Hypergraphs for Compact Closed Categories

Alessandro Di Giorgio
Tallinn University of Technology, Estonia

Callum Reader
Tallinn University of Technology, Estonia

Abstract

In recent years a succession of papers have taken to representing string diagrams as hypergraphs, the advantage of which is that the structural equations of diagrams come for free from the hypergraph structure. This improves implementability and reduces the number of rewrites necessary for reasoning.

Notably, however, string diagrams for compact closed categories have escaped this treatment. In this paper we introduce a combinatorial interpretation of compact closed string diagrams, via the Int construction on hypergraphs for traced string diagrams. Using this interpretation, we characterise string diagram rewriting as a suitable adaptation of double-pushout rewriting.

2012 ACM Subject Classification Theory of computation → Rewrite systems; Theory of computation → Equational logic and rewriting

Keywords and phrases rewriting, compact closed categories, string diagrams

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.50

Funding *Alessandro Di Giorgio*: Supported by the European Union under Grant Agreement No. 101087529.
Callum Reader: Supported by the Estonian Research Council via grant PRG3215.

1 Introduction

String diagrams [20, 25, 29] are a sound and complete graphical notation for reasoning in symmetric monoidal categories (SMCs). In this language, morphisms are represented as boxes with incoming and outgoing wires; composition $(;)$ corresponds to concatenation of diagrams along the wires and the monoidal product $(\otimes)$ to juxtaposition of diagrams. As an example, given $f\colon A \otimes B \to C \otimes D$ and $g\colon D \to E$, the composite $f;(\mathrm{id}_C \otimes g)$ is represented as the first diagram on the left of Figure 1. The key insight of this formalism is that topologically equivalent diagrams represent equal morphisms, allowing complex equational reasoning to be reduced to intuitive diagrammatic manipulations. For example, the first two diagrams on the left of Figure 1 can be seen to be equal by crossing g with a wire.

On top of this general principle, one typically equips diagrams with additional structure and equations that capture the specific behavior of the system being modeled. This approach has proven particularly valuable across diverse fields, including quantum computing [14], concurrency theory [10], programming [5, 16, 18], logic [6], control theory [11], electrical circuits [4] and linguistics [27], among others.

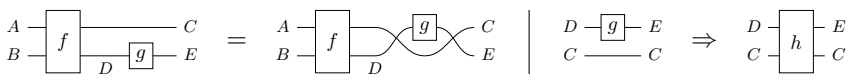
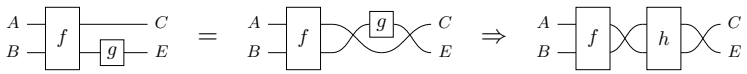


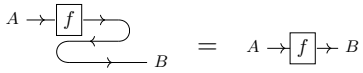
Figure 1 Two equivalent string diagrams (left) and a rewrite rule (right).

Given the compositional nature of diagrams, equational reasoning proceeds by identifying a sub-diagram matching the left-hand side of a rewrite rule and substituting it with the right-hand side. However, such a match may only become apparent after deforming the original diagram. For instance, consider the rewrite rule shown on the right of Figure 1 and observe that the first diagram below can be rewritten into the third only after a deformation.



From a practical point of view, this approach is infeasible: each rewrite step would require searching through equivalence classes of diagrams to find potential matches.

This problem is addressed by interpreting string diagrams as certain *hypergraphs* and their rewriting as *double-pushout rewriting* (DPO) of such hypergraphs. This approach has been successfully applied to string diagrams for SMCs [8] and various extensions [7, 17, 3, 24, 31, 15]. Of particular relevance to this paper is the hypergraph representation for traced monoidal categories [17], which we use as a basis for a combinatorial representation of *compact closed* categories [22]. Compact closed categories arise naturally in the study of systems with duality, providing the categorical framework for quantum protocols [2, 28], modelling reversible and bidirectional computation [13], as well as concurrent computation [1], and describing grammatical structure in natural language semantics [27]. Graphically, the duality is represented by allowing directed wires to bend, introducing additional topological deformations beyond those available in SMCs. For example, double bends in wires can be straightened, as shown below.



In this paper, we propose a combinatorial representation of compact closed string diagrams via the Int construction [21] applied to the traced hypergraphs of [17]. This construction yields a hypergraph representation that captures deformations such as the one above on the nose. We show that rewriting of compact closed string diagrams can be characterized as double-pushout rewriting on these hypergraphs.

Synopsis. Section 2 recalls the syntactic definition of string diagrams for traced monoidal and compact closed categories freely generated from a monoidal signature. Section 3 recalls the interpretation of traced string diagrams as partial monogamous cospans of hypergraphs. Section 4 uses the Int construction to define the hypergraph interpretation for compact closed string diagrams, and shows that this characterizes the free compact closed category on a monoidal signature. Finally, Section 5 leverages the interpretation to establish a correspondence between string diagram rewriting and hypergraph rewriting in the compact closed setting.

Notation. We denote by MonSig the category of monoidal signatures and their morphisms; by TrCat the 2-category of traced monoidal categories, traced monoidal functors and monoidal natural transformations between them; and by CompCat the 2-category of compact closed categories strong monoidal functors and monoidal natural transformations between them.

$$\begin{array}{c} id_X; t = t; id_Y \qquad \mathrm{id}_I \otimes t = t = t \otimes \mathrm{id}_I \\ (t_1; t_2); t_3 = t_1; (t_2; t_3) \quad (t_1 \otimes t_2) \otimes t_3 = t_1 \otimes (t_2 \otimes t_3) \\ (t_1; t_2) \otimes (t_3; t_4) = (t_1 \otimes t_3); (t_2 \otimes t_4) \\ (t \otimes \mathrm{id}_Z); \sigma_{Y,Z} = \sigma_{X,Z}; (\mathrm{id}_Z \otimes t) \\ \sigma_{A,B}; \sigma_{B,A} = \mathrm{id}_A \otimes \mathrm{id}_B \end{array} \qquad \begin{array}{c} \mathrm{tr}_Z((\mathrm{id}_Z \otimes t_1); t_2; (\mathrm{id}_Z \otimes t_3)) = t_1; \mathrm{tr}_Z(t_2); t_3 \\ \mathrm{tr}_Z(t_1 \otimes t_2) = \mathrm{tr}_Z(t_1) \otimes t_2 \\ \mathrm{tr}_W(\mathrm{tr}_Z(t)) = \mathrm{tr}_{ZW}(t) \quad \mathrm{tr}_I(t) = t \\ \mathrm{tr}_W(t_1; (t_2 \otimes id_Y)) = \mathrm{tr}_Z((t_2 \otimes \mathrm{id}_X); t_1) \\ \mathrm{tr}_A(\sigma_{A,A}) = \mathrm{id}_A \end{array}$$

Figure 2 Axioms of Symmetric Monoidal (left) and Traced Monoidal Categories (right).

2 String Diagrammatic Syntax

In traditional algebraic theories, terms are generated from a set of operations $f(x_1, \ldots, x_n)$, each with an arity n specifying the number of arguments it requires. Such terms can be depicted as syntax trees, with variables at the leaves and operations at the nodes. Monoidal theories generalize this framework by allowing operations to have multiple outputs – that is, a *coarity* in addition to their arity. This generalization moves from the cartesian setting of algebraic theories to the broader framework of symmetric monoidal categories, where terms are naturally represented as string diagrams rather than trees.

As for terms of an algebraic theory, string diagrams can be defined inductively starting from a signature, that is a pair of sets of basic types and operations.

