambda^{\mathbb{C}}, \rho^{\mathbb{C}}) \text{ and } (\mathbb{D}, \boxtimes, J, \alpha^{\mathbb{D}}, \lambda^{\mathbb{D}}, \rho^{\mathbb{D}})$$

be monoidal categories. A *monoidal functor* (sometimes called *strong monoidal functor*) is a triple (F, ε, μ) consisting of a functor $F \colon \mathbb{C} \to \mathbb{D}$ and two natural isomorphisms $\varepsilon \colon J \cong F(I)$ and $\mu \colon F(A \otimes B) \cong F(A) \boxtimes F(B)$; such that

- the associators satisfy $\alpha^{\mathbb{D}}_{FA,FB,FC} ; (\mathrm{id}_{FA} \otimes \mu_{B,C}) ; \mu_{A,B \otimes C} = (\mu_{A,B} \otimes \mathrm{id}_{FC}) ; \mu_{A \otimes B,C} ; F(\alpha^{\mathbb{C}}_{A,B,C})$,
- the left unitor satisfies $(\varepsilon \otimes \mathrm{id}_{FA}) ; \mu_{I,A} ; F(\lambda^{\mathbb{C}}_{A}) = \lambda^{\mathbb{D}}_{FA}$
- the right unitor satisfies $(\mathrm{id}_{FA} \otimes \varepsilon) ; \mu_{A,I} ; F(\rho^{\mathbb{C}}_{FA}) = \rho^{\mathbb{D}}_{FA}$.

A monoidal functor is a *monoidal equivalence* if it is moreover an equivalence of categories. Two monoidal categories are monoidally equivalent if there exists a monoidal equivalence between them.

During most of the paper, we omit all associators and unitors from monoidal categories, implicitly using the *coherence theorem* for monoidal categories.

Theorem E.3 (Coherence theorem, [26]). *Every monoidal category is monoidally equivalent to a strict monoidal category.*

Definition E.4 (Symmetric monoidal category, [26]). A *symmetric monoidal category* $(\mathbb{C}, \otimes, I, \alpha, \lambda, \rho, \sigma)$ is a monoidal category $(\mathbb{C}, \otimes, I, \alpha, \lambda, \rho)$ equipped with a braiding $\sigma_{A,B} \colon A \otimes B \to B \otimes A$, which satisfies the hexagon equation

$$\alpha_{A,B,C} ; \sigma_{A,B \otimes C} ; \alpha_{B,C,A} = (\sigma_{A,B} \otimes \mathrm{id}) ; \alpha_{B,A,C} ; (\mathrm{id} \otimes \sigma_{A,C})$$

and additionally satisifes $\sigma_{A,B} ; \sigma_{B,A} = \mathrm{id}$.

Definition E.5 ([26]). A symmetric monoidal functor between two symmetric monoidal categories $(\mathbb{C}, \sigma^{\mathbb{C}})$ and $(\mathbb{D}, \sigma^{\mathbb{D}})$ is a monoidal functor $F \colon \mathbb{C} \to \mathbb{D}$ such that $\sigma^{\mathbb{D}} ; \mu = \mu ; F(\sigma^{\mathbb{C}})$.

E.2 Sesquifunctors

Definition E.6. Let $\mathbb{A}_1,\dots,\mathbb{A}_n$ and $\mathbb{B}$ be categories. A *sesquifunctor* $T\colon \mathbb{A}_1,\dots,\mathbb{A}_n \to \mathbb{B}$ [41] is an assignment on objects and morphisms that is independently functorial on each variable. That is, the sesquifunctor is given by a family of functors

$$T_i(X_1,\dots,\bullet_i\dots,X_n)\colon \mathbb{A}_i \to \mathbb{B} \text{ for } i = 1,\dots,n.$$

These functors coincide on objects, in that $T(X_1,\dots,X_n)$ is uniquely determined independently of the T_i we use to define it.

Remark E.7. Sesquifunctors form a multicategory with ordinary composition, which preserves single-variable functoriality. Moreover, they form a 2-multicategory with transformations between them.

