Rewriting modulo traced comonoid structure

Dan R. Ghica
School of Computer Science
University of Birmingham, UK
d.r.ghica@bham.ac.uk

George Kaye
School of Computer Science
University of Birmingham, UK
g.j.kaye@pgr.bham.ac.uk

Abstract

In this paper we adapt previous work on rewriting string diagrams using hypergraphs to the case where the underlying category has a *traced comonoid structure*, in which wires can be forked and the outputs of a morphism can be connected to its input. Such a structure is particularly interesting because any traced Cartesian (dataflow) category has an underlying traced comonoid structure. We show that certain subclasses of hypergraphs are fully complete for traced comonoid categories: that is to say, every term in such a category has a unique corresponding hypergraph up to isomorphism, and from every hypergraph with the desired properties, a unique term in the category can be retrieved up to the axioms of traced comonoid categories. We also show how the framework of double pushout rewriting (DPO) can be adapted for traced comonoid categories by characterising the valid pushout complements for rewriting in our setting. We conclude by presenting a case study in the form of recent work on an equational theory for *sequential circuits*: circuits built from primitive logic gates with delay and feedback. The graph rewriting framework allows for the definition of an *operational semantics* for sequential circuits.

1 Introduction

String diagrams constitute a useful and elegant conceptual bridge between term rewriting and graph rewriting. We will not reprise here, for lack of space, the by now impressive body of theoretical and applied work for and with string diagrams but will take it as a given. The survey [Sel11] is a suitable starting point into the literature.

The purpose of this paper is to support reasoning (via graph rewrite) in *traced categories with a comonoid structure* but without a monoid structure. Prior art on this topic exists [Kis12; DK13], but it is based on the framework of 'framed point graphs' which requires rewriting modulo so called *wire homeomorphisms*. This style of rewriting is awkward and is increasingly considered as obsolete as compared to more recent work on rewriting with hypergraphs [Bon+22a; Bon+22b; Bon+22c], possibly modulo *Frobenius* structure. The variation to just a (co)monoid structure ('half a Frobenius') has been studied as well [FL22; MZ22].

The study of rewriting for traced categories with a comonoid structure is motivated by an important application, *dataflow categories* [CŞ90; CŞ94; Has97], which represent a categorical foundation for the semantics of digital circuits [GKS22]. It is also technically challenging, as it falls in a gap between compact closed structures constructible via Frobenius and symmetric monoidal categories (without trace) so 'off the shelf' solutions cannot be currently used. In fact the gap between the kind of semantic models which use an underlying compact closed structure and those which use a traced monoidal structure is significant: the former have a *relational* nature with subtle causality (e.g. quantum or electrical circuits) whereas the latter are *functional* with clear input-output causality (e.g. digital or logical circuits) so it is not surprising that the underlying rewrite frameworks should differ.

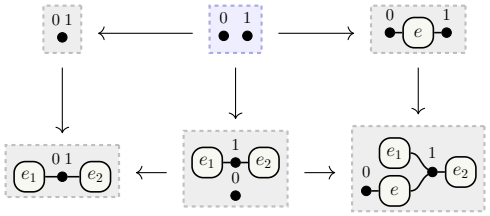
A key feature of compact closed categories is that the Cartesian product, if it exists, is degenerate and identified with the co-product. Even without invoking copying, we will see how trying to perform rewriting in a traced category with a comonoid structure can also lead to inconsistencies. This is a firm indication that a bespoke rewriting framework needs to be constructed to fill this particular situation.

Contributions. This paper makes two distinct technical contributions. The first is to show that one subclass of cospans of hypergraphs ('partial monogamous') are fully complete for traced terms, and another class ('partial left-monogamous') fully complete for traced comonoid terms. The challenge is not so much in proving the correctness of the construction but in defining precisely what these combinatorial structures should be. In particular, the extremal point of tracing the identity: $\mathrm{Tr}\left(\boxplus\right) =$, corresponding graphically to a closed loop, provides a litmus test. The way this is resolved must

be robust enough to handle the addition of the comonoid structure, which graphically corresponds to 'tracing a forking wire' $\mathrm{Tr}\left(\right) =$.

When rewriting with Frobenius using double pushout (DPO) rewriting, every pushout complement is valid [Bon+22a] whereas when rewriting with symmetric monoidal structure exactly one pushout complement is valid [Bon+22b]. For the traced case some pushout complements are valid and some are not. The second contribution is to characterise the valid pushout complements as 'traced boundary complements'.

This is best illustrated with an example in which there is a pushout complement that is valid in a Frobenius setting because it uses the monoid structure, but it is not valid neither in a traced, nor even in a traced comonoid setting. Imagine we have a rule $\langle —, -\boxed{e}- \rangle$ and a term $\boxed{e_1}\boxed{e_2}$, and rewrite it as follows.



This corresponds to the term rewrite $\boxed{e_1}\boxed{e_2} =$ $=$, which holds in a Frobenius setting but not a setting without a commutative monoid structure. On the other hand, the rewriting system for symmetric monoidal categories [Bon+22b] is too restrictive as it enforces that any matching must be mono: this prevents matchings such as $-\boxed{e}-$ in . Here again the challenge is precisely identifying the concept of traced boundary complement mathematically. The solution, although not immediately obvious, is not complicated, again requiring a generalisation from monogamy to partial monogamy.

2 Monoidal theories and hypergraphs

When modelling a system using monoidal categories, its components and properties are specified using a *monoidal theory*. A class of SMCs particularly interesting to us is that of *PROPs* [Mac65] ('categories of *PRO*ducts and *P*ermutations'), which have natural numbers as objects and addition as tensor product.

Definition 1 (Symmetric monoidal theory)**.** *A* (single-sorted) symmetric monoidal theory (*SMT*) *is a tuple* $(\Sigma, \mathcal{E})$ *where* Σ *is a set of* generators *with associated* (*co*)*arities in* $\mathsf{dom}(,)\mathsf{cod}(\in)\mathbb{N}$ *and* $\mathcal{E}$ *is a set of equations. Given a SMT* $(\Sigma, \mathcal{E})$*, let* $\mathbf{S}_\Sigma$ *be the symmetric monoidal category freely generated over* Σ *and let* $\mathbf{S}_{\Sigma,\mathcal{E}}$ *be* $\mathbf{S}_\Sigma$ *quotiented by the equations in* $\mathcal{E}$*. We write* $\mathbf{S} := \mathbf{S}_\emptyset$ *for the SMC with terms constructed solely from identities and symmetries.*

Remark 2. It is important to distinguish between the *syntactic* category $\mathbf{S}_\Sigma$ and the *semantic* category $\mathbf{S}_{\Sigma,\mathcal{E}}$. In the former, only 'structural' equalities of the axioms of SMCs hold: moving boxes around while retaining connectivity. In the latter, more equations hold that mean terms with completely different boxes and connectivity can be potentially equal.

Remark 3. One can also define a *multi-sorted* SMT, in which wires can be of multiple colours. For brevity, we will only consider the single-sorted case, but the results generalise easily using the results of [Bon+22a; Bon+22b].

While one could reason in $\mathbf{S}_\Sigma$ using the one-dimensional categorical term language, it is more intuitive to reason with *string diagrams* [JS91; Sel11], which represent *equivalence classes* of terms up to the axioms of SMCs. In the language of string diagrams, a generator $\phi\colon m \to n$ is drawn as a box $^m-\boxed{\phi}-^n$, the identity id_x as $^x-^x$, and the symmetry $\sigma_{X,Y}$ as $^{x\ \ y}_{y\ \ x}$. Composite terms will be illustrated as wider boxes $^m-\boxed{f}-^n$ to distinguish them from generators: then (diagrammatic order) composition $f \fatsemi g$ is defined as horizontal juxtaposition $^x-\boxed{f}\boxed{g}-^z$ and tensor $f \otimes g$ as vertical juxtaposition $\begin{smallmatrix} x-\boxed{f}-y \\ z-\boxed{g}-w \end{smallmatrix}$.

Example 4. The monoidal theory of *special commutative Frobenius algebras* is defined as $(\Sigma_{\mathbf{Frob}}, \mathcal{E}_{\mathbf{Frob}})$ where $\Sigma_{\mathbf{Frob}} := \{ -\boxed{}-, -\bullet\!\!-\boxed{}-, -\boxed{}\!\!-\bullet- \}$ and the equations of $\mathcal{E}_{\mathbf{Frob}}$ are listed in Figs. 1 to 3. We write $\mathbf{Frob} := \mathbf{S}_{\Sigma_{\mathbf{Frob}},\mathcal{E}_{\mathbf{Frob}}}$.

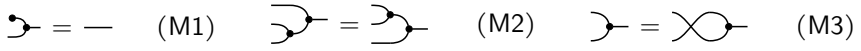


Figure 1: Equations $\mathcal{E}_{\mathbf{CMon}}$ of a *commutative monoid*.

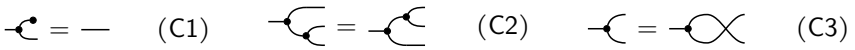


Figure 2: Equations $\mathcal{E}_{\mathbf{CComon}}$ of a *commutative comonoid*.

Reasoning equationally using string diagrams is certainly attractive as a pen-and-paper method, but for larger systems it quickly becomes intractible to do this by hand. Instead, it is desirable to perform equational reasoning *computationally*. Unfortunately, string diagrams as topological objects are not particularly suited for this purpose; instead, we require a combinatorial representation. Fortunately, this has been well studied recently, first with *string graphs* [DK13; Kis12] and later with *hypergraphs* [Bon+22a; Bon+22b; Bon+22c], a generalisation of regular graphs in which edges can be the source or target of an arbitrary number of vertices. In this paper we are concerned with the latter.

Hypergraphs are formally defined as objects in a functor category.

Definition 5 (Hypergraph)**.** *Let* $\mathbf{X}$ *be the category containing objects* (k, l) *for* $k, l \in \mathbb{N}$ *and one additional object* $\star$*. For each* (k, l) *there are* $k + l$ *morphisms* $(k, l) \to \star$*. Let* $\mathbf{Hyp}$ *be the functor category* $[\mathbf{X}, \mathbf{Set}]$*.*

An object in $\mathbf{Hyp}$ maps $\star$ to a set of vertices, and each (k, l) to a set of hyperedges with k sources and l targets, Given a hypergraph $F \in \mathbf{Hyp}$, we write $F_\star$ for its set of vertices and $F_{k,l}$ for the set of edges with k sources and l targets. A morphism of hypergraphs $f \colon F \to G \in \mathbf{Hyp}$ consists of functions $f_\star$ and $f_{k,l}$ for each $k, l \in \mathbb{N}$ preserving sources and targets in the obvious way. With such morphisms a hypergraph can be *labelled*.