▶ **Definition 1.** *A* monoidal signature $\Sigma = (\Sigma_0, \Sigma_1, ar, coar)$ *consists of: a set* Σ_0 *of* sorts*, a set* Σ_1 *of* generators *and functions* $ar, coar \colon \Sigma_1 \to \Sigma_0^\star$*, that assign to each generator in* Σ_1 *an* arity *and* coarity*, respectively, in* Σ_0^**, the free monoid on* Σ_0*.*

For convenience, we focus on string diagrams for traced monoidal categories, which are a central tool for this paper.

▶ **Definition 2.** *Given a monoidal signature* Σ*,* $\mathrm{Tr}(\Sigma)$ *is the traced monoidal category having as objects elements of* $\Sigma_0^\star$ *and as morphisms the terms generated by the following rules*

$$\mathrm{id}_I \colon I \to I \qquad \frac{A \in \Sigma_0}{\mathrm{id}_A \colon A \to A} \qquad \frac{A, B \in \Sigma_0}{\sigma_{A,B} \colon AB \to BA} \qquad \frac{f \in \Sigma_1 \quad ar(f) = X \quad coar(f) = Y}{f \colon X \to Y}$$

$$\frac{t_1 \colon X \to Y \quad t_2 \colon Y \to Z}{t_1; t_2 \colon X \to Z} \qquad \frac{t_1 \colon X \to Y \quad t_2 \colon Z \to W}{t_1 \otimes t_2 \colon XZ \to YW} \qquad \frac{t \colon ZX \to ZY}{\mathrm{tr}_Z(t) \colon X \to Y}$$

and quotiented by the axioms in Figure 2.

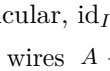
$\mathrm{Tr}(\Sigma)$ is not only traced monoidal, it is actually the *free* traced monoidal category on a monoidal signature, in the sense of Joyal and Street [20].

▶ **Definition 3.** *Let* $\mathscr{C}$ *be a monoidal category, whose objects we denote as* $\mathscr{C}_0$*. An* interpretation *of a monoidal signature* Σ *into* $\mathscr{C}$ *consists of a function* $\mathrm{I}_0 \colon \Sigma_0 \to \mathscr{C}_0$ *which extends uniquely to* $\hat{\mathrm{I}}_0 \colon (\Sigma_0)^* \to \mathscr{C}_0$ *with* $\hat{\mathrm{I}}_0(X_1 \ldots X_n) = \hat{\mathrm{I}}_0(X_1) \otimes \cdots \otimes \hat{\mathrm{I}}_0(X_n)$ *and* $\hat{\mathrm{I}}_0(I) = I$*, and, for any* $f \in \Sigma_1$*, a morphism* $\mathrm{I}_1(f) \colon \hat{\mathrm{I}}_0(ar(f)) \to \hat{\mathrm{I}}_0(coar(f))$*.*

Interpretations form a category $[\Sigma, \mathscr{C}]$ *(see [20]). Given a monoidal functor* $\mathcal{F} \colon \mathscr{C} \to \mathscr{D}$ *and an interpretation* $\mathrm{I} \in [\Sigma, \mathscr{C}]$*, we can define a composite interpretation* $\mathcal{F} \circ \mathrm{I} \in [\Sigma, \mathscr{D}]$*. For a given* I*, this precomposition induces a functor* $\mathrm{I}^* \colon \mathrm{MonCat}(\mathscr{C}, \mathscr{D}) \to [\Sigma, \mathscr{D}]$*.*

▶ **Definition 4.** *Let* Σ *be a monoidal signature and* $\mathscr{C}$ *be a traced category. Then a traced category* $\mathscr{F}(\Sigma)$ *is said to be free if there is an interpretation* $\mathrm{I} \in [\Sigma, \mathscr{F}(\Sigma)]$ *such that* $\mathrm{I}^* \colon \mathrm{TrMon}(\mathscr{F}(\Sigma), \mathscr{C}) \to [\Sigma, \mathscr{C}]$ *gives an equivalence of categories, for every* $\mathscr{C}$*. We call such an interpretation the free interpretation.*

▶ **Theorem 5.** *The functor* $\mathrm{Tr}\colon \mathrm{MonSig} \to \mathrm{TrCat}$ *gives the free traced monoidal category up to isomorphism.*

Terms of $\mathrm{Tr}(\Sigma)$ can be depicted with the usual boxes-and-wires notation of string diagrams. In particular, id_I is drawn as the empty diagram  , while identities id_A correspond to labelled wires  , symmetries $\sigma_{A,B}$ are represented as wire crossings  and generators $f\colon A_1 \dots A_n \to B_1 \dots B_m$ appear as boxes  with wires on their left and right sides corresponding to the arity and coarity, respectively. Composition $t_1; t_2$ is obtained by gluing wires together  , while the monoidal product $t_1 \otimes t_2$ corresponds to juxtaposition  and the trace $\mathrm{tr}_Z(t)$ to a loop  . Most axioms in Figure 2 are absorbed by the graphical notation, the remaining ones correspond to topological deformations.

As we recall in the next section, an equivalent, combinatorial, characterization of the free traced monoidal category over Σ was given in [17] using certain hypergraphs. Our goal is to provide an analogous characterisation for $\mathrm{CC}(\Sigma)$, the free compact closed category over Σ.

▶ **Definition 6.** *Given a monoidal signature* Σ, $\mathrm{CC}(\Sigma)$ *is the compact closed category having as objects the elements of the free involutive monoid on* Σ_0 *and as morphisms the terms generated by the rules in Definition 2, excluding the rule for trace, and the following additional rules – where* $(-)^*$ *is the involution on* Σ_0 *– for cups and caps morphisms*

$$\frac{A \in \Sigma_0}{\mathrm{cup}_A\colon I \to A^* \otimes A} \qquad \frac{A \in \Sigma_0}{\mathrm{cap}_A\colon A \otimes A^* \to I}$$

quotiened by the axioms of SMCs, on the left of Figure 2, and the yanking axioms:

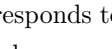
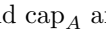
$$(\mathrm{id}_A \otimes \mathrm{cup}_A); (\mathrm{cap}_A \otimes \mathrm{id}_A) = \mathrm{id}_A \qquad (\mathrm{cup}_A \otimes \mathrm{id}_{A^*}); (\mathrm{id}_{A^*} \otimes \mathrm{cap}_A) = \mathrm{id}_{A^*}.$$

▶ Remark 7. Note that $\mathrm{CC}(\Sigma)$ is generated from a monoidal signature and *not* a compact closed signature, or autonomous signature as defined in, for example, [29]. This is, in essence, because any compact closed category generated from a compact closed signature, is strong monoidally equivalent to one generated from a monoidal signature.

▶ **Definition 8.** *Let* Σ *be a monoidal signature and* $\mathscr{C}$ *be a compact closed category. Then a compact closed category* $\mathscr{F}(\Sigma)$ *is said to be free if there is an interpretation* $\mathrm{I} \in [\Sigma, \mathscr{F}(\Sigma)]$ *such that* $\mathrm{I}^*\colon \mathrm{CompCat}(\mathscr{F}(\Sigma), \mathscr{C}) \to [\Sigma, \mathscr{C}]$ *gives an equivalence of categories, for every* $\mathscr{C}$*. We call such an interpretation the free interpretation.*

▶ **Proposition 9.** *The functor* $\mathrm{CC}\colon \mathrm{MonSig} \to \mathrm{CompCat}$ *gives the free compact closed category up to isomorphism.*

Proof. Straightforward induction. This is analogous to the proof in Section 4 of [22]. ◀

The graphical representation of morphisms of $\mathrm{CC}(\Sigma)$ is analogous to the one for $\mathrm{Tr}(\Sigma)$, however there are some differences. First, since objects have duals, wires have a direction: id_A corresponds to  , while id_{A^*} to  . Moreover, the additional morphisms cup_A and cap_A are represented as bending of wires, respectively as and . The yanking axioms correspond to straightening double bends, e.g.  .