Proposition E.8. *The multicategory of sesquiprofunctors is representable by the funny tensor product of categories* $(\square)\colon$ **Cat** $\times$ **Cat** $\to$ **Cat**. *The funny tensor product of two categories,* $\mathbb{C}\,\square\,\mathbb{D}$*, is computed as the following pushout, where* $\mathbb{C}_0$ *and* $\mathbb{D}_0$ *are the discrete categories on the objects of* $\mathbb{C}_0$ *and* $\mathbb{D}_0$*.*

$$\begin{array}{ccc} \mathbb{C}_0 \times \mathbb{D}_0 & \longrightarrow & \mathbb{C} \times \mathbb{D}_0 \\ \downarrow & & \downarrow \\ \mathbb{C}_0 \times \mathbb{D} & \longrightarrow & \mathbb{C}\,\square\,\mathbb{D} \end{array}$$

Explicitly, objects of the funny tensor product are pairs of objects. Morphisms are either morphisms in $\mathbb{C}$*, in* $\mathbb{D}$*, or formal compositions of both, as it happens with the coproduct of monoids. See Weber, [46].*

E.3 Double categories

Definition E.9 (Monoids and promonoids)**.** A *monoid* in a double category is an arrow $T\colon \mathbb{A} \to \mathbb{A}$ together with cells $m \in \mathrm{hom}(M \otimes M; 1, 1; M)$ and $e \in \mathrm{cell}(1; 1, 1; M)$, called multiplication and unit, satisfying unitality and associativity.

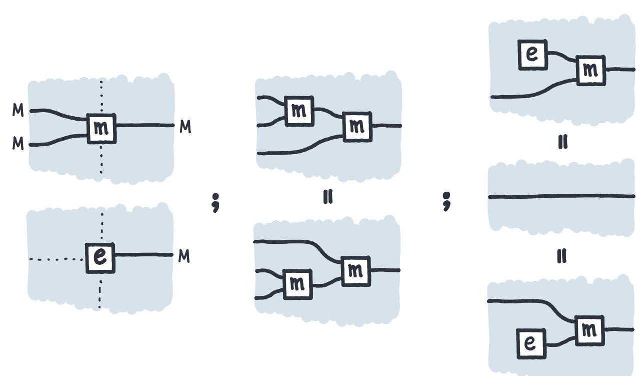

Figure 21: Data and axioms of a monoid in a double category.

A *promonoid* in a double category is a proarrow $M\colon \mathbb{A} \to \mathbb{A}$ together with cells $m \in \mathrm{cell}(1; M \otimes M, M, 1)$ and $e \in \mathrm{cell}(1; 1, M; 1)$, called promultiplication and prounit, satisfying unitality and associativity.

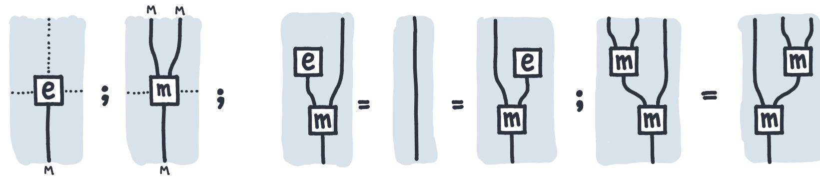

Figure 22: Data and axioms of a promonoid in a double category.

Dually, we can define *comonoids* and *procomonoids*.

A monad is a monoid on the category of categories, functors and profunctors **Cat**.

E.4 Monoidal bicategories

Definition E.10. In a monoidal bicategory, a *pseudomonoid* over a 0-cell M is a pair of 1-cells, $M \boxtimes M \to M$ and $I \to M$, together with the following triple of invertible 2-cells representing associativity and unitality (Figure 15), and satisfying the pentagon and triangle equations (Figure 24).

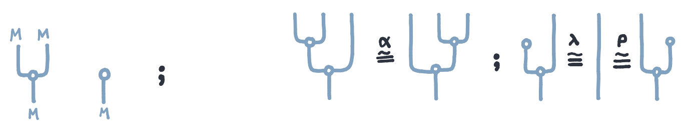

Figure 23: Data for a pseudomonoid.