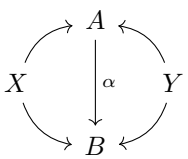
Definition 6 (Slice category [Law63])**.** *For a category* $\mathbf{C}$ *and an object* $C \in \mathbf{C}$*, the* slice category $\mathbb{C}/C$ *is the category with objects the morphisms of* $\mathbf{C}$ *with target* C*, and morphisms* $(f \colon X \to C) \to (f' \colon X' \to C)$ *are the morphisms* $g \colon X \to X' \in \mathbf{C}$ *such that* $f' \circ g = f$*.*

Definition 7 (Hypergraph signature [Bon+22a])**.** *For a given monoidal signature* Σ*, its corresponding* hypergraph signature $[\![\Sigma]\!]$ *is the hypergraph with edges* $e_\phi \in [\![\Sigma]\!]_{\mathsf{dom}(\phi),\mathsf{cod}(\phi)}$ *for each* $\phi \in \Sigma$*, and a vertex* v*. For a hyperedge* e_ϕ*,* $i < \mathsf{dom}(\phi)$ *and* $j < \mathsf{cod}(\phi)$*,* $\mathsf{s}_i(e_\phi) = \mathsf{t}_j(e_\phi) = v$*.*

Definition 8 (Labelled hypergraph [Bon+22a])**.** *For a monoidal signature* Σ*, let the category* $\mathbf{Hyp}_\Sigma$ *be defined as the slice category* $\mathbf{Hyp}/[\![\Sigma]\!]$*.*

While (labelled) hypergraphs may have dangling vertices, they do not have *interfaces* specifying the order of inputs and outputs. These can be provided using *cospans*.

Definition 9 (Categories of cospans [Bon+22b])**.** *For a finitely cocomplete category* $\mathbf{C}$*, a* cospan *from* $X \to Y$ *is a pair of arrows* $X \to A \leftarrow Y$*. A* cospan morphism $(X \xrightarrow{f} A \xleftarrow{g} Y) \to (X \xrightarrow{h} B \xleftarrow{k} Y)$ *is a morphism* $\alpha \colon A \to B \in \mathbf{C}$ *such that the following diagram commutes:*



Two cospans $X \to A \leftarrow Y$ *and* $X \to B \leftarrow Y$ *are* isomorphic *if there exists a morphism of cospans as above where* α *is an isomorphism. Composition is by pushout:*

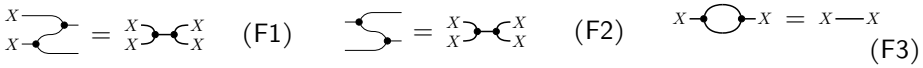


Figure 3: Equations $\mathcal{E}_{\mathbf{Frob}}$ of a *special commutative Frobenius algebra*, in addition to those in Figs. 1 and 2.

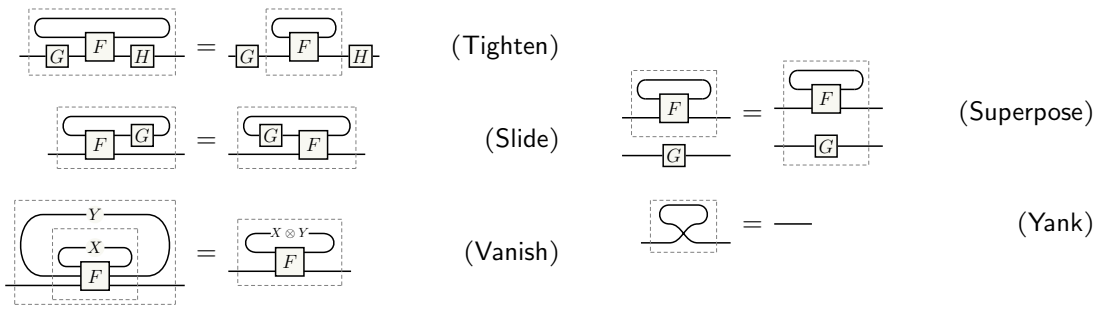
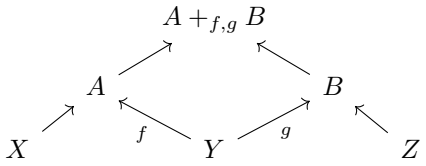


Figure 4: Equations that hold in any *symmetric traced monoidal category*.



The identity is $X \xrightarrow{\mathsf{id}_X} X \xleftarrow{\mathsf{id}_X} X$. *The category of cospans over* $\mathbf{C}$, *denoted* $\mathsf{Csp}(\mathbf{C})$ *has as objects the objects of* $\mathbf{C}$ *and as morphisms the isomorphism classes of cospans. This category has monoidal product given by the coproduct in* $\mathbf{C}$ *with unit the initial object* $0 \in \mathbf{C}$.

The interfaces of a hypergraph can be specified as cospans by having the 'legs' of the cospan pick vertices in the graph at the apex.

Definition 10 (Discrete hypergraph)**.** *A hypergraph is called* discrete *if it has no edges.*

A discrete hypergraph F with $|F_\star| = n$ is written as n when clear from context. Morphisms from discrete hypergraphs to a main graph pick out the vertices in the interface: to assign an order to these vertices some more categorical machinery is required.

Theorem 11 ([Bon+22a], Thm. 3.6)**.** *Let* $\mathbb{X}$ *be a PROP whose monoidal product is a coproduct,* $\mathbf{C}$ *a category with finite colimits, and* $F \colon \mathbb{X} \to \mathbf{C}$ *a coproduct-preserving functor. Then there exists a PROP* $\mathsf{Csp}_F(\mathbf{C})$ *whose arrows* $m \to n$ *are isomorphism classes of* $\mathbf{C}$ *cospans* $Fm \to C \leftarrow Fn$.

Theorem 12 ([Bon+22a], Thm. 3.8)**.** *Let* $\mathbb{X}$ *be a PROP whose monoidal product is a coproduct,* $\mathbf{C}$ *a category with finite colimits, and* $F \colon \mathbb{X} \to \mathbf{C}$ *a coproduct-preserving functor. Then there is a homomorphism of PROPs* $\tilde{F} \colon \mathsf{Csp}(\mathbb{X}) \to \mathsf{Csp}_F(\mathbf{C})$ *that sends* $m \xrightarrow{f} X \xleftarrow{g} n$ *to* $Fm \xrightarrow{Ff} FX \xleftarrow{Fg} Fn$. *If* F *is full and faithful, then* $\tilde{F}$ *is faithful.*

Definition 13. *Let* $\mathbb{F}$ *be the PROP with morphisms* $m \to n$ *the functions between finite sets* $[m] \to [n]$.

Definition 14 ([Bon+22a])**.** *Let* $D \colon \mathbb{F} \to \mathbf{Hyp}_\Sigma$ *be the faithful, coproduct-preserving functor that sends each object* $m \in \mathbb{F}$ *to the discrete hypergraph* $m \in \mathbf{Hyp}_\Sigma$ *and each morphism to the induced homomorphism of discrete hypergraphs.*

From this we define the category $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ with objects *discrete cospans of hypergraphs*. Since the legs of each cospan are discrete hypergraphs containing some number of vertices, the objects of this category can be viewed as natural numbers, making this another PROP.

3 Hypergraphs for traced categories

We wish to use the hypergraph framework for a setting with a *trace*.

Definition 15 (Symmetric traced monoidal category [JSV96; Has09])**.** *A* symmetric traced monoidal category (*STMC*) *is a symmetric monoidal category* $\mathbf{C}$ *with functions* $\mathrm{Tr}^X_{A,B}(-) \colon \mathbf{C}(X \otimes A, X \otimes B) \to \mathbf{C}(A, B)$ *for any objects* A, B *and* X *satisfying the axioms of STMCs listed in Fig. 4.*

In string diagrams, the trace is represented by joining some of the inputs of a term to its outputs.

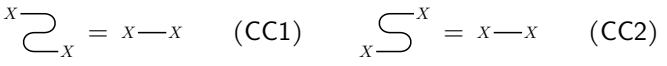


Figure 5: Equations that hold in any *compact closed category*.

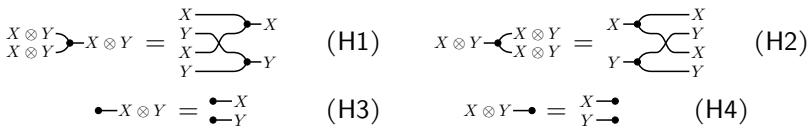
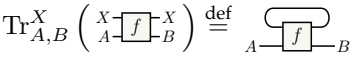


Figure 6: Equations $\mathcal{E}_{\mathbf{Hyp}}$ of a *hypergraph category*, in addition to those in Figs. 1 to 3.

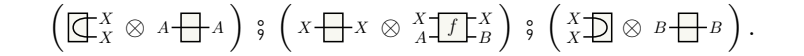


Traced monoidal categories are not the only kind of category in which wires can 'bend'.

Definition 16 (Compact closed category)**.** *A* compact closed category (*CCC*) *is a symmetric monoidal category in which every object X has a* dual X^* *equipped with morphisms called the* unit  (*'cup'*) *and the* counit  (*'cap'*) *satisfying the equations of CCCs listed in Fig. 5.*

Dual objects are conventionally drawn as wires flowing the 'other way', but in this paper this is not necessary as all categories will be *self-dual*: any object X is isomorphic to X^*.

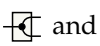
Proposition 17 (Canonical trace ([JSV96], Prop. 3.1))**.** *Any CCC has a trace* $\mathrm{Tr}^X_{A,B}\left(\begin{smallmatrix} X \\ A \end{smallmatrix} f \begin{smallmatrix} X \\ B \end{smallmatrix}\right)$ *called the* canonical trace*, defined for the self-dual case as*



The category of interfaced hypergraphs as defined in the previous section already contains the structure necessary to define a trace.

Definition 18 (Hypergraph category [FS19])**.** *A* hypergraph category *is a symmetric monoidal category in which each object X has a special commutative Frobenius structure in the sense of Example 4 satisfying the equations in Fig. 6.*

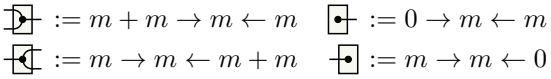
Proposition 19 ([RSW05])**.** *Any hypergraph category is self-dual compact closed.*

Proof. The cup is constructed as  and the cap as  . □

A generic 'hypergraph category' should not be confused with the category of hypergraphs **Hyp**, which is not itself a hypergraph category. However, the category of *cospans* of hypergraphs is such a category.

Proposition 20 ([CW87; Bon+22a])**.** $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ *is a hypergraph category.*

Proof. A Frobenius structure can be defined on $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ for each $m \in \mathbb{N}$ as follows:



□

Corollary 21. $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ *is compact closed.*

Corollary 22. $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ *has a trace.*

This means that a STMC freely generated over a signature faithfully embeds into a CCC generated over the same signature, mapping the trace in the former to the canonical trace in the latter. However, this mapping is not *full*: there are terms in a CCC that are not terms in a STMC, such as  . This means we must still restrict the cospans of hypergraphs in $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ we use for *traced* terms.

3.1 Monogamy

In [Bon+16], it is shown that terms in a (non-traced) symmetric monoidal category are interpreted via a faithful functor into a sub-PROP of $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$. One condition on this sub-PROP is that all hypergraphs are *acyclic*. Clearly, to model trace this condition must be dropped.

However, there is also another condition known as *monogamy*: informally, this means that every vertex has exactly one 'in' and 'out' connection, be it to an edge or an interface. For the most part, this condition also applies to the traced case: wires cannot arbitrarily fork and join. There is one nuance: the trace of the identity. This is depicted as a closed loop $\mathrm{Tr}^1\left(\boxplus\right) = \bigcirc$, and one might think that it can be discarded, i.e. $\bigcirc = \emptyset$. This is *not* always the case, such as in the category of finite dimensional vector spaces [Has97, Sec. 6.1]. These closed loops must be represented in the hypergraph framework: there is a natural representation as a lone vertex disconnected from either interface. In fact, this is exactly how the canonical trace applied to an identity is interpreted in $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$.