▶ Remark 10. Note that, although $\mathrm{CC}(\Sigma)$ lacks a primitive trace operator like $\mathrm{Tr}(\Sigma)$, every compact closed category has a trace, derived from cups and caps as follows:  .

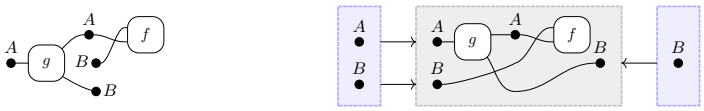


Figure 3 A Σ-labelled hypergaph (left) and a Σ-labelled hypergraph with interfaces (right).

3 Hypergraphs for Traced Monoidal Categories

In this section we recall the hypergraph interpretation of traced string diagrams from [17]. First, we give a definition of hypergraphs whose nodes and edges are labelled by the symbols of a monoidal signature Σ.

▶ **Definition 11.** *Let Σ be a monoidal signature. A Σ-labelled hypergraph G is a tuple (N, E, l_N, l_E, s, t) consisting of:*

- *finite sets of nodes N and edges E;*
- *source and target functions $s, t \colon E \to N^\star$, mapping an edge to a finite list of vertices;*
- *a node labelling function $l_N \colon N \to \Sigma_0$, assigning to each node in N a sort in Σ_0;*
- *an edge labelling function $l_E \colon E \to \Sigma_1$, assigning to each edge in E a generator in Σ_1, in a way that is compatible with the type of the generator.*

For two Σ-labelled hypergraphs $G = (N^G, E^G, l_N^G, l_E^G, s^G, t^G)$ and $H = (N^H, E^H, l_N^H, l_E^H, s^H, t^H)$, a hypergraph morphism $f \colon G \to H$ consists of a pair of functions $f_N \colon N^G \to N^H$ and $f_E \colon E^G \to E^H$, such that the following commutes

$$\begin{array}{ccccc} (N^G)^\star & \xleftarrow{s^G} & E^G & \xrightarrow{t^G} & (N^G)^\star \\ {\scriptstyle f_N^\star}\downarrow & & \downarrow{\scriptstyle f_E} & & \downarrow{\scriptstyle f_N^\star} \\ (N^H)^\star & \xleftarrow{s^H} & E^H & \xrightarrow{t^H} & (N^H)^\star \end{array}$$

where $f_N^\star$ is the inductive extension of f_N. Σ-labelled hypergraphs and their morphisms form a category.

The hypergraph on left of Figure 3 is an example of Σ-labelled hypergraph, where Σ is the monoidal signature with $\Sigma_0 = \{A, B\}$ and $\Sigma_1 = \{f \colon AB \to I, g \colon A \to AB\}$. Nodes are depicted as labelled black dots and edges as labelled boxes. For the sake of brevity, from now on we will refer to Σ-labelled hypergraphs simply as hypergraphs.

Note that hypergraphs do not have clear left and right interfaces, as opposed to string diagrams. To recover this feature, we consider *discrete cospans* of hypergraphs $X \to G \leftarrow Y$, where X and Y have empty sets of edges. An example is given on the right of Figure 3.

▶ **Definition 12.** *Discrete cospans of hypergraphs form a symmetric monoidal category* $\mathrm{CSpan}(\Sigma)$. *Composition $(X \to G \leftarrow Y); (Y \to H \leftarrow Z)$ is given by pushout of the span formed by the middle legs, while the monoidal product $(X \to G \leftarrow Y) \otimes (Z \to H \leftarrow W)$ is defined via the coproduct in the category of hypergraphs, which is given by disjoint union.*

▶ Remark 13. It is worth remarking that objects of $\mathrm{CSpan}(\Sigma)$ are equipped with a Frobenius algebra [12]. As such, $\mathrm{CSpan}(\Sigma)$ is *self-dual* compact closed, thus object and their duals are identified.

We are interested only in a certain class of hypergraphs with interfaces, namely all and only those representing traced string diagrams. In order to give a definition for them, we recall the notion of degree of a node.

▶ **Definition 14.** *Given a hypergraph G, and a node v of G, its degree,* $\deg(v)$ *is a tuple* (i, o) *where i denotes the number of edges with v in their source, and o denotes the number of edges with v in their target.*

▶ **Definition 15.** *A discrete cospan of hypergraphs,* $X \xrightarrow{f} G \xleftarrow{g} Y$*, is called* partial monogamous *if f and g are both monomorphisms and, for all nodes v of G,* $\deg(v)$ *is*

$(0,0)$	*if* $v \in \mathrm{im}(f) \cap \mathrm{im}(g)$	$(0,1)$	*if* $v \in \mathrm{im}(f) \cap \mathrm{im}(g)^c$
$(1,0)$	*if* $v \in \mathrm{im}(f)^c \cap \mathrm{im}(g)$	$(0,0)$ *or* $(1,1)$	*otherwise.*

Partial monogamy is a rather technical condition characterising which hypergraphs are meaningful in the traced setting. In particular, it allows self-loops over identities to be represented as single disconnected nodes (for the details see Definition 3.20 in [17]).

▶ **Definition 16.** *Given a monoidal signature Σ, we define* $\mathrm{PMon}(\Sigma)$ *to be the subcategory of* $\mathrm{CSpan}(\Sigma)$ *consisting of only discrete partial monogamous cospans of hypergraphs.*

Note that partial monogamy is preserved by pushout and the tensor product, and the identity cospan is partial monogamous. Hence $\mathrm{PMon}(\Sigma)$ is indeed a monoidal category. It is, furthermore, traced [17] and, in fact, the *free* traced monoidal category on Σ, in the sense of Definition 4.

Note here that we are not asking for an isomorphism as in, for example, [29], but an equivalence, as in [20]. This leads to a sort of normal form, see Remark 26 for details.

▶ **Theorem 17.** $\mathrm{PMon}(\Sigma)$ *is the free traced monoidal category over Σ.*

Proof. This follows from Corollary 3.32 of [17] we simply define the interpretation $\mathrm{I}_{\mathrm{Tr}} \colon \Sigma \to \mathrm{PMon}(\Sigma)$ to be the isomorphism from $\mathrm{Tr}(\Sigma)$ but restricted to its generators. ◀

4 Hypergraphs for Compact Closed Categories

In this section we develop a combinatorial characterization of the free compact closed category over a monoidal signature. Our approach uses the Int construction [21], which builds a canonical compact closed category from any traced monoidal category. We define our compact closed category of hypergraphs as the result of applying the Int construction to partial monogamous hypergraphs, and prove that it is indeed the free compact closed category on the signature.

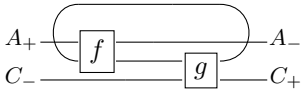
We then give an explicit account of this category of hypergraphs and its structure. Since both the category of string diagrams $\mathrm{CC}(\Sigma)$, and the category of hypergraphs $\mathrm{Int}(\mathrm{PMon}(\Sigma))$ are free compact closed categories we immediately have a strong monoidal equivalence between the two categories. This equivalence gives us the interpretation functor, which we describe explictly. It is worth noting here that our category of hypergraphs is only free *up to equivalence*. We will see that, as a consequence of this, translating between the term language and hypergraphs results in a sort of normal form for the string diagram language.

