Electronic Notes in Theoretical Computer Science 69 (2003)
URL: http://www.elsevier.nl/locate/entcs/volume69.html 19 pages

The Uniformity Principle on Traced Monoidal Categories

Masahito Hasegawa

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

Abstract

The uniformity principle for traced monoidal categories has been introduced as a natural generalization of the uniformity principle (Plotkin's principle) for fixpoint operators in domain theory. We show that this notion can be used for constructing new traced monoidal categories from known ones. Some classical examples like the Scott induction principle are shown to be instances of these constructions. We also characterize some specific cases of our constructions as suitable enriched limits.

1 Introduction

Traced monoidal categories were introduced by Joyal, Street and Verity [9] as the categorical structure for cyclic phenomena arising from various areas of mathematics. They are (balanced) monoidal categories enriched with a *trace*, which is a natural generalization of the traditional notion of trace in linear algebra, and can be regarded as an operator to create a "loop".

In computer science, specialized versions of traced monoidal categories naturally arise from considerations on feedback operators (in the fairly general sense) as well as cyclic data structures. In the middle of 90's, Martin Hyland and the author independently observed a bijective correspondence between Conway (Bekič, or dinatural diagonal) fixpoint operators [3,12] and traces on categories with finite products [6,7]. Thus, in this setting, the notion of trace precisely captures the well-behaved fixpoint operators commonly used in computer science.

An extra bonus of this trace-fixpoint correspondence is that the uniformity principle (à la Plotkin) on a fixpoint operator precisely amounts to a uniformity principle on the corresponding trace, as introduced in [6,7] (see also historical remarks at the end of Sec. 2). This uniformity principle is general enough to make sense for arbitrary traced monoidal categories. An application of this concept is found in Selinger's work on categorical models of asynchronous communications [11].

In the present paper we demonstrate that this uniformity principle on traced monoidal categories can be used for constructing new traced monoidal categories (and categories with fixpoint operators) from known ones. The construction is very simple and in some sense old – its origin can be traced back to the Scott induction principle.

Moreover, our constructions seem to enjoy characterizations by some universal property, as certain limits in an enriched sense. We study this issue for some specific cases.

The rest of this paper is organized as follows. We recall the notion of uniformity and strict maps for traced monoidal categories in Sec. 2. Sec. 3 is devoted for some basic results on the uniformity principle. In Sec. 4 we recall the correspondence between traces and fixpoint operators and explain the relationship between the uniformity principle for traces and that for Conway operators. Sec. 5 and 6 form the central part of this paper, where we show how the uniformity principle can be used for constructing new traced monoidal categories. Sec. 7 gives some observations on the uniformity principle and the **Int** construction. Appendix A explains the graphical notations for monoidal categories, which are used throughout this paper. The definition of traced symmetric monoidal categories is summarized in Appendix B.

Remark 1.1 Although many of the notions and results in this work apply to general traced balanced monoidal categories, in this paper we restrict our attention only to traced symmetric monoidal categories, firstly because for ease of presentation, and secondly because most of examples relevant to computer science seem to be instances of symmetric monoidal categories. So by a traced monoidal category, in this paper we mean a traced symmetric monoidal category unless otherwise stated.

2 Uniformity for Traces

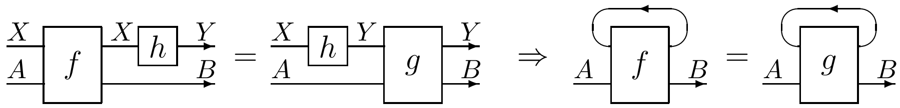
Definition 2.1 Consider a traced monoidal category $\mathcal{C}$ with trace Tr. We say $h : X \to Y$ is *strict* in $\mathcal{C}$ (with respect to the trace Tr) if the following condition holds:

For any $f : A \otimes X \to B \otimes X$ and $g : A \otimes Y \to B \otimes Y$,
$(id_B \otimes h) \circ f = g \circ (id_A \otimes h) : A \otimes X \to B \otimes Y$

$$\begin{tikzcd} A \otimes X \arrow[r, "f"] \arrow[d, "id_A \otimes h"'] & B \otimes X \arrow[d, "id_B \otimes h"] \\ A \otimes Y \arrow[r, "g"'] & B \otimes Y \end{tikzcd}$$

implies $Tr^X_{A,B}(f) = Tr^Y_{A,B}(g) : A \to B$.

In terms of the graphic notation:



Definition 2.2 Let $\mathcal{C}$ be a traced monoidal category with trace Tr, and $\mathcal{S}$ be a class of arrows of $\mathcal{C}$. We say Tr is *uniform* (or: Tr satisfies the uniformity principle) for $\mathcal{S}$ if, for any $h : X \to Y$ of $\mathcal{S}$, the condition in Def. 2.1 holds.

Thus the class of strict maps is the largest class of arrows for which the trace satisfies the uniformity principle.

As noted by Selinger [11], the uniformity principle can be seen a proof principle for showing two traces are equivalent: to prove $Tr^X(f) = Tr^Y(g)$, it suffices to find $h : X \to Y$ of $\mathcal{S}$ such that $(id \otimes h) \circ f = g \circ (id \otimes h)$ holds. In such applications, it is often convenient to give $\mathcal{S}$ a priori as a suitable subcategory containing reasonably rich class of arrows; see *ibid.* for several good examples.

However, we note that there is no reason to expect that the class of strict maps form a category – in fact there are counterexamples, as we will see shortly. For now, we shall recall some popular examples, where strict maps actually form categories.

Example 2.3

- The category of finite dimensional vector spaces over a field and linear maps, where the trace is the generalization of the standard trace (see e.g. [9]). An arrow in this category is strict if and only if it is an isomorphism.
- The category **Cppo** of ω-cpo's with bottom and continuous functions, where the trace is induced from the least fixpoint operator (see Sec. 4). An arrow is strict w.r.t. this trace iff it preserves the bottom element.
- The category of sets and partial functions, with coproducts as monoidal products and the natural feedback operator as trace. In this setting any arrow is strict. $\square$

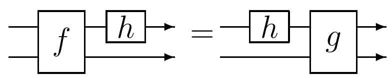
Remark 2.4 Our terminology ("uniformity" and "strictness") is motivated from that of fixpoint operators in domain theory, and will be justified in Sec. 4. The corresponding notions for various specialized versions of traced monoidal categories had appeared in the literature under various names and forms. In particular, Ştefănescu's "enzymatic rule" for his network algebras [13] precisely corresponds to our uniformity principle, where strict arrows are called "functorial arrows" (following the terminology by Arbib and Manes for partially additive categories [2]). See *ibid.* for bibliographic remarks and also several examples.