Definition 23. *For a hypergraph $F \in \mathbf{Hyp}$, the* degree *of a vertex $v \in F_\star$ is a tuple (i, o) where i is the number of pairs (e, i) where e is a hyperedge with v as its ith target, and o is similarly the number of pairs (e, j) where e is a hyperedge with v as its jth target.*

Definition 24. *For a cospan $m \xrightarrow{f} F \xleftarrow{g} n \in \mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$, we say it is* partial monogamous *if f and g are mono and, for all nodes $v \in F_\star$, the degree of v is*

$$\begin{array}{llll} (0,0) & \text{if } v \in f_\star \wedge v \in g_\star & (0,1) & \text{if } v \in f_\star \\ (1,0) & \text{if } v \in g_\star & (0,0) \text{ or } (1,1) & \text{otherwise} \end{array}$$

Lemma 25. *Given a cospan $x + m \xrightarrow{h+f} F \xleftarrow{k+g} x + n \in \mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$, let p be the map sending vertices in $F_\star$ to the corresponding vertices after constructing $\mathrm{Tr}^x\,(x + m \to F \leftarrow x + n)$. Then if the degree of $h(i)$ is (i_1, j_1) and the degree of $k(i)$ is (i_2, j_2), then the degree of $p(h(i)) = p(k(i))$ is $(i_1 + i_2, j_1 + j_2)$.*

Proof. By computing the pushouts. □

Theorem 26. *Let $m \to F \leftarrow n$, $n \to G \leftarrow p$, $p \to H \leftarrow q$ and $x + m \to K \leftarrow x + n$ be partial monogamous cospans in $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$. Then,*

- *identities and symmetries are partial monogamous;*
- *$m \to F \leftarrow n \fatsemi n \to G \leftarrow p$ is partial monogamous;*
- *$m \to F \leftarrow n \otimes p \to H \leftarrow q$ is partial monogamous; and*
- *$\mathrm{Tr}^x\,(x + m \to K \leftarrow x + n)$ is partial monogamous.*

Proof. Since any monogamous hypergraph is also partial monogamous, the first three points hold due to [Bon+22a, Prop.16], dropping the acyclicity condition. For the final condition, consider the image of f and g. For each $i \in x$, there are two cases to consider: $f(i) = g(i)$ and $f(i) \neq g(i)$.

In the former, the degree of $v' = f(v) = h(v)$ must be $(0, 0)$ by definition of partial monogamicity. Therefore in the traced hypergraph $m \to K' \leftarrow n$, v' will still have degree $(0, 0)$ by Lemma 25, and cannot be in the interfaces, so the hypergraph is still partial monogamous.

In the latter case, $f(i)$ must have degree of $(0, 0)$ if it in the image of k or $(0, 1)$ otherwise. Similarly $g(i)$ either has degree $(0, 0)$ or $(1, 0)$. Let $v := p(f(i)) = p(g(i))$; we now consider the degree of v computed using Lemma 25:

- If $f(i)$ is in the image of k and $g(i)$ is in the image of h, then v has degree $(0, 0)$. v is in the image of $h \fatsemi p$ and $h \fatsemi p$, so the cospan is partial monogamous.
- If $f(i)$ is in the image of k, then v has degree $(1, 0)$; since v is in the image of $k \fatsemi p$, the cospan is partial monogamous.
- If $g(i)$ is in the image of h, then v has degree $(0, 1)$; since v is in the image of $h \fatsemi p$, the cospan is partial monogamous.
- Otherwise, v will have degree $(1, 1)$, and is not in the image of either interface so the cospan is partial monogamous.

□

This means that the partial monogamous hypergraphs form a sub-PROP of $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$.

Definition 27. *Let* $\mathsf{PMCsp}_D(\mathbf{Hyp}_\Sigma)$ *be the sub-PROP of* $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ *containing only the partial monogamous cospans of hypergraphs.*

Crucially, while we leave $\mathsf{PMCsp}_D(\mathbf{Hyp}_\Sigma)$ in order to construct the trace using the cup and cap, the resulting cospan *is* in $\mathsf{PMCsp}_D(\mathbf{Hyp}_\Sigma)$.

3.2 From terms to graphs

Definition 28. *For a SMT* $(\Sigma, \mathcal{E})$*, let* $\mathbf{T}_\Sigma$ *be the STMC freely generated over the generators in* Σ*. Let* $\mathbf{T}_{\Sigma,\mathcal{E}}$ *be* $\mathbf{T}_\Sigma$ *quotiented by equations in* $\mathcal{E}$*.*

A (*traced*) *PROP morphism* is a strict (traced) symmetric monoidal functor between PROPs. For $\mathsf{PMCsp}_{FI}(\mathbf{Hyp}_\Sigma)$ to be suitable for reasoning with a traced category $\mathbf{T}_\Sigma$ for some given signature, there must be a *fully complete* PROP morphism $\mathbf{T}_\Sigma \to \mathsf{PMCsp}_{FI}(\mathbf{Hyp}_\Sigma)$: a full and faithful functor from terms to cospans of hypergraphs.

We exploit the interplay between compact closed and traced categories in order to use the existing PROP morphisms from [Bon+22a] for the traced case. Since $\mathbf{S}_\Sigma$ and $\mathsf{PMCsp}_D(\mathbf{Hyp}_\Sigma)$ are freely generated, to define a PROP morphism between them it suffices to define it on the generators in the former.

Definition 29 ([Bon+22a])**.** *Let* $[\![-]\!]_\Sigma \colon \mathbf{S}_\Sigma \to \mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ *be a prop morphism defined as*

$$[\![\ m \text{—}[\phi]\text{—} n\ ]\!]_\Sigma := m \to \boxed{\ m\ \ \phi\ \ n\ } \leftarrow n$$

$$[\![\ n \text{—}\boxplus\text{—} n\ ]\!]_\Sigma := n \xrightarrow{\mathsf{id}} n \xleftarrow{\mathsf{id}} n \qquad [\![\ \begin{smallmatrix} m & & n \\ n & & m \end{smallmatrix}\ ]\!]_\Sigma := m+n \xrightarrow{[\mathsf{id},\mathsf{id}]} m+n \xleftarrow{[\mathsf{id},\mathsf{id}]} n+m$$

Let $[-]_\Sigma : \mathbf{Frob} \to \mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ *be a PROP morphism defined as in Proposition 20. Then, let*

$$\langle\!\langle - \rangle\!\rangle_\Sigma \colon \mathbf{S}_\Sigma + \mathbf{Frob} \to \mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$$

be the copairing of $[\![-]\!]_\Sigma + [-]_\Sigma$*.*

Lemma 30. *Let* $m \text{—}[f]\text{—} n$ *be a term in* $\mathbf{T}_\Sigma$*. Then there exists at least one* $\begin{smallmatrix} x & & x \\ m & [g] & n \end{smallmatrix} \in \mathbf{S}_\Sigma$ *such that* $\mathrm{Tr}^x\left(\ \begin{smallmatrix} x & & x \\ & [g] & \end{smallmatrix}\ \right) = \text{—}[f]\text{—}$*.*

Proof. Any trace can be made a 'global trace' by applying (Tighten) and (Superpose). □

Proposition 31. *There exists a faithful PROP morphism* $\lfloor - \rfloor_\Sigma^{\mathbf{T}} : \mathbf{T}_\Sigma \to \mathbf{S}_\Sigma + \mathbf{Frob}$*.*

Proof. Lemma 30 is used to isolate a term in $\mathbf{S}_\Sigma$. The corresponding term in $\mathbf{S}_\Sigma + \mathbf{Frob}$ is then the canonical trace of this term. There may be many such terms in $\mathbf{S}_\Sigma$, but the canonical trace being a trace means that any possible outcomes post-trace are all equal. The equations of $\mathbf{Frob}$ do not merge any morphisms since the only use of the generators of $\mathbf{Frob}$ is in the canonical trace, to which the Frobenius equations do not apply. □

Before progressing to the main theorem, we must show a result about terms in $\mathbf{S}$, i.e. terms constructed from just symmetries and identities: there is a correspondence between $\mathbf{S}$ and *bijective* functions.

Definition 32. *Let* $\mathbb{P}$ *be the sub-PROP of* $\mathbb{F}$ *containing only the bijective functions.*

Lemma 33. $\mathbf{S} \cong \mathbb{P}$*.*

Proof. The morphism $\phi \colon \mathbf{S} \to \mathbb{P}$ is defined over generators in $\mathbf{S}$ as

$$\phi(\ \ ) = \{\} \quad \phi(\ \boxplus\ ) = \{0 \mapsto 0\} \quad \phi(\ \times\ ) = \{0 \mapsto 1, 1 \mapsto 0\}$$

Since any term in $\mathbf{S}$ can be expressed using these generators, this defines the complete transformation.

The reverse morphism $\psi \colon \mathbb{F} \to \mathbf{S}$ is inductively over the size of m. For the base case $f \colon [0] \to [0]$, let $\phi(f) := \ \ $. For $f \colon [k+1] \to [k+1]$, let i such that $f(i) = k+1$, and define the function $f' \colon \mathbb{N}_k \to \mathbb{N}_k$ as the function such that $f'(j) = f(j)$ if $j < i$, and $f(j+1)$ otherwise. Then

$$\psi(f) := \begin{smallmatrix} i & & \\ 1 & & [\psi(f')]\text{—} k \\ k-i & & \end{smallmatrix}$$

These are shown to be inverses by a simple induction in both directions. □

Lemma 34. *Given a monogamous cospan $m \xrightarrow{f} m \xleftarrow{g} m$, there exists a unique term $h\colon m \to m \in \mathbf{S}$ up to the axioms of SMCs such that $[\![h]\!]_\Sigma = m \xrightarrow{f} m \xleftarrow{g} m$.*

Proof. Since the cospan is monogamous, f and g are mono. As the cospan is also discrete, there exists a (unique) bijective function $h'\colon [m] \to [m]$ such that $h'(i) = j$ if $f(i) = g(j)$. By Lemma 33, there is a corresponding term $h \in \mathbf{S}$ that is unique up to SMC axioms: a simple induction shows that $[\![h]\!]_\Sigma = m \xrightarrow{f} m \xleftarrow{g} m$. □

Theorem 35. *A cospan $m \to F \leftarrow n$ is in the image of $\langle\!\langle - \rangle\!\rangle_\Sigma \circ \lfloor - \rfloor^{\mathbf{T}}_\Sigma$ if and only if it is partial monogamous.*

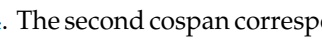
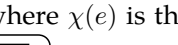
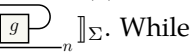
Proof. To show that $\lfloor [\![f]\!]_\Sigma \rfloor^{\mathbf{T}}_\Sigma$ is partial monogamous for any $f \in \mathbf{T}_\Sigma$ we use induction on the structure of f. Generators, identities and symmetries are partial monogamous, as semi-monogamicity is preserved by composition, tensor and trace. So $[\![f]\!]_\Sigma$ is partial monogamous.