4.1 Hypergraphs for compact closed categories, via the Int construction

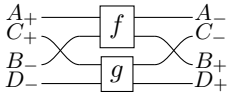
The Int construction gives the free compact closed category on a traced category. Intuitively, it "reshapes" the traced category in order to obtain the compact closed structure.

▶ **Definition 18** (Int construction)**.** *Given a traced monoidal category $\mathscr{C}$, the compact closed category* $\mathrm{Int}(\mathscr{C})$ *is defined as follows:*

- *An object in* $\mathrm{Int}(\mathscr{C})$ *is given by a pair* (A_+, A_-) *of objects in* $\mathscr{C}$*;*
- *A morphism* $(A_+, A_-) \xrightarrow{f} (B_+, B_-)$ *in* $\mathrm{Int}(\mathscr{C})$ *is given by a morphism* $\begin{smallmatrix} A_+ \\ B_- \end{smallmatrix} \boxed{f} \begin{smallmatrix} A_- \\ B_+ \end{smallmatrix}$ *in* $\mathscr{C}$*;*
- *Identities* $(A_+, A_-) \xrightarrow{id} (A_+, A_-)$ *are given by identities* $\mathrm{id}_{A_+ \otimes A_-}$ *in* $\mathscr{C}$*;*
- *Composition* $(A_+, A_-) \xrightarrow{f} (B_+, B_-) \xrightarrow{g} (C_+, C_-)$ *is given by the following diagram in* $\mathscr{C}$*;*

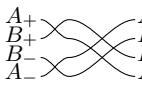


- *The monoidal product is defined pointwise on objects* $(A_+, A_-) \otimes (B_+, B_-) \coloneqq (A_+ \otimes B_+, A_- \otimes B_-)$ *and on morphisms by the following diagram in* $\mathscr{C}$



with (I, I) *as the monoidal unit;*

- *Symmetries* $(A_+, A_-) \otimes (B_+, B_-) \xrightarrow{\sigma} (B_+, B_-) \otimes (A_+, A_-)$ *are given by appropriate compositions of symmetries in* $\mathscr{C}$



;

- *Duals on objects are defined as* $(A_+, A_-)^* \coloneqq (A_-, A_+)$ *and on morphisms as*

$$\left(\begin{smallmatrix} A_+ \\ B_- \end{smallmatrix} \boxed{f} \begin{smallmatrix} A_- \\ B_+ \end{smallmatrix}\right)^* \coloneqq \begin{smallmatrix} B_- \\ A_+ \end{smallmatrix} \boxed{f} \begin{smallmatrix} B_+ \\ A_- \end{smallmatrix} .$$

Accordingly, cups and caps correspond, respectively, to $\begin{smallmatrix} A_+ & A_- \\ A_- & A_+ \end{smallmatrix}$ *and* $\begin{smallmatrix} A_- & A_+ \\ A_+ & A_- \end{smallmatrix}$ *in* $\mathscr{C}$*.*

This construction induces a 2-functor $\mathrm{Int}\colon \mathrm{TrMon} \to \mathrm{CompCat}$, where traced monoidal functors are applied pointwise to each morphism of $\mathrm{Int}(\mathscr{C})$ and similarly for monoidal natural transformations.

▶ **Theorem 19** ([21]). *The* Int *construction 2-functor is left 2-adjoint to the forgetful 2-functor from the 2-category of compact closed categories, to the 2-category of traced categories.*

$$\mathrm{TrCat} \underset{U_{\mathrm{Tr}}}{\overset{\mathrm{Int}}{\underset{\longleftarrow}{\overset{\longrightarrow}{\bot}}}} \mathrm{CompCat}$$

The forgetful functor is an embedding of 2-categories, in that

$$\mathrm{CompCat}(\mathscr{C}, \mathscr{D}) \cong \mathrm{TrMon}(U_{\mathrm{Tr}}(\mathscr{C}), U_{\mathrm{Tr}}(\mathscr{D})).$$

There is a slight mismatch of abstraction here, since we wish to form a functor out of the category of monoidal signatures, but monoidal signatures only form a 1-category. We will then, by abuse of notation, use Int to denote both the 2-functor and its underlying functor.

▶ **Definition 20.** *We define the functor* $\mathrm{CCHyp}\colon \mathrm{MonSig} \to \mathrm{CompCat}$ *to be the composite* $\mathrm{Int} \circ \mathrm{PMon}$.

▶ **Theorem 21.** *The functor* $\mathrm{CCHyp}\colon \mathrm{MonSig} \to \mathrm{CompCat}$ *gives the free compact closed category up to equivalence.*

Proof. Note that an interpretation into a traced category or a compact closed category is, by definition an interpretation into its underlying symmetric monoidal category.

Let $\mathrm{I}_{\mathrm{Tr}} \in [\Sigma, \mathrm{PMon}(\Sigma)]$ be the free interpretation and let η be the unit of the 2-adjunction in Theorem 19. Then we have a new monoidal interpretation given by the composite interpretation $\mathrm{I}_{\mathrm{CC}} \coloneqq \eta \circ \mathrm{I}_{\mathrm{Tr}} \colon \Sigma \to \mathrm{Int} \circ \mathrm{PMon}(\Sigma)$. For a given compact closed $\mathscr{C}$, precomposition by I_{CC} gives the following commuting diagram where the centre composite is an equivalence.

$$\begin{array}{ccc} \mathrm{CompCat}(\mathrm{Int} \circ \mathrm{PMon}(\Sigma), \mathscr{C}) & \xrightarrow{U_{\mathrm{Tr}}} & \mathrm{TrMon}(U_{\mathrm{Tr}} \circ \mathrm{Int} \circ \mathrm{PMon}(\Sigma), U_{\mathrm{Tr}}\mathscr{C}) \\ \downarrow{\scriptstyle \mathrm{I}^*_{\mathrm{CC}}} & \overset{\sim}{\searrow} & \downarrow{\scriptstyle \eta^*} \\ [\Sigma, \mathscr{C}] & \underset{\mathrm{I}^*_{\mathrm{Tr}}}{\overset{\sim}{\longleftarrow}} & \mathrm{TrMon}(\mathrm{PMon}(\Sigma), U_{\mathrm{Tr}}(\mathscr{C})) \end{array}$$

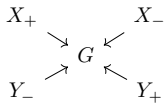
Hence we have that $\mathrm{CCHyp}(-)$ is indeed the free functor up to equivalence. ◀

In Subsection 4.3 we will see how to interpret string diagrams of $\mathrm{CC}(\Sigma)$ as morphisms in $\mathrm{CCHyp}(\Sigma)$. First, however, we give an explicit account of $\mathrm{CCHyp}(\Sigma)$.

4.2 Hypergraphs for compact closed categories, concretely

Here we unpack the definition of the Int construction to give a concrete account of $\mathrm{CCHyp}(\Sigma)$. Understanding the definition of composition requires some work, as such we leave this to a final proposition.