3 Basic Facts

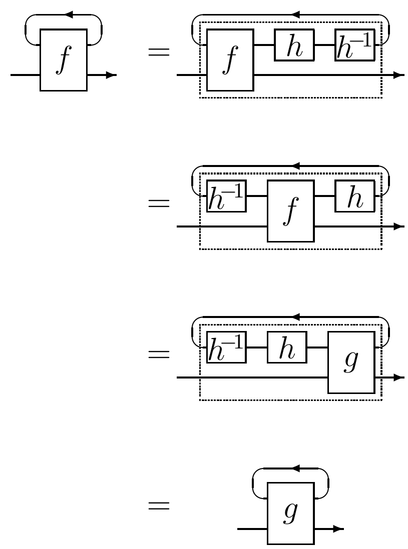
As a warming up, let us see a few basic (but fundamental) properties which strict maps in a traced monoidal category enjoy – and do *not* enjoy. In summary, we will see that (1) isomorphisms are strict, (2) strict maps are closed under tensor product, but (3) strict maps may *not* be closed under composition.

Lemma 3.1 *In a traced monoidal category, isomorphisms are strict.*

Proof. Suppose h is an isomorphism. Then



implies



$h^{-1} \circ h = id$

dinaturality

assumption

$h \circ h^{-1} = id$

Hence h is strict. □

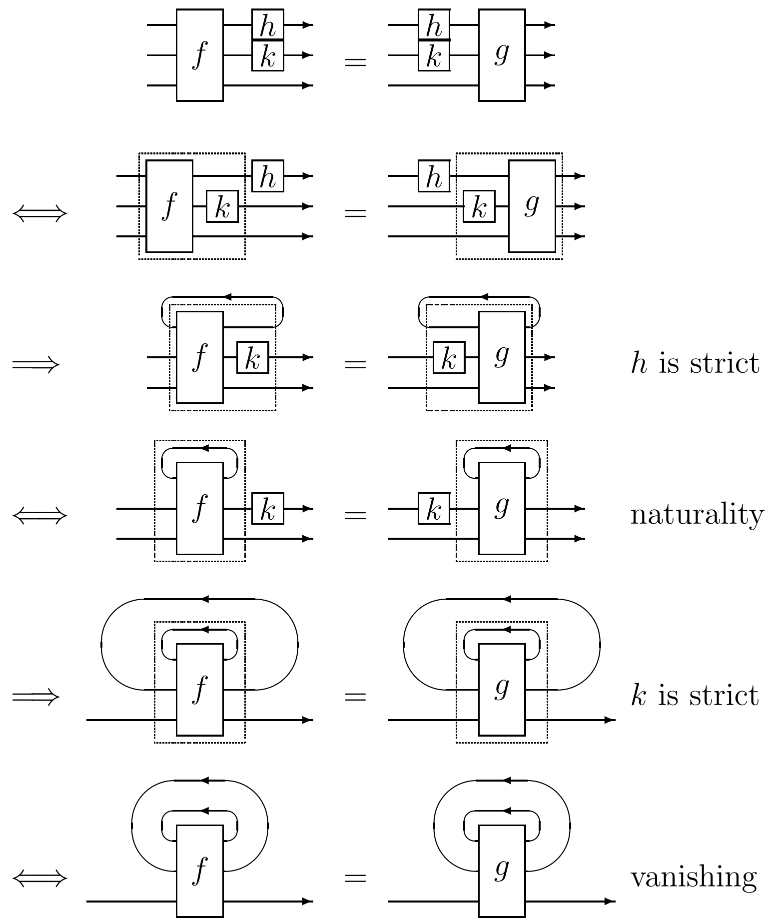
By a similar argument we also have

Lemma 3.2 *The composition of a strict map and an isomorphism is strict.* □

(This actually subsumes the previous lemma, as an identity is trivially strict.)

Lemma 3.3 *Strict maps are closed under tensor product.*

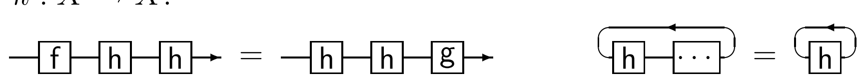
Proof. Suppose h and k are strict. Then



Hence $k \otimes h$ is also strict. □

Proposition 3.4 *There are traced monoidal categories in which strict maps are not closed under composition.*

Proof. Consider the traced monoidal category generated from an object X, arrows $\mathsf{f}, \mathsf{g}, \mathsf{h} : X \to X$ with axioms $\mathsf{h} \circ \mathsf{h} \circ \mathsf{f} = \mathsf{g} \circ \mathsf{h} \circ \mathsf{h}$ and $Tr^X(k \circ \mathsf{h}) = Tr^X(\mathsf{h})$ for any $k : X \to X$.



In this traced category, h is strict but $\mathsf{h} \circ \mathsf{h}$ is not (although $\mathsf{h} \circ \mathsf{h} \circ \mathsf{f} = \mathsf{g} \circ \mathsf{h} \circ \mathsf{h}$, $Tr^X(\mathsf{f}) \neq Tr^X(\mathsf{g})$). □

Corollary 3.5 *There are traced monoidal categories in which strict maps are not closed under trace.*

Proof. Consider the same example as the last proposition. By the previous lemmas, we know that $c_{X,X} \circ (\mathsf{h} \otimes \mathsf{h}) : X \otimes X \to X \otimes X$ is strict. However, $Tr^X(c_{X,X} \circ (\mathsf{h} \otimes \mathsf{h})) = \mathsf{h} \circ \mathsf{h}$ is not strict. □

4 The Trace-Fixpoint Correspondence

Before going into the main topic of this paper (constructions of traced monoidal categories using uniformity principle), let us recall how traces and fixpoint operators on a category with finite products are related, and see how the uniformity principle on traced monoidal categories generalizes that on categories with fixpoint operator. Later these observations enable us to specialize some of our constructions of traced monoidal categories to those of categories with fixpoint operator.

4.1 Fixpoint Operators

Let $\mathcal{C}$ be a category with finite products.

Definition 4.1 A *parameterized fixpoint operator* on $\mathcal{C}$ is a family of functions

$$(-)^\dagger : \mathcal{C}(A \times X, X) \to \mathcal{C}(A, X)$$

which is natural in A and satisfies the *fixpoint equation* $f^\dagger = f \circ \langle id_A, f^\dagger \rangle : A \to X$ for $f : A \times X \to X$.