Now we show that any partial monogamous cospan $m \xrightarrow{f} F \xleftarrow{g} n$ must be in the image of $[\![-]\!]_\Sigma \circ \lfloor - \rfloor^{\mathbf{T}}_\Sigma$. To do this, we will now construct a cospan that is isomorphic to $m \xrightarrow{f} F \xleftarrow{g} n$, but from which it is possible to read off a unique term in $\mathbf{T}_\Sigma$. The components of the new cospan are as follows:

- let L be the hypergraph containing vertices with degree $(0, 0)$ that are not in the image of f or g;
- let E be the hypergraph containing hyperedges of F and their source and target vertices, but disconnected;
- let V be the discrete hypergraph containing all the vertices of F; and
- let S and T be the discrete hypergraphs containing the source and target vertices of hyperedges in F respectively, with the ordering determined by some order $e_1, e_2, \cdots, e_n$ on the edges in F.

These parts can be composed and a trace applied to obtain the follow cospan:

$$\mathrm{Tr}^T \left(T + m \xrightarrow{\mathsf{id}+f} V \xleftarrow{\mathsf{id}+g} S + n \fatsemi \emptyset + S + n \xrightarrow{\mathsf{id}} L + E + n \xleftarrow{\mathsf{id}} \emptyset + T + n \right) \tag{1}$$

This can be checked to be isomorphic to the original cospan $m \xrightarrow{f} F \xleftarrow{g} n$ by applying the pushouts. From this we can read off a term in $\mathbf{T}_\Sigma$: Since the first cospan is monogamous, it corresponds to a term  by Lemma 34. The second cospan corresponds to   , where $\chi(e)$ is the generator in Σ that e is labelled with. Putting this all together yields $h :=$  . While there may be multiple orderings on the edges, the possible terms are equal by sliding and the naturality of symmetry, so there is one unique term  that corresponds to cospan (1). It is clear by definition that  produces (1), which is isomorphic to the original cospan $m \xrightarrow{f} F \xleftarrow{g} n$, so it is in the image of $[\![-]\!]_\Sigma \circ \lfloor - \rfloor^{\mathbf{T}}_\Sigma$. □

Proposition 36 ([Bon+22a])**.** *$[\![-]\!]_\Sigma$ and $[-]_\Sigma$ are faithful.*

Corollary 37. $\mathbf{T}_\Sigma \cong \mathsf{PMCsp}_{FI}(\mathbf{Hyp}_\Sigma)$.

4 Hypergraphs for traced commutative comonoid categories

We are interested in another element of structure in addition to the trace: the ability to *copy* and *discard* wires. This is known as a (*commutative*) *comonoid structure*: categories equipped with such a structure are also known as *gs-monoidal* (*garbage-sharing*) categories [FL22].

Definition 38. *Let $(\Sigma_{\mathbf{CComon}}, \mathcal{E}_{\mathbf{CComon}})$ be the symmetric monoidal theory of* commutative comonoids, *with $\Sigma_{\mathbf{CComon}} := \{$  ,  $\}$ and $\mathcal{E}_{\mathbf{CComon}}$ defined as in Fig. 2. We write $\mathbf{CComon} := \mathbf{S}_{\Sigma_{\mathbf{CComon}}, \mathcal{E}_{\mathbf{CComon}}}$.*

From now on, we write 'comonoid' to mean 'commutative comonoid'. There has already been work using hypergraphs for PROPs with a (co)monoid structure [FL22; MZ22] but these consider *acyclic* hypergraphs: we must ensure that removing the acyclicity condition does not lead to any degeneracies.

Definition 39 (Partial left-monogamy)**.** *For a cospan $m \xrightarrow{f} H \xleftarrow{g} n$, we say it is* partial left-monogamous *if f is mono and, for all nodes $v \in H_\star$, the degree of v is $(0, m)$ if $v \in f_\star$ and $(0, m)$ or $(1, m)$ otherwise, for some $m \in \mathbb{N}$.*

Remark 40. As with the vertices not in the interfaces with degree $(0, 0)$ in the vanilla traced case, the vertices not in the interface with degree $(0, m)$ allow for terms such as $\mathrm{Tr}\left(\boxed{\leftarrow}\right)$.

Lemma 41. *Let $m \to F \leftarrow n$, $n \to G \leftarrow p$, $p \to H \leftarrow q$ and $x + m \to K \leftarrow x + n$ be partial left-monogamous cospans. Then,*

- *identities and symmetries are partial left-monogamous;*
- *$m \to F \leftarrow n \fatsemi n \to G \leftarrow p$ is partial left-monogamous;*
- *$m \to F \leftarrow n \otimes p \to H \leftarrow q$ is partial left-monogamous; and*
- *$\mathrm{Tr}^x\,(x + m \to K \leftarrow x + n)$ is partial left-monogamous.*

Proof. Identities and symmetries are monogamous, and as such they are also partial left-monogamous. For the operations, we only need to check the in-degrees of vertices, as the out-degree can be arbitrary. For composition, only the vertices in the image of $n \to F$ and $n \to G$ have their in-degrees modified. By partial left-monogamy we can conclude that:

- the in-degree of vertices in the image of $n \to F$ is 0 if they are in the image of $m \to F$, and 0 or 1 otherwise; and
- the in-degree of vertices in the image of $n \to G$ is 0.

This means that vertices in the image of $m \to F \fatsemi G$ will have in-degree 0, and the other vertices will have 0 or 1 otherwise. For tensor, the elements of the original two graphs are unaffected so the degrees remain unchanged. For trace, we must show that for any set of vertices in the image of $x + n \to K$ merged by the trace, at most one of them can have in-degree 1. But this must be the case because any image in the image of $x + m \to K$ must have in-degree 0, and $x + m \to K$ is moreover mono so it cannot merge vertices in the image of $x + n \to K$. □

Definition 42. *Let $\mathsf{PLMCsp}_D(\mathbf{Hyp}_\Sigma)$ be the sub-PROP of $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ containing only the partial left-monogamous cospans of hypergraphs.*

This category can be equipped with a comonoid structure.

Definition 43. *Let $\lfloor - \rfloor^{\mathbf{C}} : \mathbf{CComon} \to \mathbf{Frob}$ be the obvious embedding of $\mathbf{CComon}$ into $\mathbf{Frob}$, and let $\lfloor - \rfloor_\Sigma : \mathbf{T}_\Sigma + \mathbf{Comon} \to \mathbf{S}_\Sigma + \mathbf{Frob}$ be the copairing of $\lfloor - \rfloor_\Sigma^{\mathbf{T}}$ and $\lfloor - \rfloor^{\mathbf{C}}$.*

Lemma 44. *The image of $[-]_\Sigma \circ \lfloor - \rfloor^{\mathbf{C}}$ is in $\mathsf{PLMCsp}_D(\mathbf{Hyp}_\Sigma)$*

Proof. By definition. □

Corollary 45. *The image of $\langle\!\langle - \rangle\!\rangle_\Sigma \circ \lfloor - \rfloor_\Sigma$ is in $\mathsf{PLMCsp}_D(\mathbf{Hyp}_\Sigma)$.*

Lemma 46 ([Lac04])**.** $\mathbf{CComon} \cong \mathbb{F}^{\mathsf{op}}$.

Lemma 47. *Given a partial left-monogamous cospan $m \xrightarrow{f} m \xleftarrow{g} n$, there exists a unique term $h \colon m \to n \in \mathbf{CComon}$ up to the axioms of SMCs and comonoids such that $[-]_\Sigma \circ \lfloor - \rfloor^{\mathbf{C}} = m \xrightarrow{f} m \xleftarrow{g} n$.*

Proof. Given a partial left-monogamous cospan $m \xrightarrow{f} m \xleftarrow{g} n$, there exists a unique function $h \colon [n] \to [m]$ such that $h(i) = j$ if $g(i) = f(j)$. Since $\mathbb{F} \cong \mathbb{F}^{\mathsf{op}}$, by Lemma 46, there exists a corresponding term in $\mathbf{CComon}$. □

Theorem 48. $\mathbf{T}_\Sigma + \mathbf{CComon} \cong \mathsf{PLMCsp}_{FI}(\mathbf{Hyp}_\Sigma)$.

Proof. Since $\langle\!\langle - \rangle\!\rangle_\Sigma$ and $[-]_\Sigma^{\mathbf{C}}$ are faithful, it suffices to show that a cospan $m \to F \leftarrow n$ in $\mathsf{PLMCsp}_D(\mathbf{Hyp}_\Sigma)$ can be decomposed in such a way that each component is in the image of either $\langle\!\langle - \rangle\!\rangle_\Sigma$ or $[-]_\Sigma^{\mathbf{C}}$. This is achieved by taking the construction of Theorem 35 and allowing the first component to be partial left-monogamous; by Lemma 47 a term in $\mathbf{CComon}$ can be retrieved from this. □

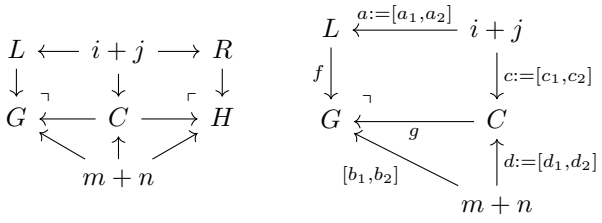


Figure 7: The DPO diagram and a pushout complement.

5 Graph rewriting

We have now shown that we can reason up to the axioms of symmetric traced categories with a comonoid structure using hypergraphs: string diagrams equal by topological deformations are translated into isomorphic hypergraphs. However, to reason about an *monoidal theory* with extra equations we must actually rewrite the components of the graph. In the syntactic realm this is performed with *term rewriting*.

Definition 49 (Term rewriting)**.** *A* rewriting system $\mathcal{R}$ *for a traced PROP* $\mathbf{T}_\Sigma$ *consists of a set of* rewrite rules $\langle\, i \boxed{l} j \,,\, i \boxed{r} j \,\rangle$*. Given terms* $m \boxed{g} n$ *and* $m \boxed{h} n$ *in* $\mathbf{T}_\Sigma$ *we write* $\boxed{g} \Rightarrow_{\mathcal{R}} \boxed{h}$ *if there exists rewrite rule* $(\, i \boxed{l} j \,,\, i \boxed{r} j \,)$ *in* $\mathcal{R}$ *and* $\overset{j}{m} \boxed{c} \overset{i}{n}$ *in* $\mathbf{T}_\Sigma$ *such that* $\boxed{g} = \underline{\boxed{l}\ \boxed{c}}$ *and* $\boxed{h} = \underline{\boxed{r}\ \boxed{c}}$ *by axioms of STMCs.*

The graph equivalent is *graph rewriting*. A common framework is that of *double pushout rewriting* (DPO rewriting). We use an extension, known as *double pushout rewriting with interfaces* (DPOI rewriting).