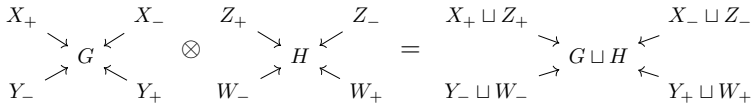
- The objects are given by pairs of Σ_0-labelled sets (X_+, X_-), where here we think of X_+ as containing the strings that flow from left-to-right and X_- as containing the strings that flow from right-to-left.
- The morphisms are given by cospans with four legs:



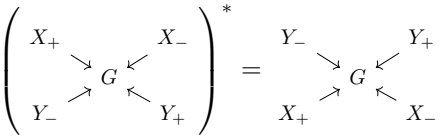
  such that when "read from left-to-right" – that is, when thought of as a cospan

$$X_+ \sqcup Y_- \to G \leftarrow X_- \sqcup Y_+$$

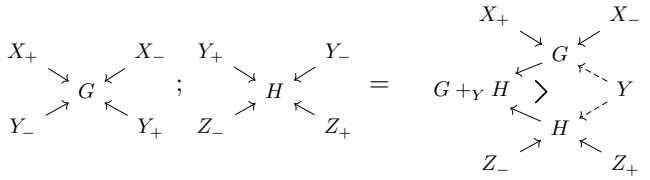
  they are partial monogamous.
- The monoidal product is given "pointwise" by disjoint union of hypergraphs. That is to say: $(X_+, X_-) \otimes (Z_+, Z_-) = (X_+ \sqcup Z_+, X_- \sqcup Z_-)$, and



- The dual of (X_+, X_-) is (X_-, X_+) and the dual of a four-legged cospan is the same four-legged cospan, but with its legs reflected in the horizontal axis:

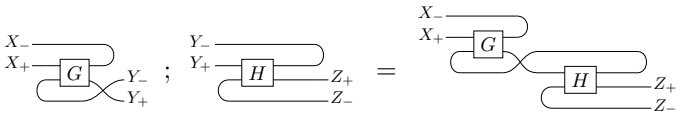


▶ **Proposition 22.** *Given two composable morphisms in* $\mathrm{CCHyp}(\Sigma)$ *as below, then the composite is given "top-to-bottom" by taking a pushout over* $Y = Y_+ \sqcup Y_-$*:*

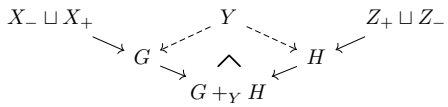


Proof. In essence, this follows from the fact that $\mathrm{PMon}(\Sigma)$ is a subcategory of $\mathrm{CSpan}(\Sigma)$. Since $\mathrm{PMon}(\Sigma)$ inherits its composition from $\mathrm{CSpan}(\Sigma)$, which is self-dual compact closed, we can use this structure to reinterpret the composition.

Suppose we have the two cospans above, G and H, viewed as cospans in $\mathrm{CSpan}(\Sigma)$. For simplicity, we represent them as string diagrams and consider the following composite

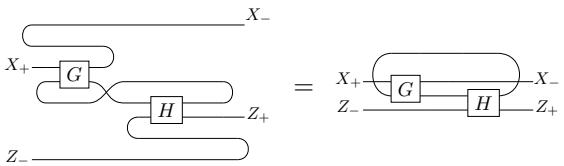


This is exactly the composite given by taking the pushout in the following diagram.



Using the self-dual compact closed structure of $\mathrm{CSpan}(\Sigma)$ this is in natural correspondence with the cospan $X_+ \sqcup Z_- \to G +_B H \leftarrow X_- \sqcup Z_+$.

This correspondence is given by pre- and postcomposing with cups and caps and as in the string diagram on the left below, but this is equal to the diagram on the right by yanking and symmetries.



Hence this is equal to the composite in the Int construction, and since the trace and symmetries preserve partial monogamy [17], the resulting composite must be partial monogamous. ◀

4.3 Interpreting string diagrams as hypergraphs

In order to give an interpretation functor from the term language to the language of hypergraphs, we will show that the term language is also free. We can then utilise the bicategorical Yoneda lemma to prove that, firstly there exists an interpretation functor, and secondly, that this interpretation functor gives an equivalence of compact closed categories. We firstly give an account of the category whose morphisms are given by the term language.

▶ **Theorem 23.** *There is a strong monoidal equivalence* $\mathrm{CC}(\Sigma) \simeq \mathrm{CCHyp}(\Sigma)$.

Proof. Note, for any compact closed $\mathscr{C}$, we have three equivalent functors, as given by our freeness results: Proposition 9 and Theorem 21.

$$\begin{array}{ccc} & \overset{\mathrm{CompCat}(\mathrm{CC}(-),\mathscr{C})}{\cong} & \\ \mathrm{MonSig}^{\mathrm{op}} & \xrightarrow{\quad [-,\mathscr{C}] \quad} & \mathrm{Cat} \\ & \underset{\mathrm{CompCat}(\mathrm{CCHyp}(-),\mathscr{C})}{\simeq} & \end{array}$$

If we think of MonSig as being a locally discrete 2-category then we may think of these functors as 2-presheaves. As such, the bicategorical Yoneda lemma (see [19], for example) tells us that we have a natural equivalence $\mathrm{CC}(-) \simeq \mathrm{CCHyp}(-)$. ◀

We want, not only to know that these functors are strong monoidal equivalent, but also to have an explicit characterisation of this equivalence. The construction of this equivalence is given via the bicategorical Yoneda lemma by considering the transpose of the identity functor, as follows.

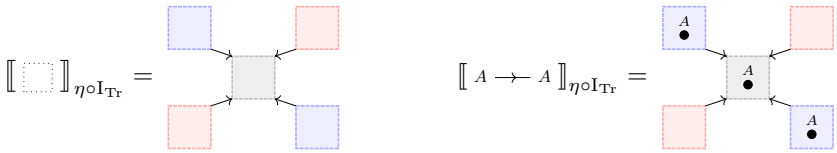
$$\begin{array}{ccccc} \mathrm{CompCat}(\mathrm{CCHyp}(\Sigma), \mathrm{CCHyp}(\Sigma)) & \xrightarrow{(\eta \circ \mathrm{I}_{\mathrm{Tr}})^*} & [\Sigma, \mathrm{CCHyp}\Sigma] & \xrightarrow{\sim} & \mathrm{CompCat}(\mathrm{CC}(\Sigma), \mathrm{CCHyp}(\Sigma)) \\ \mathrm{id} & \longmapsto & \eta \circ \mathrm{I}_{\mathrm{Tr}} & \longmapsto & [\![-]\!]_{\eta \circ \mathrm{I}_{\mathrm{Tr}}} \end{array}$$

Thus the interpretation functor, which gives the equivalence, is just given by the inductive extension $[\![-]\!]_{\eta \circ \mathrm{I}_{\mathrm{Tr}}} : \mathrm{CC}(\Sigma) \to \mathrm{CCHyp}(\Sigma)$.

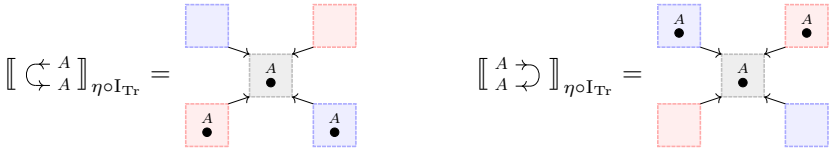
Explicitly, this means the following. For any list of sorts, $A_1 \dots A_n \in (\Sigma_0)^\star$, we have

$$[\![A_1 \dots A_n]\!]_{\eta \circ \mathrm{I}_{\mathrm{Tr}}} = (\mathrm{I}_{\mathrm{Tr}}(A_1 \dots A_n), \varnothing) = (\{0 \mapsto A_1, \dots, n \mapsto A_n\}, \varnothing).$$

The identity on the monoidal unit, and on some $A \in \Sigma_0$ are interpreted as follows:

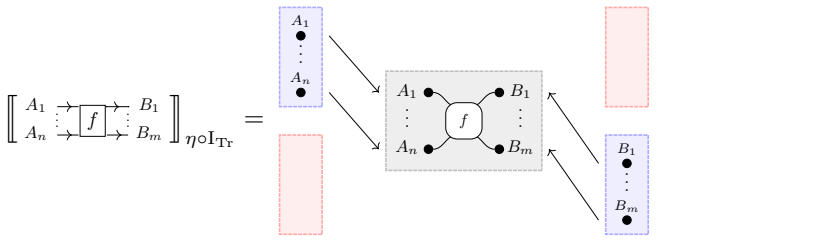


Unsurprisingly, cups and caps are interpreted similarly to identities:

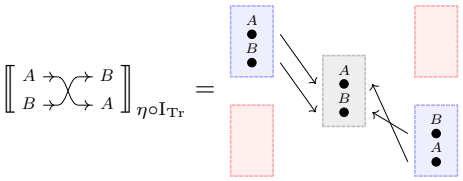


However, note how in the visual representation for morphisms of $\mathrm{CCHyp}(\Sigma)$, the left-to-right inputs of the string diagram are sent on the top-left of the four-legged cospan, and the left-to-right outputs on the bottom-right; similarly, the right-to-left inputs are sent on the top-right and the right-to-left outpus on the bottom-left.

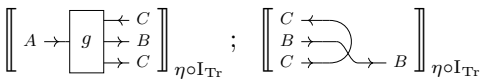
For any generator $f \colon A_1 \dots A_n \to B_1 \dots B_m \in \Sigma_1$, we have:



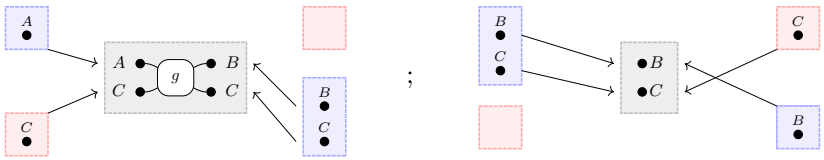
For any $A, B \in \Sigma_0$, the symmetry $\Sigma_{A,B}$ is interpreted as follows.



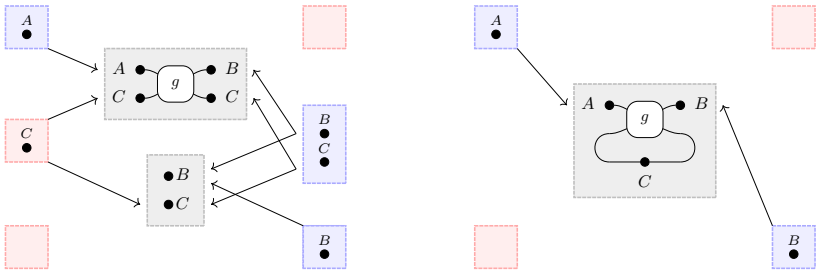
▶ **Example 24.** For the sake of illustration, let us consider how the following composition is computed in CCHyp(Σ).



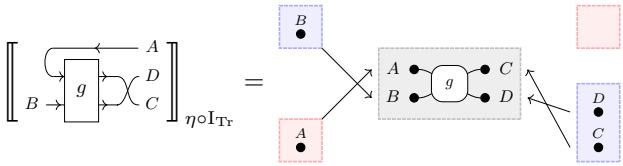
After interpretation this corresponds to the following composite



Based on the description of composition given in Proposition 22, the gluing process proceeds as follows. Firstly we swap the top two legs of the second cospan, then we stack the two cospans on top of one another this gives us the diagram below on the left. Secondly we identify along their mappings, which gives us the diagram below on the right.



▶ **Example 25.** Below we give another example of a diagram which includes both cups and symmetries.



▶ Remark 26. There is a slight subtlety here in how duals are interpreted. Whilst here we define cospans of hypergraphs to be built from finite *sets*, in practice it is typical to think of them as being built from finite cardinals so that cospans of hypergraphs form a coloured PROP. For the sake of implementation this is much simpler – it gives us an obvious choice of ordering to keep track of the monoidal product.

If we define our partial monogamous cospans in this way, however, then after applying the Int construction, the interpretation functor actually creates a sort of normal form. Consider, for example, the interpretation applied to the following two objects.

$$[\![A_1 A_2^* A_3]\!]_{\eta \circ \mathrm{I}_{\mathrm{Tr}}} = (\{0 \mapsto A_1, 1 \mapsto A_3\}, \{0 \mapsto A_2\}) = [\![A_1 A_3 A_2^*]\!]_{\eta \circ \mathrm{I}_{\mathrm{Tr}}}$$

Since the interpretation functor identifies them, the auto-equivalence induced on $\mathrm{CC}(\Sigma)$ will also identify these two objects.

If we want to avoid this, we can simply define the interpretation functor to be such that its image is pairs (A_+, A_-) such that A_+ and A_- form a partitioned cardinal. In other words we define the objects of $\mathrm{CCHyp}(\Sigma)$ to be a PROP coloured in $\Sigma_0 \times \{+, -\}$. In such a case we would have the following distinct interpretations.

$$[\![A_1 A_2^* A_3]\!]_{\eta \circ \mathrm{I}_{\mathrm{Tr}}} = (\{0 \mapsto A_1, 2 \mapsto A_3\}, \{1 \mapsto A_2\})$$

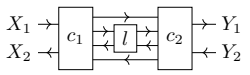
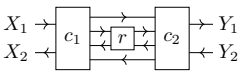
$$[\![A_1 A_3 A_2^*]\!]_{\eta \circ \mathrm{I}_{\mathrm{Tr}}} = (\{0 \mapsto A_1, 1 \mapsto A_2\}, \{2 \mapsto A_3\})$$

This actually turns our equivalence into an isomorphism, although we do not prove that here.

5 Rewriting of Compact Closed String Diagrams

As mentioned in the introduction, rewriting of string diagrams can be practically infeasible. The same also holds for the compact closed case, where the yanking equations introduce additional structural equivalences on the diagrams. In this section, we first give a definition of rewriting for morphisms in $\mathrm{CC}(\Sigma)$, and then, leveraging on the interpretation defined in Section 4, we show that it corresponds to double-pushout rewriting of hypergraphs in $\mathrm{CCHyp}(\Sigma)$.

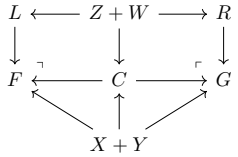
▶ **Definition 27** (String diagram rewriting). *Let $f, g \colon X_1 \otimes X_2^* \to Y_1 \otimes Y_2^*$ and $l, r \colon Z_1 \otimes Z_2^* \to W_1 \otimes W_2^*$ be pairs of morphisms in $\mathrm{CC}(\Sigma)$. We say that f rewrites into g according to the rule $\langle l, r \rangle$ – denoted as $f \to_{\langle l,r \rangle} g$ – if in $\mathrm{CC}(\Sigma)$*

$f =$  *and* $g =$  .

The interpretation $[\![-]\!]_{\eta \circ \mathrm{I}_{\mathrm{Tr}}}$ – for the sake of brevity, simply $[\![-]\!]$ – maps diagrams in $\mathrm{CC}(\Sigma)$ to hypergraphs in $\mathrm{CCHyp}(\Sigma)$, which, in turn, are represented by morphisms of $\mathrm{PMon}(\Sigma)$. Therefore, we now recall the notion of double-pushout rewriting in $\mathrm{PMon}(\Sigma)$.