Definition 4.2 A *Conway operator* on $\mathcal{C}$ is a parameterized fixpoint operator $(-)^\dagger$ which satisfies

- *dinaturality:* $(f \circ \langle \pi_{A,X}, g \rangle)^\dagger = f \circ \langle id_A, (g \circ \langle \pi_{A,Y}, f \rangle)^\dagger \rangle : A \to X$ for $f : A \times Y \to X$ and $g : A \times X \to Y$
- *diagonal property:* $(f \circ (id_A \times \Delta_X))^\dagger = (f^\dagger)^\dagger : A \to X$ for $f : A \times X \times X \to X$, where $\Delta_X : X \to X \times X$ is the diagonal map.

This axiomatization of Conway operators is taken from [12]; see [3,7] for other possible axiomatizations.

4.2 The Correspondence

The basic relationship between traces and fixpoint operators is

Theorem 4.3 (Hyland/Hasegawa) *For any category with finite products, to give a Conway operator is to give a trace (where finite products are taken as the monoidal structure).* □

This observation, noticed independently by Martin Hyland and the author, together with some implications to the study on semantics of recursive computation, was first announced in [6]; a full proof is found in [7] (but we should note that mathematically equivalent observations have been made by several authors even before the notion of traced monoidal categories was invented, in particular by Bloom and Ésik [3] and Ştefănescu [13]). Here we shall only give the constructions of this bijective correspondence:

$$\frac{f : A \times X \to X}{f^\dagger = Tr^X_{A,X}(\Delta_X \circ f) : A \to X} \quad \frac{g : A \times X \to B \times X}{Tr^X_{A,B}(g) = \pi_{B,X} \circ (g \circ (\mathsf{id}_A \times \pi'_{B,X}))^\dagger : A \to B}$$

4.3 Correspondence of the Uniformity Principles

Quite fortunately, the correspondence between traces and Conway operators smoothly extends to the uniformity principles. Let us recall the classical notion of uniformity for fixed points (Plotkin's principle):

Definition 4.4 Let $(-)^\dagger$ be a parameterized fixpoint operator on a category with finite products. We say $h : X \to Y$ is *strict* (with respect to $(-)^\dagger$) if the following condition holds:

For any $f : A \times X \to X$ and $g : A \times Y \to Y$, $h \circ f = g \circ (id_A \times h)$

$$\begin{tikzcd} A \times X \arrow[r, "f"] \arrow[d, "id_A \times h"'] & X \arrow[d, "h"] \\ A \times Y \arrow[r, "g"'] & Y \end{tikzcd}$$

implies $g^\dagger = h \circ f^\dagger : A \to Y$.

(The reader should compare this with the corresponding definition for traced monoidal categories (Def. 2.1). They are quite similar – we should confess that Def. 2.1 was inspired from Def. 4.4 – but they are also subtly different, in that the arrow h in the pre-condition of Def. 4.4 survives in the post-condition while it disappears from the post-condition in Def. 2.1.)

For instance, in the category **Cppo** an arrow is strict with respect to the standard least fixpoint operator if and only if it preserves the bottom, thus is strict in the standard sense.

It is also possible to formulate the uniformity principle with respect to a given class (quite often a subcategory) $\mathcal{S}$ of strict maps in the same way as Def. 2.2: $(-)^\dagger$ satisfies the uniformity principle for the class of arrows $\mathcal{S}$ if, for any $h \in \mathcal{S}$, the condition in Def. 4.4 holds. In [3] (cf. [12]) it is shown that a Conway operator satisfying the uniformity principle for a lluf subcategory with finite products is an iteration operator [3] – thus uniformity principle for a reasonably rich $\mathcal{S}$ does have strong consequences. (But again we shall warn that there are cases where the strict maps do not form a category! Also we note that $\mathcal{S}$ being a category is not necessary to show the above-mentioned result; it suffices to ask that $\mathcal{S}$ contains a few structural morphisms. The only reason to assume $\mathcal{S}$ to be a category seems that it is the case for all natural examples known so far.)

Theorem 4.5 *Consider a category with finite products and a Conway operator, and the corresponding trace (as given in Thm. 4.3). Then an arrow is strict w.r.t. the Conway operator if and only if it is strict w.r.t. the trace.* □

A proof is given in Appendix C. It is almost straightforward to show that the trace-strictness implies the Conway-strictness. The other direction is more non-trivial and slightly tricky; perhaps the easiest way, as taken here, is first to show that Conway-strict arrows are closed under products, using the Bekič property (which gives another axiomatization of Conway operators [6,7]).

This theorem confirms that the uniformity principles for traces and Conway operators coincide, as long as we talk about those on categories with finite products. We regard this as a strong evidence that our notion of uniformity principle on traces is a natural generalization of that on traditional fixpoint operators in the theory of computation, especially in domain theory. Technically, this result enables us to specialize the constructions of traced monoidal categories via the uniformity principle to those of categories with finite products and Conway operator, to be introduced in the following sections.

5 Constructions via Uniformity

Good constructions of categories with trace or fixpoint operator are of great value, as the recent history of knot theory (after 80's we know that many knot invariants can be constructed in a generic way) and domain theory (the progress of axiomatic and synthetic domain theory resulted some constructions of models of domain theory) has shown. The main goal of this paper is a small contribution towards this direction: to demonstrate that the uniformity principle on traced monoidal categories helps us with constructing new traced monoidal categories. The construction is of rather general nature, and naturally we cannot expect very strong results. However, we shall try to indicate how natural it is, by pointing the relationship with a classical example: the Scott induction principle in domain theory.

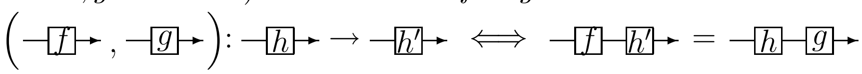
To motivate the constructions, let us start with some elementary exercises:

> Let F, G etc be functors between traced monoidal categories which preserve the structure on the nose (which we shall call strict traced functors). Can we give a trace to categories like the comma category $F \downarrow G$, categories of (co)algebras of endofunctors F-Alg, F-Coalg, or even the inserters (dialgebras) of F and G etc?

The answer depends on the cases – in general we cannot (as expected), but in some particular cases there exists a natural way to give a trace. It turns out that, for *all* of these examples, we can naturally identify a full subcategory which is *traced monoidal* – with help of the *uniformity principle*.