Definition 50 (DPO rule)**.** *Given interfaced hypergraphs* $i \xrightarrow{a_1} L \xleftarrow{a_2} j$ *and* $i \xrightarrow{b_1} R \xleftarrow{b_2} j$*, their* DPO rule *in* $\mathbf{Hyp}_\Sigma$ *is a span* $L \xleftarrow{[a_1,a_2]} i + j \xrightarrow{[b_1,b_2]} R$*.*

Definition 51 (DPO(I) rewriting)**.** *Let* $\mathcal{R}$ *be a set of DPO rules. Then, for morphisms* $G \leftarrow m + n$ *and* $H \leftarrow m + n$ *in* $\mathbf{Hyp}_\Sigma$*, there is a rewrite* $G \rightsquigarrow_{\mathcal{R}} H$ *if there exists a rule* $L \leftarrow i + j \rightarrow R \in \mathcal{R}$ *and cospan* $i + j \rightarrow C \leftarrow n + m \in \mathbf{Hyp}_\Sigma$ *such that diagram in the left of Fig. 7 commutes.*

The first thing to note is that the graphs in the DPO diagram have a *single* interface $G \leftarrow m + n$ instead of the cospans $m \rightarrow G \leftarrow n$ we are used to. Before performing DPO rewriting in $\mathbf{Hyp}_\Sigma$, the interfaces must be 'folded' into one.

Definition 52 ([Bon+22b])**.** *Let* $\ulcorner - \urcorner \colon \mathbf{S}_\Sigma + \mathbf{Frob} \rightarrow \mathbf{S}_\Sigma + \mathbf{Frob}$ *be defined as having action* $m \boxed{f} n \mapsto \overset{m}{\underset{n}{\boxed{f}}}$.

Note that the result of applying $\ulcorner - \urcorner$ is not in the image of $\langle\!\langle - \rangle\!\rangle_\Sigma \circ \lfloor - \rfloor^{\mathbf{T}}_\Sigma$ any more. This is not an issue, so long as we 'unfold' the interfaces once rewriting is completed.

Proposition 53 ([Bon+22a], Prop. 4.8)**.** *If* $\langle\!\langle\, m \boxed{f} n \,\rangle\!\rangle_\Sigma = m \xrightarrow{i} F \xleftarrow{o} n$ *then* $\ulcorner \langle\!\langle\, \boxed{f} \,\rangle\!\rangle_\Sigma \urcorner$ *is isomorphic to* $0 \rightarrow F \xleftarrow{i+o} m + n$*.*

In order to apply a given DPO rule $L \leftarrow i + j \rightarrow R$ in some larger graph $m \rightarrow G \leftarrow n$, a morphism $L \rightarrow G$ must first be identified. The next step is to 'cut out' the components of L that exist in G.

Definition 54 (Pushout complement)**.** *Let* $i + j \rightarrow L \rightarrow G \rightarrow m + n$ *be morphisms in* $\mathbf{Hyp}_\Sigma$*. Then the* pushout complement *of these morphisms is an object* C *with morphisms* $i + j \rightarrow C \rightarrow G$ *such that* $L \rightarrow G \leftarrow C$ *is a pushout and the diagram on the right of Fig. 7 commutes.*

Once a pushout complement C is computed, the pushout of $C \leftarrow i + j \rightarrow R$ can be performed to obtain the completed rewrite H. However, a pushout complement may not exist for a given rule and matching.

Definition 55 ([Bon+22a], Def. 3.16)**.** *Let* $i + j \xrightarrow{a} L \xrightarrow{f} G$ *be morphisms in* $\mathbf{Hyp}_\Sigma$*. The morphisms satisfy the* no-dangling *condition if, for every hyperedge not in the image of* f*, each of its source and target vertices is either not in the image of* f *or are in the image of* $f \circ a$*. The morphisms satisfy the* no-identification *condition if any two distinct elements merged by* f *are also in the image of* $f \circ a$*.*

Proposition 56 ([Bon+22a], Prop. 3.17)**.** *The morphisms $i+j \to L \to G$ have at least one pushout complement if and only if they satisfy the no-dangling and no-identification conditions.*

Definition 57. *Given a partial monogamous cospan $i \to L \leftarrow j$, a morphism $L \to G$ is called a* matching *if it has at least one pushout complement.*

In certain settings, known as *adhesive categories* [LS04], it is possible to be more precise about the number of pushout complements for a given matching and rewrite rule.

Proposition 58 ([LS04])**.** *In an adhesive category, pushout complements of $i + j \xrightarrow{a} L \to G$ are unique if they exist and a is mono.*

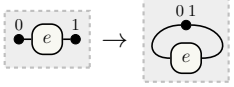
Proposition 59 ([LS05])**.** $\mathbf{Hyp}_\Sigma$ *is adhesive.*

Since a given pushout complement uniquely determines the rewrite performed, it might seem advantageous to always have exactly one pushout complement. However, as shall be detailed later, when writing modulo traced comonoid structure there are settings where having multiple pushout complements is beneficial.

5.1 Rewriting with traced structure

While in the Frobenius case considered in [Bon+22a], all valid pushout complements correspond to a valid rewrite, this is not the case for the traced monoidal case. In [Bon+22b], pushout complements that correspond to a valid rewrite in the non-traced symmetric monoidal case are identified as *boundary complements*. We will use a weakening of this definition.

Definition 60 (Traced boundary complement)**.** *A pushout complement as in Definition 54 is called a* traced boundary complement *if c_1 and c_2 are mono and $j + m \xrightarrow{[c_2,d_1]} C \xleftarrow{[d_2,c_1]} n + i$ is a partial monogamous cospan.*

Unlike [Bon+22b], we do not enforce that the matching is mono, as restricting to these matchings actually cuts off potential rewrites in the *traced* setting, such as the occurrence of a rewrite rule inside a loop: 

.

Definition 61 (Traced DPO)**.** *For morphisms $G \leftarrow m + n$ and $H \leftarrow m + n$ in $\mathbf{Hyp}_\Sigma$, there is a traced rewrite $G \rightsquigarrow_{\mathcal{R}} H$ if there exists a rule $L \leftarrow i + j \to G \in \mathcal{R}$ and cospan $i + j \to C \leftarrow n + m \in \mathbf{Hyp}_\Sigma$ such that diagram in Definition 51 commutes and $i + j \to C$ is a traced boundary complement.*

Some intuition on the construction of traced boundary complements may be required: this will be provided through a lemma and some examples.

Lemma 62. *In a traced boundary complement, let $v \in i$ and $w_0, w_1, \cdots w_k$ such that $f(a_1(v)) = f(a_2(w_0))$, $f(a_1(v)) = f(a_2(w_1))$ and so on. Then either (1) there exists exactly one w_l not in the image of d_1 such that $c_1(v) = c_2(w_l)$; (2) $c_1(v)$ is in the image of d_1; or (3) $c_1(v)$ has degree $(1, 0)$. The same also holds for $w \in j$, with the interface map as d_2 and the degree as $(0, 1)$.*

Proof. Since $c_1(v)$ is in the image of c, it must have either degree $(0, 0)$ or $(1, 0)$ by partial monogamy. For it to have degree $(0, 0)$, it must either be in the image of one of c_2 or d_2. In the case of the former, this means that there must be a w_l such that $c_1(v) = c_2(w_l)$, and only one such vertex as c_2 is mono, so (1) is satisfied. In the case of the latter, (2) is immediately satisfied. Otherwise, (3) is satisfied.

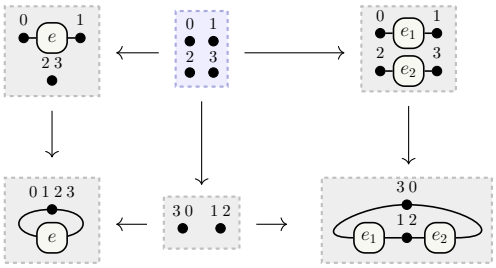
The proof for the flipped case is exactly the same. □

Often there can be valid rewrites in the realm of graphs that are non-obvious in the term language. This is because we are rewriting modulo *yanking*.

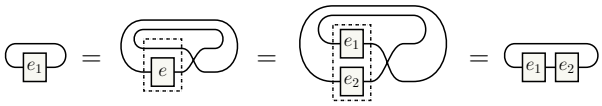
Example 63. Consider the rule $\langle$ 

, 

$\rangle$. The interpretation of this as a DPO rule in a valid traced boundary complement is illustrated below.

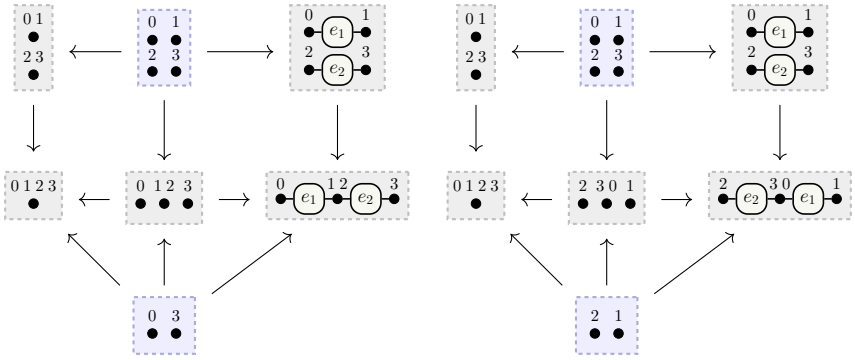


This corresponds to a valid term rewrite:

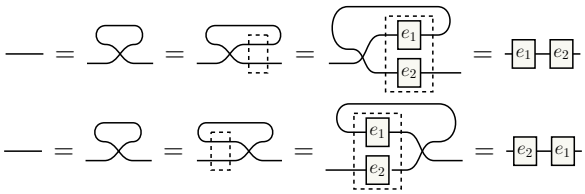


Unlike regular boundary complements, traced boundary complements need not be unique. This is not an issue, as all pushout complements can be enumerated for a given rule and matching [Heu+11].

Example 64. Consider the rule $\langle \_\_, \begin{smallmatrix} e_1 \\ e_2 \end{smallmatrix} \rangle$. Below are two valid traced boundary complements involving a matching of this rule.

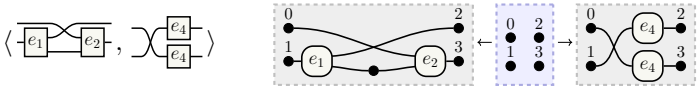


These derivations arise since we are rewriting modulo *yanking*:



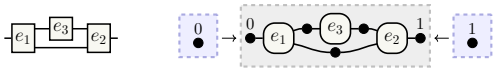
Rewriting modulo yanking also eliminates another foible of rewriting modulo (non-traced) symmetric monoidal structure. In the SMC case, the image of the matching must be *convex*: any path between vertices must also be captured. This is not necessary in the traced case.

Example 65. Consider the following rewrite rule and its interpretation.



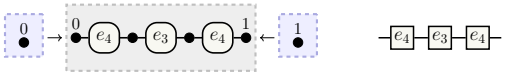
(2)

Now consider the following term and interpretation:



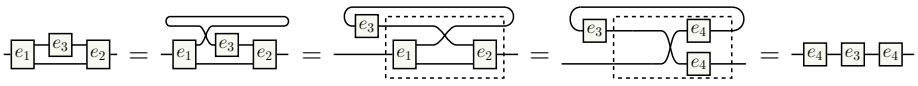
(3)

Although it is not obvious in the original string diagram, there is in fact a matching of (2) in (3). Performing the DPO procedure yields the following:



(4)

In a non-traced setting this is an invalid rule! However, it is possible with yanking.