▶ **Definition 28** (DPO rewriting). *Let $F, G \colon X \to Y$ and $L, R \colon Z \to W$ be pairs of morphisms in $\mathrm{PMon}(\Sigma)$. We say that F rewrites into G according to the rule $\langle L, R \rangle$ – denoted as $F \Rightarrow_{\langle L,R \rangle} G$ – if:*

1. *there is $Z + W \xrightarrow{[c_2, d_1]} C \xleftarrow{[d_2, c_1]} X + Y$ in $\mathrm{PMon}(\Sigma)$, with c_1 and c_2 mono, such that*
2. *the diagram below commutes and the two marked squares are pushouts.*



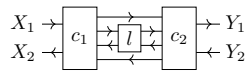
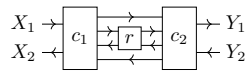
Given that hypergraphs in $\mathrm{CCHyp}(\Sigma)$ are traced hypergraphs of a certain shape, the rewriting of the former is "simulated" by rewriting of the latter. In particular, rewriting happens on the *underlying hypergraph* that represents a morphism.

▶ **Definition 29.** *Let $F \colon (X_+, X_-) \to (Y_+, Y_-)$ be a morphism in $\mathrm{CCHyp}(\Sigma)$, its* underlying hypergraph *– denoted $\lfloor F \rfloor$ – is the corresponding cospan $X_+ \sqcup Y_- \to \lfloor F \rfloor \leftarrow X_- \sqcup Y_+$ in $\mathrm{PMon}(\Sigma)$.*

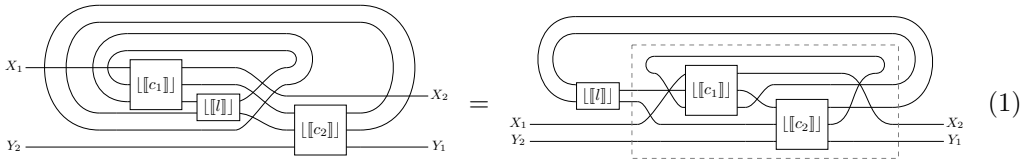
▶ **Definition 30.** *Let $F, G\colon (X_+, X_-) \to (Y_+, Y_-)$ and $L, R\colon (Z_+, Z_-) \to (W_+, W_-)$ be pairs of morphisms in* $\mathrm{CCHyp}(\Sigma)$*. We say that F rewrites into G according to the rule $\langle L, R\rangle$ – denoted as $F \rightsquigarrow_{\langle L,R\rangle} G$ – if $\lfloor F \rfloor \Rightarrow_{\langle \lfloor L \rfloor, \lfloor R \rfloor\rangle} \lfloor G \rfloor$.*

▶ **Theorem 31.** *For any pairs of morphisms $f, g\colon X_1 \otimes X_2^* \to Y_1 \otimes Y_2^*$ and $l, r\colon Z_1 \otimes Z_2^* \to W_1 \otimes W_2^*$ in* $\mathrm{CC}(\Sigma)$ *we have that $f \to_{\langle l,r\rangle} g$ if and only if $[\![f]\!] \rightsquigarrow_{\langle [\![l]\!], [\![r]\!]\rangle} [\![g]\!]$.*

Proof. For the only if part, assume that $f \to_{\langle l,r\rangle} g$, and thus, by Definition 27, we have that

$f =$  and $g =$  .

We need to show that $[\![f]\!] \rightsquigarrow_{\langle [\![l]\!], [\![r]\!]\rangle} [\![g]\!]$ which, by Definition 30, it amounts to the DPO rewrite $\lfloor [\![f]\!] \rfloor \Rightarrow_{\langle \lfloor [\![l]\!] \rfloor, \lfloor [\![r]\!] \rfloor\rangle} \lfloor [\![g]\!] \rfloor$ of the underlying hypergraphs. However, by Theorem 5.24 in [17], the latter is equivalent to the rewrite of the corresponding traced string diagrams. Therefore it is enough to observe that the partial monogamous cospan $\lfloor [\![f]\!] \rfloor$ (resp. $\lfloor [\![g]\!] \rfloor$) corresponds to the string diagram on the left below in $\mathrm{Tr}(\Sigma)$. By the axioms of traced monoidal categories this is equivalent to the string diagram on the right.

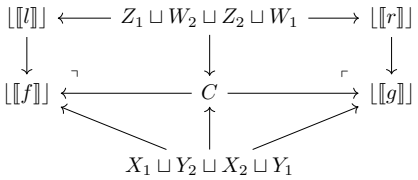


(1)

The dashed box highlights a context for a traced string diagram rewrite, as required by Definition 5.1 in [17]. In turn, by Theorem 5.24 in [17], we have a DPO rewrite of the corresponding hypergraphs.

For the other direction, suppose that $[\![f]\!] \rightsquigarrow_{\langle [\![l]\!], [\![r]\!]\rangle} [\![g]\!]$ which, by Definition 30, corresponds to $\lfloor [\![f]\!] \rfloor \Rightarrow_{\langle \lfloor [\![l]\!] \rfloor, \lfloor [\![r]\!] \rfloor\rangle} \lfloor [\![g]\!] \rfloor$. Then there exists a cospan in $\mathrm{PMon}(\Sigma)$, on the left below, such that diagram on the right is a DPO diagram.

$$Z_2 \sqcup W_1 \sqcup X_1 \sqcup Y_2 \to C \leftarrow Z_1 \sqcup W_2 \sqcup X_2 \sqcup Y_1$$



By Lemma 5.22 in [17], we have that $X_1 \sqcup Y_2 \to \lfloor [\![f]\!] \rfloor \leftarrow X_2 \sqcup Y_1$ can be factored as

$$\mathrm{tr}_{Z_1 \sqcup W_2}\left(\begin{array}{c} Z_1 \sqcup W_2 \to \lfloor [\![l]\!] \rfloor \leftarrow Z_2 \sqcup W_1 \\ \otimes \\ X_1 \sqcup Y_2 \to X_1 \sqcup Y_2 \leftarrow X_1 \sqcup Y_2 \end{array} ; Z_2 \sqcup W_1 \sqcup X_1 \sqcup Y_2 \to C \leftarrow Z_1 \sqcup W_2 \sqcup X_2 \sqcup Y_1 \right).$$

By analogous reasoning to the one above, we can decompose C as on the right of (1), and the whole cospan as on the left (1). By fullness of $\mathrm{I}_{\mathrm{Tr}}(-)$ and therefore of $[\![-]\!]$, we get a diagram in the shape of f (resp. g). ◀

6 Conclusions, Related and Future Work

In this work, we extended to the compact closed setting a line of research on string diagram rewriting initiated in [7, 8] and further developed in [17, 3, 24, 31, 15]. Our contributions are twofold. First, Theorem 21 provides a characterisation of compact closed string diagrams

as a certain kind of hypergraphs with interfaces. Second, Theorem 31 shows that, in the compact closed setting, string diagram rewriting can be realised as double pushout rewriting on the corresponding hypergraphs.

Beyond the theoretical contribution of offering an alternative presentation of freely generated compact closed categories, this work introduces a concrete combinatorial data structure suitable for implementations of string diagrammatic reasoning. This opens the possibility to extending existing diagrammatic proof assistants, such as [23], to the compact closed case.

As related work, it is worth mentioning that our interpretation in terms of hypergraphs complements the topological interpretation given as oriented 1-cobordisms in [30].