5.1 *First Example: $\mathcal{C}^{\multimap}$*

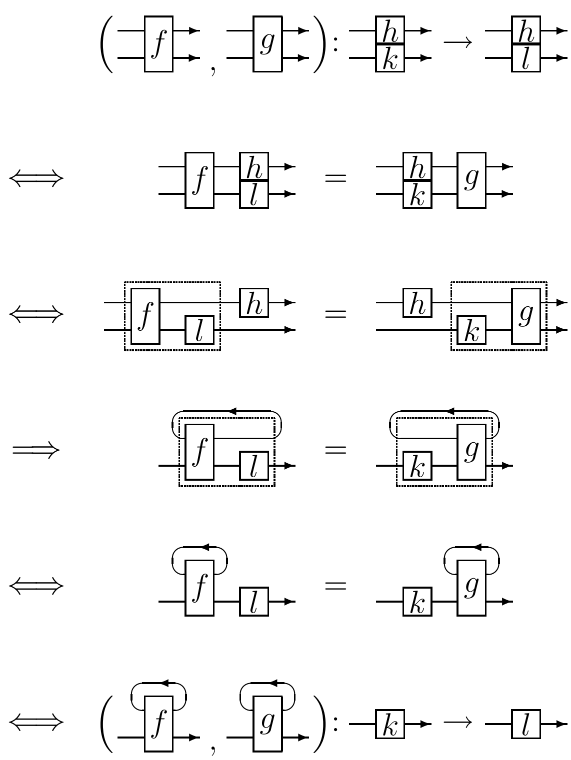
We shall look at a simple case in detail: given a traced monoidal category $\mathcal{C}$ with a trace Tr, let us consider the arrow category $\mathcal{C}^{\rightarrow}$, whose objects are arrows of $\mathcal{C}$ and a morphism from $h : X \rightarrow Y$ to $h' : X' \rightarrow Y'$ in $\mathcal{C}^{\rightarrow}$ is a pair $(f : X \rightarrow X', g : Y \rightarrow Y')$ such that $h' \circ f = g \circ h$ holds in $\mathcal{C}$.



$\mathcal{C}^{\rightarrow}$ naturally inherits a symmetric monoidal structure from $\mathcal{C}$, determined by a pointwise manner. The question is then how to give a trace, say, for $(f, g) : k \otimes h \rightarrow l \otimes h$ (i.e., $(l \otimes h) \circ f = g \circ (k \otimes h)$). The natural candidate is the pair $(Tr(f), Tr(g))$ – what is not obvious is that if this is really a morphism from k to l, i.e., $l \circ Tr(f) = Tr(g) \circ k$ holds. At this point, the reader probably notice that we can appeal to the uniformity principle: if h is *strict* w.r.t. Tr, then $(l \otimes h) \circ f = g \circ (k \otimes h)$ implies $l \circ Tr(f) = Tr(g) \circ k$! Let us define $\mathcal{C}^{\multimap}$ to be a full subcategory of $\mathcal{C}^{\rightarrow}$ whose objects are *strict maps* w.r.t. the trace. Since strict maps are closed under tensor product, $\mathcal{C}^{\multimap}$ is a symmetric monoidal full subcategory of $\mathcal{C}^{\rightarrow}$.

Proposition 5.1 *$\mathcal{C}^{\multimap}$ is a traced monoidal category.*

Proof. For $(f, g) : k \otimes h \rightarrow l \otimes h$ (i.e. $(l \otimes h) \circ f = g \circ (k \otimes h)$), define the trace on $\mathcal{C}^{\multimap}$ by $Tr^h_{k,l}(f, g) = (Tr(f), Tr(g))$. We have $Tr^h_{k,l}(f, g) : k \rightarrow l$ because



definition

h is strict

naturality

definition

The axioms of trace are trivially satisfied. $\square$

We shall note that this construction specializes to Conway operators (because the uniformity principles for traces and Conway operators agree): if $\mathcal{C}$ is a category with finite products and a Conway operator, so is $\mathcal{C}^{\multimap}$. Here is a classical example:

Example 5.2 (Scott Induction Principle) Let D be a ω-cpo with bottom $\perp$, and $f : D \to D$ be continuous. We write $\mathsf{fix}(f)$ for the least fixpoint of f. Let P be an inclusive (admissible) subset of D. If $x \in P$ implies $f(x) \in P$ and also $\perp \in P$, then $\mathsf{fix}(f) \in P$:

$$\begin{tikzcd} P \arrow[r, "\iota"] \arrow[d, "f|_P"'] & D \arrow[d, "f"] \\ P \arrow[r, "\iota"'] & D \end{tikzcd} \quad \textit{implies} \quad \begin{tikzcd} 1 \arrow[r, "="] \arrow[d, "\mathsf{fix}(f|_P)"'] & 1 \arrow[d, "\mathsf{fix}(f)"] \\ P \arrow[r, "\iota"'] & D \end{tikzcd}$$

where ι is the strict order monic associated to the inclusive subset. This can be seen an instance of the construction described above ($\mathbf{Cppo}^{\multimap}$). □

This example, although not new at all, gives a strong motivation to our study. It has been observed that many of the proof techniques on type theories like logical relations can be understood as model-construction techniques, for example categorical glueing or comma objects. The example above says that we can understand the Scott induction principle in this general context too, as a construction on traced monoidal categories.

5.2 Variations

We have seen that, in a particularly simple case, uniformity principle can be used for constructing new traced monoidal categories from known ones. Now the reader should be able to think of many variations of $\mathcal{C}^{\multimap}$: like comma categories, categories of algebras of endofunctors, as well as those of coalgebras – just by restricting the objects to be strict with respect to the trace.

Example 5.3 ($\mathbf{Cppo}_\perp$ from co-slice) It is an easy exercise to see $\mathbf{Cppo}_\perp$ as a traced full subcategory of the co-slice $1\backslash\mathbf{Cppo}$: its objects are strict maps from the one-point cpo 1 (hence the bottom elements), so arrows are precisely the bottom-preserving maps. The trace on $\mathbf{Cppo}_\perp$ is then inherited from **Cppo**. □

Example 5.4 (inserters) Let $F, G : \mathcal{C} \to \mathcal{D}$ be strict traced functors. We consider the following full subcategory of the inserter of F and G: its objects are pairs (C, h) where C is an object of $\mathcal{C}$ and $h : FC \to GC$ is a strict map in $\mathcal{D}$, while an arrow $f : (C, h) \to (C', h')$ is an arrow $f : C \to C'$ in $\mathcal{C}$ such that $h' \circ Ff = Gf \circ h$ holds. By repeating the same consideration as the case of $\mathcal{C}^{\multimap}$, this forms a traced monoidal category, with the trace inherited from $\mathcal{C}$. □

As special cases, we can apply these constructions on categories with finite products and a Conway operator, with strict maps w.r.t. the Conway operator (as we already did in Example 5.2). We already know that the resulting category is traced. If its monoidal product is cartesian, by the trace-fixpoint correspondence, we have a Conway operator on it.