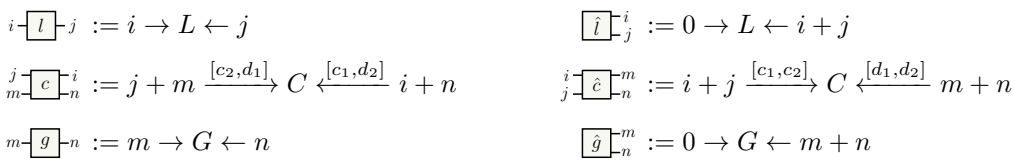
We are almost ready to show the soundness and completeness of this DPO rewriting system. The final prerequisite is a decomposition lemma, akin to a similar result in [Bon+22a].

Lemma 66 (Traced decomposition)**.** *Given partial monogamous cospans $m \xrightarrow{d_1} G \xleftarrow{d_2} n$ and $i \xrightarrow{a_1} L \xleftarrow{a_2} j$, along with a morphism $L \xrightarrow{f} G$ such that $i+j \to L \to G$ satisfies the no-dangling and no-identification conditions, then there exists $j+m \xrightarrow{[c_2,d_1]} C \xleftarrow{[c_1,d_2]} i+n$ such that $m \to G \leftarrow n$ can be factored as*

$$\mathrm{Tr}^i \left(\begin{array}{c} i \xrightarrow{a_1} L \xleftarrow{a_2} j \\ \otimes \\ m \to m \leftarrow m \end{array} \fatsemi j+m \xrightarrow{[c_2,d_1]} C \xleftarrow{[c_1,d_2]} i+n \right) \tag{5}$$

where all cospans are partial monogamous and $j+m \xrightarrow{c_2,d_1} C \xleftarrow{c_1,d_2} i+n$ is a traced boundary complement.

Proof. Let $i+j \xrightarrow{[c_1,c_2]} C \xleftarrow{[d_1,d_2]} m+n$ be defined as a traced boundary complement of $i+j \xrightarrow{[a_1,a_2]} L \xrightarrow{f} G$, which exists as the no-dangling and no-identification condition is satisfied. We assign names to the various cospans in play, and reason string diagrammatically:

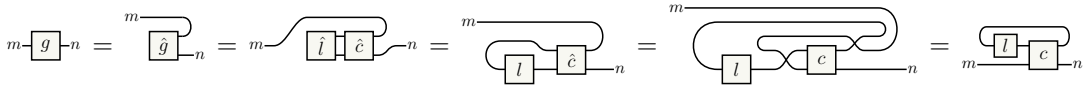


Note that the cospans in the left column are partial monogamous by definition of rewrite rules and traced boundary complements. We will show that  can be decomposed into a form using the two cospans  and  , along with identities.

By using the compact closed structure of $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$, we have the following:

Since G is the pushout of $L \xleftarrow{[a_1,a_2]} i+j \xrightarrow{[c_1,c_2]} C$ and pushout is cospan composition, we also have that  .

Putting this all together we can show that



Since the 'loop' is constructed in the same manner as the canonical trace on $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ (and is therefore identical in the graphical notation), this is a term in the form of (5). □

We write $\ulcorner \lfloor \mathcal{R} \rfloor_\Sigma^{\mathbf{T}} \urcorner$ for the pointwise map ( , ) $\mapsto$ ($\ulcorner \lfloor$ $\rfloor_\Sigma^{\mathbf{T}} \urcorner$, $\ulcorner \lfloor$  $\rfloor_\Sigma^{\mathbf{T}} \urcorner$).

Theorem 67. *Let $\mathcal{R}$ be a rewriting system on $\mathbf{T}_\Sigma$. Then,*

Proof. First the ($\Rightarrow$) direction. If  $\Rightarrow_{\mathcal{R}}$ then we have = and = .

Define the following cospans:

$$0 \to L \leftarrow i+j := \langle\!\langle \ulcorner\, l \,\urcorner \rangle\!\rangle_\Sigma = \langle\!\langle$$  $$\rangle\!\rangle_\Sigma \quad (6)$$

$$0 \to R \leftarrow i+j := \langle\!\langle \ulcorner\, r \,\urcorner \rangle\!\rangle_\Sigma = \langle\!\langle$$  $$\rangle\!\rangle_\Sigma \quad (7)$$

$$0 \to G \leftarrow m+n := \langle\!\langle \ulcorner\, f \,\urcorner \rangle\!\rangle_\Sigma = \langle\!\langle$$  $$\rangle\!\rangle_\Sigma \quad (8)$$

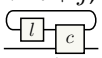
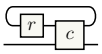
$$0 \to H \leftarrow m+n := \langle\!\langle \ulcorner\, h \,\urcorner \rangle\!\rangle_\Sigma = \langle\!\langle$$  $$\rangle\!\rangle_\Sigma \quad (9)$$

$$i+j \to C \leftarrow m+n := \langle\!\langle$$  $$\rangle\!\rangle_\Sigma \quad (10)$$

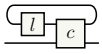
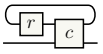
By functoriality, since $\ulcorner\, f \,\urcorner =$  then

$$0 \to G \leftarrow m+n = 0 \to L \leftarrow i+j \fatsemi i+j \to C \leftarrow m+n.$$

Cospan composition is pushout, so $L \to G \leftarrow C$ is a pushout. Using the same reasoning, $R \to G \leftarrow C$ is also a pushout: this gives us the DPO diagram. All that remains is to check that the aforementioned pushouts are traced boundary complements: this follows by inspecting components.

Now the ($\Leftarrow$) direction: assume we have a DPO diagram (51) where $L \leftarrow i+j$, $i+j \to R$, $m+n \to G$ and $m+n \to H$ are defined as in (6-9) above. We must show that $\ulcorner\, f \,\urcorner =$  and $\ulcorner\, h \,\urcorner =$  . By definition of traced boundary complement $j+m \to C \leftarrow i+n$ is a partial monogamous cospan, so by fullness of $\langle\!\langle - \rangle\!\rangle_\Sigma$, there exists a term $^{j}_{m}$  $^{i}_{n}$ $\in \mathbf{S}_\Sigma$ such that $[\![$ $]\!]_\Sigma = j+m \to C \leftarrow i+n$. By traced decomposition (Lemma 66), we have that for any traced boundary complement $i+j \to C \leftarrow m+n$ and morphism $L \to G$, $m \to G \leftarrow n$ can be factored as in (5), i.e.

$$\langle\!\langle \ulcorner\, f \,\urcorner \rangle\!\rangle_\Sigma = \mathrm{Tr}^j \left(\langle\!\langle \ulcorner\, l \,\urcorner \rangle\!\rangle_\Sigma \otimes \mathsf{id}_n \fatsemi \langle\!\langle$$ $$\rangle\!\rangle_\Sigma \right).$$

So by functoriality, we have that $\ulcorner\, f \,\urcorner =$  . The same reasoning follows for $\ulcorner\, h \,\urcorner =$  . □

5.2 Rewriting with traced comonoid structure

To extend rewriting with traced structure to the comonoid case, the traced boundary complement conditions need to be weakened to the case of *left*-monogamous cospans.

Definition 68 (Traced left-boundary complement)**.** *For partial left-monogamous cospans $i \xrightarrow{a_1} L \xleftarrow{a_2} j$ and $n \xrightarrow{b_1} G \xleftarrow{b_2} m \in \mathbf{Hyp}_\Sigma$, a pushout complement as in Definition 60 is called a* traced left-boundary complement *if c_2 is mono and $j+m \xrightarrow{[c_2,d_1]} C \xleftarrow{[c_1,d_2]} i+n$ is a partial left-monogamous cospan.*

Definition 69 (Traced comonoid DPO)**.** *For morphisms $G \leftarrow m+n$ and $H \leftarrow m+n$ in $\mathbf{Hyp}_\Sigma$, there is a traced comonoid rewrite $G \rightsquigarrow_\mathcal{R} H$ if there exists a rule $L \leftarrow i+j \to G \in \mathcal{R}$ and cospan $i+j \to C \leftarrow n+m \in \mathbf{Hyp}_\Sigma$ such that diagram in Definition 51 commutes and $i+j \to C \to G$ is a traced left-boundary complement.*

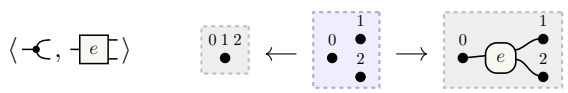
Lemma 70 (Traced comonoid decomposition)**.** *Lemma 66 holds when all cospans are partial left-monogamous and $j+m \xrightarrow{c_2,d_1} C \xleftarrow{c_1,d_2} i+n$ is a traced left-boundary complement.*

Proof. As Lemma 66, but with partial left-monogamous cospans. □

Theorem 71. *Let $\mathcal{R}$ be a rewriting system on $\mathbf{T}_\Sigma + \mathbf{CComon}$. Then,*  $\Rightarrow_\mathcal{R}$  *in $\mathbf{T}_\Sigma + \mathbf{CComon}$ if and only if $\langle\!\langle \ulcorner \lfloor$  $\rfloor_\Sigma \urcorner \rangle\!\rangle_\Sigma \rightsquigarrow \langle\!\langle \ulcorner \lfloor \mathcal{R} \rfloor_\Sigma \urcorner \rangle\!\rangle_\Sigma$* $\langle\!\langle \ulcorner \lfloor$ $\rfloor_\Sigma \urcorner \rangle\!\rangle_\Sigma$.

Proof. This is the same as for Theorem 67 but with the replacement of all traced boundary complements with traced left-boundary complements. □

Example 72. As with the traced case, there may be multiple valid rewrites given a particular interface. The comonoid structure adds more possibilities, as there are the equations of commutative comonoids to consider. Consider the following rule and its interpretation.



(11)

Two valid rewrites are as follows:

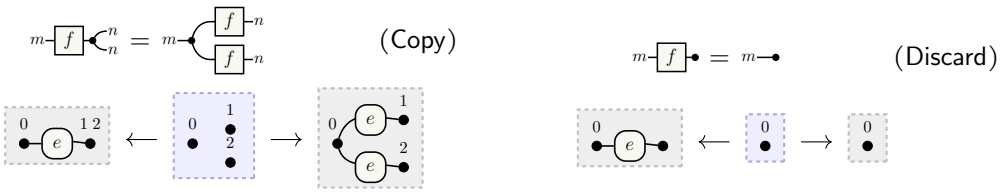
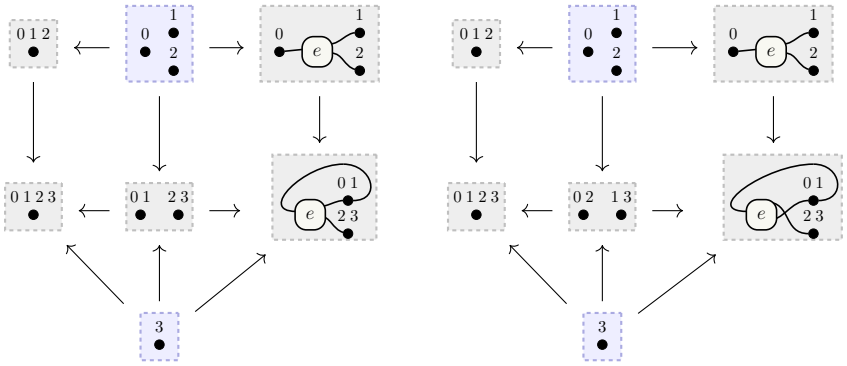
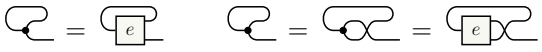


Figure 8: Equations of the monoidal theory $\mathbf{Cart}_{\mathcal{C}}$, where m–f–n is an arbitrary morphism in $\mathcal{C}$, and the interpretations of these equations as rewrite rules for an arbitrary generator e.