As future work, we plan to study properties of the hypergraph rewriting system, in particular *confluence* [9], with an emphasis on specific compact closed theories. Promising applications may arise in areas such as bidirectional programming [13] and natural language semantics [27]. Another compelling direction is the investigation of rewriting modulo additional structure, for instance in dagger compact closed categories [28], appearing in the context of quantum computation. Finally, we believe that this work lays the groundwork for a combinatorial characterisation of string diagrams for closed symmetric monoidal categories, recently introduced by the authors in [26]. Such a characterisation would provide valuable tools for studying rewriting properties of the linear λ-calculus and of the simply typed λ-calculus in the cartesian setting.

References

1 Samson Abramsky. Interaction categories. In *Theory and Formal Methods 1993: Proceedings of the First Imperial College Department of Computing Workshop on Theory and Formal Methods, Isle of Thorns Conference Centre, Chelwood Gate, Sussex, UK, 29–31 March 1993*, pages 57–69. Springer, 1993.

2 Samson Abramsky and Bob Coecke. A categorical semantics of quantum protocols. In *Proceedings of the 19th Annual IEEE Symposium on Logic in Computer Science, 2004.*, pages 415–425. IEEE, 2004. doi:10.1109/LICS.2004.1319636.

3 Mario Alvarez-Picallo, Dan Ghica, David Sprunger, and Fabio Zanasi. Rewriting for Monoidal Closed Categories. In *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*, volume 228 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 29:1–29:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022. doi:10.4230/LIPIcs.FSCD.2022.29.

4 Guillaume Boisseau and Pawel Sobocinski. String diagrammatic electrical circuit theory. *arXiv preprint*, 2021. arXiv:2106.07763.

5 Filippo Bonchi, Alessandro Di Giorgio, and Elena Di Lavore. A diagrammatic algebra for program logics. In *International Conference on Foundations of Software Science and Computation Structures*, pages 308–330. Springer Nature Switzerland Cham, 2025. doi:10.1007/978-3-031-90897-2_15.

6 Filippo Bonchi, Alessandro Di Giorgio, Nathan Haydon, and Pawel Sobocinski. Diagrammatic algebra of first order logic. In *Proceedings of the 39th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 1–15, 2024. doi:10.1145/3661814.3662078.

7 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski, and Fabio Zanasi. String diagram rewrite theory I: Rewriting with Frobenius structure. *Journal of the ACM (JACM)*, 69(2):1–58, 2022. doi:10.1145/3502719.

8 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Pawel Sobocinski, and Fabio Zanasi. String diagram rewrite theory II: Rewriting with symmetric monoidal structure. *Mathematical Structures in Computer Science*, 32(4):511–541, 2022. doi:10.1017/S0960129522000317.

9 Filippo Bonchi, Fabio Gadducci, Aleks Kissinger, Paweł Sobociński, and Fabio Zanasi. String diagram rewrite theory III: Confluence with and without frobenius. *Mathematical Structures in Computer Science*, 32(7):829–869, 2022. doi:10.1017/S0960129522000123.

10 Filippo Bonchi, Joshua Holland, Robin Piedeleu, Paweł Sobociński, and Fabio Zanasi. Diagrammatic algebra: from linear to concurrent systems. *Proceedings of the ACM on Programming Languages*, 3(POPL):1–28, 2019. doi:10.1145/3290338.

11 Filippo Bonchi, Paweł Sobociński, and Fabio Zanasi. A categorical semantics of signal flow graphs. In *International Conference on Concurrency Theory*, pages 435–450. Springer, 2014.

12 Aurelio Carboni and Robert FC Walters. Cartesian bicategories I. *Journal of pure and applied algebra*, 49(1-2):11–32, 1987.

13 Chao-Hong Chen and Amr Sabry. A computational interpretation of compact closed categories: reversible programming with negative and fractional types. *Proc. ACM Program. Lang.*, 5(POPL), January 2021. doi:10.1145/3434290.

14 Bob Coecke and Aleks Kissinger. Picturing quantum processes: A first course on quantum theory and diagrammatic reasoning. In *International conference on theory and application of diagrams*, pages 28–31. Springer, 2018. doi:10.1007/978-3-319-91376-6_6.

15 Alessandro Di Giorgio, Dan R Ghica, and Fabio Zanasi. Rewriting for traced monoidal closed categories. In *International Conference on Graph Transformation*, pages 24–43. Springer, 2025. doi:10.1007/978-3-031-94706-3_2.

16 Dan Ghica and Fabio Zanasi. String diagrams for λ-calculi and functional computation. *arXiv preprint*, 2023. arXiv:2305.18945.

17 Dan R. Ghica and George Kaye. Rewriting modulo traced comonoid structure. *Logical Methods in Computer Science*, Volume 22, Issue 1, January 2026. doi:10.46298/lmcs-22(1:2)2026.

18 Alan Jeffrey. Premonoidal categories and a graphical view of programs. *Preprint, Dec*, 1997.

19 Niles Johnson and Donald Yau. *2-Dimensional Categories*. Oxford University Press, January 2021. doi:10.1093/oso/9780198871378.001.0001.

20 André Joyal and Ross Street. The geometry of tensor calculus, i. *Advances in Mathematics*, 88(1):55–112, 1991. doi:10.1016/0001-8708(91)90003-P.

21 André Joyal, Ross Street, and Dominic Verity. Traced monoidal categories. *Mathematical Proceedings of the Cambridge Philosophical Society*, 119(3):447–468, 1996. doi:10.1017/S0305004100074338.

22 G.M. Kelly and M.L. Laplaza. Coherence for compact closed categories. *Journal of Pure and Applied Algebra*, 19:193–213, 1980. doi:10.1016/0022-4049(80)90101-2.

23 Aleks Kissinger. Chyp: Composing hypergraphs, proving theorems, 2023.

24 Aleksandar Milosavljevic, Robin Piedeleu, and Fabio Zanasi. Rewriting for symmetric monoidal categories with commutative (co)monoid structure. *Logical Methods in Computer Science*, Volume 21, Issue 1, January 2025. doi:10.46298/lmcs-21(1:12)2025.

25 Robin Piedeleu and Fabio Zanasi. *An Introduction to String Diagrams for Computer Scientists*. Cambridge University Press, 2025.

26 Callum Reader and Alessandro Di Giorgio. String Diagrams for Closed Symmetric Monoidal Categories. In Stefano Guerrini and Barbara König, editors, *34th EACSL Annual Conference on Computer Science Logic (CSL 2026)*, volume 363 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:23, Dagstuhl, Germany, 2026. Schloss Dagstuhl – Leibniz-Zentrum für Informatik. doi:10.4230/LIPIcs.CSL.2026.12.

27 Mehrnoosh Sadrzadeh, Stephen Clark, and Bob Coecke. The frobenius anatomy of word meanings I: Subject and object relative pronouns. *Journal of Logic and Computation*, 23(6):1293–1317, 2013. doi:10.1093/LOGCOM/EXT044.

28 Peter Selinger. Dagger compact closed categories and completely positive maps. *Electronic Notes in Theoretical computer science*, 170:139–163, 2007.

29 Peter Selinger. *A Survey of Graphical Languages for Monoidal Categories*, pages 289–355. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011. doi:10.1007/978-3-642-12821-9_4.

30 David I. Spivak, Patrick Schultz, and Dylan Rupel. String diagrams for traced and compact categories are oriented 1-cobordisms. *Journal of Pure and Applied Algebra*, 221(8):2064–2110, 2017. doi:10.1016/j.jpaa.2016.10.009.

31 Aleksei Tiurin, Chris Barrett, Dan R Ghica, and Nick Hu. Equivalence hypergraphs: Dpo rewriting for monoidal e-graphs. In *2025 40th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 209–222. IEEE, 2025. doi:10.1109/LICS65433.2025.00023.