6 Constructions as Enriched Limits

6.1 Some Attempts

Having observed these examples, it is then natural to ask if these constructions can be characterized by some suitable universal property. However, the category of traced monoidal categories and (strict) traced functors fails to have many interesting limits or colimits; even worse, this category is *not* monadic over **Cat** (in the sense that it is not monadic for the monad induced by the natural forgetful functor), although it is monadic over the category $\mathbf{SMCat}_s$ of small symmetric monoidal categories and strict symmetric monoidal functors (Martin Hyland and John Power, private communication). Thus this seems not the right setting to look at – in any case our constructions are in much more flavour of two-dimensional limits, and also there seems no way to accommodate the uniformity principle in this one-dimensional view.

Then a second and natural attempt is to look for a suitable enrichment, so that 2-cells somehow capture the strict maps (or natural transformations whose components are strict). As already warned before, strict maps do not form a category in general, so we cannot have a **Cat**-enrichment. However they do form *graphs* and it seems natural to consider the following **Graph**-enrichment on the category of traced monoidal categories and traced functors, for the cartesian closed category $\mathbf{Graph} \simeq \mathbf{Set}^{\cdot \rightrightarrows \cdot}$: each hom-set is equipped with a graph structure whose objects are traced functors, and arrows are monoidal natural transformations whose components are strict.

This *seems* to work well: for example, it is tempting to characterize $\mathcal{C}^{\multimap}$ as a **Graph**-cotensor of the graph $(\cdot \to \cdot)$ and the traced monoidal category $\mathcal{C}$ (see Appendix D for the notion of cotensors; for general enriched category theory see [10,4]). Thus we would like to claim

$$\mathbf{TreMon}(\mathcal{B}, \mathcal{C}^{\multimap}) \simeq [(\cdot \to \cdot), \mathbf{TreMon}(\mathcal{B}, \mathcal{C})]$$

where **TreMon** is the **Graph**-enriched category of traced monoidal categories and strict traced functors as described above.

Alas, there already exists a nasty difficulty even in this simple case. The problem is that strict maps in $\mathcal{C}^{\multimap}$ may *not* agree with those coming from strict maps in $\mathcal{C}$. This is very problematic, as morphisms in $\mathbf{TreMon}(\mathcal{B}, \mathcal{C}^{\multimap})$ depend on the strict maps in $\mathcal{C}^{\multimap}$, while those in $[(\cdot \to \cdot), \mathbf{TreMon}(\mathcal{B}, \mathcal{C})]$ depend just on the strict maps in $\mathcal{C}$.

And, unfortunately, there are counterexamples. Suppose that $\mathcal{C}$ is a traced monoidal category in which all strict maps are monic. Then it follows that $(f, g) : h \to h'$ in $\mathcal{C}^{\multimap}$ is strict whenever its second component g is strict in $\mathcal{C}$. For instance: the traced monoidal category generated from an object X, arrows $\mathsf{f}, \mathsf{g}, \mathsf{h} : X \to X$ with axioms $\mathsf{h} \circ \mathsf{h} \circ \mathsf{f} = \mathsf{g} \circ \mathsf{h} \circ \mathsf{h}$ and $Tr^X(k \circ \mathsf{h}) = Tr^X(\mathsf{h})$ for any $k : X \to X$. This has already been used as a case where strict maps do not compose – h is strict while $\mathsf{h} \circ \mathsf{h}$ is not strict. And in this category every morphism is monic. Therefore $(\mathsf{h} \circ \mathsf{h}, \mathsf{h}) : \mathsf{h} \to id_X$ is strict, although its first component is not.

6.2 A Solution

Perhaps the easiest way to remedy this is to explicitly specify a "monoidal" subgraph of strict maps (that is, a subgraph of strict maps which is closed under tensor product) for each traced monoidal category, and then give the enrichment with respect to such explicitly specified graphs of strict maps. For instance, given a traced monoidal category $\mathcal{C}$ with a monoidal subgraph $\mathcal{S}$ of strict maps of $\mathcal{C}$, we define $\mathcal{C}^{\multimap}$ as the full subcategory of $\mathcal{C}^{\to}$ whose objects are in $\mathcal{S}$, and we specify its monoidal subgraph $\mathcal{S}^{\multimap}$ as the class of strict maps whose components belong to $\mathcal{S}$.

Now we re-define **TreMon** as follows. Its objects are traced monoidal categories with a specified monoidal subgraph of strict maps. Arrows are strict traced functors. Its hom-graphs are defined as the previous version, except that we ask the each component of natural transformations stay in the specified monoidal subgraph of strict maps.

Then $(\mathcal{C}^{\multimap}, \mathcal{S}^{\multimap})$ is indeed the cotensor of the graph $(\cdot \to \cdot)$ and $(\mathcal{C}, \mathcal{S})$. In fact we have all **Graph**-cotensors:

Theorem 6.1 **TreMon** *is* **Graph**-*cotensored:*

$$\mathbf{TreMon}((\mathcal{B},\mathcal{U}), [\mathcal{G}, (\mathcal{C}, \mathcal{S})]) \simeq [\mathcal{G}, \mathbf{TreMon}((\mathcal{B},\mathcal{U}), (\mathcal{C}, \mathcal{S}))]$$

□

Explicitly, $[\mathcal{G}, (\mathcal{C}, \mathcal{S})]$ can be described as follows. Its objects are graph homomorphisms from $\mathcal{G}$ to $\mathcal{S}$. Arrows are transformations between graph homomorphisms. The symmetric monoidal structure is given by a pointwise manner, e.g. $I(X) = I$, $(F \otimes G)(X) = FX \otimes GX$ and so on. Given a transformation $\theta : F \otimes H \to G \otimes H$, we have its trace $Tr^H_{F,G}(\theta) : F \to G$ by $(Tr^H_{F,G}(\theta))_X = Tr^{HX}_{FX,GX}(\theta_X)$ (thanks to the uniformity). Finally, we specify the monoidal subgraph part of $[\mathcal{G}, (\mathcal{C}, \mathcal{S})]$ as the collection of strict maps whose components are all in $\mathcal{S}$ – in this way we exclude the nasty possibility mentioned before.

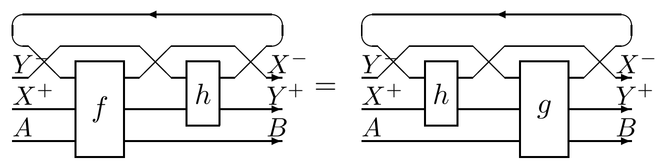
We believe that other constructions are naturally characterized as certain **Graph**-limits in this **TreMon**, though the details are yet to be spelled out. Also it still remains open how we can characterize the original constructions (without using specified classes of strict maps).