The first rewrite is the 'obvious' one, but the second also holds by cocommutativity:



6 Case studies

6.1 Cartesian structure

One important class of categories with a traced comonoid structure are *traced Cartesian categories* [CŞ90; Has97]. These categories are interesting because any traced Cartesian category admits a fixpoint operator [Has97, Thm. 3.1].

Definition 73 (Cartesian category [Fox76])**.** *A monoidal category is* Cartesian *if its tensor is given by the Cartesian product.*

As a result of this, the unit is a terminal object in any Cartesian category, and any object has a comonoid structure. Cartesian categories can also be expressed as a monoidal theory:

Definition 74. *For a given base PROP* $\mathbf{T}_{\Sigma_{\mathbf{C}}}$ *with a comonoid structure, the monoidal theory* $(\Sigma_{\mathbf{Cart}_{\mathbf{C}}}, \mathcal{E}_{\mathbf{Cart}_{\mathbf{C}}})$ *is defined with* $\Sigma_{\mathbf{Cart}_{\mathbf{C}}} := \Sigma_{\mathbf{C}}$ *and* $\mathcal{E}_{\mathbf{Cart}_{\mathbf{C}}}$ *as the equations in Fig. 8.*

As the equations in $\mathcal{E}_{\mathbf{Cart}_{\mathbf{C}}}$ are parameterised over any morphism m–f–n, a separate DPO rewrite rule is required for every combination of generators as in Fig. 8.

Remark 75. The combination of Cartesian equations with the underlying compact closed structure of $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ may prompt alarm bells, as a compact closed category in which the tensor is the Cartesian product is trivial. However, it is important to note that $\mathsf{Csp}_D(\mathbf{Hyp}_\Sigma)$ is *not* subject to these equations: it is only a setting for performing graph rewrites.

Remark 76. One might define a finite set of Cartesian rewrite rules by assuming that the edge actually represents a subgraph, but what do the sources and targets of this edge mean if we would like to match on a single vertex? One solution would be to use the *hierarchical hypergraphs* of [Alv+22] to replace the concrete generator edge with a single edge representing an arbitrary subgraph: this remains as future work.

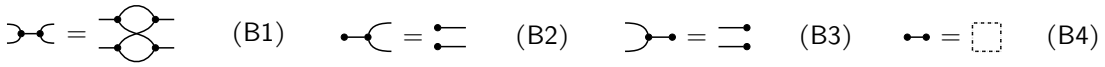


Figure 9: Equations $\mathcal{E}_{\mathbf{Bialg}}$ of a *bialgebra*, in addition to those in Figs. 1 and 2.

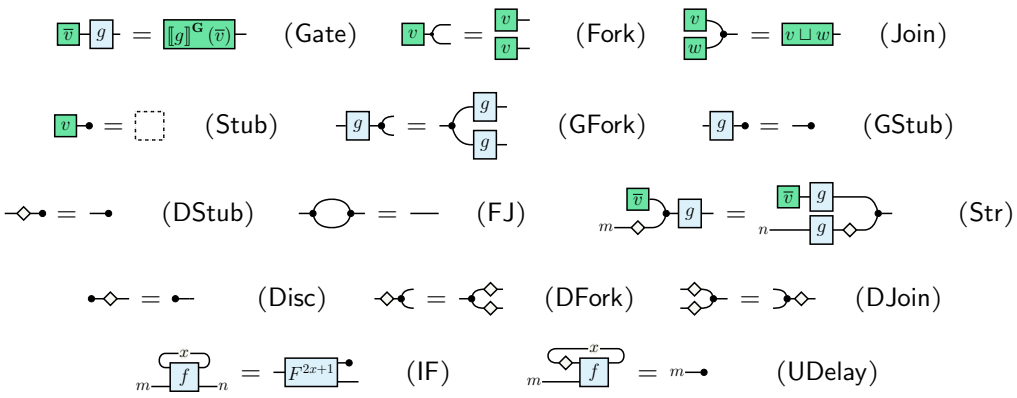


Figure 10: The equations of $\mathcal{E}_{\mathbf{SCirc}}$, from the monoidal theory of gate-level sequential circuits. Section 6.2.

6.2 Digital circuits

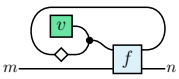
A traced monoidal theory with a comonoid structure that is of particular interest to us is the *local theory of sequential circuits* from [GKS22, Sec. VI].

Definition 77 (Gate-level circuits)**.** *Let the monoidal theory of* gate-level sequential circuits *be defined as* $(\Sigma_{\mathbf{SCirc}}, \mathcal{E}_{\mathbf{SCirc}})$*, where*

$$\Sigma_{\mathbf{SCirc}} := \{ \; , \; , \; , \; , \; , \; , \; , \; \mathsf{t} \; , \; \mathsf{f} \; , \; \mathsf{b} \; , \; \}$$

and the equations of $\mathcal{E}_{\mathbf{SCirc}}$ *are listed in Figs. 1, 2, 9 and 10, where* $[\![-]\!]^{\mathbf{G}}$ *maps gates to the corresponding truth table in Appendix A,* $\sqcup$ *is the join in the information lattice in Appendix A, and* F^n *is defined inductively as* $F^0 := \mathrm{id}$ *and* $F^{k+1} := F^k ; F$.

The generators in $\Sigma_{\mathbf{SCirc}}$ are, respectively: AND, OR and NOT gates; constructs for introducing, forking, joining and stubbing wires; *values* representing a true signal, a false signal, and both signals at once; and a delay of one unit of time.

The equations of $\mathcal{E}_{\mathbf{SCirc}}$ contain the equations of a commutative comonoid, so this is a perfect use case for rewriting modulo trace commutative comonoid structure. Using graph rewriting, we can sketch out an *operational semantics* for sequential circuits. For the interests of brevity, we will only consider circuits of the form  : circuits with no 'non-delay-guarded feedback' in which the registers of the circuit have been isolated from a core containing only 'blue' (*combinational*) components, which models a function. Any sequential circuit can be translated into such a form by the equational theory.

We can 'apply' such a circuit to an input as shown in the left-hand side of Fig. 11; the equations in $\mathcal{E}_{\mathbf{SCirc}}$ can be used to derive the right-hand side. The equations (Gate), (Fork), (Join) and (Stub) can then be applied to reduce the two 'new' cores down to values, representing the output and new state of the circuit.

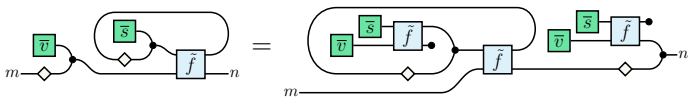


Figure 11: The cycle equation, which is derivable from the equations in $\mathcal{E}_{\mathbf{SCirc}}$

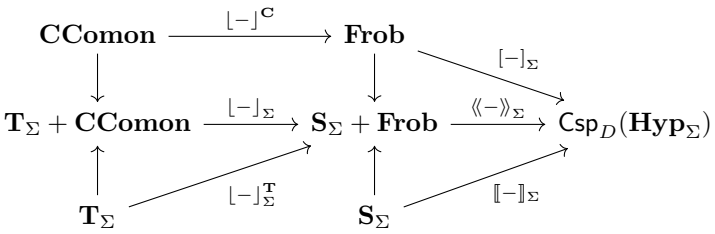


Figure 12: The various PROP morphisms at play.

When the circuits are interpreted as hypergraphs and the equations as rewrites, a computer could perform this sequence of rewrites in order to evaluate circuits in a step-by-step manner.

7 Conclusion, related and further work

We have shown how the frameworks for rewriting string diagrams modulo Frobenius [Bon+22a] and symmetric monoidal [Bon+22b] structure using hypergraphs can also be adapted for rewriting modulo traced comonoid structure, by using hypergraphs that sit between the two settings.

Graphical languages for traced categories have seen many applications, such as to illustrate cyclic lambda calculi [Has97], or to reason graphically about programs [SJ99]. The presentation of traced categories as *string diagrams* has existed since the 90s [JS91; JSV96]; a soundness and completeness theorem for traced string diagrams, folklore for many years but only proven for certain signatures [Sel11], was finally shown in [Kis14]. Combinatorial languages predate even this, having existed since at least the 80s in the guise of *flowchart schemes* [Ste90; CŞ90; CŞ94], These diagrams have also been used to show the completeness of finite dimensional vector spaces [HHP08] with respect to traced categories and, when equipped with a dagger, Hilbert spaces [Sel12].

We are not just concerned with diagrammatic languages as a standalone concept: we are interested in performing *graph rewriting* with them to reason about monoidal theories. This has been been studied in the context of traced categories before using *string graphs* [Kis12; DK13]. We have instead opted to use the *hypergraph* framework of [Bon+22a; Bon+22b; Bon+22c] instead, as it allows rewriting modulo *yanking*, is more extensible for rewriting modulo comonoid structure, and one does not need to awkwardly reason modulo wire homeomorphisms.

As mentioned during the case studies, there are still elements of the rewriting framework that are somewhat informal. One such issue involves defining rewrite spans for arbitrary subgraphs: this is hard to do at a general level because the edges must be concretely specified in DPO rewriting. However, if we performed rewriting with *hierarchical hypergraphs* [Alv+21], in which edges can have hypergraphs as labels, we could 'compress' the subgraph into a single edge that can be rewritten: this is future work.

In regular PROP notation, wires are annotated with numbers in order to avoid drawing multiple wires in parallel: when interpreted as hypergraphs a vertex is created for each wire, and simple diagrams can quickly get very large. The results of [Bon+22b] also extend to the multi-sorted case, in which vertices are labelled in addition to wires. We could use this in combination with the *strictifiers* of [WGZ22]: these are additional generators for transforming buses of wires into thinner or thicker ones. This could drastically reduce the number of elements in a hypergraph, which is ideal from a computational point of view. Work has already begun on implementing the rewriting system for digital circuits using these techniques.

References

[Alv+21] Mario Alvarez-Picallo et al. *Functorial String Diagrams for Reverse-Mode Automatic Differentiation*. July 28, 2021. DOI: 10.48550/arXiv.2107.13433. arXiv: 2107.13433 [cs].

[Alv+22] Mario Alvarez-Picallo et al. "Rewriting for Monoidal Closed Categories". In: *7th International Conference on Formal Structures for Computation and Deduction (FSCD 2022)*. Ed. by Amy P. Felty. Vol. 228. Leibniz International Proceedings in Informatics (LIPIcs). Dagstuhl, Germany: Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2022, 29:1–29:20. ISBN: 978-3-95977-233-4. DOI: 10.4230/LIPIcs.FSCD.2022.29.