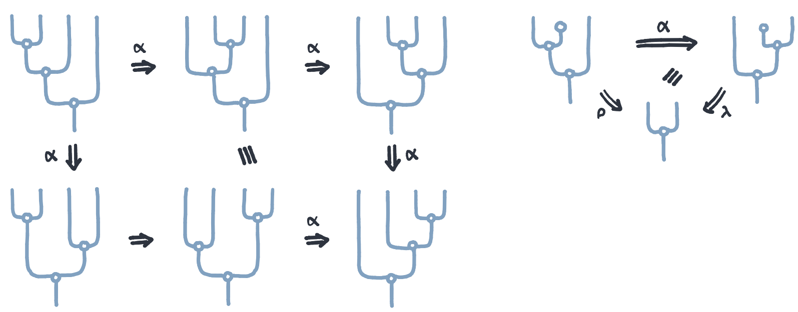

Figure 24: Pentagon and triangle axioms for a pseudomonoid.

A *symmetric* pseudomonoid is a pseudomonoid endowed with an invertible 2-cell representing commutation (Figure 25), and satisfying symmetry and the two hexagon equations (Figure 26).

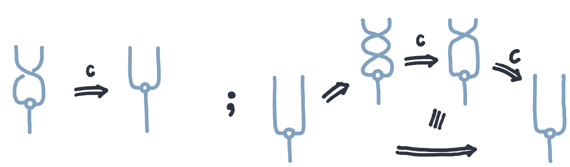

Figure 25: Commutator and symmetry for a pseudomonoid.

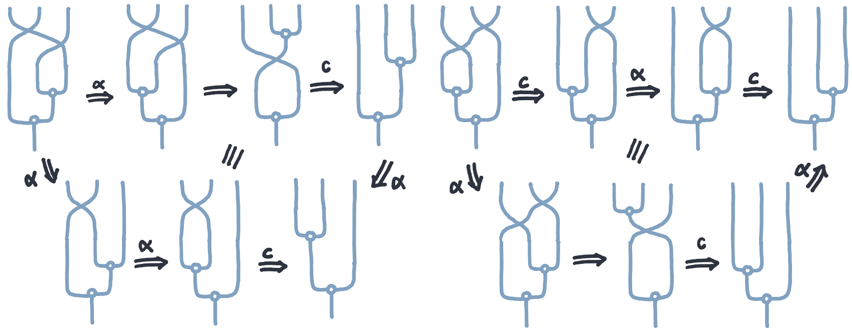

Figure 26: Hexagon equations for a symmetric pseudomonoid.

Definition E.11 (Homomorphism of pseudomonoids)**.** A homomorphism of pseudomonoids is given by a 1-cell between their underlying 0-cells and the following invertible 2-cells, representing preservation of the multiplication and the unit (Figure 15), and satisfying compatibility with associativity and unitality (Figure 28).

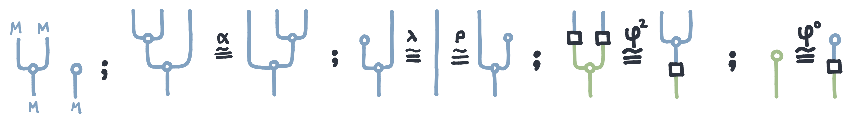

Figure 27: Data for a pseudomonoid homomorphism.

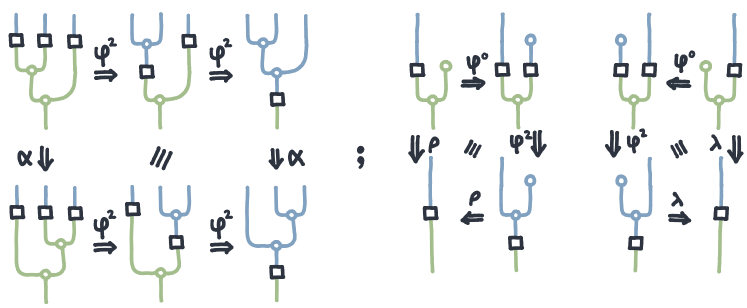

Figure 28: Axioms for a pseudomonoid homomorphism.