7 Strict Maps in $\mathbf{Int}\,\mathcal{C}$

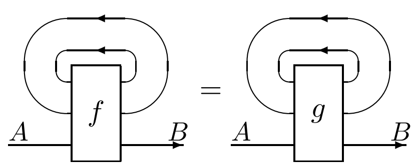
The **Int** construction, introduced in [9], turns a traced monoidal category $\mathcal{C}$ into a compact closed category $\mathbf{Int}\,\mathcal{C}$ to which $\mathcal{C}$ fully faithfully embeds (see Appendix E for a summary of the construction); its applications to computer science are studied, e.g., in [1,5]. It is natural to ask how the uniformity principles in $\mathcal{C}$ and in $\mathbf{Int}\,\mathcal{C}$ are related.

Unfortunately, the situation seems less clear than we first guess, and in this paper we can give only some elementary results and remarks. First, by a straightforward calculation, we have an obvious sort of characterization of strict maps in $\mathbf{Int}\,\mathcal{C}$ in terms of $\mathcal{C}$:

Proposition 7.1 *$h \in \mathbf{Int}\,\mathcal{C}((X^+, X^-), (Y^+, Y^-)) = \mathcal{C}(X^+ \otimes Y^-, Y^+ \otimes X^-)$ is strict in $\mathbf{Int}\,\mathcal{C}$ if and only if, for any $f \in \mathcal{C}(A \otimes X^+ \otimes X^-, B \otimes X^+ \otimes X^-)$ and $g \in \mathcal{C}(A \otimes Y^+ \otimes Y^-, B \otimes Y^+ \otimes Y^-)$,*



implies



□

From this characterization, it is immediate to see

Proposition 7.2 *If $h^+ \in \mathcal{C}(X^+, Y^+)$ and $h^- \in \mathcal{C}(Y^-, X^-)$ are strict in $\mathcal{C}$, then $h^+ \otimes h^- \in \mathcal{C}(X^+ \otimes Y^-, Y^+ \otimes X^-) = \mathbf{Int}\,\mathcal{C}((X^+, X^-), (Y^+, Y^-))$ is strict in $\mathbf{Int}\,\mathcal{C}$.* □

Therefore the canonical embedding from $\mathcal{C} \times \mathcal{C}^{\mathrm{op}}$ to $\mathbf{Int}\,\mathcal{C}$ preserves strict maps. However, we do not know much about strict maps in $\mathbf{Int}\,\mathcal{C}$ which do not arise in this way. We even do not know an answer for the following (seemingly easy) question:

> If $h \in \mathbf{Int}\,\mathcal{C}((X^+, X^-), (Y^+, Y^-))$ is strict in $\mathbf{Int}\,\mathcal{C}$, is h also strict in $\mathcal{C}$ (as a morphism from $X^+ \otimes Y^-$ to $Y^+ \otimes X^-$)?

Note that the converse does not hold. For instance, consider the symmetry $c_{I,X} \in \mathbf{Int}\,\mathcal{C}((I, I), (X, X)) = \mathcal{C}(I \otimes X, X \otimes I)$. It is an isomorphism (hence strict) in $\mathcal{C}$, but its strictness in $\mathbf{Int}\,\mathcal{C}$ implies $Tr^X(id_X) \otimes Tr^X(id_X) = id_I$, which is not true in general.

Acknowledgement

I thank John Power and Peter Selinger for helpful suggestions, and Gheorghe Ştefănescu for pointers to related work.

References

[1] Abramsky, S., *Retracing some paths in process algebra*, in "Proc. Concurrency Theory (CONCUR'96)", Springer Lecture Notes in Computer Science **1119** (1996), pp. 1–17.

[2] Arbib, A.M. and E.G. Manes, *Partially additive categories and flow diagram semantics*, J. Algebra **62** (1980), 203–227.

[3] Bloom, S. and Z. Ésik, "Iteration Theories", EATCS Monographs on Theoretical Computer Science, Springer-Verlag, 1993.

[4] Borceux, F., "Handbook of Categorical Algebra 2: Categories and Structures", Encyclopedia of Mathematics **51**, Cambridge University Press, 1994.

[5] Haghverdi, E., "A Categorical Approach to Linear Logic, Geometry of Interaction and Full Completeness", PhD thesis, University of Ottawa, 2000.

[6] Hasegawa, M., *Recursion from cyclic sharing: traced monoidal categories and models of cyclic lambda calculi*, in "Proc. Typed Lambda Calculi and Applications (TLCA'97)", Springer Lecture Notes in Computer Science **1210** (1997), pp. 196–213.

[7] Hasegawa, M., "Models of Sharing Graphs: A Categorical Semantics of let and letrec", PhD thesis ECS-LFCS-97-360, University of Edinburgh, 1997 / Distinguished Dissertation Series, Springer-Verlag, 1999.

[8] Joyal, A. and R. Street, *Geometry of tensor calculus I*, Adv. Math. **88** (1991), 55–113.

[9] Joyal, A., R. Street and D. Verity, *Traced monoidal categories*, Math. Proc. Cambridge Phil. Soc. **119**(3) (1996), 447–468.

[10] Kelly, G.M., "Basic Concepts of Enriched Category Theory", London Mathematical Society Lecture Note Series **64**, Cambridge University Press, 1982.

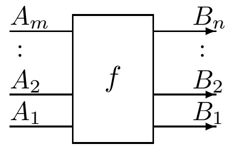
[11] Selinger, P., *Categorical structure of asynchronomy*, in "Proc. 15th Mathematical Foundations of Programming Semantics (MFPS)", Electronic Notes in Theoretical Computer Science **20** (1999).

[12] Simpson, A. and G. Plotkin, *Complete axioms for categorical fixed-point operators*, in "Proc. 15th Logic in Computer Science (LICS 2000)", pp. 30–41.

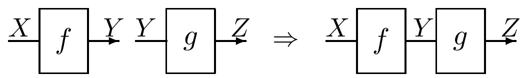
[13] Ştefănescu, G., "Network Algebra", Discrete Mathematics and Theoretical Computer Science Series, Springer-Verlag, 2000.