[Bel77] Nuel D. Belnap. "A Useful Four-Valued Logic". In: *Modern Uses of Multiple-Valued Logic*. Ed. by J. Michael Dunn and George Epstein. Episteme. Dordrecht: Springer Netherlands, 1977, pp. 5–37. ISBN: 978-94-010-1161-7. DOI: 10.1007/978-94-010-1161-7_2.

[Bon+16] Filippo Bonchi et al. "Rewriting modulo Symmetric Monoidal Structure". In: *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '16. New York, NY, USA: Association for Computing Machinery, July 5, 2016, pp. 710–719. ISBN: 978-1-4503-4391-6. DOI: 10.1145/2933575.2935316.

[Bon+22a] Filippo Bonchi et al. "String Diagram Rewrite Theory I: Rewriting with Frobenius Structure". In: *Journal of the ACM* 69.2 (Mar. 10, 2022), 14:1–14:58. ISSN: 0004-5411. DOI: 10.1145/3502719.

[Bon+22b] Filippo Bonchi et al. "String Diagram Rewrite Theory II: Rewriting with Symmetric Monoidal Structure". In: *Mathematical Structures in Computer Science* 32.4 (Apr. 2022), pp. 511–541. ISSN: 0960-1295, 1469-8072. DOI: 10.1017/S0960129522000317.

[Bon+22c] Filippo Bonchi et al. "String Diagram Rewrite Theory III: Confluence with and without Frobenius". In: *Mathematical Structures in Computer Science* (June 13, 2022), pp. 1–41. ISSN: 0960-1295, 1469-8072. DOI: 10.1017/S0960129522000123.

[CŞ90] Virgil Emil Căzănescu and Gheorghe Ştefănescu. "Towards a New Algebraic Foundation of Flowchart Scheme Theory". In: *Fundamenta Informaticae* 13.2 (Jan. 1, 1990), pp. 171–210. ISSN: 0169-2968. DOI: 10.3233/FI-1990-13204.

[CŞ94] Virgil Emil Căzănescu and Gheorghe Ştefănescu. "Feedback, Iteration, and Repetition". In: *Mathematical Aspects of Natural and Formal Languages*. Vol. Volume 43. World Scientific Series in Computer Science Volume 43. World Scientific, Oct. 1, 1994, pp. 43–61. ISBN: 978-981-02-1914-7. DOI: 10.1142/9789814447133_0003.

[CW87] A. Carboni and R. F. C. Walters. "Cartesian Bicategories I". In: *Journal of Pure and Applied Algebra* 49.1 (Nov. 1, 1987), pp. 11–32. ISSN: 0022-4049. DOI: 10.1016/0022-4049(87)90121-6.

[DK13] Lucas Dixon and Aleks Kissinger. "Open-Graphs and Monoidal Theories". In: *Mathematical Structures in Computer Science* 23.2 (Apr. 2013), pp. 308–359. ISSN: 0960-1295, 1469-8072. DOI: 10.1017/S0960129512000138.

[FL22] Tobias Fritz and Wendong Liang. *Free Gs-Monoidal Categories and Free Markov Categories*. Apr. 17, 2022. DOI: 10.48550/arXiv.2204.02284. arXiv: 2204.02284 [cs, math, stat].

[Fox76] Thomas Fox. "Coalgebras and Cartesian Categories". In: *Communications in Algebra* 4.7 (Jan. 1, 1976), pp. 665–667. ISSN: 0092-7872. DOI: 10.1080/00927877608822127.

[FS19] Brendan Fong and David I. Spivak. "Hypergraph Categories". In: *Journal of Pure and Applied Algebra* 223.11 (Nov. 1, 2019), pp. 4746–4777. ISSN: 0022-4049. DOI: 10.1016/j.jpaa.2019.02.014.

[GKS22] Dan R. Ghica, George Kaye, and David Sprunger. *A Compositional Theory of Digital Circuits*. Aug. 1, 2022. DOI: 10.48550/arXiv.2201.10456. arXiv: 2201.10456 [cs, math].

[Has09] Masahito Hasegawa. "On Traced Monoidal Closed Categories". In: *Mathematical Structures in Computer Science* 19.2 (Apr. 2009), pp. 217–244. ISSN: 1469-8072, 0960-1295. DOI: 10.1017/S0960129508007184.

[Has97] Masahito Hasegawa. "Recursion from Cyclic Sharing: Traced Monoidal Categories and Models of Cyclic Lambda Calculi". In: *Typed Lambda Calculi and Applications*. Ed. by Philippe de Groote and J. Roger Hindley. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1997, pp. 196–213. ISBN: 978-3-540-68438-1. DOI: 10.1007/3-540-62688-3_37.

[Heu+11] Marvin Heumüller et al. "Construction of Pushout Complements in the Category of Hypergraphs". In: *Electronic Communications of the EASST* 39.0 (0 Sept. 20, 2011). ISSN: 1863-2122. DOI: 10.14279/tuj.eceasst.39.647.

[HHP08] Masahito Hasegawa, Martin Hofmann, and Gordon Plotkin. "Finite Dimensional Vector Spaces Are Complete for Traced Symmetric Monoidal Categories". In: *Pillars of Computer Science: Essays Dedicated to Boris (Boaz) Trakhtenbrot on the Occasion of His 85th Birthday*. Ed. by Arnon Avron, Nachum Dershowitz, and Alexander Rabinovich. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2008, pp. 367–385. ISBN: 978-3-540-78127-1. DOI: 10.1007/978-3-540-78127-1_20.

[JS91] André Joyal and Ross Street. "The Geometry of Tensor Calculus, I". In: *Advances in Mathematics* 88.1 (1991), pp. 55–112. ISSN: 0001-8708. DOI: 10.1016/0001-8708(91)90003-P.

[JSV96] André Joyal, Ross Street, and Dominic Verity. "Traced Monoidal Categories". In: *Mathematical Proceedings of the Cambridge Philosophical Society* 119.3 (Apr. 1996), pp. 447–468. ISSN: 1469-8064, 0305-0041. DOI: 10.1017/S0305004100074338.

[Kis12] Aleks Kissinger. "Pictures of Processes: Automated Graph Rewriting for Monoidal Categories and Applications to Quantum Computing". PhD thesis. University of Oxford, Mar. 22, 2012. DOI: 10.48550/arXiv.1203.0202. arXiv: 1203.0202 [quant-ph].

[Kis14] Aleks Kissinger. "Abstract Tensor Systems as Monoidal Categories". In: *Categories and Types in Logic, Language, and Physics: Essays Dedicated to Jim Lambek on the Occasion of His 90th Birthday*. Ed. by Claudia Casadio et al. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2014, pp. 235–252. ISBN: 978-3-642-54789-8. DOI: 10.1007/978-3-642-54789-8_13.

[Lac04] Stephen Lack. "Composing PROPs". In: *Theory and Applications of Categories* 13.9 (2004), pp. 147–163. ISSN: 1201-561X.

[Law63] F. William Lawvere. "Functorial Semantics of Algebraic Theories". In: *Proceedings of the National Academy of Sciences* 50.5 (Nov. 1963), pp. 869–872. DOI: 10.1073/pnas.50.5.869.

[LS04] Stephen Lack and Paweł Sobociński. "Adhesive Categories". In: *Foundations of Software Science and Computation Structures*. Ed. by Igor Walukiewicz. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, pp. 273–288. ISBN: 978-3-540-24727-2. DOI: 10.1007/978-3-540-24727-2_20.

[LS05] Stephen Lack and Paweł Sobociński. "Adhesive and Quasiadhesive Categories". In: *RAIRO - Theoretical Informatics and Applications* 39.3 (July 2005), pp. 511–545. ISSN: 0988-3754, 1290-385X. DOI: 10.1051/ita:2005028.

[Mac65] Saunders MacLane. "Categorical Algebra". In: *Bulletin of the American Mathematical Society* 71.1 (1965), pp. 40–106. ISSN: 0002-9904, 1936-881X. DOI: 10.1090/S0002-9904-1965-11234-4.

[MZ22] Aleksandar Milosavljevic and Fabio Zanasi. *String Diagram Rewriting Modulo Commutative Monoid Structure*. Apr. 8, 2022. DOI: 10.48550/arXiv.2204.04274. arXiv: 2204.04274 [cs, math].

[RSW05] R. Rosebrugh, N. Sabadini, and R. F. C. Walters. "Generic Commutative Separable Algebras and Cospans of Graphs." In: *Theory and Applications of Categories* 15 (2005), pp. 164–177. ISSN: 1201-561X.

[Sel11] Peter Selinger. "A Survey of Graphical Languages for Monoidal Categories". In: *New Structures for Physics*. Ed. by Bob Coecke. Lecture Notes in Physics. Berlin, Heidelberg: Springer, 2011, pp. 289–355. ISBN: 978-3-642-12821-9. DOI: 10.1007/978-3-642-12821-9_4.

[Sel12] Peter Selinger. "Finite Dimensional Hilbert Spaces Are Complete for Dagger Compact Closed Categories". In: *Logical Methods in Computer Science* 8.3 (Aug. 10, 2012). DOI: 10.2168/LMCS-8(3:6)2012.

[SJ99] Ralf Schweimeier and Alan Jeffrey. "A Categorical and Graphical Treatment of Closure Conversion". In: *Electronic Notes in Theoretical Computer Science*. MFPS XV, Mathematical Foundations of Progamming Semantics, Fifteenth Conference 20 (Jan. 1, 1999), pp. 481–511. ISSN: 1571-0661. DOI: 10.1016/S1571-0661(04)80090-2.

[Ște90] Gheorghe Ștefănescu. "Feedback Theories (A Calculus for Isomorphism Classes of Flowchart Schemes)". In: *Romanian Journal of Pure and Applied Mathematics* 35.1 (1990), pp. 73–79.

[WGZ22] Paul Wilson, Dan Ghica, and Fabio Zanasi. *String Diagrams for Non-Strict Monoidal Categories*. Jan. 29, 2022. DOI: 10.48550/arXiv.2201.11738. arXiv: 2201.11738 [math].

A Belnap logic

The equational theory of gate-level sequential circuits in Section 6.2 is based on the four-valued *Belnap logic* [Bel77]. The four values in question, •, f, t and b form a lattice in two ways. The first is the lattice of *information order* shown in the left of Fig. 13, which dictates how much 'information content' a particular value has. The second is the lattice of *logical order*, which can be seen as the lattice in Fig. 13 rotated by 90

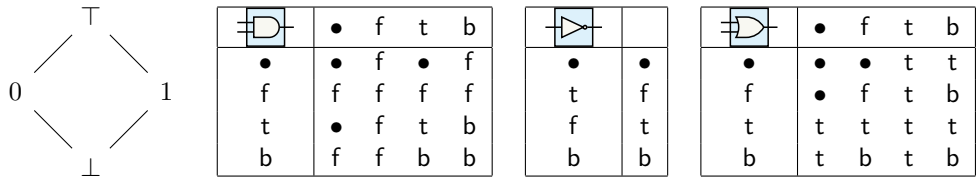


Figure 13: The lattice structure on Belnap values, and the truth tables of Belnap logic gates [Bel77].

degrees clockwise. This lattice determines the outputs of the gates: the AND gate is the meet in this lattice and the OR gate is the join. The NOT gate acts in the usual way on f and t, and as the identity otherwise. The truth tables are shown in the right of Fig. 13.