A Graphic Presentation of Monoidal Categories

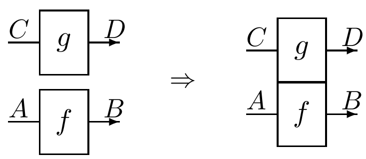
In this paper, an arrow $f : A_1 \otimes A_2 \otimes \ldots \otimes A_m \to B_1 \otimes B_2 \otimes \ldots \otimes B_n$ in a monoidal category (pretended as if it is strict for brevity) is drawn as (from left to right)



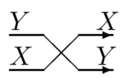
The identity arrow is drawn just as a straight line. The composition of $f : X \to Y$ and $g : Y \to Z$ is represented as a sequential composition



while the tensor $f \otimes g : A \otimes C \to B \otimes D$ of $f : A \to B$ and $g : C \to D$ is drawn as a parallel composition



The symmetry $c_{X,Y} : X \otimes Y \to Y \otimes X$ in a symmetric monoidal category is represented by a cross wiring:



For the correctness of these graphical presentations, see [8].

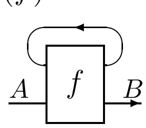
B Traced Monoidal Categories

A *trace* on a symmetric monoidal category $\mathcal{C}$ is a family of functions $Tr^X_{A,B} : \mathcal{C}(A \otimes X, B \otimes X) \to \mathcal{C}(A, B)$ subject to the following conditions:

- it is natural in A and B (*left/right tightening*), and dinatural in X (*sliding*)
- *vanishing*: $Tr^I_{A,B}(f) = f$ and $Tr^{X \otimes Y}_{A,B}(f) = Tr^X_{A,B}(Tr^Y_{A \otimes X, B \otimes X}(f))$
- *superposing:* $Tr^X_{C \otimes A, C \otimes B}(id_C \otimes f) = id_C \otimes Tr^X_{A,B}(f)$
- *yanking*: $Tr^X_{X,X}(c_{X,X}) = id_X$

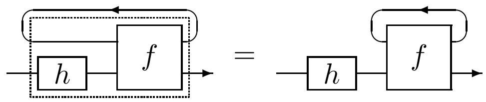
where, for brevity, we pretend as if $\otimes$ is strictly associative. A *traced symmetric monoidal category* is a symmetric monoidal category equipped with a (specified) trace – note that there can be many ways of giving traces.

Trace admits a natural graphical presentation as a "feedback": for $f : A \otimes X \to B \otimes X$, its trace $Tr^X_{A,B}(f) : A \to B$ can be drawn as

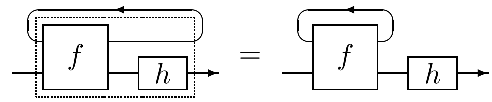


Using this notation, the axioms for trace given above can be graphically presented as follows.

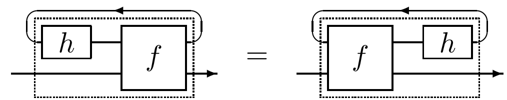
Naturality (Left Tightening)



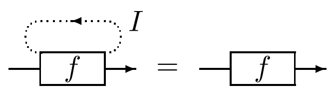
Naturality (Right Tightening)



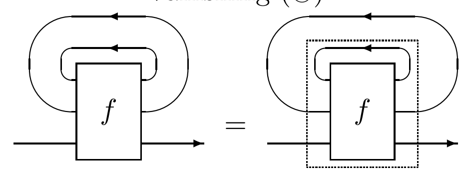
Dinaturality (Sliding)



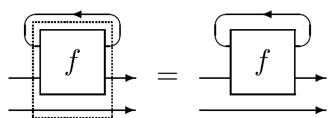
Vanishing (I)



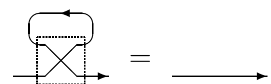
Vanishing ($\otimes$)



Superposing



Yanking



C Proof of Theorem 4.5

C.1 From the Trace-Strictness to the Conway-Strictness

Assume that the diagram

$$\begin{tikzcd} A \times X \arrow[r, "f"] \arrow[d, "A\times h"'] & X \arrow[d, "h"] \\ A \times Y \arrow[r, "g"'] & Y \end{tikzcd}$$

commutes and that h is strict w.r.t. the trace. Then the following diagram

$$\begin{tikzcd} A \times X \arrow[r, "(h\times X)\circ\Delta\circ f"] \arrow[d, "A\times h"'] & Y \times X \arrow[d, "Y\times h"] \\ A \times Y \arrow[r, "\Delta\circ g"'] & Y \times Y \end{tikzcd}$$

also commutes. From the uniformity for the trace, we have

$$Tr^X((h \times X) \circ \Delta \circ f) = Tr^Y(\Delta \circ g).$$

By Right Tightening, the left hand side is equal to $h \circ Tr^X(\Delta \circ f)$. Since $f^\dagger = Tr^X(\Delta \circ f)$ and $g^\dagger = Tr^Y(\Delta \circ g)$, we get $h \circ f^\dagger = g^\dagger$. Therefore h is strict w.r.t. the Conway operator.

C.2 From the Conway-Strictness to the Trace-Strictness

Assume that the diagram

$$\begin{tikzcd} A \times X \arrow[r, "f"] \arrow[d, "A\times h"'] & B \times X \arrow[d, "B\times h"] \\ A \times Y \arrow[r, "g"'] & B \times Y \end{tikzcd}$$

commutes and that h is strict w.r.t. the Conway operator. Then the following diagram

$$\begin{tikzcd} A \times B \times X \arrow[r, "f\circ(A\times\pi')"] \arrow[d, "A\times B\times h"'] & B \times X \arrow[d, "B\times h"] \\ A \times B \times Y \arrow[r, "g\circ(A\times\pi')"'] & B \times Y \end{tikzcd}$$

also commutes. Since h is strict w.r.t. the Conway operator, so is $B \times h$, by Lemma C.1 below. Thus we have

$$(B \times h) \circ (f \circ (A \times \pi'))^\dagger = (g \circ (A \times \pi'))^\dagger.$$

Since $Tr^X(f) = \pi \circ (f \circ (A \times \pi'))^\dagger$ and $Tr^Y(g) = \pi \circ (g \circ (A \times \pi'))^\dagger$, we get $Tr^X(f) = Tr^Y(g)$. Hence h is strict w.r.t. the trace.

Lemma C.1 *If $h : X \longrightarrow Y$ and $h' : X' \longrightarrow Y'$ are strict w.r.t. the Conway operator, so is $h \times h' : X \times X' \longrightarrow Y \times Y'$.*

Proof. Assume that the diagram

(C.1)
$$\begin{tikzcd} A \times X \times X' \arrow[r, "f"] \arrow[d, "A\times h\times h'"'] & X \times X' \arrow[d, "h\times h'"] \\ A \times Y \times Y' \arrow[r, "g"'] & Y \times Y' \end{tikzcd}$$

commutes. Our aim is to show $(h \times h') \circ f^\dagger = g^\dagger$. By the Bekič property (which holds for any Conway operator), this is equivalent to equations

(C.2) $$h \circ (f_1 \circ \langle A \times X, f_2^\dagger \rangle)^\dagger = (g_1 \circ \langle A \times Y, g_2^\dagger \rangle)^\dagger$$

(C.3) $$h' \circ (f'_2 \circ \langle A \times X', f'^\dagger_1 \rangle)^\dagger = (g'_2 \circ \langle A \times Y', g'^\dagger_1 \rangle)^\dagger$$

where $f_1 = \pi \circ f : A \times X \times X' \longrightarrow X$, $f_2 = \pi' \circ f : A \times X \times X' \longrightarrow X'$, $f'_i = f_i \circ (A \times c_{X',X})$, and so on. We shall show C.2. C.3 is proved in the same way.

By C.1, the diagrams

$$\begin{tikzcd} A \times X \times X' \arrow[r, "f_1"] \arrow[d, "A\times h\times h'"'] & X \arrow[d, "h"] \\ A \times Y \times Y' \arrow[r, "g_1"'] & Y \end{tikzcd} \tag{C.4}$$

$$\begin{tikzcd} A \times X \times X' \arrow[r, "f_2"] \arrow[d, "A\times X\times h'"'] & X' \arrow[d, "h'"] \\ A \times X \times Y' \arrow[r, "g_2\circ(A\times h\times Y')"'] & Y' \end{tikzcd} \tag{C.5}$$

commute. From C.5 and the strictness of h',

$$h' \circ f_2^\dagger = (g_2 \circ (A \times h \times Y'))^\dagger.$$

By naturality, the right hand side is equal to $g_2^\dagger \circ (A \times h)$. Thus we have a commutative diagram

$$\begin{tikzcd} A \times X \arrow[r, "f_2^\dagger"] \arrow[d, "A\times h"'] & X' \arrow[d, "h'"] \\ A \times Y \arrow[r, "g_2^\dagger"'] & Y' \end{tikzcd} \tag{C.6}$$

From C.4 and C.6,

$$\begin{tikzcd} A \times X \arrow[r, "\langle A\times X, f_2^\dagger\rangle"] \arrow[d, "A\times h"'] & A \times X \times X' \arrow[r, "f_1"] \arrow[d, "A\times h\times h'"] & X \arrow[d, "h"] \\ A \times Y \arrow[r, "\langle A\times Y, g_2^\dagger\rangle"'] & A \times Y \times Y' \arrow[r, "g_1"'] & Y \end{tikzcd}$$

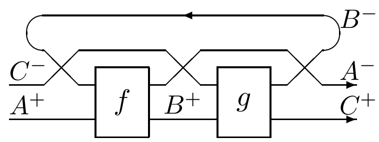
commutes. Since h is strict, we obtain C.2. $\square$

D Cotensors

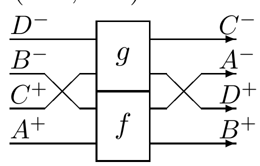
Let $\mathcal{V}$ be a symmetric monoidal closed category and $\mathcal{C}$ be a $\mathcal{V}$-category. We say the *cotensor* of $V \in \mathcal{V}$ and $C \in \mathcal{C}$ exists if there is an object $[V, C] \in \mathcal{C}$ together with isomorphisms $\mathcal{C}(D, [V, C]) \simeq [V, \mathcal{C}(D, C)]$ which are $\mathcal{V}$-natural in D in $\mathcal{C}$ (note that the square bracket in the right hand side is the internal hom of $\mathcal{V}$; indeed the internal hom can be regarded as the special case of cotensor with $\mathcal{C} = \mathcal{V}$). $\mathcal{C}$ is $\mathcal{V}$-*cotensored* when the cotensor of V and C exists for all V and C.

E The Int Construction

Let $\mathcal{C}$ be a traced monoidal category. The compact closed category $\mathbf{Int}\,\mathcal{C}$ is given as follows. Its objects are pairs of those of $\mathcal{C}$, and an arrow from (A^+, A^-) to (B^+, B^-) in $\mathbf{Int}\,\mathcal{C}$ is an arrow from $A^+ \otimes B^-$ to $B^+ \otimes A^-$. The identity arrow on (A^+, A^-) is $id_{A^+\otimes A^-} \in \mathcal{C}(A^+ \otimes A^-, A^+ \otimes A^-)$. The composition of $f \in \mathbf{Int}\,\mathcal{C}((A^+, A^-), (B^+, B^-)) = \mathcal{C}(A^+ \otimes B^-, B^+ \otimes A^-)$ and $g \in \mathbf{Int}\,\mathcal{C}((B^+, B^-), (C^+, C^-)) = \mathcal{C}(B^+ \otimes C^-, C^+ \otimes B^-)$ is given by



The unit of the monoidal structure is (I, I). The tensor product on objects is $(A^+, A^-) \otimes (B^+, B^-) = (A^+ \otimes B^+, A^- \otimes B^-)$, and on arrows $f : (A^+, A^-) \to (B^+, B^-)$ and $g : (C^+, C^-) \to (D^+, D^-)$ we have



The symmetry from $(A^+, A^-) \otimes (B^+, B^-)$ to $(B^+, B^-) \otimes (A^+, A^-)$ in $\mathbf{Int}\,\mathcal{C}$ is $c_{A^+,B^+} \otimes c_{A^-,B^-}$ in $\mathcal{C}$. The duality $(-)^* : (\mathbf{Int}\,\mathcal{C})^{\mathrm{op}} \to \mathbf{Int}\,\mathcal{C}$ is given by $(A^+, A^-)^* = (A^-, A^+)$ and $f^* = c_{B^+,A^-} \circ f \circ c_{B^-,A^+}$ for $f : (A^+, A^-) \to (B^+, B^-)$. The unit and counit

$$\eta_{(A^+,A^-)} : (I, I) \to (A^+, A^-) \otimes (A^+, A^-)^* = (A^+ \otimes A^-, A^- \otimes A^+)$$

$$\varepsilon_{(A^+,A^-)} : (A^+, A^-)^* \otimes (A^+, A^-) = (A^- \otimes A^+, A^+ \otimes A^-) \to (I, I)$$

are given by the suitable isomorphisms in $\mathcal{C}$.

Like any compact closed category, $\mathbf{Int}\,\mathcal{C}$ has a unique trace, called *canonical trace* in [9]. To be explicit, for $f : (A^+, A^-) \otimes (X^+, X^-) \to (B^+, B^-) \otimes (X^+, X^-)$, its trace $Tr^{(X^+,X^-)}(f) : (A^+, A^-) \to (B^+, B^-)$ is given as